## Useful financial functions in Excel

<table>
<thead>
<tr>
<th>Abbreviation</th>
<th>Description</th>
<th>Abbreviation</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>PV</td>
<td>Present value of an annuity</td>
<td>RATE</td>
<td>Rate of return of an annuity</td>
</tr>
<tr>
<td>NPV</td>
<td>Net present value of periodic cash flows</td>
<td>IRR</td>
<td>Internal rate of return of periodic cash flows</td>
</tr>
<tr>
<td>FV</td>
<td>Future value of an annuity</td>
<td></td>
<td></td>
</tr>
<tr>
<td>PRICE</td>
<td>Price of a coupon bond</td>
<td>YIELD</td>
<td>Yield of coupon bond</td>
</tr>
<tr>
<td>PRICEDISC</td>
<td>Price of a discount bond</td>
<td>YIELDDISC</td>
<td>Yield of discount bond</td>
</tr>
<tr>
<td>TBILLPRICE</td>
<td>Price of T-bill (special case of PRICEDISC)</td>
<td>TBILLYIELD</td>
<td>Yield of T-bill</td>
</tr>
<tr>
<td>ACCRINT</td>
<td>Accrued interest</td>
<td>COUPDAYS</td>
<td>Number of days in current coupon</td>
</tr>
<tr>
<td>COUPNUM</td>
<td>Number of coupons remaining</td>
<td>COUPDAYBS</td>
<td>Number of days between previous settlement</td>
</tr>
<tr>
<td>COUPNCD</td>
<td>Next coupon date</td>
<td>COUPDAYSNC</td>
<td>Number of days between settlement coupon</td>
</tr>
<tr>
<td>COUPPCD</td>
<td>Previous coupon date</td>
<td></td>
<td></td>
</tr>
<tr>
<td>DURATION</td>
<td>Duration of a coupon bond</td>
<td>MDURATION</td>
<td>Modified duration</td>
</tr>
<tr>
<td>EFFECT</td>
<td>Effective annual interest rate</td>
<td>TBILLEQ</td>
<td>Bond equivalent yield of a T-bill</td>
</tr>
</tbody>
</table>
The Formula Auditing Toolbar enables you to trace graphically the relationships between cells. It also allows you to monitor cell contents by placing them in a Watch Window.

To display the formula auditing toolbar

View > Toolbars > Formula auditing

To trace a cell's precedents

1. Select a cell containing a formula
2. Click on the Trace Precedents button
3. Click on the Trace Precedents button again to display the previous level of precedents.
4. Remove tracer arrows one level at a time by clicking Remove Precedent Arrows

To trace a cell's dependents

1. Select a cell containing a formula
2. Click on the Trace Dependents button
3. Click on the Trace Dependents button again to display the previous level of dependents.
4. Remove tracer arrows one level at a time by clicking Remove Dependent Arrows

To select the cell at the other end of an arrow

Double click the arrow

To remove all tracer arrows

Click the Remove All Arrows button.

To display all the relationships in a worksheet

1. In an empty cell, type =
2. Then click the Select All button and evaluate the cell with Ctrl-Enter
3. Click the Trace Precedents button twice.

To display a formula in a cell

Select the cell and press F2
To display all formulae

Click Ctrl-~

To add a cell to the watch window

1. Open the Watch Window by clicking on the Watch Window button in the Formula Auditing Toolbar.
2. Select the cells you want to monitor.
3. Click on the Add Watch button in the Watch Window.
Basic bond pricing

In principal, pricing a risk-free bond is deceptively simple - the price or value of a bond is the present value of the future cash flows, discounted at the prevailing rate of interest, which is known as the **yield**.

\[ P = \sum_{i=1}^{T} C \left( \frac{1}{1 + r} \right)^i + R \left( \frac{1}{1 + r} \right)^T \]

where \( P \) is the price, \( C \) is the coupon, \( R \) is the redemption value (principal) and \( T \) is the term. Alternatively, the yield of a bond is the internal rate of return at which the discounted value is equal to market price. Bonds are known as **fixed income assets**, because the timing and magnitude of the future cash flows are fixed. Their value however varies inversely with the yield. Bonds of similar risk and term will attract similar yields.

In practice, bond pricing is more complicated because

- coupons are paid more frequently than annually, typically every six months.
- a price is required between coupon periods necessitating discounting for fractional periods.
- interest rates (yields) may be expected to change during the term of the bond.

The first complication is dealt with by treating the coupon period (e.g. 6 months) as the discounting period. If there are \( m \) coupons per year,

\[ P = \sum_{i=1}^{mT} C \left( \frac{1}{1 + \frac{r}{m}} \right)^{i} + R \left( \frac{1}{1 + \frac{r}{m}} \right)^{mT} \]

Treatment of fractional periods is a matter of market convention. In particular, various markets employ different **day count conventions** for calculating the fraction of the coupon period which as elapsed on a given day. Similar conventions are employed for pricing zero coupon bonds. However, zero coupon bonds issued with a maturity less than one year (**notes**) are priced with yet another convention. Computation of bond prices and yields requires being familiar with the prevailing conventions.

Changing interest rates (**the yield curve**) can be accommodated by discounting each cash flow at the appropriate **spot rate**. Credit risk can be incorporated in a simple way by discounting at a higher rate than the yield on risk-free bonds. This difference, known as the **spread**, depends upon the credit rating of the issuer. More sophisticated measures employ credit risk models to allow for the possibility of default and ratings changes during a given horizon. Sophisticated measures will also account directly for the options embedded in many bonds, as for example in a callable bond.
Day count conventions

There are two aspects to calculating the fraction \( \alpha \) of a year to which an interest payment applies - the day count convention.

- determining the year fraction \( \alpha(d_1, d_2) \) given \( d_1 \) and \( d_2 \). This is known as the basis.
- determining \( d_1 \) and \( d_2 \).

The most frequently encountered of the bases are:

- **actual/365** \( \alpha(d_1, d_2) = \frac{d_2 - d_1}{365} \)
- **actual/360** \( \alpha(d_1, d_2) = \frac{d_2 - d_1}{360} \)
- **actual/actual** \( \alpha(d_1, d_2) = \frac{f_{i+1} - f_i}{f_{i+1} - f_i} + \frac{m_{n-1} - m_i}{m_{n-1} - m_i} \)
- **30/360** \( \alpha(d_1, d_2) = \frac{1}{12} \left( \frac{m_2 - d_1}{30} + (n - 3) + \frac{d_2 - m_{n-1}}{30} \right) \)

where \( f_i \) are year end dates with \( f_1 \leq d_1 \leq f_2 \leq f_{n-1} \leq d_2 \leq f_n \) and where \( m_i \) are month end dates with \( m_1 \leq d_1 \leq m_2 \leq m_{n-1} \leq d_2 \leq m_n \). Markets differ in how they treat situations such as \( d_2 = m_n \) and \( m_n = 29 \) or \( m_n = 31 \). \( d_2 - d_1 \) means the number of days from, and including, \( d_1 \) until, but excluding, \( d_2 \).

Actual/actual can be expressed alternatively as

\[ \alpha(d_1, d_2) = \max(30 - d_1, 0) + 360 \times (y_2 - y_1) + 30 \times (m_2 - m_1 - 1) + \min(30, d_2) \]

We can convert from actual/365 to actual/360 by multiplying by 365/360. But conversion between the other bases is more complicated and arbitrary.

- **Government bonds**
  
  **Basis:** CAD, GBP and JPY use actual/365. Other European currencies use 30/360. USD uses actual/actual. INR uses the 30/360 basis for instruments with residual maturity exceeding a year, and the actual/365 basis otherwise.

- **Money market (Libor and cash rates)**
  
  **Basis:** AUD, BEF, CAD, and GBP use actual/365. Most others use actual/360. Euribor and Euribor both use 30/360.

  **Start date:** AUD, CAD, and GBP use same day. FRF uses same day +1. Most others including Euribor and Euribor use same day +2.

  **End date:** Many currencies use the modified following business day convention. The end date is on the following business day, unless it is in a different month, in which case it is on the previous business day. But if the start date was the last business day of a month then end date is the last business day in the corresponding month.

- **Swaps**
  
  **Basis:** Typically the same as government bonds.

  **Start and date:** Floating side uses Libor conventions. Fixed side uses the same start date as the floating side.
Duration and convexity

**Duration and Sensitivity**

Assuming annual coupons, the price of a coupon bond is the discounted value of cash flows

\[
P = \sum_{t=1}^{T} C \left( \frac{1}{1+r} \right)^t + R \left( \frac{1}{1+r} \right)^T
\]

where \( P \) is the full or dirty price, \( C \) is the annual coupon, \( R \) is the redemption value and \( T \) is the term. This can be rewritten as

\[
P = \sum_{t=1}^{T} C (1 + r)^{-t} + R (1 + r)^{-T}
\]

Differentiating with respect to the yield gives

\[
\frac{\partial P}{\partial r} = \sum_{t=1}^{T} -t C (1 + r)^{-t-1} - TR (1 + r)^{-T-1}
\]

which can be written as

\[
\frac{\partial P}{\partial r} = -\frac{1}{1+r} \sum_{t=1}^{T} C \left( \frac{1}{1+r} \right)^t + R \left( \frac{1}{1+r} \right)^T
\]

which is precisely the term inside the brackets in equation (1). Substituting equation (2) into equation (1) gives

\[
\frac{\partial P}{\partial r} = -\frac{1}{1+r} \text{Dur} \times P
\]
With \( m \) coupons per year, this becomes (see below)

\[
\frac{\partial P}{\partial r} = - \left( \frac{1}{1 + \frac{r}{m}} \right) \times P
\]

To simplify, we call the product on the left modified duration. That is, defining

\[
MDur = \frac{1}{1 + \frac{r}{m}} \text{ Dur}
\]

we have

\[
\frac{dP}{dr} = -\text{MDur} \times P
\]

For small changes in interest rate, we have

\[
\frac{\Delta P}{\Delta r} \approx -\text{MDur} \times P
\]

or

\[
\frac{\Delta P}{P} \approx -\text{MDur} \times \Delta r
\]

A one percentage point increase in yield will lead to (approx.) MDur fall in price.

Practitioners often express duration (that is, interest-rate sensitivity) in terms of the dollar value of a basis point (DV01) or more generally price value of a basis point (PV01). This is defined as

\[
PV01 = \text{MDur} \times P \times 0.01 \times 0.01
\]

Note that, strictly speaking, it is the invoice or dirty price that should be used for \( P \) in this calculation.

---

**Multiannual coupons**

If there are \( m \) coupons per year, the price of a bond is

\[
P = \sum_{i=1}^{mT} C \left( \frac{1}{1 + \frac{r}{m}} \right)^i \left( \frac{1}{1 + \frac{r}{m}} \right)^{mT}
\]

\[
= \sum_{i=1}^{mT} C \left( \frac{1 + \frac{r}{m}}{1 + \frac{r}{m}} \right)^{-i} + R \left( \frac{1}{1 + \frac{r}{m}} \right)^{mT}
\]

Differentiating with respect to the yield

\[
\frac{\partial P}{\partial r} = - \sum_{i=1}^{mT} \frac{C}{m} \left( \frac{1 + \frac{r}{m}}{1 + \frac{r}{m}} \right)^{-i-1} - mT \left( \frac{r}{m} \right)^{-mT-1} \frac{1}{m}
\]

\[
= - \left( \frac{1}{1 + \frac{r}{m}} \right) \left( \sum_{i=1}^{mT} \frac{C}{m} \left( \frac{1}{1 + \frac{r}{m}} \right)^{i} + TR \left( \frac{1}{1 + \frac{r}{m}} \right)^{mT} \right)
\]
A closed formula for duration

Inverting the previous equation, the duration of a bond is

\[
\text{Dur} = -\left(1 + \frac{\ell}{m}\right) \frac{\partial P}{\partial r}
\]

where

\[
P = \sum_{t=1}^{mT} \frac{C}{m} \left( \frac{1}{1 + \frac{\ell}{m}} \right)^t + R \left( \frac{1}{1 + \frac{\ell}{m}} \right)^{mT}
\]

By summing the geometric series, the price of the bond can be written in closed form as

\[
P = \frac{C}{r} \left(1 - \frac{1}{\left(1 + \frac{\ell}{m}\right)^{mT}}\right) + \frac{1}{\left(1 + \frac{\ell}{m}\right)^{mT}} R
\]

Differentiating this expression and substituting in (2), we obtain a closed formula for the duration of a bond

\[
\text{Dur} = \frac{1 + \frac{\ell}{m}}{r} - \frac{T \left(\frac{C}{R} - r\right) + \left(1 + \frac{\ell}{m}\right)}{\left(\frac{C}{R} \left(\left(1 + \frac{\ell}{m}\right)^{mT} - 1\right) + r\right)}
\]

When the bond is at par, \(C/R = r\), and this simplifies to

\[
\text{Dur} = \frac{1 + \frac{\ell}{m}}{r} \left(1 - \frac{1}{\left(1 + \frac{\ell}{m}\right)^{mT}}\right)
\]

The limit of duration for long term bonds

As \(T\) goes to infinity, the second term in equation (3) goes to zero. Therefore, the duration of a long-term bond converges to
For example, with a yield of 5%, the duration of a biannual converges to $\frac{1 + \frac{5}{2}}{5\%} = 20.5$

### Convexity

Duration is related to the first derivative of bond price with respect to yield. Convexity is a measure of the second derivative, normalised by bond price.

$$C = \frac{\frac{d^2 P}{dr^2}}{P}$$

Though it is possible to derive a formula for convexity, by differentiating the above formula for $dP/dr$, we would need to incorporate the complications date count conventions for mid-coupon bonds. Alternatively, we can estimate convexity accurately by numerical differentiation

$$C = \frac{P(r + dr) - 2P(r) + P(r - dr)}{P \cdot dr^2}$$

where $dr$ is a small change in interest rate (e.g. 0.0001 for 1 basis point).

Alternatively, we can compute convexity from the first derivative of duration (this is useful if we have a formula for duration, as in Excel). From above

$$PC = \frac{d^2 P}{dr^2} = \frac{d\left(\frac{dP}{dr}\right)}{dr}$$

But

$$\frac{dP}{dr} = -PD$$

where $D$ is modified duration. Substituting and using the product rule

$$PC = \frac{d(-PD)}{dr} = -D\left(\frac{dP}{dr}\right) - P\left(\frac{dD}{dr}\right)$$

$$= PD^2 - P\left(\frac{dD}{dr}\right)$$

so that
\[ C = D^2 - \left( \frac{dD}{dr} \right) \]

\( dD/dr \) can itself be calculated by numerical differentiation.
Bootstrapping

With annual compounding, the price of a unit par bond with \( n \) years remaining is given by

\[
c P_1 + c P_2 + \ldots + c P_{n-1} + (1 + c) P_t = 1
\]

where \( c \) is the coupon (yield) and \( P_t \) is the discount factor (price of a \( t \)-year zero-coupon bond). This can be solved successively to give the prices of zero-coupon bonds to match a given yield curve.

\[
P_t = \frac{1 - c \sum_{i=1}^{n-1} P_i}{1 + c}
\]

For semi-annual coupons, the analogous equations are

\[
\frac{c}{2} P_1 + \frac{c}{2} P_1 + \ldots + \frac{c}{2} P_{t-\frac{1}{2}} + \left(1 + \frac{c}{2}\right) P_t = 1
\]

and

\[
P_t = \frac{1 - \frac{c}{2} \sum_{i=\frac{1}{2}}^{n-\frac{1}{2}} P_i}{1 + \frac{c}{2}}
\]
Estimating spot rates

Michael Carter

The basic bond pricing equation is

\[
P + AI = \sum_{i=1}^{n} \frac{C}{(1 + s_i / m)^{mi}} + \frac{R}{(1 + s_n / m)^{mn}}
\]  

(1)

where

\[P = \text{clean price}\]
\[AI = \text{accrued interest}\]
\[C = \text{annual coupon}\]
\[R = \text{redemption payment (principal)}\]
\[m = \text{frequency of coupons}\]
\[n = \text{number of remaining coupons}\]

This can be written in terms of the discount factors

\[P + AI = \frac{C}{m} \sum_{i=1}^{n} \delta_{t_i} + \delta_{t_n} R\]

where

\[\delta_{t_i} = \left(\frac{1}{1 + s_i / m}\right)^{mt_i}\]

The spot rates or discount factors also determine the forward rates. Let \(r_t\) denote the forward (short) rate

\[(1 + \frac{s}{m})^{mt_i} = (1 + \frac{s}{m})^{mt_{i-1}} \left(1 + \frac{r_t}{m}\right)\]

so that

\[(1 + \frac{r_t}{m}) = \frac{(1 + \frac{s}{m})^{mt_i}}{(1 + \frac{s}{m})^{mt_{i-1}}} = \frac{\delta_{t_{i-1}}}{\delta_t}\]

\[r_t = m \frac{\Delta \delta_{t_i}}{\delta_t} = m \frac{\Delta \delta_{t_i}}{\delta_t}\]

If there is an active market in zero-coupon bonds, these can be used to give immediate market estimates of the discount rate at various terms. However, such instruments are traded only in the U.K. and U.S. treasury markets. Moreover, even in these markets, they are usually disregarded because of restricted maturities, limited liquidity and tax complications.
In principle, discount factors \( h_t \) can be inferred from the prices of coupon bonds by inverting (1). In turn, these can be used to infer the spot rate \( z_t \) and forward rate \( r_t \) curves. The inversion process is known as bootstrapping.

In practice, estimation of the spot rate curve is complicated by two basic problems:

- Bonds of the same maturity may be selling at different yields, due to market imperfections, limited liquidity, tax etc.
- There may be no data on bonds of other maturities.

These problems are tackled (with varying degrees of success) by statistical estimation and interpolation.

The basic approach is to assume a specific functional form for the forward rate or discount function, and then adjust the parameters until the best fit is obtained. Simple polynomial functions such as

\[
f(t) = a_0 + a_1 t + a_2 t^2 + a_3 t^3
\]

have been found not to be very suitable, since they imply that rates go to plus or minus infinity as \( t \to \infty \).

Two basic generalizations are found - exponential functions and polynomial or exponential splines.

### Parsimonious functional forms

The most straightforward generalization of (2) is to substitute an exponential for each power of \( t \), fitting a model of the form

\[
f(t) = a_0 + a_1 e^{-k_1 t} + a_2 e^{-k_2 t} + a_3 e^{-k_3 t} + \ldots
\]

This is the exponential yield model adopted by J.P Morgan.

The most popular model of this form is due to Nelson and Siegel (1987). They observe that the second order exponential model is the general solution to a second-order differential equation (assuming real unequal roots)

\[
f(t) = b_0 + b_1 e^{-\tau_1 t} + b_2 e^{-\tau_2 t}
\]

where \( \tau_1, \tau_2 \) are the rates of decay. Finding that this is overparameterized, they adopt the general solution for the case of equal roots

\[
f(t) = b_0 + \frac{t}{\tau} e^{-\frac{t}{\tau}}
\]

The short rate is \( b_0 + b_1 \), while the long rate is \( \lim_{t \to \infty} f(t) = b_0 \). \( b_1 \) can be interpreted as the weight attached to the short term component, and \( b_2 \) as the weight of the medium term. \( \tau \) determines the rate of decay.

The spot rate, the average of the forward rates, can be obtained by integrating this equation, giving

\[
s(t) = b_0 + (b_1 + b_2) \frac{\tau}{t} \left(1 - e^{-\frac{t}{\tau}}\right) - b_2 e^{-\frac{t}{\tau}}
\]

This is the model adopted by the National Stock Exchange of India for estimating its published spot rate series.
Svennson (1994) extended this specification by adding an additional term for greater flexibility, specifically

$$f(t) = \beta_0 + \beta_1 e^{-\frac{t}{\tau_1}} + \beta_2 \frac{t}{\tau_1} e^{-\frac{t}{\tau_1}} + \beta_3 \frac{t}{\tau_2} e^{-\frac{t}{\tau_2}}$$

The corresponding spot rate curve is

$$s(t) = \beta_0 + (\beta_1 + \beta_2) \frac{t}{\tau_1} \left(1 - e^{-\frac{t}{\tau_1}}\right) - \beta_2 e^{-\frac{t}{\tau_1}} + \beta_3 \frac{t}{\tau_2} \left(1 - e^{-\frac{t}{\tau_2}}\right) - \beta_3 e^{-\frac{t}{\tau_2}}$$

This is the model used by the Deutsche Bundesbank for estimating its published spot rate series.

A recent comprehensive review by Ioannides (2003) found that the parsimonious functional forms out-performed corresponding spline methods, with the Svennson specification preferred over that of Nelson and Siegel. However, we note that the Bank of England recently drew the opposite conclusion, switching from Svensson's method to a spline method (Anderson and Sleath, 1999).

**Example: National Stock Exchange of India**

Estimating the Nelson-Siegel model for bonds traded on 26 June 2004 yields the following parameter estimates

$$\beta_0 = 0.0727, \beta_1 = -0.0231, \beta_2 = -0.0210, \tau = 2.8601$$

**Example: Deutsche Bundesbank**

For the 15 September 2004, the Deutsche Bundesbank estimated the following parameters for the Svennson model:

$$\beta_0 = 5.4596, \beta_1 = -3.53042, \beta_2 = -0.37788, \beta_3 = -0.98812, \tau_1 = 2.70411, \tau_2 = 2.53479$$

These parameters imply the following spot rates.
The spot and forward curves are illustrated in the following graph.
'Implementation of Nelson-Siegel method for estimating forward rate curve
'Michael Carter, 2004

Function GetFormula(ThisCell)
    GetFormula = ThisCell.Formula
End Function

'Discount function
Function df(t As Double, b0 As Double, b1 As Double, b2 As Double, tau As Double) As Double
    df = Exp(-t * (b0 + (b1 + b2) * (1 - Exp(-t / tau)) * (tau / t) - b2 * Exp(-t / tau)))
End Function

'Bond price function
Function Pr(t As Double, C As Double, n As Integer, b0 As Double, b1 As Double, b2 As Double, tau As Double) As Double
Dim i As Integer
Dim P As Double
P = 0
For i = 1 To n
    P = P + df(t + (i - 1) / 2, b0, b1, b2, tau) * (100 * C / 2)
Next i
Pr = P + df(t + (n - 1) / 2, b0, b1, b2, tau) * 100
End Function
Table 1

The term structure of interest rates - estimation details

<table>
<thead>
<tr>
<th>Central bank</th>
<th>Estimation method</th>
<th>Minimised error</th>
<th>Shortest maturity in estimation</th>
<th>Adjustments for tax distortions</th>
<th>Relevant maturity spectrum</th>
</tr>
</thead>
<tbody>
<tr>
<td>Belgium</td>
<td>Svensson or Nelson-Siegel</td>
<td>Weighted prices</td>
<td>Treasury certificates: &gt; few days Bonds: &gt; one year</td>
<td>No</td>
<td>Couple of days to 16 years</td>
</tr>
<tr>
<td>Canada</td>
<td>Merrill Lynch Exponential Spline</td>
<td>Weighted prices</td>
<td>Bills: 1 to 12 months Bonds: &gt; 12 months</td>
<td>Effectively by excluding bonds</td>
<td>3 months to 30 years</td>
</tr>
<tr>
<td>Finland</td>
<td>Nelson-Siegel</td>
<td>Weighted prices</td>
<td>≥ 1 day</td>
<td>No</td>
<td>1 to 12 years</td>
</tr>
<tr>
<td>France</td>
<td>Svensson or Nelson-Siegel</td>
<td>Weighted prices</td>
<td>Treasury bills: all Treasury Notes: : ≥ 1 month Bonds: : ≥ 1 year</td>
<td>No</td>
<td>Up to 10 years</td>
</tr>
<tr>
<td>Germany</td>
<td>Svensson</td>
<td>Yields</td>
<td>&gt; 3 months</td>
<td>No</td>
<td>1 to 10 years</td>
</tr>
<tr>
<td>Italy</td>
<td>Nelson-Siegel</td>
<td>Weighted prices</td>
<td>Money market rates: O/N and Libor rates from 1 to 12 months Bonds: &gt; 1 year</td>
<td>No</td>
<td>Up to 30 years</td>
</tr>
<tr>
<td>Japan</td>
<td>Smoothing splines</td>
<td>Prices</td>
<td>≥ 1 day</td>
<td>Effectively by price adjustments for bills</td>
<td>1 to 10 years</td>
</tr>
<tr>
<td>Norway</td>
<td>Svensson</td>
<td>Yields</td>
<td>Money market rates: &gt; 30 days Bonds: &gt; 2 years</td>
<td>No</td>
<td>Up to 10 years</td>
</tr>
<tr>
<td>Spain</td>
<td>Svensson Nelson-Siegel (before 1995)</td>
<td>Weighted prices</td>
<td>≥ 1 day</td>
<td>≥ 1 day</td>
<td>Yes</td>
</tr>
<tr>
<td>Sweden</td>
<td>Smoothing splines and Svensson</td>
<td>Yields</td>
<td>≥ 1 day</td>
<td>No</td>
<td>Up to 10 years</td>
</tr>
<tr>
<td>Switzerland</td>
<td>Svensson</td>
<td>Yields</td>
<td>Money market rates: ≥ 1 day Bonds: ≥ 1 year</td>
<td>No</td>
<td>1 to 30 years</td>
</tr>
</tbody>
</table>
Table 1 cont

The term structure of interest rates - estimation details

<table>
<thead>
<tr>
<th>Central bank</th>
<th>Estimation method</th>
<th>Minimised error</th>
<th>Shortest maturity in estimation</th>
<th>Adjustments for tax distortions</th>
<th>Relevant maturity spectrum</th>
</tr>
</thead>
<tbody>
<tr>
<td>United Kingdom(^1)</td>
<td>VRP (government nominal)</td>
<td>Yields</td>
<td>1 week (GC repo yield)</td>
<td>No</td>
<td>Up to around 30 years</td>
</tr>
<tr>
<td></td>
<td>VRP (government real/implied inflation)</td>
<td>Yields</td>
<td>1.4 years</td>
<td>No</td>
<td>Up to around 30 years</td>
</tr>
<tr>
<td></td>
<td>VRP (bank liability curve)</td>
<td>Yields</td>
<td>1 week</td>
<td>No</td>
<td>Up to around 30 years</td>
</tr>
<tr>
<td>United States</td>
<td>Smoothing splines (two curves)</td>
<td>Bills: weighted prices</td>
<td>–</td>
<td>No</td>
<td>Up to 1 year</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Bonds: prices</td>
<td>≥ 30 days</td>
<td>No</td>
<td>1 to 10 years</td>
</tr>
</tbody>
</table>

\(^1\) The United Kingdom used the Svensson method between January 1982 and April 1998.

3. Zero-coupon yield curves available from the BIS

Table 2 provides an overview of the term structure information available from the BIS Data Bank. Most central banks estimate term structures at a daily frequency. With the exception of the United Kingdom, central banks which use Nelson and Siegel-related models report estimated parameters to the BIS Data Bank. Moreover, Germany and Switzerland provide both estimated parameters and spot rates from the estimated term structures. Canada, the United States and Japan, which use the smoothing splines approach, provide a selection of spot rates. With the exception of France, Italy and Spain, the central banks report their data in percentage notation. Specific information on the retrieval of term structure of interest rates data from the BIS Data Bank can be obtained from BIS Data Bank Services.
Treasury Yield Curve Methodology

This description was revised and updated on February 9, 2006.

The Treasury’s yield curve is derived using a quasi-cubic hermite spline function. Our inputs are the COB bid yields for the on-the-run securities. Because the on-the-run securities typically trade close to par, those securities are designated as the knot points in the quasi-cubic hermite spline algorithm and the resulting yield curve is considered a par curve. However, Treasury reserves the option to input additional bid yields if there is no on-the-run security available for a given maturity range that we deem necessary for deriving a good fit for the quasi-cubic hermite spline curve. In particular, we are currently using inputs that are not on-the-run securities. These are two composite rates in the 20-year range reflecting market yields available in that time tranche. Previously, a rolled-down 10-year note with a remaining maturity nearest to 7 years was also used as an additional input. That input was discontinued on May 26, 2005.

More specifically, the current inputs are the most recently auctioned 4-, 13- and 26-week bills, plus the most recently auctioned 2-, 3-, 5-, and 10-year notes and the most recently auctioned 30-year bond, plus the off-the-runs in the 20-year maturity range. The quotes for these securities are obtained at or near the 3:30 PM close each trading day. The long-term composite inputs are the arithmetic averages of the bid yields on bonds with 18 - 22 years remaining to maturity; and those with 20 years and over remaining to maturity, each inputted at their average maturity. The inputs for the three bills are their bond equivalent yields.

To reduce volatility in the 1-year CMT rate, and due to the fact that there is no on-the-run issue between 6-months and 2-years, Treasury uses an additional input to insure that the 1-year rate is consistent with on-the-run yields on either side of it’s maturity range. Thus, Treasury interpolates between the secondary bond equivalent yield on the most recently auctioned 26-week bill and the secondary market yield on the most recently auctioned 2-year note and inputs the resulting yield as an additional knot point for the derivation of the daily Treasury Yield Curve. The result of this step is that the 1-year CMT is generally the same as the interpolated rate. Treasury has used this interpolated methodology since August 6, 2004.

Treasury does not provide the computer formulation of our quasi-cubic hermite spline yield curve derivation program. However, we have found that most researchers have been able to reasonably match our results using alternative cubic spline formulas.

Treasury reviews its yield curve derivation methodology on a regular basis and reserves the right to modify, adjust or improve the methodology at its option. If Treasury determines that the methodology needs to be changed or updated, Treasury will revise the above description to reflect such changes.

Yield curve rates are normally available at Treasury’s interest rate web sites as early as 5:00 PM and usually no later than 6:00 PM each trading day.

Office of Debt Management
Department of the Treasury
Interpolation

Michael Carter

Frequently we are given some data regarding some relationship, and we want to infer the relationship at new points. Common examples in computational finance include estimating volatility using the volatility smile estimated from traded options and estimating spot interest rates for various maturities from the market prices of traded bonds. Unless we are blessed with instruments matching exactly the parameters we are interested in (same maturity, strike etc.), it is necessary to estimate the relationship using the available data. This process is called interpolation.

Interpolation is less obviously an integral part of numerical methods for integration, solving systems of non-linear equations, optimization and dynamic programming.

Polynomial interpolation

Introduction

The following table lists the volatility of the S&P500 index implied by June 2005 calls as quoted on the market on 11 November 2004.

<table>
<thead>
<tr>
<th>Strike</th>
<th>Volatility</th>
</tr>
</thead>
<tbody>
<tr>
<td>1005</td>
<td>0.213</td>
</tr>
<tr>
<td>1050</td>
<td>0.194</td>
</tr>
<tr>
<td>1100</td>
<td>0.173</td>
</tr>
<tr>
<td>1150</td>
<td>0.156</td>
</tr>
<tr>
<td>1200</td>
<td>0.142</td>
</tr>
<tr>
<td>1250</td>
<td>0.13</td>
</tr>
<tr>
<td>1300</td>
<td>0.121</td>
</tr>
<tr>
<td>1350</td>
<td>0.118</td>
</tr>
<tr>
<td>1400</td>
<td>0.117</td>
</tr>
<tr>
<td>1500</td>
<td>0.12</td>
</tr>
</tbody>
</table>

These data illustrate an archetypical volatility smile. (Actually, more data points were available. I have selected a limited range of strikes for illustration.)
Suppose we wish to price an option with a strike of 1275. The simplest estimate of the appropriate volatility would be to take the volatility of the nearest traded option. In this case, there are two: 1250 and 1300 with volatilities of 13% and 12.1% respectively. We could reasonably take either, but a better alternative would be to take the average, namely 12.55%.

Taking the average is a specific case of linear interpolation, in which we estimate the value at a given point by taking a weighted average of the neighbouring points, with the weights proportional to the respective distances of the given point to neighbouring points. Specifically, given two data \((x_1, y_1)\) and \((x_2, y_2)\), we estimate the value \(y\) at \(x\) as the weighted average of \(y_1\) and \(y_2\):

\[
y = \frac{x_2 - x}{x_2 - x_1} y_1 + \frac{x - x_1}{x_2 - x_1} y_2
\]

The equation of the linear function interpolating the points \((x_1, y_1)\) and \((x_2, y_2)\) is

\[
L_1(x) = \frac{x_2 - x}{x_2 - x_1} y_1 + \frac{x - x_1}{x_2 - x_1} y_2
\]

It is clear that \(L_1(x_1) = y_1\) and \(L_2(x_2) = y_2\).

Geometrically, linear interpolants lie along the straight line joining the two data points. Since there is a unique straight line joining any two distinct points, there is a unique linear interpolant at any arbitrary \(x\) given two data points \((x_1, y_1)\) and \((x_2, y_2)\).

It is evident that the volatility smile displays a degree of curvature, which a linear interpolation cannot capture. Curvature can be captured by a higher degree polynomial. Specifically, if we utilize a third point, we can define a quadratic function which passes through the three points. The following form defines a quadratic function through \((x_1, y_1)\) and \((x_2, y_2)\):

\[
L_2(x) = L_1(x) + a_2 (x - x_1) (x - x_2)
\]

By construction, \(L_2(x_1) = L_1(x_1) = y_1\) and \(L_2(x_2) = L_1(x_2) = y_2\). We can choose \(a_2\) so that it also interpolates at \(x_3\), that is such that \(L_2(x_3) = y_3\). Setting \(x = x_3\)

\[
L_2(x_3) = y_3 = L_1(x_3) + a_2 (x_3 - x_1) (x_3 - x_2)
\]

and solving for \(a_2\) gives
\[ a_2 = \frac{y_3 - L_2(x_3)}{(x_3 - x_1)(x_3 - x_2)} \]

Therefore \( L_2 \) is the unique quadratic function passing through \((x_1, y_1), (x_2, y_2), \) and \((x_3, y_3).\)

Similarly, we can make use of a fourth point \((x_4, y_4)\) to define a cubic function as follows

\[ L_3(x) = L_2(x) + a_3 (x - x_1) (x - x_2) (x - x_3) \]

By construction, this interpolates at \( x_1, x_2 \) and \( x_3. \) We can choose \( a_3 \) so that it eliminates the error in the quadratic function \( L_2 \) at \( x_4. \) That is, we choose

\[ a_3 = \frac{y_4 - L_2(x_4)}{(x_4 - x_1)(x_4 - x_2)(x_4 - x_3)} \]

With this choice of \( a_3, \) \( L_3(x) \) is the unique cubic function that interpolates the four points \((x_1, y_1), (x_2, y_2), (x_3, y_3), \) and \((x_4, y_4).\)

Proceeding in this way, we can construct polynomials of higher orders by adding successive points. In other words, we have outlined an algorithm for constructing interpolating polynomials of any order. This is known as Newton’s method. This algorithm lends itself to a recursive implementation, as in the following function. It duplicates the in-built function \texttt{InterpolatingPolynomial}.

\begin{verbatim}
InterpPoly1[{{X_, Y_}}, X_] := Y
InterpPoly1[data_, x_] := Module[{X, Y, newtonPoly},
  {X, Y} = Transpose[data];
  newtonPoly[xx_] := Times @@ (xx - Most[X]);
  InterpPoly1[Most[data], x] +
  Last[Y] - (InterpPoly1[Most[data], x] /. x -> Last[X])
  newtonPoly[Last@X]
  ]

In general, using more information improves the interpolation up to a point. Using all the data to construct a single high-degree polynomial can be a very bad idea, as we will now illustrate. There is a unique 9 degree polynomial that passes through all 10 data points of the volatility smile. Its equation is

\[ v = -230\ 103 + 1728.12\ K - 5.75956\ K^2 + 0.0111802\ K^3 - 0.0000139294\ K^4 + 1.15507\times10^{-8}\ K^5 - 6.37464\times10^{-12}\ K^6 + 2.25767\times10^{-15}\ K^7 - 4.65586\times10^{-19}\ K^8 + 4.25945\times10^{-23}\ K^9 \]

But forcing it to fit all the points requires a spurious oscillation in the last interval. Another example is given by the Gamma function discussed below.
If we need to interpolate at a single point, then it is probably appropriate to utilize the three or four nearest points to construct a quadratic or cubic interpolant. If repeated interpolation over the whole range is required, then it is perhaps more appropriate to construct an interpolant piece by piece, as is done with splines. As usual, the advice of Press et. al. (1992) is succinct and apposite. "Unless there is solid evidence that the interpolating function is close in form to the true function $f$, it is a good idea to be cautious about high-order interpolation. We enthusiastically endorse interpolations with 3 or 4 points, we are perhaps tolerant of 5 or 6; but we rarely go higher than that unless there is quite rigorous monitoring of the estimated errors." (p. 107)

More efficient algorithms are available both to compute a polynomial interpolant, and to compute a single interpolated value at a specific point. We shall discuss these in turn. But first we present another example that will we use in illustrating the algorithms.

**Example: Bessel function**

Suppose we want to interpolate a value for $x = 1.5$, given the following tabulated data.

<table>
<thead>
<tr>
<th>$x$</th>
<th>$y$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.</td>
<td>0.765198</td>
</tr>
<tr>
<td>1.3</td>
<td>0.620086</td>
</tr>
<tr>
<td>1.6</td>
<td>0.455402</td>
</tr>
<tr>
<td>1.9</td>
<td>0.281819</td>
</tr>
<tr>
<td>2.2</td>
<td>0.110362</td>
</tr>
</tbody>
</table>

The data are tabulated values of the zero-order Bessel function, and the true value at $x = 1.5$ is 0.511828. The following table details the interpolated values and absolute interpolation error (times $10^6$) for various orders of interpolation.

<table>
<thead>
<tr>
<th></th>
<th>Value</th>
<th>Error ($\times 10^6$)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Linear</td>
<td>0.510297</td>
<td>1530.9</td>
</tr>
<tr>
<td>Quadratic</td>
<td>0.511286</td>
<td>542.032</td>
</tr>
<tr>
<td>Cubic</td>
<td>0.511813</td>
<td>15.0026</td>
</tr>
<tr>
<td>Quartic</td>
<td>0.51182</td>
<td>7.70291</td>
</tr>
</tbody>
</table>

In general, using the closest points gives the most accurate interpolation, although this is not necessarily the case. A cubic interpolation at $x = 1.5$ using the points 1.3, 1.6, 1.9, and 2.2 is more accurate than one using
the closest points 1.0, 1.3, 1.6 and 1.9, and indeed is more accurate than a quartic approximation using all four points.

\[
\begin{array}{|c|c|}
\hline
\text{Second Cubic} & \text{Value} \quad \text{Error (x 10^6)} \\
\hline
& 0.51183 \quad 2.51661 \\
\hline
\end{array}
\]

In practice, of course, we do not know the true value otherwise there would be no necessity for interpolation. All we can do is look for convergence of successive approximations.

- **Computation**

  - **Monomials**

    We have seen that for any set of set of \( n + 1 \) data points \((x_0, y_0), (x_1, y_1), \ldots, (x_n, y_n)\), there is a unique \( n \)-degree polynomial \( P_n(x) \) that interpolates the data, that is for which

    \[
P_n(x_i) = y_i, \quad i = 0, 1, \ldots, n
    \]

    An arbitrary \( n \)-degree polynomial can be written as

    \[
P_n(x) = a_0 + a_1 x + a_2 x^2 + \ldots + a_n x^n
    \]

    and the data points provide system of \( n + 1 \) linear equations in the \( n + 1 \) unknown coefficients \( a_0, a_1, \ldots, a_n \).

    \[
P_n(x_0) = a_0 + a_1 x_0 + a_2 x_0^2 + \ldots + a_n x_0^n = y_0
    \]

    \[
P_n(x_1) = a_0 + a_1 x_1 + a_2 x_1^2 + \ldots + a_n x_1^n = y_1
    \]

    \[
    \vdots
    \]

    \[
P_n(x_n) = a_0 + a_1 x_n + a_2 x_n^2 + \ldots + a_n x_n^n = y_n
    \]

    Provided that the evaluation points \( x_0, x_1, \ldots, x_n \) are distinct, these \( n + 1 \) linear equations are independent. In principle, they can be solved to yield the coefficients \( a_0, a_1, \ldots, a_n \) of the unique interpolating polynomial. In practice, this is a very bad idea, since the coefficient matrix

    \[
    V = \begin{pmatrix}
    1 & x_0 & x_0^2 & \ldots & x_0^n \\
    1 & x_1 & x_1^2 & \ldots & x_1^n \\
    \vdots & \vdots & \vdots & \ddots & \vdots \\
    1 & x_n & x_n^2 & \ldots & x_n^n
    \end{pmatrix}
    \]

    tends to be ill-conditioned for large \( n \). Incidentally, the coefficient matrix is known as a *Vandermonde* matrix.

  - **Lagrange polynomials**

    A more promising approach is to construct a set of \( n + 1 \) polynomials \( L_0(x), L_1(x), \ldots, L_n(x) \) of degree \( n \) with the property that

    \[
    L_i(x_j) = \begin{cases}
    1, & \text{if } i = j \\
    0, & \text{otherwise}
    \end{cases}
    \]

    Then a linear combination of these polynomials with weights \( y_i \) is a polynomial of degree \( n \) that interpolates the data points. That is
\[ P_n(x) = y_0 L_0(x) + y_1 L_1(x) + \ldots + y_n L_n(x) \]  

(2)
satisfies

\[ P_n(x_i) = y_i, \ i = 0, 1, \ldots, n \]

How can we construct such a set of basic polynomials? Simple. The function \((x - x_1)(x - x_2)\ldots(x - x_n)\) is an \(n\)-degree polynomial that takes the value 0 at each of the points \(x_1, x_2, \ldots, x_n\) and takes a non-zero value \((x_0 - x_1)(x_0 - x_2)\ldots(x_0 - x_n)\) at \(x_0\). Dividing by this value \((x_0 - x_1)(x_0 - x_2)\ldots(x_0 - x_n)\) gives the desired basic polynomial

\[ L_0(x) = \frac{(x - x_1)(x - x_2)\ldots(x - x_n)}{(x_0 - x_1)(x_0 - x_2)\ldots(x_0 - x_n)} \]  

(3)

with the required property

\[ L_0(x_j) = \begin{cases} 
1, & \text{if } j = 0 \\
0, & \text{otherwise} 
\end{cases} \]

The remaining polynomials \(L_1(x), L_2(x), \ldots, L_n(x)\) are constructed analogously. These basic polynomials are known as the Lagrange polynomials. Note that they depend only on the \(x\)-values. So a practical approach to interpolation given a data set \([(x_0, y_0), (x_1, y_1), \ldots, (x_n, y_n)]\) is to compute the Lagrange polynomials using (3) and then to compute the interpolating polynomial using (2). "It is not terribly wrong to implement the Lagrange formula straightforwardly, but it is not terribly right either. The resulting algorithm gives no error estimate, and it is also somewhat awkward to program. A much better algorithm (for constructing the same, unique, interpolating polynomial is Neville's algorithm." (Press et. al. 1982:108). Furthermore, the polynomial expressed in the form (2) is expensive to evaluate and difficult to differentiate.

- **Neville's method**

If all that is required is interpolation at a single point, it is more efficient to compute the interpolated value directly rather than first computing the interpolating polynomial and then evaluating it at the required point. **Neville’s method** provides a very straightforward algorithm for computing polynomial interpolations using successive linear interpolations. It is detailed in the accompanying workbook *interpolation.xls*. The algorithm can be very elegantly expressed recursively. Press et. al. (1982) provide an iterative implementation in C.

Given a set of \(n + 1\) data points \([(x_0, y_0), (x_1, y_1), \ldots, (x_n, y_n)]\), the interpolated value based upon an \(n\)-degree polynomial is a weighted average of the values provided by the \(n - 1\) degree polynomials interpolating respectively the first and the last \(n\) data points.

\[ \text{Interp}(x; [(x_0, y_0), (x_1, y_1), \ldots, (x_n, y_n)]) = \frac{x - x_0}{x_n - x_0} \text{Interp}(x, [(x_1, y_1), \ldots, (x_n, y_n)]) + \frac{x_n - x}{x_n - x_0} \text{Interp}(x, [(x_0, y_0), (x_1, y_1), \ldots, (x_{n-1}, y_{n-1})]) \]

The following table lists the successive orders of interpolation of the Bessel function at \(x = 1.5\). Each successive row is a weighted average of the entries in the preceding row. Note that this is the transpose of the format in *Interpolation.xls*. 


\textbf{Newton polynomials and divided differences}

The set \( \mathcal{P}_n \) of all polynomials of degree \( \leq n \) is a linear or vector space of dimension \( n + 1 \). The set of monomial functions \( 1, x, x^2, \ldots, x^n \) constitutes a basis for this space. The set of Lagrange polynomials defined in (3) provides another basis for \( \mathcal{P}_n \). Equations (1) and (2) represent the same polynomial using different bases. The Newton polynomials \( N_0(x), N_1(x), \ldots, N_n(x) \) defined by

\[
N_0(x) = 1, \quad N_1(x) = x - x_0, \quad N_2(x) = (x - x_0)(x - x_1), \\
N_3 = (x - x_0)(x - x_1)(x - x_2), \ldots, \quad N_n(x) = (x - x_0)(x - x_1) \ldots (x - x_{n-1})
\]

provide yet another basis for \( \mathcal{P}_n \). Note that the Newton polynomials can be defined recursively

\[
\begin{align*}
N_0(x) &= 1 \\
N_i(x) &= (x - x_{i-1}) N_{i-1}(x), \quad i = 1, 2, \ldots, n
\end{align*}
\]

which facilitates computation.

Any \( n \)-degree polynomial \( P_n(x) \) can be written as a linear combination of this basis

\[
P_n(x) = b_0 N_0(x) + b_1 N_1(x) + b_2 N_2(x) + \ldots + b_n N_n(x)
\]

(4)

The actual coefficients \( b_0, b_1, \ldots, b_n \) to specify that unique polynomial which interpolates a given set of distinct data points \( \{(x_0, y_0), (x_1, y_1), \ldots, (x_n, y_n)\} \) can be computed by the method of divided differences.

Given a set of tabulated data \( \{(x_0, y_0), (x_1, y_1), \ldots, (x_n, y_n)\} \), their divided differences are defined recursively

\[
\text{DD}(\{(x_0, y_0), (x_1, y_1), \ldots, (x_n, y_n)\}) =
\text{DD}(\{(x_1, y_1), \ldots, (x_n, y_n)\}) - \text{DD}(\{(x_0, y_0), (x_1, y_1), \ldots, (x_{n-1}, y_{n-1})\})
\]

\[
x_n = y_n
\]

with \( \text{DD}(\{(x_i, y_i)\}) = y_i \).
The successive divided differences provide the coefficients $b_0$, $b_1$, ..., $b_n$ required to represent the interpolating polynomial using the Newton basis. The divided differences for the Bessel function data are

\[
\text{dividedDifferences = Table}[DD[H] & /@ Partition[data, i + 1, 1],
{i, 0, Length[data] - 1}] // TableForm
\]

\[
\begin{array}{cccccc}
0.765198 & 0.620086 & 0.455402 & 0.281819 & 0.110362 \\
-0.483706 & -0.548946 & -0.578612 & -0.571521 \\
-0.108734 & -0.049443 & 0.011814 \\
0.0658786 & 0.0680685 \\
0.00182492
\end{array}
\]

The coefficients $b_0$, $b_1$, ..., $b_4$ are listed in the first column of the previous table, namely

\[
\text{coefficients = First} /\!\!/ \text{dividedDifferences}
\]

\[
\{0.765198, -0.483706, -0.108734, 0.0658786, 0.00182492\}
\]

Note that each row of the table is computed from the previous row, so that the last entry depends upon all the entries in the table. There is no redundancy. In this example, the \(x\) values are 1.0, 1.3, 1.6, 1.9 and 2.2. The Newton basis is \([1, x - 1, (x - 1)(x - 1.3), (x - 1)(x - 1.3)(x - 1.6), (x - 1)(x - 1.3)(x - 1.6)(x - 1.9)]\) and the interpolating quartic polynomial is

\[
p_4(x) = b_0 + b_1(x - 1) + b_2(x - 1)(x - 1.3) + \\
b_3(x - 1)(x - 1.3)(x - 1.6) + b_4(x - 1)(x - 1.3)(x - 1.6)(x - 1)
\]

\[
p_4[x_] := \text{coefficients}.\{(1, x - 1, (x - 1)(x - 1.3), (x - 1)(x - 1.3)(x - 1.6), (x - 1)(x - 1.3)(x - 1.6)(x - 1.9))
\]

\[
p_4[x]
\]

\[
0.765198 - 0.483706 (-1 + x) - 0.108734 (-1.3 + x) (-1 + x) + \\
0.0658786 (-1.6 + x) (-1.3 + x) (-1 + x) + \\
0.00182492 (-1.9 + x) (-1.6 + x) (-1.3 + x) (-1 + x)
\]

We can confirm that this is the correct polynomial by verifying that it interpolates the data points.

\[
\text{TableForm}[[H[[1]], H[[2]], p_4[H[[1]]]] & /@ data, \\
TableHeadings \rightarrow \{\text{None}, \{"X", "Data", "Interpolated"}\}]
\]

\[
\begin{array}{ccc}
X & \text{Data} & \text{Interpolated} \\
1. & 0.765198 & 0.765198 \\
1.3 & 0.620086 & 0.620086 \\
1.6 & 0.455402 & 0.455402 \\
1.9 & 0.281819 & 0.281819 \\
2.2 & 0.110362 & 0.110362
\end{array}
\]

Interpolation with divided differences is usually attributed to Newton. However, quadratic interpolation via divided difference for computing sines is found in an astronomical treatise by the Indian mathematician Brahmagupta published 1000 years earlier (Joseph 2000).
- Illustrating the basis functions

We can get an impression of the efficiency of alternative methods for computing interpolating polynomials by illustrating the different bases. The first plot depicts the monomial basis \(1, x, x^2, x^3, x^4\) on the interval \([0,1]\).

![Monomial Basis](image1)

The next shows the Lagrange basis for five equally-spaced points on the interval \([0,1]\).

```math
\text{LagrangeBasis}[\text{vals}_-, \text{var}_-] := \text{Module}[\{n = \text{Length}[\text{vals}]\},
\quad \text{Times} @@ \text{Subsets}[\text{var} - \text{vals}, \{n - 1\}] / (\text{Times} @@ \{\text{Take}[\text{vals}, \{\#\}][[1]] - \text{Drop}[\text{vals}, \{\#\}]\} & / @ \text{Range}[n] // \text{Flatten})
\text{Plot}[\text{Evaluate}[\text{LagrangeBasis}[\text{Range}[0, 1, 1/4], x]], \{x, 0, 1\},
\quad \text{PlotLabel} \rightarrow \text{"Lagrange Basis"},
\quad \text{PlotStyle} \rightarrow \{\text{Blue}, \text{Red}, \text{Green}, \text{RGBColor}[1, 1, 0], \text{RGBColor}[0, 1, 1]\},
\quad \text{Ticks} \rightarrow \{\text{Automatic}, \{0, 0.5, 1\}\}]
```

![Lagrange Basis](image2)

The Newton basis functions are

\[1, x - x_0, (x - x_0)(x - x_1), \ldots, (x - x_0)(x - x_1) \ldots (x - x_{n-1})\]
The next shows the Newton basis for five equally-spaced points on the interval [0,2].

**Newton Basis**

---

### Piecewise polynomial interpolation

We have seen that high-order polynomial interpolation can introduce spurious oscillations into the results. An obvious alternative strategy is to apply different interpolating functions to different segments of the domain.

Why do we use polynomials?

- Weierstrass approximation theorem - any continuous function can be approximated arbitrarily closely by a polynomial.
- Easy to compute with. In particular, easy to differentiate and integrate.

---

#### Piecewise linear interpolation ("connect the dots")

Perhaps the simplest practical method of interpolation is piecewise linear interpolation, whereby any desired point is linearly interpolated from its two adjacent neighbours. Given \( n + 1 \) data points \((x_0, y_0), (x_1, y_1), \ldots, (x_n, y_n)\) with \(x_0 < x_1 < \ldots < x_n\), piecewise linear interpolation is defined by

\[
p(x) = \alpha_j y_j + (1 - \alpha_j) y_{j+1} \text{ for } x_j \leq x \leq x_{j+1}
\]

where

\[
\alpha_j = \frac{x_{j+1} - x}{x_{j+1} - x_j}
\]

Piecewise linear interpolation has the desirable properties that it preserves mononicity and convexity. The problem with piecewise linear interpolation is that the resulting curve is jagged, with zero curvature between the data points and abrupt changes in slope at those points. Mathematically speaking, a piecewise linear interpolant has zero second derivative between the data points, and undefined or indefinite second derivative at those points. Smooth interpolation requires using higher order polynomials in each segment.
### Piecewise cubic interpolation

Given two data points \((x_1, y_1)\) and \((x_2, y_2)\), there are many cubic functions that pass through both points. However, there is a unique cubic function that interpolates both points and has a specified slope at each end. That is, there is a unique cubic function \(p(x)\) satisfying the conditions

\[
p(x_1) = y_1, \quad p(x_2) = y_2, \quad p'(x_1) = s_1, \quad p'(x_2) = s_2
\]

The interpolating cubic has the formula

\[
p(x) = b_0 + b_1(x-x_1) + b_2(x-x_1)^2 + b_3(x-x_1)^3 (x-x_2)
\]

with

\[
b_0 = y_1, \quad b_1 = s_1, \quad b_2 = \frac{m-s_1}{x_2-x_1}, \quad b_3 = -\frac{m-s}{(x_2-x_1)^2}
\]

where \(m = \frac{y_2-y_1}{x_2-x_1}\) is the slope of the line joining the two data points, and \(s = \frac{s_1+s_2}{2}\) is the average of the slopes at the two endpoints. The coefficients can be computed using an extended version of the divided differences table, but this seems unnecessary, since the coefficients can easily be computed directly using the above formula.

We can verify this representation directly by substitution and evaluation. Differentiating, we have

\[
p'(x) = b_1 + 2b_2(x-x_1) + 2b_3(x-x_1)(x-x_2) + 2b_3(x-x_1)^2
\]

At \(x_1\), we clearly have \(p(x_1) = y_1\) and \(p'(x_1) = b_1 = s_1\). It is a little more tedious to manually verify \(p(x_2) = y_2\) and \(p'(x_2) = s_2\), but straightforward for Mathematica.

```mathematica
p[x_] := b0 + b1 (x - x1) + b2 (x - x1)^2 + b3 (x - x1)^3 (x - x2)
{p[x1], p'[x1], p[x2], p'[x2]} /. {b0 -> y1, b1 -> s1,
  b2 -> m - s1, b3 -> -2 (m - (s1 + s2)/2)/(x2 - x1)^2} /. m -> Y2 - Y1
// Simplify
{y1, s1, y2, s2}
```

For computation, it is more appropriate to express the polynomial in terms of the given data

\[
p(x) = \alpha y_1 + (1-\alpha) y_2 + \beta s_1 + \delta s_2
\]

where

\[
\alpha = \frac{(x_2-x)^2(2(x-x_1)+h)}{h^3}, \quad 1 - \alpha = -\frac{(x-x_1)^2(2(x_2-x)+h)}{h^3}, \\
\beta = \frac{(x-x_1)(x_2-x)^2}{h^2}, \quad \delta = \frac{(x-x_1)^2(x_2-x)}{h^2}, \quad h = x_2 - x_1
\]

Observe that
when \( x = x_1, \alpha = 1 \) while \( \beta = \delta = 0 \) so that \( p(x_1) = y_1 \)

- when \( x = x_2, \alpha = 0 \) while \( \beta = \delta = 0 \) so that \( p(x_2) = y_2 \)

Similarly, we can show that

- \( p'(x_1) = s_1 \) and \( p'(x_2) = s_2 \)

Therefore, \( p \) correctly interpolates the data.

\[
\begin{align*}
p[xx_] &: = \alpha y_1 + (1 - \alpha) y_2 + \beta s_1 + \delta s_2 / . \{ \\
& \alpha \to \frac{(x_2 - xx)^2}{h^3} (2 (xx - x_1) + h), \\
& \beta \to \frac{(xx - x_1) (x_2 - xx)^2}{h^2}, \delta \to - \frac{(xx - x_1)^2 (x_2 - xx)}{h^2} \} / . h \to x_2 - x_1
\end{align*}
\]

\( \{ y_1, y_2, s_1, s_2 \} \)

We can interpolate a set of \( n + 1 \) data points \((x_0, y_0), (x_1, y_1), \ldots, (x_n, y_n)\) with \( x_0 < x_1 < \ldots < x_n \) by piecing together cubic interpolants between successive pairs, analogous to piecewise linear interpolation. Specifically, the piecewise cubic approximation is

\[
p(x) = \alpha_j y_j + (1 - \alpha_j) y_{j+1} + \beta_j s_j + \delta_j s_{j+1} \quad \text{for } x_j \leq x \leq x_{j+1} \quad (7)
\]

\[
\alpha_j = \frac{(x_{j+1} - x)^2 (2(x - x_j) + h_j)}{h_j^3}, \beta_j = \frac{(x - x_j)(x_{j+1} - x)^2}{h_j^2}, \delta_j = \frac{(x - x_j)^2 (x_{j+1} - x)}{h_j^2}, h_j = x_{j+1} - x_j \quad (8)
\]

Note that the resulting interpolant is \( C^1 \), that is differentiable with continuous first derivative. Given the formulae, all that is required is to specify the slopes \( s_i, i = 0, 1, \ldots, n \) at the interpolation points, which are conventionally terms knots. A number of suggestions have been made.

### Cubic Hermite interpolation

Where the interpolation is approximating a function the value of derivatives of which are known at the knots, these can be used to specify the slopes. In other words, if the data comprises \( n + 1 \) triples \((x_0, y_0, y'_0), (x_1, y_1, y'_1), \ldots, (x_n, y_n, y'_n)\), an appropriate piecewise cubic interpolant is given by (7) and (8) with \( s_i = y'_i, i = 0, 1, \ldots, n \). This is known as Hermite interpolation.

### Cubic Bessel interpolation

Set the slope \( s_i \) at \( x_i \) to be the slope of the unique cubic that interpolates \( (x_{i-1}, y_{i-1}), (x_i, y_i), (x_{i+1}, y_{i+1}) \). Note that the slopes at the external knots have to be determined in some other fashion.

### Akima's interpolation

Akima suggests specific formula to determine the slopes to as to produce a “pleasing” curve.

The U. S. Treasury uses a quasi-cubic hermite spline function for estimating the zero-coupon yield curve in
the United States. What exactly this means is unclear, as it does not disclose the precise methodology. There is a suggestion that it utilizes the Akima methodology.

- **Cubic spline interpolation**

We have already noted that the piecewise cubic function specified by (7) and (8) is automatically $C^1$ by construction. In cubic spline interpolation, the slopes $s_i$ at the knots are chosen in order to make the resulting function $C^2$, that is so that it has a continuous second derivative as well.

The segments

$$
p_{j-1}(x) = \alpha_{j-1} y_{j-1} + (1 - \alpha_{j-1}) y_j + \beta_{j-1} s_{j-1} + \delta_{j-1} s_j \quad \text{and} \quad p_j(x) = \alpha_j y_j + (1 - \alpha_j) y_{j+1} + \beta_j s_j + \delta_j s_{j+1}
$$

meet at $x_j$. Their second derivatives evaluated at $x_j$ are

$$
p_{j-1}''(x) = 2 \frac{s_{j-1} + 2 s_j}{h_{j-1}^2} - 6 \frac{y_j - y_{j-1}}{h_{j-1}^2} \quad \text{and} \quad p_j''(x) = -2 \frac{2 s_j + s_{j+1}}{h_j} + 6 \frac{y_{j+1} - y_j}{h_j^2}
$$

Equating the second derivatives at the internal knots $x_1$, $x_2$, ..., $x_{n-1}$ gives us a system of $n-1$ linear equations in the $n+1$ unknown slopes $s_0$, $s_1$, ..., $s_n$.

$$
\frac{1}{h_{j-1}} s_{j-1} + 2 \left( \frac{1}{h_{j-1}} + \frac{1}{h_j} \right) s_j + \frac{1}{h_j} s_{j+1} = 3 \left( \frac{y_j - y_{j-1}}{h_{j-1}^2} + \frac{y_{j+1} - y_j}{h_j^2} \right), \quad j = 1, 2, ..., n-1
$$

This simplifies when the knots are equally spaced ($h_{j-1} = h_j = h$) to

$$
s_{j-1} + 4 s_j + s_{j+1} = 3 \frac{y_{j+1} - y_{j-1}}{h}, \quad j = 1, 2, ..., n-1
$$

Provided the slopes $s_0$ and $s_n$ at the external knots $x_0$ and $x_n$ are determined in some way, (9) and (10) are tridiagonal systems of $n-2$ equations in $n-2$ unknowns with a strictly dominant diagonal. This system has a unique solution, which can be computed easily by Gaussian elimination. Cubic spline interpolation reduces to solving a system of linear equations.

- **Boundary conditions**

The requirement of continuity of the second derivative is sufficient to determine the slopes at each of the internal knots. It remains to determine the slopes at the external knots or boundaries $x_0$ and $x_n$. Similar specifications are required by Bessel interpolation and Akima’s method. The following boundary conditions are found in practice.

- Specify the slopes $s_0$ and $s_n$.
- Set $s_0$ and $s_n$ so that $p''(x_0) = p''(x_n) = 0$. This is the so-called natural spline.
- Choose the slopes $s_0$ and $s_n$ to match the slopes of a cubic interpolating the points closest to the boundary point.
- Enforce a not-a-knot condition at the extreme internal knots. We choose a single polynomial to interpo-
late the boundary point and first two interior knots points plus the slope at the second knot. That is, between $x_0$ and $x_2$

$$p(x) = b_0 + b_1(x - x_1) + b_2(x - x_1)^2 + b_3(x - x_1)^2 (x - x_2)$$

with

$$p(x_0) = y_0, \quad p(x_1) = y_1, \quad p(x_2) = y_2, \quad p_3(x_2) = s_2$$

Note that we are free to use different conditions at the two boundaries.

For natural spline in particular, it is common practice to formulate equations analogous to (9) and (10) with the second derivatives as unknowns. de Boor (2001) writes: "Apart from its positive sounding name, natural spline interpolation has little to recommend it from an approximation-theoretic point of view... If one knows nothing about the end point derivatives, then one should try the not-a-knot condition."

- **Shape-preserving quadratic splines**

  Piecewise linear interpolation preserves monotonicity and convexity of the data, but the interpolating function is not differentiable. Cubic spline interpolation is (twice) differentiable, but it does not preserve monotonicity and convexity. For example, the interpolant may have decreasing segments although the data is strictly increasing. With quadratic splines, it is possible to have both differentiability and shape-preservation.

  Given two data points $(x_1, y_1)$ and $(x_2, y_2)$ and corresponding slopes $s_1$, $s_2$, and some interior point $\xi \in (x_1, x_2)$, there is a unique quadratic spline $p(x)$ with knot at $\xi$ that matches the data, that is with $p(x_1) = y_1$, $p(x_2) = y_2$, $p'(x_1) = s_1$ and $p'(x_2) = s_2$. Further, it is possible to choose the knot $\xi$ in such a way as to preserve monotonicity and convexity.

  Applying this procedure successively, it is possible to interpolate a set of $n+1$ data points $[(x_0, y_0), (x_1, y_1), ..., (x_n, y_n)]$ with a sequence of quadratic splines that are differentiable, and that preserve mononicity and convexity. For details, see Judd 1998: 231-235.

- **Some examples**

  Needs["Splines"]

  The gamma function

  The gamma function extends the factorial function to non-integer and complex arguments, and is defined by

  $$\Gamma(z) = \int_0^\infty t^{z-1} e^{-t} dt$$

  Amongst other places, it is found in certain probability distributions.

  For integer arguments, the gamma function has the value

  $$\Gamma(n) = (n - 1)!$$

  So, an obvious question is how accurately can we compute the values of the gamma function by interpola-
tion. The first 5 integer values of the Gamma function are

\[
\begin{array}{c|c}
 n & \Gamma(n) \\
\hline
 1 & 1 \\
 2 & 1 \\
 3 & 2 \\
 4 & 6 \\
 5 & 24 \\
\end{array}
\]

The quartic interpolating polynomial is

\[
\text{approx5} = \text{InterpolatingPolynomial}[	ext{data}, x] \quad \text{// Expand} \quad \text{// N}
\]

\[
9.0 - 16.5833 x + 11.625 x^2 - 3.41667 x^3 + 0.375 x^4
\]

depicted below. The blue curve shows the true value, and the red curve is the approximation.

Suppose we add another point (6, 6!) and plot over the same range. The approximation, depicted by the green line, deteriorates dramatically.
On the other hand, suppose we subtract the last point and plot over the same range. This appears to improve the approximation almost uniformly over this range. Admittedly, this is a special example, but it does illustrate that more data is not necessarily better when it comes to interpolation.

The following graph compares the "natural" cubic spline with the quartic interpolating polynomial. Neither method dominates.
Akima's example

Here is some pseudo - experimental data on waveform distortion in electronic circuits analysed by Akima (1971).

\[
data = \text{Transpose}[\begin{pmatrix} 0 & 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 & 10 \\ 10 & 10 & 10 & 10 & 10 & 10.5 & 15 & 50 & 60 & 85 \end{pmatrix}];
\]

The 9th order polynomial interpolant is:

The natural cubic spline is
A Bezier spline interpolates only the end points, using the other data points as control points.

A composite Bezier spline alternates data points and control points.
Interpolation in two or more dimensions

Multidimensional interpolation is frequently encountered. In finance, we might want to fit a volatility surface to both strike and time to maturity. In solving for general equilibria of economies and equilibria of games, simplicial approximation involves interpolation in many dimensions.

Assuming a complete grid of points, the simplest interpolation method is bilinear or multilinear interpolation, in which the interpolated value at any point is a weighted average of the values at its surrounding grid points. This is the usual practice in simplicial approximation. This is simply piecewise linear interpolation applied in multiple dimensions, and it suffers from the same drawback of discontinuous derivatives at the boundary.

To achieve greater accuracy, we can decompose a multidimensional interpolation problems into a series of one dimensional problems. For example, to determine a volatility surface, we might interpolate separately a volatility smile at different maturities, and then interpolate the smiles to estimate the volatility at a particular maturity.

There are two methods in common use for achieving greater smoothness. In two dimensions, these are:

- **bicubic interpolation.** In this procedure, gradients and cross-derivatives are specified at each grid point. Then a cubic interpolant can be found that
  - matches the specified derivatives at the grid points
  - has continuous derivative at the boundaries

- **bicubic spline.** Perform one dimensional cubic spline interpolation in each dimension successively.
Approximation and estimation

Interpolation requires precise matching at each data point. Often, we do not require precise matching but seek the best fit to the data from a specific functional form. Estimation of the zero coupon interest rates from traded bond prices provides a good example. There are two approaches in common use:

- **parametric estimation**

  We start with a specific functional form for spot rates, for example (Nelson and Spiegel)

  \[ s(t) = \beta_0 + (\beta_1 + \beta_2) \frac{\tau}{t} (1 - e^{-\frac{t}{\tau}}) - \beta_2 e^{-\frac{t}{\tau}} \]

  The parameters \( \beta_0, \beta_1, \beta_2 \) and \( \tau \) are chosen so that the implied prices for coupon bonds best match market prices on a given day.

- **splines**

  Given arbitrarily chosen knots \( (s_0, s_1, \ldots, s_n) \), a cubic spline is fitted and the knots adjusted until the implied prices for coupon bonds best match market prices on a given day.

Both methods requires solution of an optimization problem.

The first approach is analogous to polynomial interpolation, using a single functional form to fit the entire data set. The second method is a straightforward application of spline techniques. In practice it is found that the splines can have excessive variation, so some additional smoothing criteria are often applied.
The binomial model

*Michael Carter*

A derivative is an asset the value of which depends upon another *underlying* asset. Consider the simplest possible scenario, in which the underlying has two possible future states "up" and "down". The value of the derivative in these two states is \( V_u \) and \( V_d \) respectively.

The current value of the derivative is enforced by the possibility of arbitrage between the derivative and the underlying asset. Consider a portfolio comprising \( x \) shares and short one option.

By choosing \( x \) appropriately, we can make the portfolio risk-free. That is, choosing \( x \) so that

\[
x u S - V_u = x d S - V_d
\]
we have
\[ xS = \frac{V_u - V_d}{u - d} \]

**Exercise**

Suppose \( S = 100, \ u = 1.05, \ d = 0.95, \ V_u = 5 \) and \( V_d = 0 \). Calculate the risk-free hedge. Show that it is risk-free by comparing the value of the portfolio in the two states.

Substituting for \( xS \), the value of the portfolio at time \( T \) in either state is

\[
uxS - V_u = u \left( \frac{V_u - V_d}{u - d} \right) - V_u
\]
\[
= \frac{u V_u - u V_d - u V_u + d V_u}{u - d}
\]
\[
= \frac{d V_u - u V_d}{u - d}
\]

The value of the portfolio at time 0 is

\[
xS - V = \delta (uxS - V_u) = \delta \left( \frac{d V_u - u V_d}{u - d} \right)
\]

where \( \delta \) is the discount factor. Let \( R = 1/\delta \). Solving for \( V \)

\[
V = xS - \delta \left( \frac{d V_u - u V_d}{u - d} \right)
\]
\[
= \frac{V_u - V_d}{u - d} - \frac{1}{R} \left( \frac{d V_u - u V_d}{u - d} \right)
\]
\[
= \frac{1}{R} \left( \frac{R V_u - R V_d - d V_u + u V_d}{u - d} \right)
\]
\[
= \frac{1}{R} \left( \frac{R - d}{u - d} V_u + \frac{u - R}{u - d} V_d \right)
\]

Letting

\[
p = \frac{R - d}{u - d} \quad \text{and} \quad 1 - p = 1 - \frac{R - d}{u - d} = \frac{u - d - R + d}{u - d} = \frac{u - R}{u - d}
\]

we obtain the fundamental option valuation equation

\[
V = \frac{1}{R} (p V_u + (1 - p) V_d)
\]

The value of the option at time 0 is the discounted expected value of the payoff, where the expectation is taken with respect to the synthetic or risk-neutral probabilities (defined above) and discounted at the risk-free rate.
This value is enforced by arbitrage. To see this, suppose that option is selling at a premium above its true value.

\[ V > \frac{1}{R} (pV_u + (1-p)V_d) \]

An arbitrageur can sell \( n \) options and buy \( nx \) shares, borrowing the net cost \( n(xS-V) \). At time \( T \), the portfolio is worth \( n(xuS-V_u) \) in the "up" state and (equally) \( n(xdS-V_d) \) in the "down" state. Repaying the loan plus interest of \( Rn(xS-V) \), the arbitrageur makes a risk-free profit of

\[
\text{profit} = \text{payoff} - \text{loan} \\
= n(xuS-V_u) - Rn(xS-V) \\
= n\left( \frac{dV_u-uV_d}{u-d} \right) - Rn\left( \frac{V_u-V_d}{u-d} - V \right) \\
= nR\left( V - \frac{1}{R} (pV_u + (1-p)V_d) \right)
\]

Conversely, if the option is selling at a discount, a risk-free profit can be made by reversing this transaction, buying options and selling shares.

- **Exercise**

  Suppose \( S = 100, u = 1.05, d = 0.95, V_u = 5, V_d = 0 \) and \( R = 1.01 \). Calculate the true value of the option. Suppose that the option is priced at 3.10. Find a profitable arbitrage.

- **Remarks**

  - \( R \) is the risk-free total return for the period \( T \). It is given either by \( R = 1 + rT \) or \( R = e^{rT} \) where \( r \) is the risk-free (spot) rate for the period \( T \). It is common to use continuous compounding in option evaluation, although discrete compounding is convenient (and appropriate) for the binomial model.

  - The risk-neutral probabilities \( p \) and \( 1-p \) are those probabilities at which the expected growth rate of the underlying asset is equal to the risk-free rate, that is

    \[ puS + (1-p)dS = RS \]

    Solving for \( p \),

    \[
p(u-d)S + dS = RS \\
p(u-d)S = (R-d)S \\
p = \frac{R-d}{u-d}
\]

    In the language of probability, \( p \) makes the discounted asset price a *martingale*. 
Exercise

What condition is required to ensure the existence of this equivalent martingale measure (probability)?

- The current asset price $S$ will depend upon the real probabilities $q$. The expected rate of return

$$\mu = \frac{q u S + (1 - q) d S - S}{S} = q u + (1 - q) d - 1$$

must be sufficient to induce investors to hold the asset.

- The hedge ratio $x$ is equal to delta of the option, the sensitivity of the option price to changes in the price of the underlying

$$x = \frac{V_u - V_d}{(u - d) S} = \frac{\Delta V}{\Delta S}$$

- For a vanilla call option at maturity with a strike price of $K$

$$V_u = \max(u S - K, 0) \quad \text{and} \quad V_d = \max(d S - K, 0)$$

For a vanilla put option at maturity with a strike price of $K$

$$V_u = \max(K - u S, 0) \quad \text{and} \quad V_d = \max(K - d S, 0)$$

For a vanilla European option prior to maturity, $V_u$ and $V_d$ are the discounted expected values of the option in the "up" and "down" states respectively.

For a vanilla American option prior to maturity, $V_u$ and $V_d$ are the maximum of the intrinsic values and discounted expected values of the option in the "up" and "down" states respectively.
A SYNTHESIS OF
BINOMIAL OPTION PRICING MODELS
FOR LOGNORMALLY DISTRIBUTED ASSETS

Don M. Chance

Department of Finance
Louisiana State University
2163 CEBA
Baton Rouge, LA  70803

November 20, 2007

225-578-0372; dchance@lsu.edu. The author thanks Jim Hilliard, Bob Brooks, Tung-Hsiao Yang, Tom Arnold, and Adam Schwartz for comments. A document containing all of the proofs of this paper is available on the author’s web site at http://www.bus.lsu.edu/academics/finance/faculty/dchance/Research/Proofs.pdf.
A SYNTHESIS OF
BINOMIAL OPTION PRICING MODELS
FOR LOGNORMALLY DISTRIBUTED ASSETS

Abstract

The finance literature has revealed no fewer than 11 alternative versions of the binomial option pricing model for pricing options on lognormally distributed assets. These models are derived under a variety of assumptions and in some cases require unnecessary information. This paper provides a review and synthesis of these models, showing their commonalities and differences and demonstrating how 11 diverse models all produce the same result in the limit. Some of the models admit arbitrage with a finite number of time steps and some fail to capture the correct volatility. This paper also examines the convergence properties of each model and finds that none exhibit consistently superior performance over the others. Finally, it demonstrates how a general model that accepts any arbitrage-free risk neutral probability will reproduce the Black-Scholes-Merton model in the limit.
A SYNTHESIS OF
BINOMIAL OPTION PRICING MODELS
FOR LOGNORMALLY DISTRIBUTED ASSETS

Option pricing theory has become one of the most powerful tools in economics and finance. The celebrated Black-Scholes-Merton model not only garnered a Nobel Prize for Scholes and Merton but completely redefined the financial industry. Its sister model, the binomial or two-state model, has also attracted much attention and acclaim, both for its ability to provide a clear illustration of the essential ideas behind option pricing theory with a minimum of mathematics and for its flexibility in accommodating many path-dependent options.

The origins of the binomial model are somewhat unclear. Options folklore has it that around 1975 William Sharpe, later to win a Nobel Prize for his seminal work on the Capital Asset Pricing Model, suggested to Mark Rubinstein that option valuation should be feasible under the assumption that the underlying stock price can change to one of only two possible outcomes. Sharpe subsequently formalized the idea in the first edition of his textbook. Perhaps the best-known and most widely cited original paper on the model is Cox, Ross, and Rubinstein (1979), but almost simultaneously, Rendleman and Bartter (1979) presented the same model in a slightly different manner.

Over the years, there has been an extensive body of research designed to improve the model. An examination of the literature will reveal that the binomial model has appeared in a variety of forms. Anyone attempting to understand the model can become bewildered by the array of formulas that all purport to accomplish the desired result of showing how to value an option and hedge an option position. These formulas have many similarities but notable differences. Another source of some confusion is that some

---

1 Not surprisingly, this story does not appear formally in the options literature but is related by Mark Rubinstein in RiskBooks (2003), p. 581.
presentations use opposite notation.\textsuperscript{4} But more fundamentally, the obvious question is how so many different candidates for the inputs of the binomial model can exist and how each of them can technically be correct.

The objective of this paper is to clear up some of this confusion by synthesizing the different approaches and providing a coherent and unified treatment of the model. We establish a uniform set of symbols applied consistently across all models. Each model is presented with the distinct assumptions used by its creators. Detailed mathematical derivations are omitted but are available on the author’s web site.

Some would contend that it is wasteful to study a model that for European options in the limit equals the Black-Scholes-Merton model. Use of the binomial model, they would argue, serves only a pedagogical purpose. The Black-Scholes-Merton model would always be preferred for actual applications. But it is difficult to consider the binomial model as a method for deriving path-dependent option values without knowing how well it works for the one scenario in which the true continuous limit is known. An unequivocal benchmark is rare in finance.

We show that for options on lognormally distributed assets the literature contains no less than eleven distinct versions of the binomial model, each version characterized by up and down factors and a probability of the up-move. We show that some of the models are improperly specified and can lead to arbitrage profits for a finite number of time steps. Moreover, some of the models do not correctly capture the exogenous volatility. Several papers focus first on fitting the binomial model to the physical process, rather than the risk neutral process, thereby requiring that the expected return on the stock be known, an unnecessary requirement in arbitrage-free pricing. As we show, the translation from the physical to the risk neutral process has produced some misleading results. The paper concludes with an examination of the behavior of the models as the number of time steps is increased.

Our focus is exclusively on models for pricing options on lognormally distributed assets and not on interest rates. Hence, these models can be used for options on stocks, indices, currencies, and possibly commodities. We ignore any cash flows on the underlying, but these can be easily added. The paper begins with a brief overview of the model that serves to establish the notation and terminology.

\textbf{I. Basic Review of the Binomial Model}

\textsuperscript{4}For example, in some papers and books the drift of the stochastic process for the return is sometimes denoted as $\alpha$ while the drift of the log return is denoted as $\mu$. In others the opposite notation is used. Although there is no notational standard in the options literature, the inconsistent use of these symbols is a significant cost to comparing the models.
Let the continuously compounded risk-free rate per annum be \( r \). Consider a risky asset priced at \( S \) that can move up to state \( “+” \) for a value of \( uS \) or down to state \( “-” \) for a value of \( dS \). Let there be a call option expiring in one period with exercise price \( X \). The value of the option in one period is \( cu \) if the \( “+” \) state occurs and \( cd \) if the \( “-” \) state occurs.

**I.a. Deriving the Binomial Model**

Now construct a portfolio consisting of \( \Delta \) units of the asset and \( B \) dollars invested in the risk-free asset. This portfolio replicates the call option if its outcomes are the same in both states, that is,

\[
\begin{align*}
\Delta S_u + B &= cu, \\
\Delta S_d + B &= cd.
\end{align*}
\]

The unknowns are \( B \) and \( \Delta \). Rearranging to isolate \( B \), setting the results equal to each other, and solving for \( B \) gives

\[
\Delta = \frac{c_u - c_d}{S(u - d)}.
\]

Since both values, \( c_u \) and \( c_d \), are known, we then substitute for \( \Delta \) in either equation and solve for \( B \). Then, given knowledge of \( \Delta \), \( S \), and \( B \), we obtain the call option value where

\[
c = \frac{\pi c_u + (1 - \pi) c_d}{\exp(\rho h)}
\]

is the value of the option, and

\[
\pi = \frac{\exp(\rho h) - d}{u - d}
\]

is the risk-neutral probability, sometimes referred to as the pseudo-probability or equivalent martingale probability, with \( h \) as the period length defined as the time to expiration, \( T \), divided by the number of binomial time periods, \( N \). Extension to the multiperiod case follows and leads to the same result that the option value at a node, given the option values at the next possible two nodes, is given by equation (1).

**I.b. Specification of the Binomial Parameters**

At times we will need to work with raw or discrete returns and at others times, we will work with continuous or log returns. Let the concept of return refer to the future price divided by the current price, or technically one plus the rate of return. Let the expected price one period later be \( E(S_1) \) and the expected raw return be \( E(S_1)/S \). The true probability of an up move is \( q \). Thus, the per-period expected raw return is

\[
E\left( \frac{S_1}{S} \right) = qu + (1 - q)d.
\]
The per-period expected log return is

$$E\left( \ln \left( \frac{S_t}{S_0} \right) \right) = q \ln u + (1-q) \ln d. \quad (4)$$

The variance of the raw return is,

$$E\left( \frac{S_t}{S_0} \right)^2 - E\left( \frac{S_t}{S_0} \right)^2 = (u - \alpha)^2 q + (d - \alpha)^2 (1-q)$$

$$= (u - d)^2 q (1-q). \quad (5)$$

The variance of the log return is,

$$E \left[ \ln \left( \frac{S_t}{S_0} \right) \right]^2 - E \left[ \ln \left( \frac{S_t}{S_0} \right) \right]^2 = (\ln u - \mu)^2 q + (\ln d - \mu)^2 (1-q)$$

$$= (\ln (u/d))^2 q (1-q). \quad (6)$$

These parameters describe the actual probability distribution of the stock return, or the physical process. Of course, option valuation requires transformation of the physical process to the risk neutral process. Typically, the user of an option model knows the volatility of the log return as given by the physical process, a value that may have been estimated using historical data or obtained as an implied volatility. In any case, we assume that volatility is exogenous and constant, as is usually assumed in continuous-time option pricing.

II. Fitting the Binomial Model

In early research on the binomial model, several papers examined fitting a binomial model to a continuous-time process, and each provided different prescriptions on how to do so. Before examining these models, let us review the basic concepts from the continuous-time models that are needed to fit the binomial model.

II.a Basic Continuous-Time Concepts for the Binomial Model

The results in this section are from the Black-Scholes-Merton model. It starts by proposing that the log return is normally distributed with mean $\mu$ and variance $\sigma^2$. Given that $\ln(S_{t+\Delta}/S_t) = \ln(S_{t+\Delta}) - \ln(S_t)$, the stochastic process is proposed as

$$d \ln(S_t) = \mu dt + \sigma dW_t, \quad (7)$$

where $\mu$ and $\sigma^2$ are the annualized expected return and variance, respectively, as given by $E[\ln(S_t)] = \mu dt$ and $Var[\ln(S_t)] = \sigma^2 dt$ and $dW_t$ is a Weiner process.

Now we examine the raw return, $dS_t/S_t$. Letting $G_t = \ln(S_t)$, we have $S_t = e^{G_t}$. We will need the partial derivatives, $\partial S_t/\partial G_t = e^{G_t}$, and $\partial^2 S_t/\partial G_t^2 = e^{G_t}$. Applying Itô's Lemma to $S_t$, we have
\[ dS_t = \frac{\partial S_t}{\partial G_t} dG_t + \frac{1}{2} \frac{\partial^2 S_t}{\partial G_t^2} dG_t^2. \]

Noting that \( dG_t = \mu dt + \sigma dW_t \), then \( dG_t^2 = \sigma^2 dt \). Substituting these results and the partial derivatives, we have

\[ \frac{dS_t}{S_t} = (\mu + \sigma^2 / 2) dt + \sigma dW_t. \]

Define \( \alpha \) as the expected value of the raw return so that

\[ \frac{dS_t}{S_t} = \alpha dt + \sigma dW_t, \quad (8) \]

and \( \alpha = \mu + \sigma^2 / 2 \). The expectation of \( dS_t / S_t \) is \( E[dS_t / S_t] = \alpha dt \) and \( \text{Var}[dS_t / S_t] = \sigma^2 dt \).

We see that the model assumes no difference in the volatilities of the raw and logarithmic processes in continuous time. This result is the standard assumption and derives from the fact that Itô’s Lemma is used to transform the log process to the raw process. Technically, the variance of the raw process is

\[ \text{Var}\left( \frac{dS_t}{S_t} \right) = (e^{\sigma^2 dt} - 1)e^{\alpha dt}, \quad (9) \]

which is adapted from Aitchison and Brown (1957). The difference in the variance defined as \( \sigma^2 dt \) and in (9) lies in the fact that the stochastic process for \( dS_t / S_t \) is an approximation. This subtle discrepancy is the source of some of the differences in the various binomial models.

One final result is needed. The expected value of \( S \) at the horizon \( T \) is given as

\[ E[S_T] = S \exp[(\mu + \sigma^2 / 2)T] = S \exp[\alpha T]. \quad (10) \]

II.b Fitting the Binomial Model to a Continuous-Time Process

Several of the papers on the binomial model proceed to fit the model to the continuous-time process by finding the binomial parameters, \( u, d, \) and \( q \) that force the binomial model mean and variance to equal the continuous-time model mean and variance. Thus, in this approach, the binomial model is fit to the physical process. These parameters are then used as though they apply to the risk neutral process when valuing the option. As we show shortly, this is a dangerous step.

The binomial equations for the physical process are

\[ q \ln u + (1 - q) \ln d = \mu h \quad (11) \]

and

\[^5\text{For proof see the appendix on p. 112 of Jarrow and Turnbull (2000).}\]
\[ q(1-q)(\ln(u/d))^2 = \sigma^2 h, \]  
\[ (u-d)^2 q(1-q) = (e^{\sigma^2 h} - 1)e^{\alpha h}, \]  depending on whether the user wishes to fit the log variance or raw variance. The volatility defined in the Black-Scholes-Merton model is the log volatility so the log volatility specification would seem more appropriate. But because the variance of the raw return is deterministically related to the variance of the log return, fitting the model to the variance of the raw return will still give the appropriate values of \( u \) and \( d \).

To convert the physical process to the risk-neutral process, a small transformation is needed. The mean raw return \( \alpha \) is set to the risk-free rate, \( r \). Alternatively, the mean log return \( \mu \) is set to \( r - \sigma^2/2 \). But fitting the model to the equations for the physical process is at best unnecessary and at worst, misleading. Recall that the Black-Scholes-Merton model requires knowledge of the stock price, exercise price, risk-free rate, time to expiration, and log volatility but not the true expected return on the stock. Fitting the binomial model to the physical process is unnecessary and imposes the added requirement that the expected return be known, thereby eliminating the principle advantage of arbitrage-free option pricing over preference-based pricing.

As is evident from basic option pricing theory, the arbitrage-free and correct price of the option is derived from knowledge of the volatility with the condition that the expected return equals the risk-free rate. It follows that correct specification of the binomial model should require only that these two conditions be met. Let \( \pi \) be the risk neutral probability. The correct mean specification is

\[ \pi u + (1-\pi)d = e^{rh}. \]  
This expression is then turned around to isolate \( \pi \):

\[ \pi = \frac{e^{rh} - d}{u - d}. \]

---

6 One would, however, need to exercise some care. Assume that the user knows the log variance. Then the raw variance can be derived from the right-hand side of (9), which then becomes the right-hand side of (13). If the user knows the log variance, then it becomes the right-hand side of (12). If the user has empirically estimated the raw and log variances, the former can be used as the right-hand side of (13) and the latter can be used as the right-hand side of (12). But then equations (12) and (13) might lead to different values of \( u \) and \( d \), because the empirical raw and log stochastic processes are unlikely to conform precisely to the forms specified by the underlying theory.

7 See Jarrow and Turnbull (2000) for an explanation of this transformation.
Either equation (14) or (15) is a necessary condition to guarantee the absence of arbitrage. Surprisingly, not all binomial option pricing models satisfy (14). Note that this condition is equivalent to, under risk neutrality, forcing the binomial expected raw return, not the expected log return, to equal the continuous risk-free rate. In other words, the correct value of \( \pi \) should come by specifying (14), not \( \frac{\pi \ln u + (1 - \pi) \ln d = (r - \sigma^2/2)h}{\pi \ln u + (1 - \pi) \ln d = (r - \sigma^2/2)h} \), which comes from adapting (11) to the risk neutral measure and setting the log expected return \( \mu \) to its risk neutral analog, \( r - \sigma^2/2 \). Surprisingly, many of the binomial models in the literature use this improper specification.

The no-arbitrage condition is a necessary but not sufficient condition for the binomial model to yield the correct option price. The model must be calibrated to the correct volatility. This constraint is met by using the risk-neutral analog of (5),

\[
(u - d)^2 \pi (1 - \pi) = (e^{\sigma^2 h} - 1)e^{rh}
\]

or (6),

\[
(\ln(u/d))^2 \pi (1 - \pi) = \sigma^2 h.
\]

Either condition will suffice because both return the correct volatility, either the raw or log volatility.

**II.c Convergence of the Binomial Model to the Black-Scholes-Merton Model**

Three of the most widely cited versions of the binomial model, Cox-Ross-Rubinstein (1979), Rendleman and Bartter (1979), and Jarrow-Rudd (1983), provide proofs that their models converge to the BSM model when \( N \to \infty \). Recall that each model is characterized by formulas for \( u, d \), and the probability. Hsia (1983) has provided a proof that demonstrates that convergence can be shown under less restrictive assumptions. For risk neutral probability \( \pi \), Hsia’s proof shows that the binomial model converges to the BSM model provided that \( N \pi \to \infty \) as \( N \to \infty \). To meet this requirement, we need only have \( 0 < \pi < 1 \). This result may seem surprising for it suggests that we can set the risk neutral probability at any arbitrary value such as 0.1 or

---

8The proof is widely known but can be quickly presented. The strategy of financing a stock purchase with a risk-free bond has a worst-case return per dollar invested of \( d - e^{rh} \). Arbitrage is possible if \( d > e^{rh} \). The strategy of financing a bond purchase by shorting the stock has a worst-case return of \( e^{rh} - u \). Arbitrage is possible if \( e^{rh} > u \). Hence, to prevent arbitrage, we require that \( d < e^{rh} < u \). This statement implies that there exists a measure, such as \( \pi \), such that \( \pi u + (1 - \pi)d = e^{rh} \), which is equation (14).

9A correct logarithmic specification of the no-arbitrage condition would be done by taking the log of equation (14). If this modified version of (14) were solved, the model would be correct. We found no instances in the literature in which this alternative approach is used.

10The other requirements not noted by Hsia are that the choice of \( u, d \), and \( \pi \) must force the binomial model volatility to equal the true volatility and the mean must guarantee no arbitrage.
0.8. In the literature some versions of the binomial model constrain the risk neutral probability to $\frac{1}{2}$ and as we show later, all versions of the model have risk neutral probabilities that converge to $\frac{1}{2}$. But Hsia’s proof shows that any probability other than zero or one will lead to convergence. We will examine this interesting result later.

II.d Alternative Binomial Models

We now examine the 11 different binomial models that have appeared in the literature.

Cox-Ross-Rubinstein

John Cox, Steve Ross, and Mark Rubinstein (1979), henceforth CRR, is arguably the seminal article on the model. Their equations (2) and (3) (p. 234) show the option value as given by our equation (1) with the risk neutral probability specified as our equation (2). CRR then proceed to examine how their formula behaves when $N \to \infty$ (their pp. 246-251). They do this by choosing $u$, $d$, and $q$ so that their model converges in the limit to the expected value and variance of the physical process. Thus, they solve for $u$, $d$, and $q$ using the physical process, our equations (11) and (12). Note that (11) and (12) constitute a system of two equations and three unknowns. CRR propose a solution while implicitly imposing a necessary third condition, that $ud = 1$, an assumption frequently found in the literature. Upon obtaining their solution, they then assume the limiting condition that $h^2 = 0$. This condition is necessary so that the correct variance is recovered, though the mean is recovered for any $N$. Their solutions are the well-known formulas for $u$ and $d$,

$$u = e^{\sigma \sqrt{h}}, \quad d = e^{-\sigma \sqrt{h}},$$

with physical probability

$$q = \frac{1}{2} + \frac{1}{2} \frac{\mu}{\sigma} \sqrt{h}.$$

As we discussed, for option valuation $q$ and $\mu$ are unnecessary. CRR recognize that the only condition required to prevent arbitrage is the equivalent of our equation (15). To guarantee the absence of arbitrage, they discard their solution for $q$, accept (15) as the solution for $\pi$, and retain their formulas for $u$ and $d$. But their formulas for $u$ and $d$ are the solutions based on the log mean, as given in (11), not the raw, arbitrage-prohibitive mean as in (15). Thus, their formulas for $u$ and $d$ are technically incorrect for finite $N$ and do not recover the correct volatility.

As it turns out, however, their solutions for $u$ and $d$ are correct in the limit, because in that case $u$ and $d$ converge to 1 and are infinitesimally different from their log values. Also, the risk neutral probability, equation (15), converges to $\frac{1}{2}$ using CRR’s
expressions for $u$ and $d$. CRR acknowledge, and it has been noted in the literature, that their solution recovers the volatility only in the limit, but the reason has not always been clear. Their reference to the volatility has always referred to the volatility obtained using the physical measure.\textsuperscript{11} We now see that the volatility computed using the risk neutral probabilities is also incorrect except in the limit. The problem arises from the simple fact that CRR fit the binomial model to the physical process, simultaneously deriving the physical probability $q$, and then substitute the arbitrage-free formula for $\pi$ as $q$. Had they imposed the arbitrage-free condition directly into the solution, they would have obtained different formulas, as we will see in another model.

\textit{Rendleman-Bartter \& Jarrow-Rudd-Turnbull}

Because of their similarities, discussion of the Rendleman-Bartter (RB) approach is combined with discussion of the Jarrow-Rudd (JR) approach and later appended with the Jarrow-Turnbull (JT) approach. Like CRR, these approaches also fit the binomial model to the physical process. The RB approach specifies the log mean and log variance equations, (11) and (12), and solves these two equations to obtain

$$u = e^{\mu k + \frac{\sigma \sigma}{2}}, \quad d = e^{\mu k - \frac{\sigma \sigma}{2}}.$$

Because these formulas do not specify the value of $q$, they are too general to be of use. In risk neutralizing the model, RB assume that $\mu = r - \sigma^2/2$ and a probability of $1/2$. It is important to note that this probability does not guarantee the absence of arbitrage because it is arbitrarily established and not derived by conditioning on the arbitrage-free specification, equation (14). We will need to make clear distinctions between the arbitrage-free risk-neutral probability, which results from equation (14), and the risk neutral probability obtained by solving whatever condition is imposed, such as specifying the log mean, equation (11). We will denote this latter probability $\pi^*$ and call it the risk neutral proxy probability. Hence, for RB $\pi^* = 1/2$. At this point let us turn to the JR approach, because it starts with basically these conditions. There is a great deal of confusion over this version of the binomial model. We will examine it carefully and attempt to bring some clarity.

JR solve the same two equations, the log mean and log variance, and assume that $q$, the physical probability, is $1/2$, which leads to the following formulas for $u$ and $d$,

$$u = e^{\mu k + \sigma \sqrt{k}}, \quad d = e^{\mu k - \sigma \sqrt{k}}.$$

\textsuperscript{11}See Cox, Ross, and Rubinstein (1979), pp. 248-249.
They then proceed to show that these formulas result in convergence of their model to the BSM model. Note that \( q \) is the physical probability. Normally the binomial model would be shown to converge to the BSM model using the risk neutral probability. These values for \( u \) and \( d \) are not consistent with the absence of arbitrage because they are derived by constraining the log mean, not the raw mean. That is, they are consistent with the risk neutral proxy probability \( \pi^* \) but not with the risk neutral probability \( \pi \). In the limit, it can be shown that \( \pi \) converges to \( \frac{1}{2} \) when JR’s formulas for \( u \) and \( d \) are used and with \( \pi^* = \frac{1}{2} \), all is well and the JR model is arbitrage-free in the limit. Combined with Hsia’s proof, the JR model clearly converges to BSM.

Thus, for finite \( N \), JR’s formulas do not prohibit arbitrage, but there is yet another problem. JR make the interesting comment (p. 188) that their formulas result in recovery of the volatility for any \( N \), while Cox, Ross, and Rubinstein’s parameters recover the volatility only in the limit. As we will show, the first portion of this statement is misleading. Their choice of \( \frac{1}{2} \) as a probability does indeed recover the correct volatility for finite \( N \), but this result is obtained only by using the physical probability. A risk neutral probability of \( \frac{1}{2} \) is obtained only in the limit. Hence, the volatility is not recovered for finite \( N \) when the risk neutral probability is used. For option pricing, of course, it is the risk neutral probability that counts.

JR risk neutralize their formulas by specifying that \( \mu = r - \sigma^2/2 \) (p. 190), thereby leading to their solutions:

\[
 u = e^{(r-\sigma^2/2)h + \sigma \sqrt{h}}, \quad d = e^{(r-\sigma^2/2)h - \sigma \sqrt{h}},
\]  

but again, these formulas are consistent only with a probability of \( \frac{1}{2} \) and risk neutrality as specified by \( \mu = r - \sigma^2/2 \). Simply converting the mean is not sufficient to ensure risk neutrality for a finite number of time steps.\(^{13} \)

A number of years later, JT derive the same model but make a much clearer distinction between the physical and risk neutral processes. They fix \( \pi^* \) at its arbitrage-free value and show for their up and down parameters that

\[
 \pi^* = \pi = \frac{e^{\sigma \sqrt{h}} - d}{u - d} = \frac{e^{\sigma \sqrt{h}/2} - e^{-\sigma \sqrt{h}}}{e^{\sigma \sqrt{h}} - e^{-\sigma \sqrt{h}}}. \tag{21}
\]

\(^{12}\)These are JR’s equation (13-18). In their models the up and down factors are denoted as \( u \) and \( v \), respectively, with \( S^u = 5e^v \) and \( S^v = 5e^v \).

\(^{13}\)A close look at JR shows that \( q \) is clearly the physical probability. On p. 187, they constrain \( q \) to equal the risk neutral probability, with their symbol for the latter being \( \phi \). But this constraint is not upheld in subsequent pages whereupon they rely on convergence in the limit to guarantee the desired result that arbitrage is prevented and BSM is obtained. This point has been recognized in a slightly different manner by Nawalkha and Chambers (1995), p. 608.
Like CRR, the correct specification of $\pi$ ensures that their model does not admit arbitrage. But, because their solutions for $u$ and $d$ were obtained by specifying the log mean, these solutions are not technically correct for finite $N$. The mean constraint is met, so there must be an error somewhere, which has to be in the variance. Thus, their model does not recover the variance for finite $N$ using the risk neutral probabilities. It returns the correct variance either when the physical probability is used or in the limit with the risk neutral probability converging to $\frac{1}{2}$.

For future reference, we will call this model the RBJRT model and refer only to the last version of the model in which the no-arbitrage constraint is applied to obtain $\pi$. We have shown that it does not recover the correct volatility for finite $N$. Now let us consider a model that fits a binomial tree to the physical process but does prevent arbitrage and recovers the correct volatility.

**Chriss**

Neil Chriss’s model (1996) specifies the raw mean and log variance of the physical process. The former is given by

$$q u + (1 - q) d = e^{\alpha h},$$

and the latter by equation (12). He then assumes that $q = \frac{1}{2}$. The solutions are

$$u = \frac{2e^{\alpha h + 2\sigma \sqrt{h}}}{e^{2\sigma h} + 1}, \quad d = \frac{2e^{\alpha h}}{e^{2\sigma h} + 1}.$$  

The risk-neutralized analogs are found by substituting $r$ for $\alpha$:

$$u = \frac{2e^{rh + 2\sigma \sqrt{h}}}{e^{2\sigma h} + 1}, \quad d = \frac{2e^{rh}}{e^{2\sigma h} + 1}.$$  

Note that because Chriss’ mean specification is the raw mean, transformation to risk neutrality by $\alpha = r$ correctly returns the no-arbitrage condition, equation (15). Thus, for the Chriss model, $\pi = \pi^* = \frac{1}{2}$ for all $N$, and the model correctly preserves the no-arbitrage condition and recovers the volatility for any number of time steps.

**Trigeorgis**

The Trigeorgis model transforms the original process into a log process. That is, let $X = \ln S$ and specify the binomial process as the change in $X$, or $\Delta X$. The solutions for the physical process are

$$u = e^{(\delta + \mu)n}, \quad d = e^{-\delta n}, \quad q = \frac{1}{2} + \frac{\mu}{2\sqrt{\delta^2 + \mu^2}}, \quad p = \frac{1}{2} - \frac{\mu}{2\sqrt{\delta^2 + \mu^2}}.$$  

Note that if $\delta^2 = 0$, the Trigeorgis model is the same as the CRR model. Trigeorgis then risk neutralizes the model by assuming that $\mu = r - \sigma^2/2$. The results are
Trigeorgis’ risk neutral probability comes simply from substitution of \( r - \sigma^2/2 \) for \( \mu \) in the formula for \( q \), thereby obtaining

\[
\pi^* = \frac{1}{2} + \frac{1}{2} \frac{(r - \sigma^2/2)h}{\sqrt{\sigma^2 h + (r - \sigma^2/2)h^2}}
\]

Of course this is the risk neutral proxy probability and is not given by the no-arbitrage condition. Therefore, it is not arbitrage-free for finite \( N \), though it does recover the correct volatility. In the limit, Trigeorgis’s risk neutral proxy probability, \( \pi^* \), converges to \( \frac{1}{2} \) and the arbitrage-free risk neutral probability, \( \pi \), converges to \( \frac{1}{2} \), so the Trigeorgis model is arbitrage-free in the limit.

**Wilmott1 and Wilmott2**

Paul Wilmott (1998) derives two binomial models. He specifies the raw mean and raw variance of the physical process, equations (22) and (17). His first model, which we shall call Wil1 assumes \( ud = 1 \). The solutions for the physical process are

\[
u = \frac{1}{2} \left(e^{\alpha h} + e^{(\alpha + \sigma^2)h}\right) + \frac{1}{2} \sqrt{\left(e^{\alpha h} + e^{(\alpha + \sigma^2)h}\right)^2 - 4},
\]

\[
d = \frac{1}{2} \left(e^{\alpha h} + e^{(\alpha + \sigma^2)h}\right) - \frac{1}{2} \sqrt{\left(e^{\alpha h} + e^{(\alpha + \sigma^2)h}\right)^2 - 4}.
\]

The physical probability \( q \) is found easily from the mean condition,

\[
q = \frac{e^{\alpha h} - d}{u - d}.
\]

Risk neutralizing the model is done by simply substituting \( r \) for \( \alpha \):

\[
u = \frac{1}{2} \left(e^{\alpha h} + e^{(r + \sigma^2)h}\right) + \frac{1}{2} \sqrt{\left(e^{\alpha h} + e^{(r + \sigma^2)h}\right)^2 - 4},
\]

\[
d = \frac{1}{2} \left(e^{\alpha h} + e^{(r + \sigma^2)h}\right) - \frac{1}{2} \sqrt{\left(e^{\alpha h} + e^{(r + \sigma^2)h}\right)^2 - 4}
\]

and the risk neutral probability \( \pi \) is correctly given by equation (14). Because the raw mean constraint is upheld, this model prohibits arbitrage, and it also recovers the volatility. In addition, \( \pi \) converges to \( \frac{1}{2} \) in the limit.

The second version of the model, which we shall call Wil2, assumes that \( q = \frac{1}{2} \). The solutions for the physical process are

\[
u = e^{\alpha h} \left(1 + \sqrt{e^{\sigma^2 h} - 1}\right),
\]

\[
d = e^{\alpha h} \left(1 - \sqrt{e^{\sigma^2 h} - 1}\right).
\]

Risk neutralizing the model gives the solutions
\[ u = e^{ah} \left( 1 + \sqrt{e^{\sigma^2 h} - 1} \right), \quad d = e^{ah} \left( 1 - \sqrt{e^{\sigma^2 h} - 1} \right). \tag{26} \]

Here \( \pi \) is forced to a value of \( \frac{1}{2} \) and this specification correctly prevents arbitrage because equation (14) is upheld. In addition, the volatility is recovered.

\textit{Jabbour-Kramin-Young}

Jabbour, Kramin, and Young (2001), henceforth JKY, provide a review of several well-known binomial models and introduce some new ones. They classify the models into three families. They describe one family as the Rendleman-Bartter approach, and we will refer to these models as JKYRB models. JKY call the second family alternative binomial models or ABMC models. We will refer to these models as JKYABMC models. They describe the third family as discrete-time Geometric Brownian motion or ABMD models. We will refer to this family of models as JKYABMD models. Each family is identified by its specification of the mean and variance and a conditioning constraint.

For the physical process, JKYRB models specify the mean and variance of the log process, equations (11) and (18). JKYABMC models specify the mean of the raw process, equation (22), and an approximation of the volatility of the raw process,

\[ q(1-q)(u-d)^2 \approx \sigma^2 h. \tag{27} \]

Recall that the volatility of the raw process should technically have \( e^{ah} (e^{\sigma^2 h} - 1) \) on the right-hand side.\textsuperscript{14} JKYABMD models specify an approximation of the raw mean as

\[ qu + (1-q)d \approx 1 + \alpha h, \tag{28} \]

where \( 1 + \alpha h \) is an approximation of \( e^{ah} \).

Risk neutralizing these models requires changing either \( \alpha \) to \( r \) or \( \mu \) to \( r - \sigma^2/2 \). Because the JKYRB and JKYABMD models specify the mean of the log process instead of the mean of the raw process, those models admit arbitrage for finite \( N \). Because the JKYABMC and JKYABMD models use an approximation of the raw volatility, they do not precisely recover the volatility.

As previously noted, the mean and volatility specifications establish two equations, but three unknowns. A third assumption, the conditioning constraint, is required to obtain solutions for \( u, d \), and either \( q \) or \( \pi \). We will append each model class with the letter “1,” “2,” or “3” to indicate the third assumption. Models appended with

\textsuperscript{14}We showed earlier that this approximation is obtained from derivation of the stochastic process of the raw return by applying Itô’s Lemma to the log process. It can also be shown to arise from application of the expression \( e^x \approx 1 + x + x^2/2 + \ldots \)
a “1” assume \( ud = 1 \). Models appended with a “2” assume \( ud = e^{\alpha h} \). Models appended with a “3” simply assume that \( q = \frac{1}{2} \) or \( \pi = \frac{1}{2} \).

Under these assumptions, model JKYRB1 is equivalent to Trigeorgis. Model JKYRB2 is unique and has solutions for the physical process of

\[
\begin{align*}
  u &= e^{\mu h - \frac{1-q}{\sqrt{q(1-q)}} \frac{\sigma}{\sqrt{h}}} , \\
  d &= e^{\mu h - \frac{-q}{\sqrt{q(1-q)}} \frac{\sigma}{\sqrt{h}}}
\end{align*}
\]

Many of the JKY models have the same solution for the probability, \( q \), which is

\[
q = \frac{1}{2} \left[ 1 - \frac{m}{\sqrt{4 + m^2}} \right].
\]

(29)

For JKYRB2,

\[
m = \sigma \sqrt{h}.
\]

(30)

Risk neutralizing by setting \( \mu = r - \frac{\sigma^2}{2} \) gives

\[
\begin{align*}
  u &= e^{(r - \frac{\sigma^2}{2}) h + \frac{1-q}{\sqrt{q(1-q)}} \frac{\sigma}{\sqrt{h}}} , \\
  d &= e^{(r - \frac{\sigma^2}{2}) h - \frac{q}{\sqrt{q(1-q)}} \frac{\sigma}{\sqrt{h}}}
\end{align*}
\]

(31)

\[
\pi^* = \frac{1}{2} \left[ 1 - \frac{m}{\sqrt{4 + m^2}} \right]
\]

(32)

with \( m \) given in (30). Because \( \pi^* \) is the probability obtained by the log mean constraint, it does not equal the arbitrage-free risk neutral probability, \( \pi \). Both, however, converge to \( \frac{1}{2} \) in the limit, so arbitrage is prohibited in the limit.

Model JKYRB3 is equivalent to RBJRT.

JKYAMBC1 is equivalent to Willmott2. JKYABMC2 is unique and has the following solutions for the physical process:

\[
\begin{align*}
  u &= e^{\alpha h} \left( 1 + \frac{1-q}{\sqrt{q(1-q)}} \sqrt{e^{\sigma^2 h} - 1} \right) , \\
  d &= e^{\alpha h} \left( 1 - \frac{q}{\sqrt{q(1-q)}} \sqrt{e^{\sigma^2 h} - 1} \right)
\end{align*}
\]

with \( q \) given by equation (29) and \( m \) given by

\[
m = \sqrt{e^{\sigma^2 h} - 1}.
\]

(33)

Risk neutralizing by setting \( \alpha \) to \( r \) gives

\[
\begin{align*}
  u &= e^{rh} \left( 1 + \frac{1-q^*}{\sqrt{q^*(1-q^*)}} \sqrt{e^{\sigma^2 h} - 1} \right) , \\
  d &= e^{rh} \left( 1 - \frac{q^*}{\sqrt{q^*(1-q^*)}} \sqrt{e^{\sigma^2 h} - 1} \right)
\end{align*}
\]

(34)

With \( \pi^* \) the same as in (32) and with \( m \) given in Equation (33). Because \( \pi^* \) is not obtained by meeting the arbitrage-free constraint, it does not equal the arbitrage-free
risk neutral probability, $\pi$. Both, however, converge to $\frac{1}{2}$ in the limit, so arbitrage is prohibited in the limit.

JKYABMC3 is equivalent to Willmott2.

The solutions for JKYABMD1 are

$$u = 1 + \alpha h + \frac{1-q}{\sqrt{q(1-q)}} \sigma \sqrt{h}, \quad d = 1 + \alpha h - \frac{q}{\sqrt{q(1-q)}} \sigma \sqrt{h}$$

with $q$ given by equation (29) and $m$ as

$$m = \frac{1 + \sigma^2 h - (1 + \alpha h)^2}{(1 + \alpha h) \sigma \sqrt{h}}.$$  

Risk neutralizing leads to the solution,

$$u = 1 + rh + \frac{1-\pi^*}{\sqrt{\pi^*(1-\pi^*)}} \sigma \sqrt{h}, \quad d = 1 + rh - \frac{\pi^*}{\sqrt{\pi^*(1-\pi^*)}} \sigma \sqrt{h} \quad (35)$$

with $\pi^*$ as given in (32) and $^{15}$

$$m = \frac{1 + \sigma^2 h - (1 + \alpha h)^2}{(1 + rh) \sigma \sqrt{h}}. \quad (36)$$

Because $\pi^*$ is not obtained by meeting the arbitrage-free constraint, it does not equal the arbitrage-free risk neutral probability, $\pi$. Both, however, converge to $\frac{1}{2}$ in the limit, so arbitrage is prohibited in the limit.

JKYABMD2 has solutions for the physical process of

$$u = 1 + \alpha h + \frac{1-q}{\sqrt{q(1-q)}} \sigma \sqrt{h}, \quad d = 1 + \alpha h - \frac{q}{\sqrt{q(1-q)}} \sigma \sqrt{h}$$

with $q$ given by equation (29) and $m$ as

$$m = \frac{e^{2\alpha h} + \sigma^2 h - (1 + \alpha h)^2}{(1 + \alpha h) \sigma \sqrt{h}}.$$  

These formulas are not exactly as reported in JKY. They make a further approximation using $e^x \cong 1 + x$. We report the solution without this approximation. As a result of this adjustment, we refer to this model as JKYABMD2c.

Risk neutralizing gives the solutions

$$u = 1 + rh + \frac{1-\pi^*}{\sqrt{\pi^*(1-\pi^*)}} \sigma \sqrt{h}, \quad d = 1 + rh - \frac{\pi^*}{\sqrt{\pi^*(1-\pi^*)}} \sigma \sqrt{h} \quad (37)$$

$^{15}$The formula for $\pi$ in equation (32) comes from solving a quadratic equation. As is well-known, such equations have two roots. For some of the JKY models, it can be shown that one sign is the correct one. For the JKYABMD1 model, both signs are acceptable solutions. JKY report the formula with the minus sign as the correct one, and we shall use it from this point on, but we should be aware that yet another solution exists.
with $\pi^*$ given by (32), and

$$m = \frac{e^{2rh} + \sigma^2h - (1 + rh)^2}{(1 + rh)\sigma\sqrt{h}}$$

(38)

Because $\pi^*$ is not obtained by meeting the arbitrage-free constraint, it does not equal the arbitrage-free risk neutral probability, $\pi$. Both, however, converge to $\frac{1}{2}$ in the limit, so arbitrage is prohibited in the limit.

JKYABMD3 has solutions for the physical process of

$$u = 1 + \alpha h + \sigma\sqrt{h}, \quad d = 1 + \alpha h - \sigma\sqrt{h}, \quad q = \frac{1}{2}. $$

Risk neutralizing gives the solutions

$$u = 1 + rh + \sigma\sqrt{h}, \quad d = 1 + rh - \sigma\sqrt{h}, \quad \pi^* = \frac{1}{2}. $$

(39)

Because $\pi^*$ is not obtained by meeting the arbitrage-free constraint, it does not equal the arbitrage-free risk neutral probability, $\pi$. Both, however, converge to $\frac{1}{2}$ in the limit, so arbitrage is prohibited in the limit.

**Avanelleda and Laurence**

We will take a look at one additional model that appears unique but is not. Avanelleda and Laurence (1999) (AL) take a notably different approach to solving for $u$, $d$, and $q$. For the risk neutral-process, the expected return is specified for the raw return as in equation (14). For the volatility, they specify the ratio of $u$ to $d$ in terms of a constant $\omega$,

$$u/d = e^{\omega\sqrt{h}}. $$

They specify the log volatility as in equation (12). Their solutions are

$$u = \frac{e^{rh + \omega\sqrt{h}}}{\pi e^{\omega\sqrt{h}} + (1 - \pi)e^{-\omega\sqrt{h}}}, \quad d = \frac{e^{rh - \omega\sqrt{h}}}{\pi^* e^{\omega\sqrt{h}} + (1 - \pi^*)e^{-\omega\sqrt{h}}}, \quad \pi^* = \frac{1}{2} \left(1 \pm \sqrt{1 - \sigma^2/\omega^2}\right).$$

Of course, these solutions contains an unknown $\omega$. AL note that if $\omega$ is set to $\sigma$, then $\pi^* = \frac{1}{2}$ and

$$u = \frac{e^{(r - \sigma^2/2)h + \sigma\sqrt{h}}}{\cosh(\sigma\sqrt{h})}, \quad d = \frac{e^{(r - \sigma^2/2)h - \sigma\sqrt{h}}}{\cosh(\sigma\sqrt{h})}.$$

These formulas have occasionally appeared in the literature.\(^{16}\) Nonetheless, the model is not unique as algebraic rearrangement shows that it is equivalent to Chriss.

**II.e. Model Recap**

\(^{16}\)See, for example, Carpenter (1998).
We have seen that there are 11 unique models for binomial option pricing with each model based on a different set of assumptions. Table 1 summarizes the models showing the risk neutral versions, as these are the forms used in option pricing. Note that arbitrage is possible for finite $N$ if a model fails to meet the appropriate mean constraint, equation (14). Trigeorgis and four of the five JKY models use either the log mean constraint of equation (11) or an approximation of equation (14), where $1 + rh \cong e^{rh}$. We classify the latter as admitting arbitrage, even though some would argue that the approximation is acceptable. At this point, we wish to be precise.

Any model that correctly specifies the volatility equation should recover the volatility but CRR and RBJRT do not. This result occurs because CRR and RBJRT solve the equations for the physical process, one of which is the log mean constraint. Recognizing that arbitrage must be prohibited, they adopt the arbitrage-free constraint to establish the risk neutral probability but retain the same solutions for $u$ and $d$, simply replacing $\mu$ with $r - \sigma^2/2$ where necessary. As a result, the formulas for $u$ and $d$ cannot recover the correct volatility for finite $N$.

III. Some Anomalous Cases

Let us consider four desirable conditions. We would want the risk neutral probability to range between 0 and 1. We would also want $u$ to exceed 1 and $d$ to be less than 1. As we have seen, some of the models use the arbitrage-free risk neutral probability, $\pi$, and others use the risk neutral proxy probability, $\pi^*$. We will examine the properties of whichever formula the authors propose as the risk neutral probability. Thus, we examine four conditions: (a) $\pi$ (or $\pi^*) < 1$, (b) $\pi$ (or $\pi^*) > 0$, (c) $u > 1$, and (d) $d < 1$. Conditions (a) and (b) are necessary to establish that $\pi$ (or $\pi^*)$ is a probability measure. The last two conditions are intuitively desirable. If $u < 1$, the asset goes down when intuition suggests that it should go up. If $d > 1$, the asset goes up when intuition suggests that it should go down.

For $\pi$, condition (a) is equivalent to the following:

\[
\pi < 1 \Rightarrow \frac{\exp(rh) - d}{u - d} < 1
\]

\[
\Rightarrow \exp(rh) < u
\]

Given that $u$ is always greater than $d$, which is true for each model, condition (b) is met as:

\[
\pi > 0 \Rightarrow \frac{\exp(rh) - d}{u - d} > 0
\]

\[
\Rightarrow \exp(rh) > d
\]
Conditions (c) and (d) are met simply by examining the formulas for $u$ and $d$. When the author proposes $\pi^*$ as the risk neutral probability, we examine it directly in relation to 0 and 1.

Table 2 summarizes whether each condition is always met. Footnotes show examples in which the violation occurs. While all of the models pass the test that the risk neutral probability or its proxy exceeds zero, the CRR and RBJRT models can have $\pi > 1$. This result for CRR is well-known and has been cited in the literature.\textsuperscript{17} It arises when $h > (\sigma/r)^2$, which is likely to occur with low volatility, high interest rates, and a long time to expiration. Sufficiently low volatility is unlikely to apply when modeling stock prices, but exchange rate volatility is often less than 0.1. Thus, long-term foreign exchange options where the interest rate is high can have a risk neutral probability greater than 1.\textsuperscript{18} For RBJRT, the risk neutral probability can exceed 1 if $h < 4/\sigma^2$. Although the volatility of some commodities has been known to exceed 100%, the volatility of most stocks is less than 100%. Therefore, for most stocks we have $4/1^2 = 4$, so the problem exists only if $h > 4$. For very low volatility, as in the foreign exchange market, the time step would have to be extremely large. Thus, it would take exceptionally large volatility and a very small number of time steps relative to the option maturity for the risk neutral probability to exceed 1 for the RBJRT model.

It is interesting to note, however, that even if the risk neutral probability exceeds 1, a model could still correctly provide the value of the option. But, as we have noted, the CRR and RBJRT models use the $u$ and $d$ formulas from the physical process, which is derived by constraining the log mean, not the raw mean. It is the raw mean that guarantees no arbitrage.

The other two desirable conditions are that the up factor exceeds 1 and the down factor is less than 1. These conditions are not mandatory for correctly pricing the option, but they are intuitively desirable. Only the RBJRT methodology permits an up factor that can be less than 1. Interestingly, seven of the eleven models permit a down factor greater than 1. Only the models of Trigeorgis, Wilmott1, and the JKYABMD1 model have no possibility of any anomalies.

These anomalies are interesting but usually occur only with extreme values of the inputs and/or an extremely small number of time steps relative to the option maturity.

\textsuperscript{17}See, for example, Chriss (1998), p. 239.
\textsuperscript{18}For example, suppose $r = .1$, $\sigma = .05$, and $T = 5$. In that case, the number of time steps would need to exceed 20.
They can usually be avoided when actually pricing an option. The greatest risk they pose is probably when the model is used for illustrative purposes.

IV. Model Comparisons

In Table 3 we illustrate an example for valuing a call option in which the asset is priced at 100, the exercise price is 100, and the volatility is 30%. The continuous risk-free rate is 5% and the option expires in one year. In all cases, we use the probability $\pi$ or $\pi^*$ as specified by the authors of the respective models. We show the values for 1, 5, 10, 20, 30, 50, 75 and 100 time steps. The correct value, as given by the Black-Scholes-Merton formula, is 14.23. At 50 times steps all of the prices are within 0.06. At 100 time steps, all of the prices are within 0.03.

To further investigate the question of which models performs best, we conduct additional analyses using a wide range of parameters. We let the volatility be 0.1, 0.3, and 0.5, the time to expiration be 0.25, 1.0, and 4.0, the moneyness be 10% out-of-the-money, at-the-money, and 10% in-the-money. These inputs comprise 27 unique combinations. We examine several characteristics of the convergence of these models to the Black-Scholes-Merton value.

IV.a An Initial Look at Convergence

Let $b(N)$ be the value computed for a given binomial model with $N$ time steps and BSM be the true Black-Scholes-Merton value. Binomial models are commonly described as converging in a pattern referred to as “odd-even.” That is, when the number of time steps is odd (even), the binomial price tends to be above (below) the true price. We will call this phenomenon the “odd-even” property. Interestingly, the numerical analyses show that the odd-even phenomenon never occurs for any model when out-of-the-money options are being priced. For at-the-money options, the odd-even phenomenon always occurs for the JKYABMD1 model and occasionally for some other models. Odd-even convergence never occurs for any combination of inputs for JKYRB2, JKYABMC2, and JKYABMD2c. Thus, the odd-even property is not a consistent phenomenon across models.

We next examine whether a model exhibits monotonic convergence, defined as

$$|e(N)| < |e(N-1)| \forall N > 1,$$

where $|e(N)| = |b(N) - \text{BSM}|$. Monotonic convergence, thus, means that each error is smaller than the previous one. Only the Trigeorgis model exhibits monotonic convergence and it does so for only one of the 27 combinations of inputs examined. Because monotonic convergence is virtually non-existent, we examine a slight variation.
It is possible that each alternative error is smaller than the previous one. We call this phenomenon *alternating monotonic convergence*, and it is defined as

\[ |\epsilon(N)| < |\epsilon(N-2)| \forall N > 2.\]

Unfortunately, alternating monotonic convergence does not occur for any model for all combinations of inputs.

We then attempt to identify at which step a model is deemed to have acceptably converged. For a given time step, we compute the average of the current computed price and the previous computed price. We then identify the time step at which this average price is within 0.01 of the BSM price with the added criterion that the difference must remain less than 0.01 through step 100. The results are presented in Tables 4, 5 and 6. One consistent result in all three tables is that the RBJRT and Chriss models produce the same results. Further examination shows that the values of \( u \) and \( d \) are not precisely equal for both models for all values of \( N \), but they are very close and become equal quickly for fairly small values of \( N \).

We see in Table 4 that for at-the-money options, the Trigeorgis models perform best followed by CRR and Wil1. The worst model is JKYABMD2c followed by JKYABMC2 and JKYRB2. For in-the-money options, the best model is CRR followed by Chriss-RBJRT with Wil2 very close behind. The worst model is Wil1 followed by JKYABMD3 and Trigeorgis. For out-of-the-money options, the best models are RBJRT-Chriss followed by JKYABMD1. The worst model is Wil1 followed by CRR and Wil2.

Table 5 shows that convergence is always faster with a shorter time to expiration. This result should not be surprising. With a fixed number of time steps, a shorter time to expiration means that the time step is smaller. For the medium maturity, the fastest convergence is achieved by the Trigeorgis model followed by JKYABMD1 and CRR. The worst performance is by Wil2 followed by JKYABMC2 and JKYRB2. For the shortest maturity, the best performance is by CRR followed by JKYABMD1 and Trigeorgis, and the worst performance is by JKYABMC2 followed by JKYRB2 and Wil2. For the longest maturity, the best performance is by RBJRT-Chriss followed by JKYRB2 and JKYABMC2 (tied). The worst performance is by JKYABMD2c followed by JKYABMD1 and JKYABMD3.

In Table 6 we observe that convergence is slower with higher volatility for all models. For the lowest volatility, the fastest models are RBJRT-Chriss (tied) followed by JKYRB2 and JKYABMC2 (tied). The slowest model is JKYABMD3 followed by JKYABMD2c and JKYABMD1. For the medium volatility, the fastest model is
JKYABMD3 followed by CRR and Trigeorgis, and the slowest is JKYABMC2 followed by JKYRB2 and Wil2. For the highest volatility, the fastest models are Trigeorgis followed by CRR and JKYABMD1, while the slowest is JKYABMC2 followed by JKYRB2 and Wil2 (tied).

It is difficult to draw conclusions about which are the fastest and slowest models. Each model finishes in the top or bottom four at least once. Although the tests are not independent, we can gain some insight by assigning a simple ranking (1 = best, 11 = worst) and tally the performance across all nine groupings. In that case CRR has the best performance with the lowest overall score of 36, while Trigeorgis is closely behind at 37, and RBJRT and Chriss are at 38. The next best score is somewhat further down at 45. The highest scores and, thus, worst performance are JKYABMC2 at 71.5 followed by Wil2 at 69.5 and JKYABMD2c at 67.5. These rankings are useful and could suggest that CRR, Trigeorgis, RBJRT, and Chriss might be the best class of models, but they are not sufficient to declare a single winner.

Whether a model converges acceptably can be defined by whether the error is within a tolerance for a given time step. We calculate the error for the 100th time step. These results also reveal no consistent winner among the models. Most model values are within four cents of the true value on the 100th time step, and the differences are largest with long maturity and/or high volatility, consistent with our previous finding that short maturity and low volatility options are the fastest to price.

IV.b A More Detailed Look at Convergence

Leisen and Reimer (1996) (LR) provide an extensive analysis of the convergence of the CRR and RBJRT models. They rely on the notion of order of convergence. A model converges more rapidly, the higher the order of convergence. Therefore, it is important to determine the order of convergence of these competing binomial models.

Convergence of a binomial model is defined to occur with order $\rho$ if there exists a constant $k$ such that

$$|e(N)| \leq \frac{k}{N^\rho}.$$ 

Visual examination of the errors on a log-log graph can reveal the order of convergence. LR further show, however, that a better measure of convergence can be derived using the difference between the moments of the binomial and continuous-time distributions. These moments are defined as follows:

---

19Leisen and Reimer also examine a model by Tian that incorporates skewness. We do not address adjustments for skewness in this paper, because we focus only on mean-variance binomial models.
The moments $\bar{m}^2(N)$ and $\bar{m}^3(N)$ are obviously related to the second and third moments. The third term is referred to as a pseudo-moment. Let $\rho(.)$ represent the order of convergence of the above moments and the pseudo-moment. LR show that the order of convergence of the binomial series is

$$\rho(N) := \pi \ln(u-1)^3 + (1 - \pi) \ln(d-1)^3,$$

In other words, the order of convergence is the minimum of the orders of convergence of the two moments and the pseudo-moment minus one with an overall minimum order of convergence of one. They show that the order of convergence of these terms can be derived mathematically and they do so for the three models they examine. These proofs, however, are quite detailed and cumbersome and, as they note, visual inspection is equally effective.

We examine the order of convergence using the moments and pseudo-moment of each of the eleven models. Because of the excessive space required, we do not present the results for all models. For illustrative purposes, we show the Chriss model. Figures 1-3 illustrate various characteristics of the convergence of the Chriss model for the case of a stock price of 100, exercise price of 100, risk-free rate of 0.05, time to expiration of one year, and volatility of 0.30. Because the LR error analysis uses common logs, we show only the time steps starting with 10.

Figure 1 is the option price graphed against the number of time steps, with the BSM value represented by the horizontal line. The convergence is oscillatory, exhibiting the odd-even pattern noted above. Figure 2 shows the absolute value of the error, which exhibits a wavy pattern. The solid line was created by proposing values for $k$ and $\rho$ such that the error bound always lies above the absolute value of the error. The value of $k$ is not particularly important, but the value of $\rho$ indicates the order of convergence. The value of $k$ is not particularly important, but the value of $\rho$ indicates the order of convergence. In this case, $\rho = 1$. A value of $\rho = 2$ would force the bound below the wavy error line. Thus, the order of convergence is clearly 1. Figure 3 shows the moments and pseudo-moment as defined by Liesen and Reimer. The pseudo-moment $\varphi(N)$ and $\bar{m}^2(N)$ are almost indistinguishable. The heavy solid line is the simple function $1/N^\rho$ where $\rho$ is the order of convergence of the moments and pseudo-moment. In this case, $\rho = 2$ provides

$\rho(N) := \pi \ln(u-1)^3 + (1 - \pi) \ln(d-1)^3$.

Figure 3 shows the moments and pseudo-moment as defined by Liesen and Reimer. The pseudo-moment $\varphi(N)$ and $\bar{m}^2(N)$ are almost indistinguishable. The heavy solid line is the simple function $1/N^\rho$ where $\rho$ is the order of convergence of the moments and pseudo-moment. In this case, $\rho = 2$ provides

$\rho(N) := \pi \ln(u-1)^3 + (1 - \pi) \ln(d-1)^3$.
the best fit. Therefore, following Theorem 1 of Leisen and Reimer, the order of convergence model is 1, confirming our direct examination of the error.

These graphs were generated for the remaining 10 models and all indicate an order of convergence of 1. Thus, it appears that no particular model outperforms the others. In the limit all models produce the Black-Scholes-Merton value, but of course limit analysis make \( N \) approach infinite. As we saw earlier, seven of the eleven models admit arbitrage with finite \( N \), but these arbitrage opportunities vanish in the limit. We also saw that the values of \( \pi \) and \( \pi^* \) converge to \( \frac{1}{2} \) in the limit. These results suggest that a model that correctly prevents arbitrage for all \( N \) and sets the risk neutral probability \( \pi \) at \( \frac{1}{2} \) for any \( N \) might be superior. That model is the Chriss model. And yet, there is no evidence that the Chriss model consistently performs best for finite \( N \).

IV.c Why the Models Converge

We have shown that all the models converge, but it is not clear why. As Hsia’s proof shows, the requirement for convergence is not particularly demanding, but clearly one cannot arbitrarily choose formulas for \( u, d, \) and \( \pi \).

As we noted, it is possible to prove that all of the formulas for either \( \pi \) or \( \pi^* \) converge to \( \frac{1}{2} \) in the limit. Let us examine why this result occurs. Focusing on \( \pi \), we divide the models into four categories: (1) models that assume \( \pi = \frac{1}{2} \) (Chriss, Wil2, JKYABMD3), (2) models that assume \( ud = e^{2rh} \) (JKYRB2, JKYABMC2, JKYABMD2c), (3) models that assume \( ud = e^{2(r-\sigma^2/2)h} \) (RBJRT), and (4) models that assume \( ud = 1 \) (CRR, Trigeorgis, Wil1, JKYABMD1).\(^{21}\) For (1), we need not examine the limiting condition. For (2), (3) and (4), general convergence is shown in the Appendix.

Thus, all of the models either have \( \pi \) or \( \pi^* \) converge to \( \frac{1}{2} \). The other requirements are that the models return the correct mean and volatility in the limit. Let us look at how they achieve this result. Re-classify the models according to their assumptions about the mean. Group (1) includes all models that correctly use the arbitrage-free specification of the raw mean, equation (14), (CRR, RBJRT, Chriss, Wil1, Wil2, JKYABMC2), (2) includes all models that correctly use the raw mean specification but use \( 1+ rh \) instead of \( e^h \) (JKYABMD1, JKYABMD2c, and JKYABMD3), and (3) includes the models that specify the log mean, equation (16), (Trigeorgis and JKYRB2).

\(^{21}\)It is important to understand why RBJRT is classified in this manner and not in any other group. RB and JR obtain their solutions by setting the physical probability \( q \) to \( \frac{1}{2} \). Their solution derives from using the mean of the log process. Hence, the solution is not arbitrage-free. JT impose the arbitrage-free condition and, hence, correctly use \( \pi \) for the risk neutral probability, but this constraint cannot lead to their formulas for \( u \) and \( d \). Their formulas can be obtained only by imposing a third condition, which can be inferred to be the one stated here.
Obviously Group (1) will correctly converge to the proper mean. Group (2) will do so as well, because $e^{rh}$ is well approximated by $1 + rh$ in the limit. Group (3) uses the specification (equation (16)), $\pi^* \ln u + (1 - \pi^*) \ln d = (r - \sigma^2/2)h$. Using the approximation $\ln u \cong u - 1$ and likewise for $d$, we have

$$\pi^*(u - 1) + (1 - \pi^*)(d - 1) = (r - \sigma^2/2)h$$

$$\Rightarrow \pi^* u + (1 - \pi^*)d = 1 + rh - \sigma^2 h/2$$

This specification is extremely close to that of Group (2), differing only by the variance term on the RHS, which goes to zero in the limit.

Now we need to consider the volatility. Let Group (1) consist of models that correctly specify the log volatility (CRR, RBJRT, Trigeorgis, JKRYB2, Chriss), (2) consist of models that correctly specify the raw volatility (Wil1, Wil2, and JKYABMC2), and (3) consist of models that use an approximation of the raw volatility, $\sigma^2 h = e^{2rh} (e^{\sigma^2 h - 1})$ (JKYABMD1, JKYABMD2c, and JKYABMD3). Group (1) will obviously return the correct log volatility, and Group (2) will return the correct raw volatility. Either specification suffices because constraining the one volatility automatically constrains the other. Group (3) can be shown to be based on an acceptable approximation by using the Taylor series for the exponential function and assuming $hk^k = 0$ for all $k$ of power 2 or more.

Hence, all of the models work because in the limit they all have a binomial probability of $\frac{1}{2}$, and they all return the risk-free rate as the mean and the correct volatility in the limit. Thus, any model with these characteristics will work. As we show in the next section, however, the constraints are not nearly that severe. Any probability, except 0 or 1, will suffice.

V. A General Binomial Formula

As noted earlier, Hsia’s proof of the convergence of the binomial model to the Black-Scholes-Merton model shows that any probability is acceptable provided that $u$ and $d$ return the correct mean and volatility. This result suggests that any value of the risk neutral probability would lead to convergence if the correct mean and volatility are upheld. We now propose a general binomial model with arbitrary $\pi$ that prohibits arbitrage and recovers the correct volatility for all $N$. Let the mean and variance be specified as follows:

$$\pi u + (1 - \pi) d = e^{rh}$$

$$\left(\ln(u/d)\right)^2 \pi (1 - \pi) = \sigma^2 h.$$
Of course, these are equations (14) and (18). The mean equation guarantees no arbitrage profits for all $N$. Now let us assume that $\pi$ is known but its value is left unspecified. Solving for $u$ and $d$ gives

$$u = \frac{e^{rh} + \sigma \sqrt{h} / \sqrt{1 - \pi}}{\pi e^{\sigma \sqrt{h} / \sqrt{1 - \pi}} + (1 - \pi)}, \quad d = \frac{e^{rh}}{\pi e^{\sigma \sqrt{h} / \sqrt{1 - \pi}} + (1 - \pi)}.$$ 

and, of course,

$$\pi = \frac{e^{rh} - d}{u - d}.$$

For the special case where $\pi = 1/2$, the equations are equivalent to those of Chriss.

These equations tell us that we can arbitrarily set $\pi$ to any value between 0 and 1 and be assured that the model will converge to the BSM value. This result is observed in Figure 4. Note that while convergence appears much smoother and faster with $\pi = 1/2$, the results are not much different for probabilities of $1/4$ and $3/4$. For $N = 100$, a probability of $1/4$ gives an option value of 14.27, while a probability of $3/4$ gives an option value of 14.15. The correct BSM value is 14.23.\(^{22}\)

While yet one more binomial formula is not necessary, this model shows that binomial option pricing is a remarkably flexible procedure that makes only minimum demands on its user and the choice of probability is not one of them.

**VI. Summary**

While some would argue that studying the binomial model as it applies to standard European options is not a productive exercise, such endeavors are in fact quite instructive. Standard European options have an unequivocal benchmark, the Black-Scholes-Merton model. Benchmarking is difficult for models in which numerical methods are the only means of obtaining the value of the option. Indeed the binomial model itself is often taken as the benchmark for more complex models, particular in the extensive body of research on American option pricing. As shown in this paper, the binomial model is subject to a variety of interpretations and no particular variation uniformly stands head and shoulders above the others.

On a theoretical basis, however, it would seem that an acceptable binomial model should prohibit arbitrage for a finite number of time steps and should recover the correct

---

\(^{22}\)A general formula of this type even means that extreme probabilities, say 0.01 and 0.99, would also correctly price the option in the limit. We tested these extreme values, however, and the results are not impressive. For example, with a probability of 0.01 we obtain an option value of 13.93, while a probability of 0.99 gives an option value of 13.01 after 100 time steps. Convergence is extremely erratic and the order of convergence is difficult to determine. Nonetheless, in the limit, the correct option value is obtained.
volatility. As shown here and in Hsia’s elegant proof, the choice of the actual risk neutral probability is meaningless in the limit, but clearly a risk neutral probability of \( \frac{1}{2} \) assures the fastest convergence.

This paper has shown that of the 11 models that have appeared in the literature, some admit arbitrage for a finite number of time steps. But in the limit, all of the models return the correct risk-neutral mean and volatility and, therefore, correctly price the option.

The binomial model is clearly not a single model but rather a family of interpretations of a discrete-time process that converges to the geometric Brownian motion process in the continuous limit and accurately prices options. That there are no less than 11 such members of this family does, however, seem surprising. The fact that all 11 models seem to perform equally, even though some admit arbitrage for a finite number of time steps, is a testament to the extremely general nature of the Black-Scholes-Merton model and its modest requirements.
Table 1. Summary of Measures for the Three Principal Classes of Binomial Models
These results present the solutions for the risk neutral binomial option pricing model.

(a) Log mean and log variance matching

<table>
<thead>
<tr>
<th>Key assumptions</th>
<th>CRR</th>
<th>RBJRT</th>
<th>Trigeorgis</th>
<th>JKYRB2</th>
</tr>
</thead>
<tbody>
<tr>
<td>$h^2 \to 0$, $ud = 1$ (implicitly)</td>
<td>$q = \frac{1}{2}$</td>
<td>$ud = 1$ $(\ln u = -\ln d)$</td>
<td>$ud = e^{2h}$</td>
<td></td>
</tr>
<tr>
<td>mean $\pi \ln u + (1-\pi) \ln d = e^{rh}$</td>
<td>$\pi \ln u + (1-\pi) \ln d = (r-\sigma^2/2)h$</td>
<td>$\pi \ln u + (1-\pi) \ln d = (r-\sigma^2/2)h$</td>
<td>$\pi \ln u + (1-\pi) \ln d = (r-\sigma^2/2)h$</td>
<td></td>
</tr>
<tr>
<td>variance $\pi(1-\pi)[\ln(u/d)]^2 = \sigma^2 h$</td>
<td>$\pi(1-\pi)[\ln(u/d)]^2 = \sigma^2 h$</td>
<td>$\pi(1-\pi)[\ln(u/d)]^2 = \sigma^2 h$</td>
<td>$\pi(1-\pi)[\ln(u/d)]^2 = \sigma^2 h$</td>
<td></td>
</tr>
<tr>
<td>probability ($\pi$ or $\pi^*$) $\frac{e^{rh} - e^{-\sigma \sqrt{h}}}{e^{\sigma \sqrt{h}} - e^{-\sigma \sqrt{h}}}$</td>
<td>$\frac{e^{\sigma h/2} - e^{-\sigma \sqrt{h}}}{e^{\sigma \sqrt{h}} - e^{-\sigma \sqrt{h}}}$</td>
<td>$\frac{1 + 1}{2} \frac{(r-\sigma^2/2)h}{2 \sqrt{\sigma^2 h + (r-\sigma^2/2)h^2}}$</td>
<td>$\frac{1}{2} \left[ 1 - \frac{m}{\sqrt{4 + m^2}} \right]$</td>
<td></td>
</tr>
<tr>
<td>$m$</td>
<td>NA</td>
<td>NA</td>
<td>NA</td>
<td>$\sigma \sqrt{h}$</td>
</tr>
<tr>
<td>$u$</td>
<td>$e^{\sigma \sqrt{h}}$</td>
<td>$e^{(r-\sigma^2/2)h+\sigma \sqrt{h}}$</td>
<td>$e^{\sigma h + (r-\sigma^2/2^2)h^2}$</td>
<td>$e^{(r-\sigma^2/2)h+\frac{1-\pi}{2\sqrt{1-\pi}} \sigma \sqrt{h}}$</td>
</tr>
<tr>
<td>$d$</td>
<td>$e^{-\sigma \sqrt{h}}$</td>
<td>$e^{(r-\sigma^2/2)h-\sigma \sqrt{h}}$</td>
<td>$e^{-\sigma h + (r-\sigma^2/2^2)h^2}$</td>
<td>$e^{(r-\sigma^2/2)h-\frac{1-\pi}{2\sqrt{1-\pi}} \sigma \sqrt{h}}$</td>
</tr>
<tr>
<td>Recovers $\sigma$ for finite $N$?</td>
<td>No</td>
<td>No</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Prohibits arbitrage for finite $N$?</td>
<td>Yes</td>
<td>Yes</td>
<td>No</td>
<td>No</td>
</tr>
</tbody>
</table>
(b) Raw mean and log variance matching

<table>
<thead>
<tr>
<th>Chriss</th>
<th>Chriss</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Key assumptions</strong></td>
<td>$\pi = \frac{1}{2}$</td>
</tr>
<tr>
<td><strong>mean</strong></td>
<td>$\pi u + (1 - \pi)d = e^{rh}$</td>
</tr>
<tr>
<td><strong>variance</strong></td>
<td>$\pi (1 - \pi) [\ln(u / d)]^2 = \sigma^2 h$</td>
</tr>
<tr>
<td><strong>probability</strong></td>
<td>$\frac{1}{2}$</td>
</tr>
<tr>
<td>$(\pi \text{ or } \pi^*)$</td>
<td>$\frac{1}{2}$</td>
</tr>
<tr>
<td>$m$</td>
<td>NA</td>
</tr>
<tr>
<td>$u$</td>
<td>$\frac{2e^{rh} + 2e^{2rh}}{e^{2rh} + 1}$</td>
</tr>
<tr>
<td>$d$</td>
<td>$\frac{2e^{rh}}{e^{2rh} + 1}$</td>
</tr>
<tr>
<td><strong>Recovers $\sigma$ for finite $N$?</strong></td>
<td>Yes</td>
</tr>
<tr>
<td><strong>Prohibits arbitrage for finite $N$?</strong></td>
<td>Yes</td>
</tr>
</tbody>
</table>
### (c) Raw mean and raw variance matching

<table>
<thead>
<tr>
<th>Key assumptions</th>
<th>Wil1</th>
<th>Wil2</th>
<th>JKYABMC2</th>
</tr>
</thead>
<tbody>
<tr>
<td>$ud = 1$</td>
<td>$\pi = \frac{1}{2}$</td>
<td>$ud = e^{2rh}$</td>
<td></td>
</tr>
<tr>
<td>$\pi u + (1 - \pi)d = e^{rh}$</td>
<td>$\pi u + (1 - \pi)d = e^{rh}$</td>
<td>$\pi u + (1 - \pi)d = e^{rh}$</td>
<td></td>
</tr>
<tr>
<td>$\pi(1 - \pi)(u - d)^2 = e^{2rh}(\sigma^2_k - 1)$</td>
<td>$\pi(1 - \pi)(u - d)^2 = e^{2rh}(\sigma^2_k - 1)$</td>
<td>$\pi(1 - \pi)(u - d)^2 = e^{2rh}(\sigma^2_k - 1)$</td>
<td></td>
</tr>
<tr>
<td>Probability ($\pi$ or $\pi^*$)</td>
<td>$e^{rh} - \frac{1}{2} \left( e^{-rh} + e^{(r+\sigma^2_k)h} \right) - \frac{1}{2} \sqrt{\left( e^{-rh} + e^{(r+\sigma^2_k)h} \right)^2 - 4}$</td>
<td>$\frac{1}{2}$</td>
<td>$\frac{1}{2} \left[ 1 - \frac{m}{\sqrt{4 + m^2}} \right]$</td>
</tr>
<tr>
<td>$m$</td>
<td>NA</td>
<td>NA</td>
<td>$\sqrt{e^{\sigma^2_k} - 1}$</td>
</tr>
<tr>
<td>$u$</td>
<td>$\frac{1}{2} \left( e^{-rh} + e^{(r+\sigma^2_k)h} \right) + \frac{1}{2} \sqrt{\left( e^{-rh} + e^{(r+\sigma^2_k)h} \right)^2 - 4}$</td>
<td>$e^{rh} \left( 1 + \sqrt{e^{\sigma^2_k} - 1} \right)$</td>
<td>$e^{rh} \left( 1 + \frac{1 - \pi}{\sqrt{\pi(1 - \pi)}} \sqrt{e^{\sigma^2_k} - 1} \right)$</td>
</tr>
<tr>
<td>$d$</td>
<td>$\frac{1}{2} \left( e^{-rh} + e^{(r+\sigma^2_k)h} \right) - \frac{1}{2} \sqrt{\left( e^{-rh} + e^{(r+\sigma^2_k)h} \right)^2 - 4}$</td>
<td>$e^{rh} \left( 1 - \sqrt{e^{\sigma^2_k} - 1} \right)$</td>
<td>$e^{rh} \left( 1 - \frac{\pi}{\sqrt{\pi(1 - \pi)}} \sqrt{e^{\sigma^2_k} - 1} \right)$</td>
</tr>
<tr>
<td>Recovers $\sigma$ for finite $N$?</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Prohibits arbitrage for finite $N$?</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
</tbody>
</table>

Recovers $\sigma$ for finite $N$? | Yes | Yes | Yes |
Prohibits arbitrage for finite $N$? | Yes | Yes | Yes |
(d) Approximate raw mean and approximate raw variance matching

<table>
<thead>
<tr>
<th></th>
<th>JKYABMD1</th>
<th>JKYABMD2c</th>
<th>JKYABMD3</th>
</tr>
</thead>
<tbody>
<tr>
<td>Key assumptions</td>
<td>$ud = 1$</td>
<td>$ud = e^{2rh}$</td>
<td>$\pi = \frac{1}{2}$</td>
</tr>
<tr>
<td>mean</td>
<td>$\pi u + (1 - \pi)d = 1 + rh$</td>
<td>$\pi u + (1 - \pi)d = 1 + rh$</td>
<td>$\pi u + (1 - \pi)d = 1 + rh$</td>
</tr>
<tr>
<td>variance</td>
<td>$\pi(1 - \pi)(u - d)^2 = \sigma^2 h$</td>
<td>$\pi(1 - \pi)(u - d)^2 = \sigma^2 h$</td>
<td>$\pi(1 - \pi)(u - d)^2 = \sigma^2 h$</td>
</tr>
<tr>
<td>probability ($\pi$ or $\pi^*$)</td>
<td>$\frac{1}{2} \left[ 1 - \frac{m}{\sqrt{4 + m^2}} \right]$</td>
<td>$\frac{1}{2} \left[ 1 - \frac{m}{\sqrt{4 + m^2}} \right]$</td>
<td>$\frac{1}{2}$</td>
</tr>
<tr>
<td>$m$</td>
<td>$\frac{1 + \sigma^2 h - (1 + rh)^2}{(1 + rh)\sigma \sqrt{h}}$</td>
<td>$\frac{e^{2rh} - (1 + rh)^2 + \sigma^2 h}{(1 + rh)\sigma \sqrt{h}}$</td>
<td>NA</td>
</tr>
<tr>
<td>$u$</td>
<td>$1 + rh + \frac{1 - \pi}{\sqrt{\pi(1 - \pi)}} \sigma \sqrt{h}$</td>
<td>$1 + rh + \frac{1 - \pi}{\sqrt{\pi(1 - \pi)}} \sigma \sqrt{h}$</td>
<td>$1 + rh + \sigma \sqrt{h}$</td>
</tr>
<tr>
<td>$d$</td>
<td>$1 + rh - \frac{\pi}{\sqrt{\pi(1 - \pi)}} \sigma \sqrt{h}$</td>
<td>$1 + rh - \frac{\pi}{\sqrt{\pi(1 - \pi)}} \sigma \sqrt{h}$</td>
<td>$1 + rh - \sigma \sqrt{h}$</td>
</tr>
<tr>
<td>Recovers $\sigma$ for finite $N$?</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Prohibits arbitrage for finite $N$?</td>
<td>No</td>
<td>No</td>
<td>No</td>
</tr>
</tbody>
</table>
### Table 2. Anomalies

This table reports on whether four possible anomalies are met. “Yes” indicates that the condition in the column is always met. “No” indicates that the condition is not always met. Proofs are available. Footnotes indicate examples for all “No” cases.

<table>
<thead>
<tr>
<th>Anomalies</th>
<th>$\pi$ (or $\pi^*$) &lt; 1</th>
<th>$\pi$ (or $\pi^*$) &gt; 0</th>
<th>$u &gt; 1$</th>
<th>$d &lt; 1$</th>
</tr>
</thead>
<tbody>
<tr>
<td>CRR</td>
<td>No(^1)</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>RBJRT</td>
<td>No(^2)</td>
<td>Yes</td>
<td>No(^2)</td>
<td>No(^1)</td>
</tr>
<tr>
<td>Chriss</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>No(^1)</td>
</tr>
<tr>
<td>Trigeorgis</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Wil1</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Wil2</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>No(^1)</td>
</tr>
<tr>
<td>JKYABMD1</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>JKYRB2</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>No(^1)</td>
</tr>
<tr>
<td>JKYABMC2</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>No(^1)</td>
</tr>
<tr>
<td>JKYABMD2c</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>No(^1)</td>
</tr>
<tr>
<td>JKYABMD3</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>No(^1)</td>
</tr>
</tbody>
</table>

\(^1\) $r = 0.05$, $\sigma = 0.03$, $T = 1$, $N = 2$

\(^2\) $r = 0.05$, $\sigma = 3.00$, $T = 1$, $N = 2$
Table 3. Some Numerical Examples

The table contains the binomial option value for various time steps ($N$) for a call option with stock price of 100, exercise price of 100, volatility of 0.3, risk-free rate of 0.05, and time to expiration of one year for each of the 11 binomial models. The risk neutral probability is $\pi$ or $\pi^*$ as specified by the authors of the models. The Black-Scholes-Merton option value is 14.23.

<table>
<thead>
<tr>
<th>$N$</th>
<th>CRR</th>
<th>RBJRT</th>
<th>Chriss</th>
<th>Trigeorgis</th>
<th>Wil1</th>
<th>Wil2</th>
<th>JKYABMD1</th>
<th>JKYRB2</th>
<th>JKYABMC2</th>
<th>JKYABMD2c</th>
<th>JKYABMD3</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>16.96</td>
<td>17.00</td>
<td>17.00</td>
<td>16.97</td>
<td>17.79</td>
<td>17.78</td>
<td>16.69</td>
<td>17.17</td>
<td>17.24</td>
<td>16.15</td>
<td>16.65</td>
</tr>
</tbody>
</table>
Table 4. Convergence Time Step for Binomial Models by Moneyness

The table shows the average time step $N$ at which convergence is achieved where the error is defined as $|(b(N) + b(N-1))/2 - BSM|$ where $b(N)$ is the value computed by the given binomial model for time step $N$, BSM is the correct value of the option as computed by the Black-Scholes-Merton model, and convergence is defined as an error of less than 0.01 for all remaining time steps through 100. The exercise price is 100, the risk-free rate 0.05, the volatilities are 0.1, 0.3, and 0.5, and the times to expiration are 0.25, 1.0, and 4.0. Out-of-the-money options have a stock price 10% lower than the exercise price, and in-the-money options have a stock price 10% higher than the exercise price. These parameters combine to create nine options for each moneyness class. A maximum of 100 time steps is used. For models that did not converge by the 100th time step, a value of 100 is inserted.

<table>
<thead>
<tr>
<th>Model</th>
<th>Out-of-the-Money</th>
<th>At-the-Money</th>
<th>In-the-Money</th>
</tr>
</thead>
<tbody>
<tr>
<td>CRR</td>
<td>66.86</td>
<td>40.25</td>
<td>54.29</td>
</tr>
<tr>
<td>RBJRT</td>
<td>55.57</td>
<td>54.30</td>
<td>55.02</td>
</tr>
<tr>
<td>Chriss</td>
<td>55.57</td>
<td>54.30</td>
<td>55.02</td>
</tr>
<tr>
<td>Trigeorgis</td>
<td>62.57</td>
<td>31.91</td>
<td>63.66</td>
</tr>
<tr>
<td>Wil1</td>
<td>66.93</td>
<td>48.08</td>
<td>69.08</td>
</tr>
<tr>
<td>Wil2</td>
<td>66.64</td>
<td>59.85</td>
<td>55.06</td>
</tr>
<tr>
<td>JKYABMD1</td>
<td>59.20</td>
<td>50.51</td>
<td>63.24</td>
</tr>
<tr>
<td>JKYRB2</td>
<td>61.67</td>
<td>60.39</td>
<td>57.34</td>
</tr>
<tr>
<td>JKYABMC2</td>
<td>61.92</td>
<td>60.47</td>
<td>58.58</td>
</tr>
<tr>
<td>JKYABMD2c</td>
<td>61.92</td>
<td>63.55</td>
<td>63.17</td>
</tr>
<tr>
<td>JKYABMD3</td>
<td>63.47</td>
<td>56.88</td>
<td>63.93</td>
</tr>
</tbody>
</table>
Table 5. Convergence Time Step for Binomial Models by Time to Expiration

The table shows the average time step $N$ at which convergence is achieved where the error is defined as $|b(N) + b(N-1)/2 - BSM|$ where $b(N)$ is the value computed by the given binomial model for time step $N$, BSM is the correct value of the option as computed by the Black-Scholes-Merton model, and convergence is defined as an error of less than 0.01 for all remaining time steps through 100. The exercise price is 100, the risk-free rate 0.05, the volatilities are 0.1, 0.3, and 0.5, and the moneyness is 10% out-of-the-money, at-the-money, and 10% in-the-money. The times to expiration are shown in the columns. These parameters combine to create nine options for each time to expiration. A maximum of 100 time steps is used. For models that did not converge by the 100th time step, a value of 100 is inserted.

<table>
<thead>
<tr>
<th>Model</th>
<th>Time to Expiration (T)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>0.25</td>
</tr>
<tr>
<td>CRR</td>
<td>33.33</td>
</tr>
<tr>
<td>RBJRT</td>
<td>43.33</td>
</tr>
<tr>
<td>Chriss</td>
<td>43.33</td>
</tr>
<tr>
<td>Trigeorgis</td>
<td>33.78</td>
</tr>
<tr>
<td>Wil1</td>
<td>38.33</td>
</tr>
<tr>
<td>Wil2</td>
<td>48.44</td>
</tr>
<tr>
<td>JKYABMD1</td>
<td>33.67</td>
</tr>
<tr>
<td>JKYRB2</td>
<td>51.44</td>
</tr>
<tr>
<td>JKYABMC2</td>
<td>53.22</td>
</tr>
<tr>
<td>JKYABMD2c</td>
<td>46.78</td>
</tr>
<tr>
<td>JKYABMD3</td>
<td>43.44</td>
</tr>
</tbody>
</table>
Table 6. Convergence Time Step for Binomial Models by Volatility

The table shows the average time step $N$ at which convergence is achieved where the error is defined as $|(b(N) + b(N-1))/2 - \text{BSM}|$ where $b(N)$ is the value computed by the given binomial model for time step $N$, BSM is the correct value of the option as computed by the Black-Scholes-Merton model, and convergence is defined as an error of less than 0.01 for all remaining time steps through 100. The exercise price is 100, the risk-free rate 0.05, the times to expiration are 0.25, 1.00, and 4.00, and the moneyness is 10% out-of-the-money, at-the-money, and 10% in-the-money. The volatilities are shown in the columns. These parameters combine to create nine options for each time to expiration. A maximum of 100 time steps is used. For models that did not converge by the 100th time step, a value of 100 is inserted.

<table>
<thead>
<tr>
<th>Model</th>
<th>Volatility ((\sigma))</th>
<th>0.10</th>
<th>0.30</th>
<th>0.50</th>
</tr>
</thead>
<tbody>
<tr>
<td>CRR</td>
<td></td>
<td>40.22</td>
<td>61.89</td>
<td>73.44</td>
</tr>
<tr>
<td>RBJRT</td>
<td></td>
<td>25.78</td>
<td>68.89</td>
<td>95.44</td>
</tr>
<tr>
<td>Chriss</td>
<td></td>
<td>25.78</td>
<td>68.89</td>
<td>95.44</td>
</tr>
<tr>
<td>Trigeorgis</td>
<td></td>
<td>38.00</td>
<td>62.33</td>
<td>71.44</td>
</tr>
<tr>
<td>Wil1</td>
<td></td>
<td>40.11</td>
<td>74.67</td>
<td>89.22</td>
</tr>
<tr>
<td>Wil2</td>
<td></td>
<td>29.78</td>
<td>80.11</td>
<td>97.78</td>
</tr>
<tr>
<td>JKYABMD1</td>
<td></td>
<td>42.78</td>
<td>63.22</td>
<td>80.67</td>
</tr>
<tr>
<td>JKYRB2</td>
<td></td>
<td>27.33</td>
<td>82.22</td>
<td>97.78</td>
</tr>
<tr>
<td>JKYABMC2</td>
<td></td>
<td>27.33</td>
<td>82.44</td>
<td>99.67</td>
</tr>
<tr>
<td>JKYABMD2c</td>
<td></td>
<td>45.33</td>
<td>67.78</td>
<td>94.78</td>
</tr>
<tr>
<td>JKYABMD3</td>
<td></td>
<td>48.33</td>
<td>60.22</td>
<td>93.78</td>
</tr>
</tbody>
</table>
Figure 1. Convergence of the Chriss Model to the Black-Scholes-Merton Model

This figure shows the option price obtained by the Chriss model against the Black-Scholes-Merton model (indicated by the solid line) for time steps 10 to 100. The stock price is 100, the exercise price is 100, the risk-free rate is 0.05, the time to expiration is one year, and the volatility is 0.3.
Figure 2. Absolute Value of the Convergence Error for the Chriss Model and Its Order Bound Function

This figure shows the absolute value of the error for the option price obtained by the Chriss model against the Black-Scholes-Merton model for time steps 10 to 100. The stock price is 100, the exercise price is 100, the risk-free rate is 0.05, the time to expiration is one year, and the volatility is 0.3. Because the error bound is linear in logs, a log-log scale is used. The upper bound is the dark shaded line based on an order of convergence of one.
Figure 3. Absolute Value of the Moments and Pseudo-moments for the Chriss Model and Its Order Bound Function

This figure shows the absolute value of the error for the second and third moments and the pseudo-moment as defined by Leisen and Reimer for the option price obtained by the Chriss model against the Black-Scholes-Merton model for time steps 10 to 100. The stock price is 100, the exercise price is 100, the risk-free rate is 0.05, the time to expiration is one year, and the volatility is 0.3. Because the error bound is linear in logs, a log-log scale is used. The upper bound is the dark shaded line based on an order of convergence of 2, which is consistent of order of convergence of the model of one.
Figure 4. Convergence of a General Binomial Model that Prohibits Arbitrage and Allows any Probability between 0 and 1

These figures show the value of the option computed from a general binomial model that assures the absence of arbitrage, recovery of the correct log volatility, and in which the probability can be arbitrarily chosen as indicated. The stock price is 100, the exercise price is 100, the risk-free rate is 0.05, the volatility is 0.30, and the option expires in one year. The horizontal line is the BSM value of 14.23.

(a) $\pi = 1/4$

(b) $\pi = 1/2$

(c) $\pi = 3/4$
Appendix: Why the Probabilities in all of the Models Converge to ½

Let $R = e^{rh}$. A principal requirement for these results is that $\lim_{h \to 0} R = \lim_{h \to 0} u = \lim_{h \to 0} d = 1$. Clearly this result holds for $R$. It must also hold for all definitions of $u$ and $d$ because the volatility per time step must be proportional to $\sqrt{h}$.

(1) For the class of models that assume $\pi = ½$, it is unnecessary to prove convergence. These models include Chriss, Wil2, and JKYABMD3.

(2) For the class of models that assume $ud = e^{2rh}$, proof is shown below. These models include JKYRB2, JKYABMC2, JKYABMD2c.

Given $ud = e^{2rh}$, $\pi$ is then given as

$$\pi = \frac{R - e^{2rh}/u}{u - e^{2rh}/u} = \frac{Ru - e^{2rh}}{u^2 - e^{2rh}}$$

The limit of this expression is undefined. Using L'Hôpital's rule, we require the derivatives of the numerator and denominator with respect to $h$:

$$f = Ru - e^{2rh}$$

$$f' = Ru' + uR' - 2rR^2$$

$$g = 2uu' - 2r^2$$

$$g' = \frac{1}{2} uu' - rR^2.$$ 

We know that $\lim_{h \to 0} R = \lim_{h \to 0} u = 1$, and $\lim_{h \to 0} R' = r$, so

$$\lim_{h \to 0} \pi = \frac{1}{2} \left( \frac{lim u' + r - 2r}{lim u' - r} \right) = \frac{1}{2}.$$ 

Two of the models in this class, JKYRB2 and JKYABMD2c use their own specifications of the risk neutral probability, that is, $\pi^*$. For JKYRB2,

$$\pi^* = \frac{(r - \sigma^2/2)h - \ln d}{\ln u - \ln d}.$$ 

This expression can be shown to converge to ½ even without using L'Hôpital's rule.

$$\pi^* = \frac{(r - \sigma^2/2)h - \ln d}{\ln u - \ln d} = \frac{(r - \sigma^2/2)h - 2rh + \ln u}{2ln u - 2rh}$$

$$= \frac{1}{2} \left( \frac{\ln u - rh - \sigma^2 h/2}{\ln u - rh} \right) = \frac{1}{2} \left( \frac{\ln u - rh - \sigma^2 h/2}{\ln u - rh} \right)$$

$$\lim_{h \to 0} = \frac{1}{2} (1 - 0) = \frac{1}{2}.$$ 

For JKYABMD2c, we have
Here they use $1 + rh$ instead of $e^{rh}$. Here $R = 1 + rh$ and $R' = r$. Substituting these results into the above equation for $f'/g'$ and taking the limit gives $\frac{1}{2}$.

(3) For the class of models $ud = e^{2(r-\sigma^2/2)h}$, proof is shown below. This class is the RBJRT model. We have

$$\pi^* = \frac{1 + rh - d}{u - d}.$$ 

The limit of this expression is undefined. Using L'Hôpital's rule,

$$f' = Ru' + uR' - e^{2(r-\sigma^2/2)h} 2(r - \sigma^2 / 2)$$

$$g' = 2uu' - e^{2(r-\sigma^2/2)h} 2(r - \sigma^2 / 2)$$

To make this expression go to $\frac{1}{2}$, we need $u'$ of the form $c + j\sigma / 2\sqrt{h}$ where $c$ and $j$ are constants. Working with the term in parentheses, we can reduce it to the form $\frac{(2\sqrt{h}(c - r + \sigma^2) + j\sigma) / (2\sqrt{h}(c - r + \sigma^2 / 2) + j\sigma)}$. Taking the limit gives one and the overall limit is, therefore $\frac{1}{2}$. To obtain such a derivative requires that $u$ be linear or approximately linear in $\sigma\sqrt{h}$, which applies to the RBJRT form for $u$, equation (20). This result can occur when $u$ is linear in $\sigma\sqrt{h}$ or exponential as in $e^{\sigma\sqrt{h}}$.

(4) For the class of models that assume $ud = 1$, proof is shown below. This class includes the CRR, Trigeorgis, Wil1, and JKYABMD1 models. Here we have

$$\pi = \frac{R - 1/u}{u - 1/u} = \frac{R - 1}{u - 1}. $$

This expression is undefined in the limit. Using L'Hôpital’s rule,
\[
\pi = \frac{Ru-1}{u^2-1} = \frac{f}{g}
\]
\[
f' = Ru' + uR', \quad g' = 2uu'
\]
\[
\frac{f'}{g'} = \frac{1}{2} \left( \frac{Ru' + uR'}{uu'} \right)
\]

We know that \( \lim_{h \to 0} R = \lim_{u \to 1} = 1 \), and \( \lim_{h \to 0} R' = r \), so
\[
\lim_{h \to 0} \left( \frac{f'}{g'} \right) = \lim_{h \to 0} \left( \frac{1}{2} \left( \frac{u' + r}{u'} \right) \right) = \lim_{h \to 0} \left( \frac{1}{2} + \frac{1}{2} \frac{r}{u'} \right) = \frac{1}{2} + \frac{1}{2} \lim_{h \to 0} \left( \frac{r}{u'} \right)
\]

This expression is \( \frac{1}{2} \) if \( r/u' \) converges to 0. Using the Taylor series approximation of the exponential function and letting \( h^k = 0 \) for all \( k \geq 2 \), we find that

\( CRR, Wil2, Trigeorgis : u \equiv 1 + \sigma \sqrt{h} + \sigma^2 h / 2 \)

\( JKYABMD1 : u \equiv 1 + \sigma \sqrt{h} + rh \)

The derivatives of these two expressions are of the form \( c + \sigma / 2 \sqrt{h} \) where \( c \) is a constant. Hence,
\[
\lim_{h \to 0} \left( \frac{r}{u'} \right) = \lim_{h \to 0} \left( \frac{r}{\sigma + \frac{2}{2\sqrt{h}}} \right) = \frac{0}{\sigma} = 0
\]

Thus, these models work because the up factor is linear or exponential in \( \sigma \sqrt{h} \).

The JKYABMD1 model uses its own version of the risk neutral probability,
\[
\pi^* = \frac{1 + rh - d}{u - d},
\]
where \( 1 + rh \) is used instead of \( e^{rh} \). Following the same steps as before and recognizing that \( R' = r \) gives the desired proof.

The Trigeorgis models uses its own risk neutral probability,
\[
\pi^* = \frac{(r - \sigma^2 / 2)h - \ln d}{\ln u - \ln d}
\]
Convergence is easy to show:
\[
\pi^* = \frac{(r - \sigma^2 / 2)h - \ln d}{\ln u - \ln d} = \frac{(r - \sigma^2 / 2)h + \ln u}{2\ln u}
\]
\[
= \frac{1}{2} \left( \frac{(r - \sigma^2 / 2)h}{\ln u} + \frac{\ln u}{\ln u} \right).
\]
The limit is
$$\lim_{h \to 0} \left( \frac{1}{2} \left( \frac{(r - \sigma^2 / 2)h}{\ln u} + \frac{\ln u}{\ln u} \right) \right) = \frac{1}{2}.$$
References


The Black-Scholes formula for stock indices, currencies and futures

**Michael Carter**

The standard Black-Scholes formula is

\[ c = S_0 N(d_1) - K e^{-rT} N(d_2) \]

where

\[ d_1 = \frac{\ln(F_0/K) + \sigma^2 T/2}{\sigma \sqrt{T}}, \quad d_2 = d_1 - \sigma \sqrt{T} \]

This can be rewritten as

\[
\begin{align*}
\text{Continuous dividend} & \quad \text{Foreign currency} \\
\end{align*}
\]

If the underlying assets pays a continuous dividend yield at the rate \( q \), its forward price is

\[ F_0 = S_0 e^{(r-q)T} \]

and therefore the call option value is

\[
\begin{align*}
\text{Foreign currency} & \quad \text{Foreign currency} \\
\end{align*}
\]

The forward price of a foreign currency is given by

\[ F_0 = S_0 e^{(r-r_f)T} \]
which is known as covered interest parity. Therefore, the value of a foreign currency option is

\[ c = e^{-rT} \left( S_0 e^{(r-f)T} N(d_1) - K N(d_2) \right) = S_0 e^{-rT} N(d_1) - K e^{-rT} N(d_2) \]

with

\[ d_1 = \frac{\ln(F_0/K) + \sigma^2 T/2}{\sigma \sqrt{T}}, \quad d_2 = d_1 - \sigma \sqrt{T} \]

In effect, the foreign currency is a dividend yield \( q = r_f \).

**Future**

The value of a call option on a future is given directly by

\[ c = e^{-rT} \left( F_0 N(d_1) - K N(d_2) \right) \]

with

\[ d_1 = \frac{\ln(F_0/K) + \sigma^2 T/2}{\sigma \sqrt{T}}, \quad d_2 = d_1 - \sigma \sqrt{T} \]

**Generalized Black-Scholes formula**

All these cases can be subsumed in a generalized Black-Scholes formula

\[ c = S_0 e^{(b-r)T} N(d_1) - K e^{-rT} N(d_2) \]

where

\[ d_1 = \frac{\ln(F_0/K) + \sigma^2 T/2}{\sigma \sqrt{T}}, \quad d_2 = d_1 - \sigma \sqrt{T} \]

where \( b \) is the cost-of-carry of holding the underlying security, with

\[ b = r \quad \text{non-dividend paying stock} \]
\[ b = r - q \quad \text{stock with dividend yield} \ q \]
\[ b = r_f \quad \text{currency option} \]
\[ b = 0 \quad \text{futures options} \]

Put-call parity gives

\[ p + S_0 e^{(b-r)T} = c + K e^{-rT} \]

so that
Traditionally, the Black-Scholes model is implemented in dividend yield form

\[ p = (S_0 e^{(b-r)T} N(d_1)) - K e^{-rT} N(d_2)) + K e^{-rT} - S_0 e^{(b-r)T} \]

\[ = K e^{-rT} K(1 - N(d_2)) - S_0 e^{(b-r)T}(1 - N(d_1)) \]

\[ = K e^{-rT} N(-d_1) - S_0 e^{(b-r)T} N(-d_1) \]

with the specific cases being obtained with the following substitutions

- \( q = 0 \) non-dividend paying stock
- \( q = q \) stock with dividend yield \( q \)
- \( q = r_f \) currency option
- \( q = r \) futures options

Note that even if the dividend yield is not constant, the formulae still hold with \( q \) equal to the average annualized dividend yield during the life of the option.

**Appendix**

**THEOREM.** If \( S \) is lognormally distributed and the standard deviation of \( \ln S \) is \( s \) then

\[ \Pr(S > K) = N(d_2) \]

and

\[ E(S | S > K) = E(S) N(d_1) \]

where

\[ d_1 = \frac{\ln(E(S)/K) + s^2/2}{s} \]

\[ d_2 = \frac{\ln(E(S)/K) - s^2/2}{s} \]

Consequently

\[ E[\max(S - K, 0)] = E(S) N(d_1) - K N(d_2) \]  \hspace{1cm} (1)

Proof:

\[ \Pr(S > K) = \text{Prob}(\ln S > \ln K) = N(d_2) \]
For the second part, see Hull (2003: 262-263).

Recognising that (under Black-Scholes assumptions) \( E(S_T) = F_0 = S_0 e^{rT} \) and \( s = \sigma \sqrt{T} \), the Black-Scholes formula for a call option

\[
c = e^{-rT} (F_0 N(d_1) - K N(d_2)) = e^{-rT} (S_0 e^{rT} N(d_1) - K N(d_2)) = S_0 N(d_1) - K e^{rT} N(d_2)
\]

is immediate.
Implementation of Black-Scholes option pricing
Michael Carter, 2004

Option Explicit

'******************************************************************************
' Option values

Function BSCall(S As Double, K As Double, r As Double, q As Double, sigma As
Double, T As Double) As Double
Dim d1 As Double
Dim d2 As Double
d1 = (Log(S / K) + (r - q + sigma * sigma / 2) * T) / (sigma * Sqr(T))
d2 = d1 - sigma * Sqr(T)
BSCall = S * Exp(-q * T) * Application.NormSDist(d1) - K * Exp(-r * T) * Application.NormSDist(d2)
End Function

Function BSPut(S As Double, K As Double, r As Double, q As Double, sigma As
Double, T As Double) As Double
Dim d1 As Double
Dim d2 As Double
d1 = (Log(S / K) + (r - q + sigma * sigma / 2) * T) / (sigma * Sqr(T))
d2 = d1 - sigma * Sqr(T)
BSPut = K * Exp(-r * T) * Application.NormSDist(-d2) - S * Exp(-q * T) * Application.NormSDist(-d1)
End Function

'******************************************************************************
' The Greeks

Function BSCallDelta(S As Double, K As Double, r As Double, q As Double, sigma As
Double, T As Double) As Double
Dim d1 As Double
d1 = (Log(S / K) + (r - q + sigma * sigma / 2) * T) / (sigma * Sqr(T))
BSCallDelta = Exp(-q * T) * Application.NormSDist(d1)
End Function

Function BSPutDelta(S As Double, K As Double, r As Double, q As Double, sigma As
Double, T As Double) As Double
Dim d1 As Double
d1 = (Log(S / K) + (r - q + sigma * sigma / 2) * T) / (sigma * Sqr(T))
BSPutDelta = Exp(-q * T) * (Application.NormSDist(d1) - 1)
End Function
Function BSCallGamma(S As Double, K As Double, r As Double, q As Double, sigma As Double, T As Double) As Double
Dim d1 As Double
d1 = (Log(S / K) + (r - q + sigma * sigma / 2) * T) / (sigma * Sqr(T))
Debug.Print d1
BSCallGamma = Exp(-q * T) * Application.NormDist(d1, 0, 1, False) / (S * sigma * Sqr(T))
End Function

Function BSPutGamma(S As Double, K As Double, r As Double, q As Double, sigma As Double, T As Double) As Double
BSPutGamma = BSCallGamma(S, K, r, q, sigma, T)
End Function
Dealing with dividends

Michael Carter

European options

The Black-Scholes formula is readily adapted to continuous dividends yields (see The Black-Scholes formula for stock indices, currencies and futures).

The price of a dividend paying stock typically falls when the stock goes ex-dividend. A common approach to dealing with discrete dividends is to subtract the present value of the dividends from the current stock price before applying the Black-Scholes formula (Hull 2003: 253). For example, if dividends $d_1$, $d_2$, ..., $d_n$ are anticipated at times $t_1$, $t_2$, ..., $t_n$, the present value of the dividends is

$$D = \sum_{i=1}^{n} e^{r t_i} d_i$$

and the option is valued as

$$c(S - D, K, r, \sigma, t) \text{ or } p(S - D, K, r, \sigma, t)$$

where $c$ and $p$ are the Black-Scholes formulae for call and put options respectively.

This is problematic, not the least because historical volatility measures refer to the stock price including dividends (Fischling 2002).

Bos and Vandemark (2002) propose a simple modification that closely matches numerical results. Instead of subtracting the full present value of future dividends from the current stock price, they propose apportioning each dividend between the current price and the strike price in proportion to the relative time. Specifically, if dividends $d_1$, $d_2$, ..., $d_n$ are anticipated at times $\tau_1$, $\tau_2$, ..., $\tau_n$, they compute "near" and "far" components

$$D_n = \sum_{i=1}^{n} \frac{T - \tau_i}{T} e^{r \tau_i} d_i \quad \text{and} \quad D_f = \sum_{i=1}^{n} \frac{\tau_i}{T} e^{r \tau_i} d_i$$

The option is valued as

$$c(S - D_n, K + D_f, r, \sigma, t) \text{ or } p(S - D_n, K + D_f, r, \sigma, t)$$

where $c$ and $p$ are the Black-Scholes formulae for call and put options respectively.

American options

Dealing with dividends for American options is more complicated, since dividends are closely interwined with the incentives for early exercise. This is discussed in the complementary note American options.
The binomial model

In a risk-neutral world, the total return from the stock must be \( r \). If dividends provide a continuous yield of \( q \), the expected growth rate in the stock price must be \( r - q \). The risk-neutral process for the stock price therefore is

\[
dS = (r - q) S \, dt + \sigma S \, dz
\]

The can be approximated in the simple binomial model by adjusting the risk-neutral probabilities, so that

\[
p u S_0 + (1 - p) d S_0 = S_0 e^{(r-q)\Delta t}
\]

or

\[
p = \frac{e^{(r-q)\Delta t} - d}{u - d}
\]

With this amendment, the binomial model can be used to value European and American options on indices, currencies and futures.

Discrete proportional dividends are also straightforward to incorporate into the binomial model. Whenever the stock pays a proportional dividend, the stock price tree must be adjusted downwards when the stock goes ex-dividend (Hull 2003: 402).

Discrete cash dividends are more difficult, since the adjusted tree becomes non-recombining for nodes after dividend date. This leads to an impractical increase in the number of nodes. We can finesse this problem in an analogous way to the treatment of cash dividends with the Black-Scholes formula.

Assume that the stock price \( S \) has two components - a risky component \( S^* \) with volatility \( \sigma^* \) and the dividend stream \( e^{-rt} D \). Develop a binomial tree to represent the stochastic part \( S^* \) with

\[
S^*_0 = S_0 - e^{-rt} D, \quad p = \frac{e^{\sigma^* \Delta t} - d}{u - d}, \quad u = e^{\sigma^* \Delta t}, \quad d = e^{-\sigma^* \Delta t}
\]

Then add back the present value of the dividends to each node (prior to the ex-dividend date) to obtain a binomial tree representation of \( S \), which can then be used to value contingent claims in the usual way.

This procedure could be enhanced by apportioning the dividends between current price and strike price according to the procedure of Bos and Vandemark discussed above.
Hedging strategies

Michael Carter

- Preliminaries

Introduction

Consider a derivative (or portfolio of derivatives) on a single underlying asset. Its value depends upon the current asset price $S$ and its volatility $\sigma$, the risk-free interest rate $r$, and the time to maturity $t$. That is, $V = f(S, r, \sigma, t)$. (It also depends upon constants like the strike price $K$.) Taking a Taylor series expansion, the change in value over a small time period can be approximated by

$$dV \approx \frac{\partial f}{\partial S} dS + \frac{\partial f}{\partial r} dr + \frac{\partial f}{\partial \sigma} d\sigma + \frac{\partial f}{\partial t} dt + \frac{1}{2} \frac{\partial^2 f}{\partial S^2} dS^2 + \text{other second order terms} + \text{higher order terms}$$

(1)

The partial derivatives in this expansion are known collectively as "the Greeks". They measure the sensitivity of a portfolio to changes in the underlying parameters. Specifically

$$\Delta = \frac{\partial f}{\partial S} \quad \text{Delta measures the sensitivity of the portfolio value to changes in the price of the underlying}$$

$$\rho = \frac{\partial f}{\partial r} \quad \text{Rho measures the sensitivity of the portfolio value to changes in the interest rate}$$

$$\nu = \frac{\partial f}{\partial \sigma} \quad \text{Vega measures the sensitivity}$$

of the portfolio value to changes in the volatility of the underlying

$$\Theta = \frac{\partial f}{\partial t} \quad \text{Theta measures the sensitivity of the portfolio value to the passage of time}$$

$$\Gamma = \frac{\partial^2 f}{\partial S^2} = \frac{\partial \Delta}{\partial S} \quad \text{Gamma measures the sensitivity of delta to changes in the price of the underlying,}$$

or the curvature of the $S - V$ curve.

Substituting in (1), the change in value of the portfolio can be approximated by

$$dV \approx \Delta dS + \rho dr + \nu d\sigma + \Theta dt + \frac{1}{2} \Gamma dS^2$$

(2)

Because differentiation is a linear operator, the hedge parameters of a portfolio are equal to a weighted average of the hedge parameters of its components. In particular, the hedge parameters of a short position are the negative of the hedge parameters of a long position. Consequently, (2) applies equally to a portfolio as to an individual asset. The sensitivity of a portfolio to the risk factors ($S, r, \sigma$) can be altered by changing the composition of the portfolio. It can be reduced by adding assets with offsetting parameters.
The Greeks are not independent. Any derivative (or portfolio of derivatives) \( V = f(S, r, \sigma, t) \) must satisfy the Black-Scholes differential equation

\[
\frac{\partial f}{\partial t} + r S \frac{\partial f}{\partial S} + \frac{1}{2} \sigma^2 S^2 \frac{\partial^2 f}{\partial S^2} = r V
\]

Substituting

\[
\frac{\partial \Pi}{\partial t} = \Theta, \quad \frac{\partial \Pi}{\partial S} = \Delta, \quad \frac{\partial^2 \Pi}{\partial S^2} = \Gamma
\]

it follows that the Greeks must satisfy the following relationship

\[
\Theta + r S \Delta + \frac{1}{2} \sigma^2 S^2 \Gamma = r V
\]

(3)

**Computing the Greeks**

The Greeks of vanilla European options have straightforward formulae, which can be derived from the Black-Scholes formula. The generalized Black-Scholes formulae for European options are

\[
c = S e^{-q T} N(d_1) - K e^{-r T} N(d_2)
\]

\[
p = K e^{-r T} N(-d_2) - S e^{-q T} N(-d_1)
\]

where

\[
d_1 = \frac{\ln(S / K) + (r - q + \sigma^2 T) / 2}{\sigma \sqrt{T}}, \quad d_2 = \frac{\ln(S / K) + (r - q - \sigma^2 T) / 2}{\sigma \sqrt{T}} = d_1 - s
\]

The partial derivatives ("the Greeks") are

<table>
<thead>
<tr>
<th>Delta</th>
<th>Call</th>
<th>Put</th>
</tr>
</thead>
<tbody>
<tr>
<td>( e^{-q T} N(d_1) )</td>
<td>( e^{-q T} (N(d_1) - 1) )</td>
<td></td>
</tr>
<tr>
<td>Gamma</td>
<td>( \frac{e^{-q T} N'(d_1)}{S \sigma \sqrt{T}} )</td>
<td>( \frac{e^{-q T} N'(d_1)}{S \sigma \sqrt{T}} )</td>
</tr>
<tr>
<td>Rho</td>
<td>( K T e^{-r T} N(d_2) )</td>
<td>( -K T e^{-r T} N(-d_2) )</td>
</tr>
<tr>
<td>Vega</td>
<td>( e^{-q T} S \sqrt{T} N'(d_1) )</td>
<td>( e^{-q T} S \sqrt{T} N'(d_1) )</td>
</tr>
<tr>
<td>Theta</td>
<td>( -\frac{e^{-q T} S \sigma N'(d_1)}{2 \sqrt{T}} )</td>
<td>( -\frac{e^{-q T} S \sigma N'(d_1)}{2 \sqrt{T}} )</td>
</tr>
<tr>
<td></td>
<td>( + q e^{-q T} S N(d_1) )</td>
<td>( -q e^{-q T} S N(-d_1) )</td>
</tr>
<tr>
<td></td>
<td>( -r K e^{-r T} N(d_2) )</td>
<td>( -r K e^{-r T} N(-d_2) )</td>
</tr>
</tbody>
</table>
As an example of the derivation, for a call option

\[ \Gamma = \frac{\partial \Delta}{\partial S} = e^{-qT} N'(d_1) \frac{\partial d_1}{\partial S} = e^{-qT} N'(d_1) \frac{1}{S \sigma \sqrt{T}} \]

Calculating vega from the Black-Scholes formula is an approximation, since the formula is derived under the assumption that volatility is constant. Fortunately, it can be shown that it is a good approximation to the vega calculated from a stochastic volatility model (Hull 2003: 318).

Some exotic options (e.g. barrier options) have analogous formulae. However, for most exotic options and vanilla options, the Greeks must be estimated by numerical techniques. Since these are the type of options for which institutions require such information, this motivates interest in the accurate computation of option values and sensitivities.

In principle, the Greeks can be estimated by numerical differentiation. For example,

\[ \Delta = \frac{c(S_1) - c(S_0)}{S_1 - S_0} \quad \text{and} \quad \Gamma = \frac{\Delta(S_1) - \Delta(S_0)}{S_1 - S_0} \]

However, this is not always the most appropriate method, as the small size of the denominator in the limit magnifies errors in the numerator.

**Hedging**

In the previous section, we showed the sensitivity of the value of a portfolio of derivatives of a single underlying to its risk factors can be approximated by

\[ dV \approx \Delta dS + \rho dr + \gamma d\sigma + \Theta dt + \frac{1}{2} \Gamma ds^2 \]

Hedging is the process of modifying the portfolio to reduce or eliminate the stochastic elements on the right-hand side. **Delta-hedging** eliminates the first-term on the right-hand side by making the portfolio delta neutral (\( \Delta = 0 \)). This can be done by taking an offsetting position in the underlying asset, as represented by the tangent to the portfolio at the current asset price.
Delta-gamma hedging also eliminates the last term in (4) by making the portfolio gamma neutral ($\Gamma = 0$). Since the underlying is gamma neutral, delta-gamma hedging requires the addition of other derivatives to the portfolio. Curvature (Gamma) increases as an option approaches maturity, especially for at-the-money options.
Recall the fundamental relationship (3)

$$\Theta + rS \Delta + \frac{1}{2} \sigma^2 S^2 \Gamma = rV$$

For a delta-gamma-neutral portfolio, this reduces to

$$\Theta = rV$$

The portfolio earns the risk-free rate.

The closer that hedging option matches the target option, the more robust will be the hedge provided (i.e. the wider the range of parameter variation that will be neutralised). The hedge may be improved by combining two or more options. For example, combining two options, one with a shorter and one with a longer time to maturity.
would a more accurate match to the gamma of the target option. There is a tradeoff between the robustness of
the hedge (the frequency of hedge adjustments) and the number of options that must be purchased and managed.
The actual performance of a hedge may not reach its theoretical potential (for example, because of model errors
and transaction costs). Consequently, adding too many options to the hedge may give results that are better on
paper than in reality.

A hedge comprising at least two derivatives, in addition to the underlying, can be used to eliminate three terms
in equation (3). A hedge comprising three derivatives, in addition to the underlying, can be used to neutralize all
four stochastic terms in equation (3), eliminating all risk to a first-order approximation.

In principle, a hedge can be found by solving a system of linear equations. Suppose there are $m$ potential hedg-
ing instruments. Let $x_1, x_2, \ldots, x_m$ denote the amount of hedging instrument $j$, and let $x_S$ denote the amount
invested in the underlying. Then, we seek a solution to the following system of equations.

$$
x_S + x_1 \Delta_1 + x_2 \Delta_2 + \ldots + x_m \Delta_m = \Delta
$$
$$
x_1 \Gamma_1 + x_2 \Gamma_2 + \ldots + x_m \Gamma_m = \Gamma
$$
$$
x_1 \nu_1 + x_2 \nu_2 + \ldots + x_m \nu_m = \nu
$$
$$
x_1 \rho_1 + x_2 \rho_2 + \ldots + x_m \rho_m = \rho
$$

Provided that the Greeks of the hedging instruments are linearly independent, there will be a unique solution if
$m = 3$ and multiple solutions if $m > 3$. However, the solutions may not be economically sensible.

Since $T$ appears explicitly in the formula for vega, options of different maturities will be most effective in
hedging against volatility risk. Although interest rate risk can be hedged by options, it may be cost-effective and
certainly more straightforward to hedge interest rate risk by trading bond future contracts, since they are pure
rho instruments, with no impact on delta, gamma or vega.

- **Example**
Rules of thumb

Consider a call option that is at-the-money forward, that is

\[ K = F_0 = S_0 e^{rT} \]

Then the Black-Scholes formula (assuming no dividend yield) simplifies to

\[ c = S_0 (N(d_1) - N(d_2)) \]

where

\[ d_1 = \frac{\ln \left( \frac{S_0}{F_0} \right) + (r + \sigma^2 / 2) T}{\sigma \sqrt{T}} = \frac{\ln \left( \frac{S_0}{S_0} \right) - r T + (r + \sigma^2 / 2) T}{\sigma \sqrt{T}} = \frac{1}{2} \sigma \sqrt{T} \]

\[ d_2 = d_1 - \sigma \sqrt{T} = -\frac{1}{2} \sigma \sqrt{T} \]

Therefore the call option value is

\[ c = S_0 \left( N\left( \frac{1}{2} \sigma \sqrt{T} \right) - N\left( -\frac{1}{2} \sigma \sqrt{T} \right) \right) \]

Provided that \( \sigma \sqrt{T} \) is small, this can be approximated by

\[ c = S_0 \times 0.4 \sigma \sqrt{T} \]

Since the peak of the standard normal density function is \( 1 / \sqrt{2\pi} \approx 0.4 \), the area can be approximated by a rectangle of height 0.4.

The standard normal distribution

![Image of standard normal distribution]

This formula can be inverted to obtain a "rough and ready" estimate of the implied velocity from quoted option prices, using the average of the two nearest-the-money call options.

\[ \sigma = 2.5 \frac{c}{S_0} \frac{1}{\sqrt{T}} \]
Greeks and the binomial method

Numerical differentiation

Delta measures the sensitivity of the option value to changes in the price of the underlying. It is defined as

$$\Delta = \frac{\partial V(S)}{\partial S} = \lim_{dS \to 0} \frac{V(S + dS) - V(S)}{dS}$$

An obvious method to evaluate $\Delta$ is to compute

$$\Delta = \frac{V(S + dS) - V(S)}{dS}$$

for small $dS$. This is known as the forward difference. A better alternative (though more costly to compute) is

$$\Delta = \frac{V(S + dS) - V(S - dS)}{2dS}$$

which is known as the central difference. The other first-order Greeks (rho, theta and vega) can be estimated similarly.

Gamma is the derivative of delta, or the second derivative of $v(S)$. Using central differences, gamma can be estimated by

$$\Gamma = \frac{\Delta(S + \frac{1}{2}dS) - \Delta(S - \frac{1}{2}dS)}{dS} = \frac{\frac{V(S + dS) - V(S)}{dS} - \frac{V(S) - V(S - dS)}{dS}}{dS} = \frac{\frac{V(S + dS) - V(S)}{dS}}{dS} = \frac{\frac{V(S) - V(S - dS)}{dS}}{dS}$$

Numerical differentiation and the binomial tree

Numerical differentiation is not the best method to be applied to the binomial tree. The problem is illustrated in the following diagram.
Improving the binomial method

Michael Carter

- Preliminaries

Introduction
As the number of steps are increased, the binomial method converges to the true value (by the Central Limit Theorem), but the convergence is slow and awkward. This is illustrated in the following graph for an American out-of-the-money put option \( S = 100, K = 90, r = 5\%, \sigma = 30\%, T = 1/2 \). The horizontal axis represents the true value as calculated with a 50,000 step tree.

This pattern repeats indefinitely as the number of steps is increased.
The next graph illustrates the same option with  $K = 110$.  

Clearly, there is a tradeoff between accuracy and efficiency (speed). Various methods are available for improving the performance of the binomial model. These can be classified into two groups depending on whether aimed at

- improving accuracy
- improving efficiency

Typically, success on one front implies a sacrifice on the other.
Improving accuracy

■ Successive averages
A popular technique in practice is to average the results of successive integers, \( n \) and \( n + 1 \).

![Binomial convergence – successive averages](image)

■ Parameterization
In class, we used the simple parameterization

\[
 u = e^{r \sqrt{\Delta t}}, \quad d = e^{-\sigma \sqrt{\Delta t}}, \quad p = \frac{e^{(r-q) \Delta t} - d}{u - d}
\]

where \( \Delta t = \frac{t}{n} \). Some slight improvement in accuracy (at negligible computational cost) can be attained by modifying the parameterization. Two possibilities are:

\[
 u = e^{r \Delta t + \sigma \sqrt{\Delta t}}, \quad d = e^{-\sigma \sqrt{\Delta t}}, \quad p = \frac{e^{(r-q) \Delta t} - d}{u - d}
\]

and

\[
 u = e^{\Delta x}, \quad d = e^{-\Delta x}, \quad \Delta x = \sqrt{\sigma^2 \Delta t + \nu^2 \Delta t^2}, \quad p = \frac{1}{2} \left( 1 + \nu \frac{\Delta t}{\Delta x} \right)
\]

where \( \nu = r - q - \frac{1}{2} \sigma^2 \).
- **Magic numbers**

The oscillations arise from the relationship between the strike price and the terminal nodes of the tree. The graphs reveal that there are particular choices of $n$ that minimize the error in their neighbourhood. These magic numbers depend upon the precise parameters of the option. By tailoring the size of the tree to the particular option, we might obtain more accurate results with smaller trees. This becomes especially important in applying the binomial method to barrier options.

- **Binomial-Black-Scholes**

Convergence can be significantly enhanced using the Black-Scholes formula to evaluate the penultimate nodes. This is known as the Binomial-Black-Scholes method.

![Binomial convergence – out-of-the-money put](https://www.example.com/figure.png)

- **Richardson extrapolation**

Richardson extrapolation is a method to improve an approximation that depends on a step size. Applied to the binomial model, extrapolation attempts to estimate and incorporate the improvement of higher $n$. For example, suppose we assume that errors decline inversely with $n$, so that

$$
P_{n_1} \approx P + \frac{C}{n_1}
$$

$$
P_{n_2} \approx P + \frac{C}{n_2}
$$

where $P$ is the (unknown) true value, $P_{n_1}$ and $P_{n_2}$ are estimates with step size $n_1$ and $n_2$ respectively, and $C$ is an unknown constant. Solving for $P$, we have

$$
P \approx \frac{n_2 P_2 - n_1 P_1}{n_2 - n_1}
$$

In particular, when $n_2 = 2n_1 = n$, we have
which can be alternatively expressed as

\[ P \approx 2 P_n - P_{n/2} = P_n + (P_n - P_{n/2}) \]

It is not helpful when applied to the pure binomial model, but is very effective when applied after Black-Scholes smoothing.
Control variable

A simple and effective technique is to use the binomial method to estimate the early exercise premium, as measured by the difference between the estimated prices of identical American and European options. This estimate is added to the Black-Scholes value, to give the estimated value of the American option.

\[ P = p + (A_b - E_b) \]

where \( p \) is the Black-Scholes value, \( A_b \) is the binomial estimate of the American option, and \( E_b \) is the binomial estimate of a corresponding European option. This is known as the control variable technique.

Rewriting the previous equation

\[ P = A_b + (p - E_b) \]

we observe that the effectiveness of this approach depends upon the degree to which the binomial error in the American option matches that of the European option in sign and magnitude. Chung and Shackleton (2005) explore this issue, provide a methodology for determining the optimal control, and discuss other potential control variables.
Improving efficiency

The diagonal algorithm

Curran (1995) proposed an innovative diagonal algorithm for evaluating binomial trees, which significantly reduced the number of nodes that needed to be evaluated. He reported a 10 to 15-fold increase in speed (with identical accuracy) over the corresponding standard tree. Note that this algorithm achieves a pure increase in efficiency, returning the same result as the standard method. It is equally applicable to extrapolation and control variable techniques.
The diagonal algorithm

*Michael Carter*

Curran (1995) proposed an innovative diagonal algorithm for evaluating binomial trees, which significantly reduced the number of nodes that needed to be evaluated. He reported a 10 to 15-fold increase in speed (with identical accuracy) over the corresponding standard tree. Note that this algorithm achieves a pure increase in efficiency, returning the same result as the standard method. It is equally applicable to extrapolation and control variable techniques.

The diagonal algorithm depends upon two propositions regarding the evolution of option values in a binary tree. They can be illustrated diagramatically as follows:

**Proposition 1**
If it pays to exercise the option in the next period, it pays to exercise immediately.

```
  exercise
  / \                /
exercise/ \           / \
exercise   exercise
```

**Proposition 2**
If it pays to hold the option at some time and asset price, then it pays to hold the option at the same asset price at every earlier price.

```
  ?                        ?
  / \                    / \
hold \ \    hold \ \    hold
  \ \                        \ \\
   ?                        ?
```

Proposition 1 applies provided \( q < r \). The intuition is that, on average, the asset price will grow, and therefore the implicit value will decline. If it is worth exercising in the future, it is worth exercising now. Proposition 2 applies irrespective of the dividend yield (provided that \( u d = 1 \)).

These properties of a binary tree enable two forms of acceleration in the tree.

- By Proposition 2, once an entire diagonal of no exercise (or hold) nodes has been computed, we can jump immediately to the origin, since there are no further exercise nodes. We can evaluate the initial value of the option by computing the discounted expected value of the implicit values along the no-exercise diagonal in a manner similar to computing expected values of the terminal nodes of a European option.

- Provided that \( q < r \), we can start evaluation along the diagonal starting immediately below the strike price, since we know that all nodes below this diagonal will be exercise nodes (Proposition 1), and therefore their value will be equal to the implicit value.
Proof of Proposition 1: Let $S$ denote the current asset price. Assume that both subsequent nodes are exercise nodes. Then the expected future value is

$$\text{FV} = (p(K - u S) + (1 - p)(K - d S)) = K - (p u + (1 - p) d) S$$

Recall that the risk-neutral probability $p$ is such that

$$p u + (1 - p) d = e^{(r-q)\Delta t}$$

Substituting, the expected future value at the subsequent node is

$$\text{FV} = K - e^{(r-q)\Delta t} S$$

Provided that $q \leq r$, the expected future value is less than the current implicit value, that is

$$\text{FV} = K - e^{(r-q)\Delta t} S \leq K - S$$

A fortiori, the discounted future value is less than the current implicit value. That is, $e^{-r\Delta t}(K - S)$ is more than you can expect by waiting. Consequently, the option should be exercised immediately. Note that this is not necessarily the case if $q > r$. In this case, expected capital gains are negative. So the option may become more valuable.

Proposition 2 applies irrespective of the dividend yield. It depends upon the following lemma.

Lemma. Ceteris paribus, the value of an American option increases with time to maturity (Lyuu, Lemma 8.2.1).

Proof of lemma. Suppose otherwise. Sell the more expensive shorter option and buy the one with the longer maturity for a positive cash flow. Let $t$ denote the time at which the shorter option is exercised or expires, and $P_t$ the value of the longer option at this time (assuming a put for example).

Case 1: $P_t > \max(K - S_t, 0)$. Sell the longer option.

Case 2: $P_t \leq \max(K - S_t, 0)$. In this case, the short option will be exercised. Offset this by exercising the longer option.

In either case, we have a positive cash flow at time zero, and a nonnegative cash flow at time $t$.

Proof of Proposition 2. Let $P_u$ and $P_d$ denote the possible values of the option given an asset price of $S$, and let $P_u^{+2}$ and $P_d^{+2}$ denote the possible values of the option two periods later. By assumption, the holding value at time $+2$ is greater than the exercise value. That is

$$e^{-r\Delta t}(p P_u^{+2} + (1 - p) P_d^{+2}) \geq K - S$$

By the lemma, the possible values are at least as great as they will be two periods later.

$$P_u \geq P_u^{+2} \text{ and } P_d \geq P_d^{+2}$$

Therefore, the current holding is at least as great as the current exercise value.

$$e^{-r\Delta t}(p P_u + (1 - p) P_d) \geq e^{-r\Delta t}(p P_u^{+2} + (1 - p) P_d^{+2}) \geq K - S$$
THE CONVERGENCE OF BINOMIAL TREES FOR PRICING
THE AMERICAN PUT

MARK S. JOSHI

Abstract. We study 20 different implementation methodologies for each of 11 different choices of parameters of binomial trees and investigate the speed of convergence for pricing American put options numerically. We conclude that the most effective methods involve using truncation, Richardson extrapolation and sometimes smoothing. We do not recommend use of a European as a control. The most effective trees are the Tian third order moment matching tree and a new tree designed to minimize oscillations.

1. Introduction

There are three main approaches to developing the prices of derivative contracts: Monte Carlo, PDE methods and tree methods. The last are conceptually appealing in that they have a natural financial interpretation, are easy to explain and converge in the limit to the Black–Scholes value. They are also well-adapted to the pricing of derivatives with early exercise features. Whilst tree methods can be shown to be special cases of explicit finite difference methods, the fact that when implementing them we are trying to approximate a probability measure rather than a PDE gives rise to different ideas for acceleration and parameter choices.

Whilst it follows from a suitably modified version of the Central Limit theorem that tree prices converge to the Black–Scholes price, one would also like to know in what way the convergence occurs. In addition, one would like to be able to pick the tree in such a way as to accelerate convergence. This problem has been solved for the European call and put options with Diener and Diener, [8], and Walsh, [21], providing detailed analyzes of convergence, and their work was extended by this author, [13], to show that for a given European option, a binomial tree with arbitrarily high order of convergence exists.

Date: October 9, 2008.
1991 Mathematics Subject Classification. 91B24, 60G40, 60G44. JEL Classification: G13.
Key words and phrases. binomial trees, Richardson extrapolation, options, rate of convergence.
However, for American options only limited progress has been made. This is an important problem in that trading houses may need to price thousands of contracts for book revaluation and VAR calculations. One therefore wishes to be able to obtain a fast accurate price in a minimal amount of time. The crucial issue for such calculations is to find a methodology that achieves a sufficiently accurate price quickly rather than which is asymptotically best. Staunton [18] has examined various methodologies for approximating American put including explicit finite differences, implicit finite differences and analytic approximations, as well as trees. He conclude that the Leisen–Reimer tree with the acceleration techniques of extrapolation and truncation is best. However, he does not consider other tree methodologies: the motivation for this tree choice seems to be that the Leisen–Reimer tree is the most effective tree without acceleration techniques and that these make it faster. However, this does not address the possibility that a tree does poorly without acceleration may do better with it. Our objective here is to find a fast binomial tree by examining many choices of parameters and accelerations in order to find which is fastest.

It is known that for certain trees that the American put option has order 1 convergence, [15] [17], but higher order convergence has not been established for any choice of tree. Since the only real requirements on a binomial tree are that the mean and variance in the risk-neutral measure are asymptotically correct, even for a self-similar tree in which every node is the same, there are infinite number of possible trees. For example, one can discretize the real-world measure and then pass to the risk-neutral measure and gain a different tree for each choice of the real-world drift. These will all converge to the true price but will differ for any finite number of steps. There are by now a large number of choices of parameters for trees, in this paper, we focus on eleven of these which we believe have the most interesting features, to attempt to do all possibilities would have resulted in an impossibly bloated paper.

There is also the option of using trinomial trees and one can ask similar questions in that case. We defer that work to the sequel [3] where similar conclusions are drawn and, in particular, we see that the best binomial tree found here is better than the best trinomial tree.

Many suggestions have been made for methodologies for improving convergence for individual trees. The ability to use these is independent of the choice of tree. We discuss some of the acceleration suggestions that have been made. The first is due to Hull and White, [9], with this approach one prices a European option with the same characteristics as the American option on the same tree, and then adjusts the American option price by assuming it has the same error as the European option. This can be viewed as a control variate technique. We can expect it to do well
(in terms of speed/accuracy tradeoff) when the European option is poorly priced and badly when it is priced very accurately.

Broadie and Detemple, [2], suggested two modifications. The first of these is to replace the price at the second last layer of nodes with the price given by the Black–Scholes formula. The idea being that since one is allowing no exercise opportunities between steps and we are approximating the Black–Scholes model, this ought to give a more accurate price. In addition, the Black–Scholes formula should give a price that smoothly varies and so this should make the price smoother as a function of steps. We shall refer to this as the smoothing technique.

Their second suggestion was to use Richardson extrapolation (RE) to remove the first order term as far as possible. One therefore extrapolates as if the lead term was of the form $A/n$ although it is not. Broadie and Detemple showed that the two techniques of smoothing and RE together resulted in effective speed-ups for the CRR tree.

Staunton, [18], examined the convergence of binomial trees using truncation. In particular, the tree is pruned so that nodes more than 6 standard deviations from the mean in log space are not evaluated. This results in an acceleration since it take less time to develop the tree for a given number of steps, whilst behaviour more than six standard deviations has very little effect on the price. He shows that the Leisen–Reimer tree with Richardson extrapolation and truncation is very effective. Staunton's work followed on from that of Andricopoulos, Widdicks, Duck, and Newton, [1], who had previously suggested curtailing the range of a tree according to distance from mean and strike.

Since all these techniques can be implemented independently, we therefore have $2^4$ different ways to improve each binomial tree. In addition, there is a question when using Richardson extrapolation and smoothing together whether one matches the smoothing times between the small and large numbers of steps. This means that there are a total of $20$ different ways to implement each tree.

In addition, there is now a large number of different ways to choose the parameters of a binomial tree, depending upon what characteristics one wishes to emphasize. For example, one can attempt to match higher moments, or to obtain smooth convergence, or achieve higher order convergence for a specific European option. We will examine 11 of these choices in this paper.

This results in $220$ different ways to price an American put option. It is not at all obvious which will perform best since some trees will perform well in combination with some acceleration techniques and badly with others. In this paper, we perform a comparison of all these methods.
running a large number of options for each case, and using a Leisen–
Reimer tree with a large number of steps and Richardson extrapolation as
a benchmark.

We find that the best choice of tree depends on how one defines error,
but that the two best trees are the Tian third moment-matching tree with
smoothing, Richardson extrapolation and truncation, and a new tree using
a time-dependent drift with extrapolation and truncation.

The structure of binomial trees and our eleven choices of parameters
are discussed in Section 2. The different ways these can be accelerated
is discussed in Section 3. We present numerical results in Section 4 and
conclude in Section 5.

I am grateful to Chris Beveridge, Mark Broadie, Nick Denson, Christo-
pher Merrill, Ken Palmer and Mike Staunton for their comments on an
earlier version of this paper.

2. Choices of binomial tree parameters

We quickly review our 11 choices of tree. A node in a tree is specified
by three things:

1. the probability of an up move $p$,
2. the multiplier on the stock price for an up move, $u$,
3. the multiplier on the stock price for a down move, $d$.

Typically, trees are self-similar in that every node is the same in a relative
sense. Only one of our choices, the split tree, will not be self-similar. A
sequence of trees is therefore a specification of $p$, $u$ and $d$ as a function
of the number of steps. If we require the tree to be risk-neutral then $p$
is determined by $u$ and $d$ via the usual formula

$$ p = \frac{e^{r\Delta T} - d}{u - d}, $$

with

$$ \Delta T = \frac{T}{N}. $$

(Only one of our trees, the Jarrow–Rudd tree, is not risk neutral.) A risk-
neutral tree is therefore a pair of sequences $u_n$ and $d_n$. To keep $p$
between zero and one, we must have

$$ d_n < e^{r\Delta T} < u_n. $$

We work in the Black–Scholes model with the usual parameters: $T$ is
maturity, $r$ is the continuously compounding risk-free rate, $S_t$ is the stock
price and $\sigma$ is the volatility. We can also use $\mu$ the real-world drift when
constructing the tree if we choose: its choice may affect how convergence
occurs although it does not affect the limit.
The choice of $u_n$ and $d_n$ is constrained to ensure that the limiting tree is the Black–Scholes model. Since $p_n$ constrains that the mean is correct, we have one essential condition left: the variances must converge correctly. Since we have two sequences and only one condition, there is still quite a lot of flexibility.

We first discuss the 10 trees that are self-similar. The Cox–Ross–Rubinstein (CRR) tree, [7], is the oldest tree:

$$u_n = e^{\sigma \sqrt{\Delta T}}, \quad d_n = e^{-\sigma \sqrt{\Delta T}}. \quad (2.3)$$

The Tian tree, [19], uses the extra degree of freedom to match the first three moments exactly for all $n$ rather than just the first two in the limit. It takes

$$u_n = \frac{1}{2} r_n v_n \left( v_n + 1 + \left(v_n^2 + 2v_n - 3\right)^{\frac{1}{2}} \right), \quad (2.5)$$
$$d_n = \frac{1}{2} r_n v_n \left( v_n + 1 - \left(v_n^2 + 2v_n - 3\right)^{\frac{1}{2}} \right), \quad (2.6)$$
$$r_n = e^{r \Delta T}, \quad (2.7)$$
$$v_n = e^{\sigma^2 \Delta T}. \quad (2.8)$$

The Jarrow–Rudd (JR), [10], tree is not a risk-neutral tree and, in fact, seems to be the only non-risk-neutral tree in common use:

$$u_n = e^{\mu \Delta T + \sigma \sqrt{\Delta T}}, \quad (2.9)$$
$$d_n = e^{\mu \Delta T - \sigma \sqrt{\Delta T}}, \quad (2.10)$$
$$\mu = r - \frac{1}{2} \sigma^2, \quad (2.11)$$
$$p = \frac{1}{2}. \quad (2.12)$$

A simple modification of the Jarrow–Rudd tree is to take the value of $p$ that makes the tree risk-neutral. We shall refer to this as the Jarrow–Rudd risk-neutral tree (JRRN). This has also been studied by Jarrow and Turnbull, [11].

It follows from the standard analysis of the binomial tree that one can modify the CRR tree by taking an arbitrary real-world drift $\mu$ so

$$u_n = e^{\mu \Delta T + \sigma \sqrt{\Delta T}}, \quad (2.13)$$
$$d_n = e^{\mu \Delta T - \sigma \sqrt{\Delta T}}.$$
(See for example, [12].) One choice is to take $\mu = \frac{1}{T}(\log K - \log S_0)$, thus guaranteeing that the tree is centred on the strike in log space. This was done in [13] and we shall refer to that tree as the \textit{adjusted tree}.

A similar approach has previously been suggested by Tian, [20], who suggested moving the tree slightly so that the strike of the option would land on a node in such a way as to minimize distortion. We shall refer to this as the \textit{flexible} tree.

Chang and Palmer, [5], also suggest a similar tree but make the strike lie half-way between two nodes to obtain smoother convergence for European options. We shall refer to this as the \textit{CP} tree.

Leisen and Reimer, [16], suggested changing point of view to first specifying probabilities of an up move in both stock and bond measures. These two quantities then determine the up and down moves. The probabilities are chosen by using inversions of normal approximations to binomials to get binomial approximations of normals. They suggest three different trees and we will use the one they label (C) here; since that is the one which appears to be in most common use [18]. Their tree had the features of only being defined for odd numbers of steps and being approximately centred on the option strike. This tree is known to have second order convergence for European options, [14].

In [14], the analysis of Diener and Diener was extended and a tree with third order convergence for European options, and a very small third order lead term is explicitly constructed. We shall refer to this tree as \textit{J4}. It is only defined for odd numbers of steps. This tree agrees with the Leisen–Reimer (C) tree to order 2.5 in the way the probabilities are specified. Since American options typically have first order convergence, we can expect the two trees to have similar convergence behaviour.

Another choice due to Chriss, [6], is to modify the \textit{u} and \textit{d} in the Jarrow–Rudd model. We let

$$X = \frac{2e^{r\Delta T}}{u + d}$$

and multiply \textit{u} and \textit{d} by $X$. This can be viewed as a symmetrized version of JRRN. The tree is risk-neutral.

Our final tree is the only one that is not self-similar. Our motivation is that whilst it is known that the Leisen–Reimer (C) tree has second order convergence for European options, it can actually perform worse for in-the-money American options [16]. This suggests that there is some odd interaction between the exercise boundary and the tree in the money. We therefore modify the adjusted tree above to use a time-dependent drift. In
particular, if the integer part of \( n/2 \) is \( k \), then we set

\[
t_1 = \frac{tk}{n},
\]
\[
\mu_1 = \frac{\log K - \log S_0}{t_1},
\]
\[
\mu_2 = 0
\]

and for the first \( k \) steps, we use drift \( \mu_1 \) and for the rest we use \( \mu_2 \). The up and down moves are then defined as in equation (2.13). The idea here is that in the first half we use a strong time-dependence to get the centre of the tree at the same level as strike, and then in the second half, we have no drift. We shall refer to this tree as the split tree.

It is worth noting that the trees designed to have smooth and/or higher order convergence have node placement determined by the strike of the option, and for those trees, we therefore have to build a different tree for each option. This is not, however, true for the others including, in particular, the Tian 3rd moment matching tree.

We remark that there are other possible choices and for a review of a different set of 11 choices for pricing European options we refer the reader to [4]. Our choices here were motivated by the desire to include

- higher order convergence for Europeans trees;
- the most popular and oldest trees e.g. CRR, Jarrow–Rudd, and JRRN;
- the theoretically nicest trees, e.g. the higher order moment matching;
- trees with nice lead order terms, e.g. the Chang–Palmer tree, the adjusted tree, and the flexible tree of Tian.

Whilst 10 of our 11 trees have previously been studied most of them have not been studied in combination with acceleration techniques so of our 220 trees, we estimate that at least 200 have not previously been examined.

3. The implementation choices

In this section, we list the implementation choices which can be applied to any tree and define a key for our numeric results.

Our first implementation option is truncation. We only develop the tree as far as 6 standard deviations from the mean in log-space computed in the risk-neutral measure. At points on the edge of the truncated tree, we take the continuation value to be given the Black–Scholes formula for a European option. The probability of a greater than six standard deviation move is \( 1E^{-9} \). The difference between the European and American prices will be slight so far out-of-the-money, and so far in-the-money the option will generally be behind the exercise boundary. These facts together
mean that truncation has minimal effect on the price: typical effects are around $1E^{-12}$. However, for large numbers of steps it can have large effects on speed of implementation since the number of nodes no longer grows quadratically. For small numbers of nodes, it can be slightly slower because of the extra Black–Scholes evaluations. The use of truncation in tree pricing was suggested by Andicropoulos, Widdicks, Duck, and Newton, [1], and refined by Staunton [18].

We note that the location of the truncation will vary according to volatility and time. There are clearly many other ways to carry out truncation. Our motivation here was to use a methodology that was sure to have minimal impact on price and we have therefore not examined the trade-off between location of the truncation boundary and speed. Nor have we examined the issue of whether it is better to use the intrinsic value at the boundary rather than the Black–Scholes prices. A full analysis would require one to take into account the fact that one can truncate at the edge of a narrower space when using the Black–Scholes price. We leave this issue to future work.

Our second implementation option is control variates. Given a binomial tree, one prices both the American put and the European put. If $P_A$ is the tree price of the American put, $P_E$ that of the European and $P_{BS}$ that given by the Black–Scholes formula, we take the error controlled price to be

$$
\hat{P}_A = P_A + P_{BS} - P_E.
$$

Note that we can expect this to perform well when the European price is poor, but that the error will change little when it is good. It does, however, take a substantial amount of extra computational time. In particular, when the order of convergence of the European option is higher than that of the American option, we can expect little gain. This approach is due to Hull and White, [9].

Our third implementation option is Richardson extrapolation. If the price after $n$ steps is

$$
X_n = \text{TruePrice} + \frac{E}{n} + o(1/n),
$$

then taking

$$
Y_n = A_n X_n + B_n X_{2n+1}
$$

with $A_n$ and $B_n$ satisfying

$$
A_n + B_n = 1.0,
$$

$$
\frac{A_n}{n} + \frac{B_n}{2n+1} = 0.0,
$$

respectively.
then we get

\[ Y_n = \text{TruePrice} + o(1/n). \]

We therefore take

\[ A_n = 1 - \left(1 - \frac{n}{2n + 1}\right)^{-1}, \quad (3.2) \]

\[ B_n = \left(1 - \frac{n}{2n + 1}\right)^{-1}. \quad (3.3) \]

Whilst the error for an American put will not be of the form in (3.1), if it is of this form plus a small oscillatory term, Richardson extrapolation will still reduce the size of the error. One way to reduce the oscillations is to use smoothing. Broadie and Detemple, [2], suggested using smoothing and Richardson extrapolation together.

Our fourth implementation option is smoothing. Inside the tree model, there will no exercise opportunities within the final step, so the derivative is effectively European. This suggests that a more accurate price can be obtained by using the Black–Scholes formula for this final step. With this technique we therefore replace the value at each node in the second final layer with the maximum of the intrinsic and the Black–Scholes value.

Since we can use each of these techniques independently of the others, this yields \(2^4\) different choices. We also consider an extra choice which is relevant when doing both smoothing and Richardson extrapolation. It is possible that making the tree with \(n\) and \(2n + 1\) smooth at the same time will result in better extrapolation than smoothing both of them at the last possible time which will be different for the two trees. We can therefore smooth at the first step after \(\frac{(n - 1)T}{n}\). This yields an extra 4 trees which we will refer to as being matched.

### 4. Numerical results

In order to assess the speed/accuracy trade-off of various tree methodologies without being influenced by special cases, an approach based on computing the root-mean-square (rms) error was introduced by Broadie and Detemple, [2]. One picks option parameters from a random distribution and assesses the pricing error by using a model with a large number of steps as the true value. One then looks at the number of option evaluations per second against the rms error.

Since we want to be clear that our results do not depend on particular choices of random distribution, we use identical parameters to that of Leisen, [17], and proceed as follows: volatility is distributed uniformly between 0.1 and 0.6. The time to maturity is, with probability 0.75, uniform between 0.1 and 1.00 years and, with probability 0.25, uniform between
1.0 and 5.0 years. We take the strike price, $K$, to be 100 and take the initial asset price $S_0$ to be uniform between 70 and 130. The continuously compounding rate, $r$, is, with probability 0.8, uniform between 0.0 and 0.10 and, with probability 0.2, equal to 0.0.

Some authors, [22], [18], have suggested using a model set of 16 extreme cases. Whilst this is probably enough when comparing a small number of models, here we will be doing 220 different models and want the number of test cases to be greater than the number of models. We therefore used 2200 cases and used the same set of options for each of the 220 models.

When computing the rms error, Leisen following Broadie and Detemple suggests using the relative error and dropping any cases where the true value is below 0.5 in order to avoid small absolute errors on small values distorting the results. Whilst this is reasonable, it is also criticizable in that it is particularly lenient in the hardest cases. For a deeply out-of-the-money option, the value will often be less than 0.5 so these are neglected. For a deeply in-the-money option most of the value will be the intrinsic value, so a large error on the model-dependent part may translate into a small error in relative terms.
We therefore introduce a new error measure which is intended to retain the good features of the Broadie–Detemple approach whilst excising the not so good ones. We therefore take the modified relative error to be

\[ \frac{\text{TreePrice} - \text{TruePrice}}{0.5 + \text{TruePrice} - \text{IntrinsicValue}}. \]

This has the virtue of stopping small errors in small prices appearing to be large whilst still taking deeply in- and out-of-the-money options into account. We also only assess the model-dependent part of the price.

For each of the eleven trees discussed, we run the tree with each of the 20 options according to the keys in Table 3.1. We restrict to trees with odd numbers of steps, since some trees, e.g. Leisen–Reimer, are only defined in that case. For our model prices we used the Leisen–Reimer tree with 5001 steps and Richardson extrapolation; this is following the choice of Staunton [18]. All timings are done with a 3 GigaHertz single core Pentium 4 processor.

We ran each tree with the following numbers of steps

25, 51, 101, 201, 401, 801.

We then used linear interpolation of log time against log error to estimate the time required to find an absolute rms error of 1E-3, a modified relative rms error of 1E-3 and a relative rms error (Broadie-Detemple) of 0.5E-4. The difference in target values expressing the fact that the Broadie-Detemple measure is more lenient.

From studying tables 4.1, 4.2, and 4.3. We see various effects. The most marked one is that Richardson extrapolation is very effective when the tree has been smoothed either by adapting the tree to the strike, or by using the BS formula. In particular, the unadapted trees CRR, JR, JRRN, Tian and Chriss do very badly in cases 8 through 11, but do much better in cases 12 and higher, reflecting the Black–Scholes smoothing.

The control methodology is useful when the error is large, but when the price is accurate without it, adding it in merely slows things down. This suggests it is no longer a worthwhile technique for this problem. In particular, the key of 15 almost always does worse than the key of 13 with the only exceptions being the Chang–Palmer and flexible trees using the Broadie–Detemple error measure.

Depending upon on our error methodology the most effective trees for this test are Tian 13 (absolute and Broadie-Detemple) and split 8 (modified relative.) Note, however, that split 9 (i.e. with truncation) is almost as good as split 8, and, in fact, on detailed analysis, Table 4.4, we see that the reason is that 25 steps is too many to get an error of 1E − 3. The time has therefore been extrapolated giving the appearance that the untruncated
<table>
<thead>
<tr>
<th>Key</th>
<th>CP</th>
<th>CRR</th>
<th>J4</th>
<th>JR</th>
<th>JRRN</th>
<th>LR</th>
<th>Tian</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1.8</td>
<td>4.9</td>
<td>16.1</td>
<td>8.4</td>
<td>18.4</td>
<td>16.9</td>
<td>0.23</td>
</tr>
<tr>
<td>2</td>
<td>7.2</td>
<td>16.3</td>
<td>5.0</td>
<td>12.9</td>
<td>1.3</td>
<td>3.7</td>
<td>16.0</td>
</tr>
<tr>
<td>3</td>
<td>1.9</td>
<td>5.1</td>
<td>5.8</td>
<td>14.7</td>
<td>8.9</td>
<td>19.3</td>
<td>0.24</td>
</tr>
<tr>
<td>4</td>
<td>1.8</td>
<td>4.9</td>
<td>6.4</td>
<td>16.0</td>
<td>8.4</td>
<td>18.4</td>
<td>0.24</td>
</tr>
<tr>
<td>5</td>
<td>0.3</td>
<td>1.0</td>
<td>0.8</td>
<td>2.8</td>
<td>0.3</td>
<td>1.1</td>
<td>0.9</td>
</tr>
<tr>
<td>6</td>
<td>0.3</td>
<td>1.0</td>
<td>0.8</td>
<td>2.8</td>
<td>0.3</td>
<td>1.1</td>
<td>0.9</td>
</tr>
<tr>
<td>7</td>
<td>0.3</td>
<td>1.0</td>
<td>0.8</td>
<td>2.8</td>
<td>0.3</td>
<td>1.1</td>
<td>0.9</td>
</tr>
<tr>
<td>8</td>
<td>0.3</td>
<td>1.0</td>
<td>0.8</td>
<td>2.8</td>
<td>0.3</td>
<td>1.1</td>
<td>0.9</td>
</tr>
<tr>
<td>9</td>
<td>0.3</td>
<td>1.0</td>
<td>0.8</td>
<td>2.8</td>
<td>0.3</td>
<td>1.1</td>
<td>0.9</td>
</tr>
<tr>
<td>10</td>
<td>0.3</td>
<td>1.0</td>
<td>0.8</td>
<td>2.8</td>
<td>0.3</td>
<td>1.1</td>
<td>0.9</td>
</tr>
<tr>
<td>11</td>
<td>0.3</td>
<td>1.0</td>
<td>0.8</td>
<td>2.8</td>
<td>0.3</td>
<td>1.1</td>
<td>0.9</td>
</tr>
<tr>
<td>12</td>
<td>0.3</td>
<td>1.0</td>
<td>0.8</td>
<td>2.8</td>
<td>0.3</td>
<td>1.1</td>
<td>0.9</td>
</tr>
<tr>
<td>13</td>
<td>0.3</td>
<td>1.0</td>
<td>0.8</td>
<td>2.8</td>
<td>0.3</td>
<td>1.1</td>
<td>0.9</td>
</tr>
</tbody>
</table>

Table 4.1: Number of option evaluations a second with an absolute rms error of 1E-3.

<table>
<thead>
<tr>
<th>Key</th>
<th>0</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
<th>11</th>
<th>12</th>
<th>13</th>
<th>14</th>
<th>15</th>
<th>16</th>
<th>17</th>
<th>18</th>
<th>19</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>1</td>
<td>35</td>
<td>96</td>
<td>19</td>
<td>56</td>
<td>17</td>
<td>48</td>
<td>70</td>
<td>27</td>
<td>73</td>
<td>27</td>
<td>48</td>
<td>70</td>
<td>27</td>
<td>47</td>
<td>72</td>
<td>27</td>
<td>48</td>
<td>70</td>
<td>27</td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>56</td>
<td>17</td>
<td>48</td>
<td>70</td>
<td>27</td>
<td>48</td>
<td>70</td>
<td>27</td>
<td>48</td>
<td>70</td>
<td>27</td>
<td>48</td>
<td>70</td>
<td>27</td>
<td>48</td>
<td>70</td>
<td>27</td>
<td>48</td>
<td>70</td>
</tr>
<tr>
<td></td>
<td>3</td>
<td>96</td>
<td>19</td>
<td>56</td>
<td>17</td>
<td>48</td>
<td>70</td>
<td>27</td>
<td>48</td>
<td>70</td>
<td>27</td>
<td>48</td>
<td>70</td>
<td>27</td>
<td>48</td>
<td>70</td>
<td>27</td>
<td>48</td>
<td>70</td>
<td>27</td>
</tr>
<tr>
<td></td>
<td>4</td>
<td>17</td>
<td>48</td>
<td>70</td>
<td>27</td>
<td>48</td>
<td>70</td>
<td>27</td>
<td>48</td>
<td>70</td>
<td>27</td>
<td>48</td>
<td>70</td>
<td>27</td>
<td>48</td>
<td>70</td>
<td>27</td>
<td>48</td>
<td>70</td>
<td>27</td>
</tr>
<tr>
<td></td>
<td>5</td>
<td>48</td>
<td>70</td>
<td>27</td>
<td>48</td>
<td>70</td>
<td>27</td>
<td>48</td>
<td>70</td>
<td>27</td>
<td>48</td>
<td>70</td>
<td>27</td>
<td>48</td>
<td>70</td>
<td>27</td>
<td>48</td>
<td>70</td>
<td>27</td>
<td>48</td>
</tr>
<tr>
<td></td>
<td>6</td>
<td>70</td>
<td>27</td>
<td>48</td>
<td>70</td>
<td>27</td>
<td>48</td>
<td>70</td>
<td>27</td>
<td>48</td>
<td>70</td>
<td>27</td>
<td>48</td>
<td>70</td>
<td>27</td>
<td>48</td>
<td>70</td>
<td>27</td>
<td>48</td>
<td>70</td>
</tr>
<tr>
<td></td>
<td>7</td>
<td>27</td>
<td>48</td>
<td>70</td>
<td>27</td>
<td>48</td>
<td>70</td>
<td>27</td>
<td>48</td>
<td>70</td>
<td>27</td>
<td>48</td>
<td>70</td>
<td>27</td>
<td>48</td>
<td>70</td>
<td>27</td>
<td>48</td>
<td>70</td>
<td>27</td>
</tr>
<tr>
<td></td>
<td>8</td>
<td>48</td>
<td>70</td>
<td>27</td>
<td>48</td>
<td>70</td>
<td>27</td>
<td>48</td>
<td>70</td>
<td>27</td>
<td>48</td>
<td>70</td>
<td>27</td>
<td>48</td>
<td>70</td>
<td>27</td>
<td>48</td>
<td>70</td>
<td>27</td>
<td>48</td>
</tr>
<tr>
<td></td>
<td>9</td>
<td>70</td>
<td>27</td>
<td>48</td>
<td>70</td>
<td>27</td>
<td>48</td>
<td>70</td>
<td>27</td>
<td>48</td>
<td>70</td>
<td>27</td>
<td>48</td>
<td>70</td>
<td>27</td>
<td>48</td>
<td>70</td>
<td>27</td>
<td>48</td>
<td>70</td>
</tr>
<tr>
<td></td>
<td>10</td>
<td>27</td>
<td>48</td>
<td>70</td>
<td>27</td>
<td>48</td>
<td>70</td>
<td>27</td>
<td>48</td>
<td>70</td>
<td>27</td>
<td>48</td>
<td>70</td>
<td>27</td>
<td>48</td>
<td>70</td>
<td>27</td>
<td>48</td>
<td>70</td>
<td>27</td>
</tr>
<tr>
<td></td>
<td>11</td>
<td>48</td>
<td>70</td>
<td>27</td>
<td>48</td>
<td>70</td>
<td>27</td>
<td>48</td>
<td>70</td>
<td>27</td>
<td>48</td>
<td>70</td>
<td>27</td>
<td>48</td>
<td>70</td>
<td>27</td>
<td>48</td>
<td>70</td>
<td>27</td>
<td>48</td>
</tr>
<tr>
<td></td>
<td>12</td>
<td>70</td>
<td>27</td>
<td>48</td>
<td>70</td>
<td>27</td>
<td>48</td>
<td>70</td>
<td>27</td>
<td>48</td>
<td>70</td>
<td>27</td>
<td>48</td>
<td>70</td>
<td>27</td>
<td>48</td>
<td>70</td>
<td>27</td>
<td>48</td>
<td>70</td>
</tr>
<tr>
<td></td>
<td>13</td>
<td>27</td>
<td>48</td>
<td>70</td>
<td>27</td>
<td>48</td>
<td>70</td>
<td>27</td>
<td>48</td>
<td>70</td>
<td>27</td>
<td>48</td>
<td>70</td>
<td>27</td>
<td>48</td>
<td>70</td>
<td>27</td>
<td>48</td>
<td>70</td>
<td>27</td>
</tr>
<tr>
<td></td>
<td>14</td>
<td>48</td>
<td>70</td>
<td>27</td>
<td>48</td>
<td>70</td>
<td>27</td>
<td>48</td>
<td>70</td>
<td>27</td>
<td>48</td>
<td>70</td>
<td>27</td>
<td>48</td>
<td>70</td>
<td>27</td>
<td>48</td>
<td>70</td>
<td>27</td>
<td>48</td>
</tr>
<tr>
<td></td>
<td>15</td>
<td>70</td>
<td>27</td>
<td>48</td>
<td>70</td>
<td>27</td>
<td>48</td>
<td>70</td>
<td>27</td>
<td>48</td>
<td>70</td>
<td>27</td>
<td>48</td>
<td>70</td>
<td>27</td>
<td>48</td>
<td>70</td>
<td>27</td>
<td>48</td>
<td>70</td>
</tr>
<tr>
<td></td>
<td>16</td>
<td>27</td>
<td>48</td>
<td>70</td>
<td>27</td>
<td>48</td>
<td>70</td>
<td>27</td>
<td>48</td>
<td>70</td>
<td>27</td>
<td>48</td>
<td>70</td>
<td>27</td>
<td>48</td>
<td>70</td>
<td>27</td>
<td>48</td>
<td>70</td>
<td>27</td>
</tr>
<tr>
<td></td>
<td>17</td>
<td>48</td>
<td>70</td>
<td>27</td>
<td>48</td>
<td>70</td>
<td>27</td>
<td>48</td>
<td>70</td>
<td>27</td>
<td>48</td>
<td>70</td>
<td>27</td>
<td>48</td>
<td>70</td>
<td>27</td>
<td>48</td>
<td>70</td>
<td>27</td>
<td>48</td>
</tr>
<tr>
<td></td>
<td>18</td>
<td>70</td>
<td>27</td>
<td>48</td>
<td>70</td>
<td>27</td>
<td>48</td>
<td>70</td>
<td>27</td>
<td>48</td>
<td>70</td>
<td>27</td>
<td>48</td>
<td>70</td>
<td>27</td>
<td>48</td>
<td>70</td>
<td>27</td>
<td>48</td>
<td>70</td>
</tr>
<tr>
<td></td>
<td>19</td>
<td>27</td>
<td>48</td>
<td>70</td>
<td>27</td>
<td>48</td>
<td>70</td>
<td>27</td>
<td>48</td>
<td>70</td>
<td>27</td>
<td>48</td>
<td>70</td>
<td>27</td>
<td>48</td>
<td>70</td>
<td>27</td>
<td>48</td>
<td>70</td>
<td>27</td>
</tr>
<tr>
<td>key</td>
<td>0</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>4</td>
<td>5</td>
<td>6</td>
<td>7</td>
<td>8</td>
<td>9</td>
<td>10</td>
<td>11</td>
<td>12</td>
<td>13</td>
<td>14</td>
<td>15</td>
<td>16</td>
<td>17</td>
<td>18</td>
<td>19</td>
</tr>
<tr>
<td>-----</td>
<td>----</td>
<td>----</td>
<td>----</td>
<td>----</td>
<td>----</td>
<td>----</td>
<td>----</td>
<td>----</td>
<td>----</td>
<td>----</td>
<td>----</td>
<td>----</td>
<td>----</td>
<td>----</td>
<td>----</td>
<td>----</td>
<td>----</td>
<td>----</td>
<td>----</td>
<td>----</td>
</tr>
<tr>
<td>CP</td>
<td>23</td>
<td>45</td>
<td>219</td>
<td>305</td>
<td>179</td>
<td>257</td>
<td>460</td>
<td>573</td>
<td>659</td>
<td>750</td>
<td>441</td>
<td>511</td>
<td>574</td>
<td>669</td>
<td>398</td>
<td>469</td>
<td>677</td>
<td>770</td>
<td>401</td>
<td>473</td>
</tr>
<tr>
<td>CRR</td>
<td>61</td>
<td>103</td>
<td>320</td>
<td>420</td>
<td>163</td>
<td>238</td>
<td>635</td>
<td>756</td>
<td>14</td>
<td>27</td>
<td>34</td>
<td>59</td>
<td>592</td>
<td>401</td>
<td>468</td>
<td>629</td>
<td>727</td>
<td>421</td>
<td>496</td>
<td></td>
</tr>
<tr>
<td>J4</td>
<td>465</td>
<td>601</td>
<td>323</td>
<td>428</td>
<td>46</td>
<td>82</td>
<td>859</td>
<td>991</td>
<td>910</td>
<td>1007</td>
<td>634</td>
<td>715</td>
<td>780</td>
<td>694</td>
<td>405</td>
<td>460</td>
<td>680</td>
<td>747</td>
<td>470</td>
<td>522</td>
</tr>
<tr>
<td>JR</td>
<td>43</td>
<td>76</td>
<td>123</td>
<td>187</td>
<td>110</td>
<td>171</td>
<td>145</td>
<td>216</td>
<td>8</td>
<td>17</td>
<td>57</td>
<td>91</td>
<td>1647</td>
<td>711</td>
<td>1227</td>
<td>1259</td>
<td>1896</td>
<td>1919</td>
<td>1272</td>
<td>1293</td>
</tr>
<tr>
<td>JRRN</td>
<td>43</td>
<td>76</td>
<td>124</td>
<td>188</td>
<td>110</td>
<td>171</td>
<td>146</td>
<td>217</td>
<td>8</td>
<td>17</td>
<td>57</td>
<td>91</td>
<td>1703</td>
<td>1796</td>
<td>1242</td>
<td>1278</td>
<td>1847</td>
<td>1895</td>
<td>1239</td>
<td>1279</td>
</tr>
<tr>
<td>LR</td>
<td>468</td>
<td>593</td>
<td>324</td>
<td>424</td>
<td>46</td>
<td>82</td>
<td>869</td>
<td>981</td>
<td>911</td>
<td>1009</td>
<td>635</td>
<td>718</td>
<td>774</td>
<td>876</td>
<td>486</td>
<td>562</td>
<td>825</td>
<td>935</td>
<td>537</td>
<td>588</td>
</tr>
<tr>
<td>Tian</td>
<td>10</td>
<td>22</td>
<td>25</td>
<td>50</td>
<td>13</td>
<td>26</td>
<td>27</td>
<td>53</td>
<td>8</td>
<td>16</td>
<td>42</td>
<td>70</td>
<td>1516</td>
<td>1556</td>
<td>1120</td>
<td>1147</td>
<td>1334</td>
<td>1361</td>
<td>936</td>
<td>970</td>
</tr>
<tr>
<td>adjusted</td>
<td>25</td>
<td>47</td>
<td>339</td>
<td>442</td>
<td>184</td>
<td>264</td>
<td>831</td>
<td>953</td>
<td>889</td>
<td>988</td>
<td>437</td>
<td>490</td>
<td>742</td>
<td>839</td>
<td>417</td>
<td>598</td>
<td>806</td>
<td>897</td>
<td>425</td>
<td>607</td>
</tr>
<tr>
<td>Chriss</td>
<td>44</td>
<td>79</td>
<td>123</td>
<td>187</td>
<td>122</td>
<td>187</td>
<td>145</td>
<td>216</td>
<td>8</td>
<td>17</td>
<td>57</td>
<td>91</td>
<td>1399</td>
<td>1437</td>
<td>1037</td>
<td>1074</td>
<td>1488</td>
<td>1503</td>
<td>1250</td>
<td>1292</td>
</tr>
<tr>
<td>flexible</td>
<td>35</td>
<td>63</td>
<td>75</td>
<td>125</td>
<td>154</td>
<td>228</td>
<td>701</td>
<td>821</td>
<td>716</td>
<td>819</td>
<td>381</td>
<td>452</td>
<td>720</td>
<td>824</td>
<td>397</td>
<td>467</td>
<td>593</td>
<td>697</td>
<td>339</td>
<td>410</td>
</tr>
<tr>
<td>split</td>
<td>34</td>
<td>62</td>
<td>229</td>
<td>77</td>
<td>228</td>
<td>321</td>
<td>570</td>
<td>231</td>
<td>3332</td>
<td>3330</td>
<td>253</td>
<td>26</td>
<td>1528</td>
<td>1625</td>
<td>216</td>
<td>71</td>
<td>2337</td>
<td>2365</td>
<td>207</td>
<td>72</td>
</tr>
</tbody>
</table>

**Table 4.2.** Number of option evaluations a second obtainable with a modified relative error of 1E-3 using 0.5 additional weighting.
Table 4.3: Number of option evaluations a second obtainable with a relative error of 0.5E-4 with 0.5 cut-off.

<table>
<thead>
<tr>
<th>Option Type</th>
<th>Adjusted</th>
<th>Relative</th>
<th>Total</th>
<th>Relative</th>
<th>Total</th>
</tr>
</thead>
<tbody>
<tr>
<td>Table 4.3</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Key</th>
<th>0</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
</tr>
</thead>
<tbody>
<tr>
<td>CP</td>
<td>746</td>
<td>320</td>
<td>90</td>
<td>87</td>
<td>112</td>
<td>107</td>
<td>117</td>
<td>117</td>
<td>117</td>
<td>117</td>
</tr>
<tr>
<td>CP</td>
<td>746</td>
<td>320</td>
<td>90</td>
<td>87</td>
<td>112</td>
<td>107</td>
<td>117</td>
<td>117</td>
<td>117</td>
<td>117</td>
</tr>
<tr>
<td>CP</td>
<td>746</td>
<td>320</td>
<td>90</td>
<td>87</td>
<td>112</td>
<td>107</td>
<td>117</td>
<td>117</td>
<td>117</td>
<td>117</td>
</tr>
<tr>
<td>CP</td>
<td>746</td>
<td>320</td>
<td>90</td>
<td>87</td>
<td>112</td>
<td>107</td>
<td>117</td>
<td>117</td>
<td>117</td>
<td>117</td>
</tr>
<tr>
<td>CP</td>
<td>746</td>
<td>320</td>
<td>90</td>
<td>87</td>
<td>112</td>
<td>107</td>
<td>117</td>
<td>117</td>
<td>117</td>
<td>117</td>
</tr>
<tr>
<td>CP</td>
<td>746</td>
<td>320</td>
<td>90</td>
<td>87</td>
<td>112</td>
<td>107</td>
<td>117</td>
<td>117</td>
<td>117</td>
<td>117</td>
</tr>
<tr>
<td>CP</td>
<td>746</td>
<td>320</td>
<td>90</td>
<td>87</td>
<td>112</td>
<td>107</td>
<td>117</td>
<td>117</td>
<td>117</td>
<td>117</td>
</tr>
<tr>
<td>CP</td>
<td>746</td>
<td>320</td>
<td>90</td>
<td>87</td>
<td>112</td>
<td>107</td>
<td>117</td>
<td>117</td>
<td>117</td>
<td>117</td>
</tr>
<tr>
<td>CP</td>
<td>746</td>
<td>320</td>
<td>90</td>
<td>87</td>
<td>112</td>
<td>107</td>
<td>117</td>
<td>117</td>
<td>117</td>
<td>117</td>
</tr>
<tr>
<td>CP</td>
<td>746</td>
<td>320</td>
<td>90</td>
<td>87</td>
<td>112</td>
<td>107</td>
<td>117</td>
<td>117</td>
<td>117</td>
<td>117</td>
</tr>
</tbody>
</table>

---

The table above shows the number of option evaluations a second obtainable with a relative error of 0.5E-4 with 0.5 cut-off.
Detailed data for split 8 and split 9. Error is modified relative error. The time is the average time to price one option.

Other points to note are that Leisen–Reimer and J4 give almost identical results as expected, and that the adjusted tree with RE is also very similar to these trees with RE.

Another curiosity is that in certain cases the combination of truncation and control does very badly for the split tree. This suggests that the truncated split tree is doing a poor job of pricing the European option.

If one takes a key of 0, that is with no acceleration techniques, it is, in fact, the LR and J4 trees that are best, and Tian that is the worst. This demonstrates that the accuracy in the zero case is a poor predictor of accuracy after acceleration.

The contents of the final four columns and the previous four suggest that the precise choice of time to smooth is not important in that the columns are qualitatively similar with no clear trends.

Whilst these tests have been effective for seeing how much time is required to get a good level of accuracy, they do not answer the question of which tree to use when a very high level of accuracy is required. A second set of tests was therefore run for the most accurate trees. In this case, the model prices were obtained from the Leisen–Reimer tree with 10001 steps and extrapolation.

The number of steps run were 101, 201, 401, 801, 1601.

The number of option prices run was 12,000.

Examining table 4.5, we see from the column with 1601 steps that Tian 17 achieves the smallest error with split 9 close behind. The only methods which are faster with that number of steps are the 4 last ones which do not involve Richardson extrapolation. Their errors are much larger, however. We need to compare with different number of steps, this is done in Figure
### Table 4.5. rms error in absolute terms and number of option evaluations per second for 27 good cases using 12,000 evaluations.

<table>
<thead>
<tr>
<th>Name</th>
<th>Key</th>
<th>Error in Absolute Terms</th>
<th>Error in Absolute Terms</th>
<th>Error in Absolute Terms</th>
<th>Error in Absolute Terms</th>
<th>Error in Absolute Terms</th>
<th>Error in Absolute Terms</th>
<th>Error in Absolute Terms</th>
<th>Error in Absolute Terms</th>
</tr>
</thead>
<tbody>
<tr>
<td>Tian</td>
<td>17</td>
<td>3.73e-05</td>
<td>1.55</td>
<td>8.93e-05</td>
<td>5.45</td>
<td>2.15e-04</td>
<td>18.56</td>
<td>5.59e-04</td>
<td>61.40</td>
</tr>
<tr>
<td>split</td>
<td>9</td>
<td>3.78e-05</td>
<td>1.56</td>
<td>8.18e-05</td>
<td>5.51</td>
<td>2.39e-04</td>
<td>18.87</td>
<td>5.98e-04</td>
<td>62.27</td>
</tr>
<tr>
<td>split</td>
<td>17</td>
<td>3.86e-05</td>
<td>1.56</td>
<td>8.26e-05</td>
<td>5.47</td>
<td>2.37e-04</td>
<td>18.66</td>
<td>5.99e-04</td>
<td>61.32</td>
</tr>
<tr>
<td>Tian</td>
<td>15</td>
<td>3.88e-05</td>
<td>1.31</td>
<td>9.16e-05</td>
<td>4.42</td>
<td>2.24e-04</td>
<td>14.45</td>
<td>5.58e-04</td>
<td>46.02</td>
</tr>
<tr>
<td>Tian</td>
<td>13</td>
<td>4.00e-05</td>
<td>1.56</td>
<td>9.35e-05</td>
<td>5.48</td>
<td>2.24e-04</td>
<td>18.69</td>
<td>5.56e-04</td>
<td>61.76</td>
</tr>
<tr>
<td>split</td>
<td>13</td>
<td>4.10e-05</td>
<td>1.55</td>
<td>9.11e-05</td>
<td>5.45</td>
<td>2.58e-04</td>
<td>18.61</td>
<td>6.33e-04</td>
<td>61.42</td>
</tr>
<tr>
<td>adjusted</td>
<td>17</td>
<td>5.57e-05</td>
<td>1.55</td>
<td>1.57e-04</td>
<td>5.46</td>
<td>3.78e-04</td>
<td>18.60</td>
<td>8.86e-04</td>
<td>61.47</td>
</tr>
<tr>
<td>J4</td>
<td>9</td>
<td>5.58e-05</td>
<td>1.56</td>
<td>1.57e-04</td>
<td>5.46</td>
<td>3.75e-04</td>
<td>18.66</td>
<td>8.78e-04</td>
<td>61.95</td>
</tr>
<tr>
<td>J4</td>
<td>8</td>
<td>5.58e-05</td>
<td>0.57</td>
<td>1.57e-04</td>
<td>2.31</td>
<td>3.75e-04</td>
<td>9.11</td>
<td>8.78e-04</td>
<td>36.32</td>
</tr>
<tr>
<td>LR</td>
<td>9</td>
<td>5.58e-05</td>
<td>1.56</td>
<td>1.57e-04</td>
<td>5.47</td>
<td>3.75e-04</td>
<td>18.70</td>
<td>8.78e-04</td>
<td>62.03</td>
</tr>
<tr>
<td>LR</td>
<td>8</td>
<td>5.58e-05</td>
<td>0.57</td>
<td>1.57e-04</td>
<td>2.29</td>
<td>3.75e-04</td>
<td>9.12</td>
<td>8.78e-04</td>
<td>36.26</td>
</tr>
<tr>
<td>J4</td>
<td>17</td>
<td>5.63e-05</td>
<td>1.55</td>
<td>1.58e-04</td>
<td>5.45</td>
<td>3.78e-04</td>
<td>18.54</td>
<td>8.78e-04</td>
<td>61.35</td>
</tr>
<tr>
<td>LR</td>
<td>17</td>
<td>5.63e-05</td>
<td>1.55</td>
<td>1.58e-04</td>
<td>5.46</td>
<td>3.77e-04</td>
<td>18.58</td>
<td>8.78e-04</td>
<td>61.37</td>
</tr>
<tr>
<td>JRRN</td>
<td>19</td>
<td>6.68e-05</td>
<td>1.31</td>
<td>1.36e-04</td>
<td>4.42</td>
<td>3.12e-04</td>
<td>14.42</td>
<td>8.23e-04</td>
<td>45.83</td>
</tr>
<tr>
<td>Chriss</td>
<td>17</td>
<td>6.70e-05</td>
<td>1.56</td>
<td>1.36e-04</td>
<td>5.47</td>
<td>3.13e-04</td>
<td>18.60</td>
<td>8.23e-04</td>
<td>61.50</td>
</tr>
<tr>
<td>JRRN</td>
<td>17</td>
<td>6.70e-05</td>
<td>1.56</td>
<td>1.36e-04</td>
<td>5.47</td>
<td>3.13e-04</td>
<td>18.62</td>
<td>8.24e-04</td>
<td>61.57</td>
</tr>
<tr>
<td>JR</td>
<td>17</td>
<td>6.70e-05</td>
<td>1.55</td>
<td>1.36e-04</td>
<td>5.46</td>
<td>3.13e-04</td>
<td>18.61</td>
<td>8.24e-04</td>
<td>61.40</td>
</tr>
<tr>
<td>Chriss</td>
<td>13</td>
<td>6.93e-05</td>
<td>1.56</td>
<td>1.44e-04</td>
<td>5.46</td>
<td>3.30e-04</td>
<td>18.59</td>
<td>8.51e-04</td>
<td>61.66</td>
</tr>
<tr>
<td>JR</td>
<td>13</td>
<td>6.93e-05</td>
<td>1.55</td>
<td>1.44e-04</td>
<td>5.46</td>
<td>3.31e-04</td>
<td>18.62</td>
<td>8.52e-04</td>
<td>61.71</td>
</tr>
<tr>
<td>JRRN</td>
<td>13</td>
<td>6.93e-05</td>
<td>1.55</td>
<td>1.44e-04</td>
<td>5.46</td>
<td>3.31e-04</td>
<td>18.61</td>
<td>8.52e-04</td>
<td>61.71</td>
</tr>
<tr>
<td>flexible</td>
<td>9</td>
<td>1.01e-04</td>
<td>1.55</td>
<td>2.06e-04</td>
<td>5.46</td>
<td>3.67e-04</td>
<td>18.67</td>
<td>8.54e-04</td>
<td>62.00</td>
</tr>
<tr>
<td>flexible</td>
<td>13</td>
<td>1.02e-04</td>
<td>1.55</td>
<td>2.11e-04</td>
<td>5.45</td>
<td>3.83e-04</td>
<td>18.43</td>
<td>8.98e-04</td>
<td>61.32</td>
</tr>
<tr>
<td>CP</td>
<td>17</td>
<td>1.03e-04</td>
<td>1.55</td>
<td>2.05e-04</td>
<td>5.46</td>
<td>3.53e-04</td>
<td>18.59</td>
<td>8.56e-04</td>
<td>61.42</td>
</tr>
<tr>
<td>CRR</td>
<td>7</td>
<td>2.83e-04</td>
<td>5.79</td>
<td>5.58e-04</td>
<td>18.97</td>
<td>1.14e-03</td>
<td>61.44</td>
<td>2.33e-03</td>
<td>192.60</td>
</tr>
<tr>
<td>flexible</td>
<td>7</td>
<td>2.84e-04</td>
<td>5.79</td>
<td>5.63e-04</td>
<td>19.00</td>
<td>1.16e-03</td>
<td>61.47</td>
<td>2.38e-03</td>
<td>192.11</td>
</tr>
<tr>
<td>LR</td>
<td>7</td>
<td>3.45e-04</td>
<td>5.75</td>
<td>7.08e-04</td>
<td>18.91</td>
<td>1.38e-03</td>
<td>61.39</td>
<td>2.80e-03</td>
<td>192.28</td>
</tr>
<tr>
<td>J4</td>
<td>7</td>
<td>3.45e-04</td>
<td>5.79</td>
<td>7.08e-04</td>
<td>18.97</td>
<td>1.38e-03</td>
<td>61.42</td>
<td>2.80e-03</td>
<td>192.43</td>
</tr>
</tbody>
</table>

Note: Key: t/a is for the test set of absolute terms, four is for the full set of absolute terms.
<table>
<thead>
<tr>
<th>name</th>
<th>key</th>
<th>error</th>
<th>speed</th>
<th>error</th>
<th>speed</th>
<th>error</th>
<th>speed</th>
<th>error</th>
<th>speed</th>
<th>error</th>
<th>speed</th>
<th>error</th>
<th>speed</th>
<th>error</th>
<th>speed</th>
</tr>
</thead>
<tbody>
<tr>
<td>split</td>
<td>9</td>
<td>3.94E-06</td>
<td>1.56</td>
<td>9.77E-06</td>
<td>5.51</td>
<td>2.28E-05</td>
<td>18.87</td>
<td>5.43E-05</td>
<td>62.27</td>
<td>1.13E-04</td>
<td>204.19</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>split</td>
<td>17</td>
<td>4.10E-06</td>
<td>1.56</td>
<td>9.56E-06</td>
<td>5.47</td>
<td>2.20E-05</td>
<td>18.66</td>
<td>5.29E-05</td>
<td>61.32</td>
<td>1.30E-04</td>
<td>198.76</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Tian</td>
<td>15</td>
<td>4.50E-06</td>
<td>1.31</td>
<td>1.52E-05</td>
<td>4.42</td>
<td>3.26E-05</td>
<td>14.45</td>
<td>5.93E-05</td>
<td>46.02</td>
<td>1.52E-04</td>
<td>145.08</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Tian</td>
<td>13</td>
<td>4.59E-06</td>
<td>1.56</td>
<td>1.24E-05</td>
<td>5.48</td>
<td>3.25E-05</td>
<td>18.68</td>
<td>5.76E-05</td>
<td>61.76</td>
<td>1.55E-04</td>
<td>201.07</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Tian</td>
<td>17</td>
<td>4.69E-06</td>
<td>1.55</td>
<td>9.46E-06</td>
<td>5.45</td>
<td>2.35E-05</td>
<td>18.56</td>
<td>6.89E-05</td>
<td>61.40</td>
<td>1.57E-04</td>
<td>199.28</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Chriss</td>
<td>17</td>
<td>5.07E-06</td>
<td>1.56</td>
<td>1.17E-05</td>
<td>5.47</td>
<td>2.49E-05</td>
<td>18.60</td>
<td>6.94E-05</td>
<td>61.50</td>
<td>1.45E-04</td>
<td>197.87</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>JR</td>
<td>17</td>
<td>5.07E-06</td>
<td>1.55</td>
<td>1.17E-05</td>
<td>5.46</td>
<td>2.49E-05</td>
<td>18.58</td>
<td>6.94E-05</td>
<td>61.40</td>
<td>1.44E-04</td>
<td>198.93</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>JRRN</td>
<td>17</td>
<td>5.07E-06</td>
<td>1.56</td>
<td>1.17E-05</td>
<td>5.47</td>
<td>2.49E-05</td>
<td>18.62</td>
<td>6.94E-05</td>
<td>61.57</td>
<td>1.44E-04</td>
<td>199.10</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>split</td>
<td>13</td>
<td>5.45E-06</td>
<td>1.55</td>
<td>1.33E-05</td>
<td>5.45</td>
<td>3.07E-05</td>
<td>18.61</td>
<td>7.1E-05</td>
<td>61.42</td>
<td>1.65E-04</td>
<td>200.53</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Chriss</td>
<td>13</td>
<td>6.27E-06</td>
<td>1.56</td>
<td>1.51E-05</td>
<td>5.46</td>
<td>3.27E-05</td>
<td>18.59</td>
<td>7.75E-05</td>
<td>61.66</td>
<td>1.85E-04</td>
<td>200.35</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>JR</td>
<td>13</td>
<td>6.27E-06</td>
<td>1.55</td>
<td>1.52E-05</td>
<td>5.46</td>
<td>3.27E-05</td>
<td>18.62</td>
<td>7.76E-05</td>
<td>61.71</td>
<td>1.86E-04</td>
<td>201.08</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>JRRN</td>
<td>13</td>
<td>6.27E-06</td>
<td>1.55</td>
<td>1.52E-05</td>
<td>5.46</td>
<td>3.27E-05</td>
<td>18.61</td>
<td>7.76E-05</td>
<td>61.71</td>
<td>1.86E-04</td>
<td>200.90</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>J4</td>
<td>9</td>
<td>6.65E-06</td>
<td>1.56</td>
<td>1.41E-05</td>
<td>5.46</td>
<td>3.22E-05</td>
<td>18.66</td>
<td>7.27E-05</td>
<td>61.95</td>
<td>1.62E-04</td>
<td>203.82</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>J4</td>
<td>8</td>
<td>6.65E-06</td>
<td>0.57</td>
<td>1.41E-05</td>
<td>2.31</td>
<td>3.22E-05</td>
<td>9.11</td>
<td>7.27E-05</td>
<td>36.32</td>
<td>1.62E-04</td>
<td>147.28</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>LR</td>
<td>9</td>
<td>6.65E-06</td>
<td>1.56</td>
<td>1.41E-05</td>
<td>5.47</td>
<td>3.23E-05</td>
<td>18.70</td>
<td>7.30E-05</td>
<td>62.03</td>
<td>1.63E-04</td>
<td>203.82</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>LR</td>
<td>8</td>
<td>6.65E-06</td>
<td>0.57</td>
<td>1.41E-05</td>
<td>2.29</td>
<td>3.23E-05</td>
<td>9.12</td>
<td>7.30E-05</td>
<td>36.26</td>
<td>1.63E-04</td>
<td>146.11</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>adjusted</td>
<td>17</td>
<td>6.71E-06</td>
<td>1.55</td>
<td>1.43E-05</td>
<td>5.46</td>
<td>3.08E-05</td>
<td>18.60</td>
<td>7.43E-05</td>
<td>61.47</td>
<td>1.73E-04</td>
<td>199.46</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>J4</td>
<td>17</td>
<td>6.74E-06</td>
<td>1.55</td>
<td>1.43E-05</td>
<td>5.45</td>
<td>3.10E-05</td>
<td>18.54</td>
<td>7.44E-05</td>
<td>61.35</td>
<td>1.74E-04</td>
<td>199.28</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>LR</td>
<td>17</td>
<td>6.74E-06</td>
<td>1.55</td>
<td>1.43E-05</td>
<td>5.46</td>
<td>3.10E-05</td>
<td>18.58</td>
<td>7.43E-05</td>
<td>61.37</td>
<td>1.74E-04</td>
<td>199.46</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>CP</td>
<td>17</td>
<td>9.04E-06</td>
<td>1.55</td>
<td>1.88E-05</td>
<td>5.46</td>
<td>5.52E-05</td>
<td>18.59</td>
<td>1.33E-04</td>
<td>61.42</td>
<td>4.44E-04</td>
<td>198.92</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>flexible</td>
<td>9</td>
<td>1.09E-05</td>
<td>1.55</td>
<td>2.24E-05</td>
<td>5.46</td>
<td>5.55E-05</td>
<td>18.67</td>
<td>1.01E-04</td>
<td>62.00</td>
<td>3.42E-04</td>
<td>202.90</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>flexible</td>
<td>13</td>
<td>1.17E-05</td>
<td>1.55</td>
<td>2.46E-05</td>
<td>5.45</td>
<td>5.22E-05</td>
<td>18.43</td>
<td>1.08E-04</td>
<td>61.32</td>
<td>3.93E-04</td>
<td>200.18</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>CRR</td>
<td>7</td>
<td>4.75E-05</td>
<td>5.79</td>
<td>8.95E-05</td>
<td>18.97</td>
<td>1.64E-04</td>
<td>61.44</td>
<td>3.31E-04</td>
<td>192.60</td>
<td>7.39E-04</td>
<td>630.97</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>LR</td>
<td>7</td>
<td>5.10E-05</td>
<td>5.75</td>
<td>9.51E-05</td>
<td>18.91</td>
<td>1.81E-04</td>
<td>61.39</td>
<td>3.34E-04</td>
<td>192.28</td>
<td>6.12E-04</td>
<td>622.11</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>J4</td>
<td>7</td>
<td>5.10E-05</td>
<td>5.79</td>
<td>9.51E-05</td>
<td>18.97</td>
<td>1.81E-04</td>
<td>61.42</td>
<td>3.34E-04</td>
<td>192.43</td>
<td>6.12E-04</td>
<td>630.97</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>flexible</td>
<td>7</td>
<td>5.22E-05</td>
<td>5.79</td>
<td>9.04E-05</td>
<td>19.00</td>
<td>1.80E-04</td>
<td>61.5E+01</td>
<td>3.40E-04</td>
<td>192.11</td>
<td>7.23E-04</td>
<td>629.16</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table 4.6. rms error in modified relative terms with additional weight of 0.5 and number of option evaluations per second for 27 good cases using 12,000 evaluations.
<table>
<thead>
<tr>
<th>Name</th>
<th>RMS Error</th>
<th>Speed Error</th>
<th>Split Time</th>
<th>Speed Error</th>
<th>Split Time</th>
<th>Speed Error</th>
<th>Split Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>Tian</td>
<td>7.81E-06</td>
<td>1.55</td>
<td>1.57E-05</td>
<td>18.56</td>
<td>3.80E-05</td>
<td>61.40</td>
<td></td>
</tr>
<tr>
<td>Tian</td>
<td>3.17E-06</td>
<td>1.31</td>
<td>1.69E-05</td>
<td>14.45</td>
<td>4.05E-05</td>
<td>46.02</td>
<td></td>
</tr>
<tr>
<td>Tian</td>
<td>3.34E-06</td>
<td>1.56</td>
<td>1.71E-05</td>
<td>18.68</td>
<td>4.06E-05</td>
<td>61.76</td>
<td></td>
</tr>
<tr>
<td>Split</td>
<td>3.50E-06</td>
<td>1.56</td>
<td>2.09E-05</td>
<td>18.87</td>
<td>5.71E-05</td>
<td>62.27</td>
<td></td>
</tr>
<tr>
<td>JRRN</td>
<td>3.86E-06</td>
<td>1.31</td>
<td>2.09E-05</td>
<td>14.42</td>
<td>4.83E-05</td>
<td>45.83</td>
<td></td>
</tr>
<tr>
<td>Chriss</td>
<td>3.92E-06</td>
<td>1.56</td>
<td>2.10E-05</td>
<td>18.60</td>
<td>4.86E-05</td>
<td>61.50</td>
<td></td>
</tr>
<tr>
<td>JR</td>
<td>3.92E-06</td>
<td>1.55</td>
<td>2.10E-05</td>
<td>18.58</td>
<td>4.86E-05</td>
<td>61.40</td>
<td></td>
</tr>
<tr>
<td>JRRN</td>
<td>3.92E-06</td>
<td>1.56</td>
<td>2.10E-05</td>
<td>18.62</td>
<td>4.86E-05</td>
<td>61.57</td>
<td></td>
</tr>
<tr>
<td>Split</td>
<td>4.16E-06</td>
<td>1.55</td>
<td>2.46E-05</td>
<td>18.61</td>
<td>6.42E-05</td>
<td>61.42</td>
<td></td>
</tr>
<tr>
<td>Chriss</td>
<td>4.39E-06</td>
<td>1.56</td>
<td>2.29E-05</td>
<td>18.62</td>
<td>5.23E-05</td>
<td>61.66</td>
<td></td>
</tr>
<tr>
<td>JR</td>
<td>4.39E-06</td>
<td>1.55</td>
<td>2.29E-05</td>
<td>18.62</td>
<td>5.23E-05</td>
<td>61.71</td>
<td></td>
</tr>
<tr>
<td>JRRN</td>
<td>4.39E-06</td>
<td>1.55</td>
<td>2.29E-05</td>
<td>18.61</td>
<td>5.23E-05</td>
<td>61.71</td>
<td></td>
</tr>
<tr>
<td>LR</td>
<td>5.47E-06</td>
<td>1.56</td>
<td>2.48E-05</td>
<td>18.66</td>
<td>5.35E-05</td>
<td>61.95</td>
<td></td>
</tr>
<tr>
<td>LR</td>
<td>5.47E-06</td>
<td>0.57</td>
<td>2.48E-05</td>
<td>9.11</td>
<td>5.35E-05</td>
<td>36.32</td>
<td></td>
</tr>
<tr>
<td>J4</td>
<td>5.47E-06</td>
<td>1.56</td>
<td>2.48E-05</td>
<td>18.66</td>
<td>5.35E-05</td>
<td>61.95</td>
<td></td>
</tr>
<tr>
<td>LR</td>
<td>5.51E-06</td>
<td>1.55</td>
<td>2.48E-05</td>
<td>18.58</td>
<td>5.35E-05</td>
<td>61.37</td>
<td></td>
</tr>
<tr>
<td>CP</td>
<td>8.08E-06</td>
<td>1.55</td>
<td>4.74E-05</td>
<td>18.59</td>
<td>1.02E-04</td>
<td>61.42</td>
<td></td>
</tr>
<tr>
<td>Flexible</td>
<td>8.49E-06</td>
<td>1.55</td>
<td>4.47E-05</td>
<td>18.67</td>
<td>1.22E-04</td>
<td>62.00</td>
<td></td>
</tr>
<tr>
<td>Flexible</td>
<td>8.81E-06</td>
<td>1.55</td>
<td>4.64E-05</td>
<td>18.43</td>
<td>1.26E-04</td>
<td>61.32</td>
<td></td>
</tr>
<tr>
<td>CRR</td>
<td>2.81E-05</td>
<td>5.79</td>
<td>5.36E-05</td>
<td>18.97</td>
<td>2.38E-04</td>
<td>192.60</td>
<td></td>
</tr>
<tr>
<td>Flexible</td>
<td>2.86E-05</td>
<td>5.79</td>
<td>5.38E-05</td>
<td>19.00</td>
<td>2.45E-04</td>
<td>192.11</td>
<td></td>
</tr>
<tr>
<td>J4</td>
<td>3.81E-05</td>
<td>5.79</td>
<td>7.13E-05</td>
<td>18.97</td>
<td>2.40E-04</td>
<td>192.43</td>
<td></td>
</tr>
<tr>
<td>LR</td>
<td>3.81E-05</td>
<td>5.75</td>
<td>7.13E-05</td>
<td>18.91</td>
<td>2.40E-04</td>
<td>192.28</td>
<td></td>
</tr>
</tbody>
</table>

Table 4.7: RMS error in Broadie–Detemple relative terms with cut-off of 0.5 and number of option evaluations per second for 77 good cases using 12,000 evaluations.
Figure 1. Number of evaluations per second against RMS absolute error for three trees with log scale.

Figure 2. Number of evaluations per second against modified relative RMS error for four trees with log scale.
Figure 3. Number of evaluations per second against Broadie–Detemple relative rms error for five trees with log scale.

Figure 4. Number of evaluations per second against Broadie–Detemple relative rms error for five trees with log scale.
1. We see clearly that CRR 7 is substantially worse than Tian 17 and split 9.

If one’s objective is to minimize absolute error then it is clear that we should use Tian 17: that is third moment matching with smoothing, Richardson extrapolation, truncation and matching smoothing times. The choice of split 9 is also competitive. Note that the smallest error varies with number of steps and with 401 steps, it is split 9 that wins. This suggests that the trees are essentially the same in accuracy.

For modified relative error, we examine table 4.6, we see from the column with 1601 steps that split 9 has the smallest error with split 17, Tian 13, Tian 15 and Tian 17 almost as good. Again the last 4 are faster with larger errors so we plot error against speed in Figure 2. We see clearly that CRR 7 is substantially worse than Tian 15, Tian 13 and split 9. We also see that Tian 15 is worse than Tian 13. The comparison between Tian 15 and Tian 13 suggests that although the use of a control does reduce error in this case, the additional computational effort is not worth the improvement.

If one’s objective is to minimize modified rms error then it is clear that we should use split 9; Tian 13 is also a good choice.

Examining table 4.7, we see from the column with 1601 steps that Tian 17 achieves the smallest error with Tian 15, Tian 13 and split 9 almost as good. The only methods which are faster with that number of steps are yet again the 4 last ones which do not involve extrapolation and we compare with different number of steps, in Figure 3 and in Figure 4. We see clearly that CRR 7 is substantially worse than Tian 17, Tian 15, Tian 13 and split 9. We also see that Tian 15 is worse than Tian 13 and Tian 17. The comparison between Tian 15 and Tian 13 suggests that although the use of a control does reduce error in this case, the additional computational effort is not worth the improvement.

If one’s objective is to minimize Broadie-Detemple rms error then it is clear that we should use Tian 17; Tian 13 and split 9 are also viable choices.

The reader may be interested in the order of convergence as well as the size of the errors. These were estimated by regressing the log RMS error against log time taken and fitting the best straight line through the cases with 201, 401 and 801 steps. The slopes are displayed in Table 4.8. We display results for absolute errors, relative errors with modification, and the Broadie–Detemple relative errors.

CRR 0 corresponds to the original tree of Cox, Ross and Rubinstein with no acceleration techniques, and its order is roughly $-0.5$. The CRR 12 tree corresponds to the BBSR method of Broadie and Detemple. Its convergence order is about $-2/3$ as a function of time, and so $-4/3$ as
<table>
<thead>
<tr>
<th>name</th>
<th>key</th>
<th>absolute order</th>
<th>modified order</th>
<th>BD order</th>
</tr>
</thead>
<tbody>
<tr>
<td>CRR  0</td>
<td>-0.508</td>
<td>-0.454</td>
<td>-0.506</td>
<td></td>
</tr>
<tr>
<td>CRR  12</td>
<td>-0.505</td>
<td>-0.598</td>
<td>-0.676</td>
<td></td>
</tr>
<tr>
<td>CRR  13</td>
<td>-0.575</td>
<td>-0.684</td>
<td>-0.770</td>
<td></td>
</tr>
<tr>
<td>LR    9</td>
<td>-0.738</td>
<td>-0.756</td>
<td>-0.710</td>
<td></td>
</tr>
<tr>
<td>Split 9</td>
<td>-0.922</td>
<td>-0.790</td>
<td>-0.925</td>
<td></td>
</tr>
<tr>
<td>Tian  13</td>
<td>-0.829</td>
<td>-0.672</td>
<td>-0.724</td>
<td></td>
</tr>
<tr>
<td>Tian  17</td>
<td>-0.856</td>
<td>-0.906</td>
<td>-0.766</td>
<td></td>
</tr>
</tbody>
</table>

Table 4.8. Order of convergence as expressed as a power of time for a selected few interesting cases.

...a function of the number of steps (when using the BD error measure.) Curiously, the order of convergence for absolute errors does not appear to improve above that of CRR 0 although the constant is, of course, much lower. The Tian 13 and 17 methods, and the split 9 method again display more rapid convergence than the other methods.

5. Conclusion

Pulling all these results together, we see that for pricing an American put option in the Black–Scholes model with high accuracy and speed, we should always use truncation and extrapolation. We should also use a technique which reduces the oscillations in the European case: that is smoothing or modifying the lattice to take account of strike.

The best overall results have been obtained the Tian third moment matching tree together with truncation, smoothing and extrapolation, and the new split tree which uses a time-dependent drift to minimize oscillations, together with extrapolation and truncation. We have not investigated in this paper the optimal level of truncation but have instead adopted a level that has minimal effect on price. The Tian tree has the added bonus that the node placement does not depend on strike so there is the additional possibility of pricing many options simultaneously.

Interestingly, neither of the preferred trees are amongst those in popular use at the current time. This is despite the fact that the Tian tree was first introduced fifteen years ago. A possible explanation is that its virtue, matching three moments, does not have much effect when the pay-off is not smooth, and so initial tests without smoothing and extrapolation showed it to be poor.
REFERENCES


CENTRE FOR ACTUARIAL STUDIES, DEPARTMENT OF ECONOMICS, UNIVERSITY OF MELBOURNE, VICTORIA 3010, AUSTRALIA

E-mail address: mark@markjoshi.com
Timing VBA code execution

Michael Carter

Accurately timing execution speed on a multitasking computer is surprisingly difficult, since the CPU can be regularly interrupted by other processes. It is normal to record different times on repeated runs. So good practice would be to average (or minimum) over a number of runs. It is also sensible to close other applications when undertaking timing comparisons.

VBA contains a function Timer() that gives the number of seconds since midnight. By calling Timer() at the beginning and end of a lengthy computation, it is possible to estimate the time taken by the function as follows:

    Dim StartTime, EndTime, ComputationTime As Single
    StartTime = Timer
    Do lengthy computation
    EndTime = Timer
    ComputationTime = EndTime - StartTime

More accurate timing can be achieved using the Windows operating system function GetTickCount(), which returns the time in milliseconds since the system was started. It is claimed to have a resolution of 10 milliseconds (approximately). To use this function, it must first be declared as follows:

    Declare Function GetTickCount Lib "Kernel32" () As Long
    Dim StartTime, EndTime, ComputationTime As Long
    StartTime = GetTickCount
    Do lengthy computation
    EndTime = GetTickCount
    ComputationTime = EndTime - StartTime

More information is available in the Microsoft tutorial note How To Use QueryPerformanceCounter to Time Code.