M2 Course content

Introduction
- Language reminders:
  - VBA types & instructions
  - VBA Code samples comments
  - Python
- Classes in VBA
- Classes in Python
- Case #2: Option pricing with a lattice

24-Aug-24 21:50:02

Pricing model landscape

- Here are the three types of pricing models:

	Closed-form formulas	Trees	Monte-Carlo
Examples	Black-Scholes, Heston, SABR	Hull & White	
European ex.	Yes	Yes	Yes
American ex.	No	Yes	Hard
Path dependent	No	Hard	Yes
Speed	Instantaneous	Fast	Slow
Precision	Perfect	Error ↘ 1/Nb Steps	Error ↘ 1/√Nb Draws
Memory	Light	Heavy	Light

Lattice node VBA class

- Goal #1, practice PDE and trees:
  - Simulate a stock price in a recombining lattice
  - Price European and American calls and puts
  - Assess convergence with Black-Scholes
- Goal #2, code and use a VBA class:
  - Define a class for the lattice node and tree
  - Build a tree with reconnecting nodes
  - Price options on this lattice

Stock price behavior

- In the « *Derivatives Pricing and Stochastic Calculus* » course by Elie & Kharroubi we can see for Black & Scholes p. 61:

• **The risky asset.** We suppose that the risky asset S is given by the SDE

$$\begin{cases} dS_t &= S_t\,(\mu\, dt + \sigma\, dW_t)\,, \quad t \in [0,T] \\ S_0 &= s_0 \end{cases} \tag{6.2.1}$$

where μ and σ are two constants such that $\sigma > 0$.

- We can use those assumptions for the interest rate and the volatility
- But the dividend is discrete and it is simplistic to define µ as: µ = rate - dividend

Discrete dividends

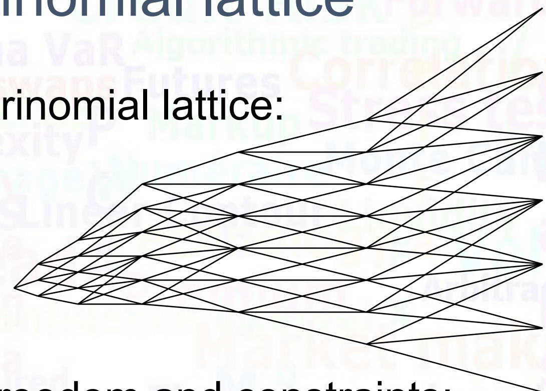
- Dividends in a near future are known
- Dividends in several years can be approximated as a % of the stock price
- Example with a stock price S_0 = 100 €
  - First annual dividend = 3 €
  - Second annual dividend = 2 € + 1% of X_{1Y}
  - Third annual dividend = 1.4 € + 1.6% of X_{2Y}
  - Fourth annual dividend = 1 € + 2% of X_{3Y}
  - ...
  - Tenth annual dividend = 3% of X_{10Y}

Dividend model

- The dividend is discrete and moves from constant to proportional to the stock price
- Example of possible functional form:
  - $D_t = \rho\,(S_0\, e^{-\lambda(t-t_0)} + S_t\,(1- e^{-\lambda(t-t_0)}))$ on ex-div. dates
  - $D_t = 0$ on all other dates
- $dS_t = S_t\, r\, dt + S_t\, \sigma\, dW_t - D_t$

Trinomial lattice

- Example of trinomial lattice:

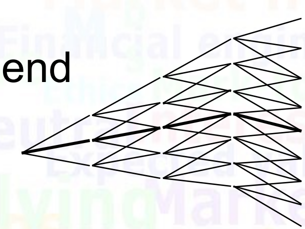


- Degrees of freedom and constraints:

Degrees of freedom	Constraints
Time steps	Match expected value
Nodes values	Match variance
Choice of next middle node	Positive probabilities
Transition probabilities	Sum of probabilities = 1

Trinomial tree assumptions

- We use the following assumptions:
  - Time steps are all equal: Δt
  - The middle node is equal to the forward price
  - Nodes values are geometric series: $\alpha = S_{i,j+1}/S_{i,j}$
  - The next middle node is the closer to the forward price
- Example with a dividend on the fifth date:



Lattice building

- Start from the root: $N_{0,0}$ with $S_{0,0} = s_0$
- Branch to the next central node: $N_{1,0}$

  $$S_{1,0} = S_{0,0} \exp(r\ \Delta t) – D_1$$
- Build other nodes on date d_1: $S_{1,j} = S_{1,0}\ \alpha^j$
- α is defined by a multiple of the standard deviation over one time step ≈ $S_t\ \sigma$ sqr(Δt):

  $$S_{i,j+1} - S_{i,j} \approx \text{sqr(3) StdDev}$$
- Divide by $S_{i,j}$: α ≈ 1+ sqr(3) StdDev / $S_{i,j}$
- The actual formula is: $\alpha = e^{\sigma\sqrt{3\Delta t}}$
  - Please note: $\sqrt{3}$ is indicative. $\sqrt{2}$ as well as 2 work too

24-Aug-24 21:50:02

Probabilities (1)

- For each node $N_{i,j}$ we calculate the probabilities:
  - Middle node on t_{i+1} is the closer to the forward: $N_{i+1,j'}$
  - The sum of the three probabilities is equal to 1:

    $$p_{up} + p_{mid} + p_{down} = 1$$
  - Matching the expected value means:

$$E_{i+1,j} = p_{up}\ S_{i+1,j'+1} + p_{mid}\ S_{i+1,j'} + p_{down}\ S_{i+1,j'-1} = E\left[S_{t_{i+1}} \middle| S_{t_i}\right]$$

  - With $S_{t_{i+1}} = S_{t_i} e^{\left(r-\frac{1}{2}\sigma^2\right)\Delta t + \sigma W_{\Delta t}} - D_{t_{i+1}}$

$$E\left[S_{t_{i+1}} \middle| S_{t_i}\right] = S_{t_i} e^{r\Delta t} - D_{t_{i+1}}$$

24-Aug-24 21:50:02

Probabilities (2)

- Matching the variance on the step after $N_{i,j}$ means:

$$V_{i+1,j} + E_{i+1,j}^2 = p_{up} S_{i+1,j'+1}^2 + p_{mid} S_{i+1,j'}^2 + p_{down} S_{i+1,j'-1}^2$$

$$V_{i+1,j} = E\left[\left(S_{t_{i+1}} - E\left[S_{t_{i+1}} \middle| S_{t_i}\right]\right)^2 \middle| S_{t_i}\right]$$

  - $V_{i+1,j} = E\left[S_{t_i}^2\left(e^{\left(r-\frac{1}{2}\sigma^2\right)\Delta t+\sigma W_{\Delta t}} - e^{r\Delta t}\right)^2 \middle| S_{t_i}\right]$
  - $V_{i+1,j} = S_{t_i}^2\, e^{2r\Delta t}\, E\left[e^{-\sigma^2\Delta t+2\sigma W_{\Delta t}} - 2e^{-\frac{1}{2}\sigma^2\Delta t+\sigma W_{\Delta t}} + 1\right]$
  - $V_{i+1,j} = S_{t_i}^2\, e^{2r\Delta t}\left(e^{\sigma^2\Delta t} - 1\right)$

24-Aug-24 21:50:02

Probabilities (3)

- The three constraints make a linear system:
  - $1 = p_{up} + p_{mid} + p_{down}$
  - $E_{i+1,j} = p_{up}\, S_{i+1,j'+1} + p_{mid}\, S_{i+1,j'} + p_{down}\, S_{i+1,j'-1}$
  - $V_{i+1,j} + E_{i+1,j}^2 = p_{up}\, S_{i+1,j'+1}^2 + p_{mid}\, S_{i+1,j'}^2 + p_{down}\, S_{i+1,j'-1}^2$
- That can be written as a matrix product:

$$\begin{pmatrix} 1 & 1 & 1 \\ S_{i+1,j'+1} & S_{i+1,j'} & S_{i+1,j'-1} \\ S_{i+1,j'+1}^2 & S_{i+1,j'}^2 & S_{i+1,j'-1}^2 \end{pmatrix} \begin{pmatrix} p_{up} \\ p_{mid} \\ p_{down} \end{pmatrix} = \begin{pmatrix} 1 \\ E_{i+1,j} \\ V_{i+1,j} + E_{i+1,j}^2 \end{pmatrix}$$

$$\begin{pmatrix} 1 & 0 & 0 \\ 0 & S_{i+1,j'} & 0 \\ 0 & 0 & S_{i+1,j'}^2 \end{pmatrix} \begin{pmatrix} 1 & 1 & 1 \\ \alpha & 1 & \alpha^{-1} \\ \alpha^2 & 1 & \alpha^{-2} \end{pmatrix} \begin{pmatrix} p_{up} \\ p_{mid} \\ p_{down} \end{pmatrix} = \begin{pmatrix} 1 \\ E_{i+1,j} \\ V_{i+1,j} + E_{i+1,j}^2 \end{pmatrix}$$

24-Aug-24 21:50:02

Probabilities (4)

- The system has a unique solution with α>1 and $S_{i+1,j'}$>0:

$$\begin{pmatrix} 1 & 1 & 1 \\ \alpha & 1 & \alpha^{-1} \\ \alpha^2 & 1 & \alpha^{-2} \end{pmatrix} \begin{pmatrix} p_{up} \\ p_{mid} \\ p_{down} \end{pmatrix} = \begin{pmatrix} 1 \\ S_{i+1,j'}^{-1} E_{i+1,j} \\ S_{i+1,j'}^{-2}\left(V_{i+1,j} + E_{i+1,j}^2\right) \end{pmatrix}$$

- Solve by substitutions:

$$\begin{pmatrix} 1 & 1 & 1 \\ \alpha - 1 & 0 & \alpha^{-1} - 1 \\ \alpha^2 - 1 & 0 & \alpha^{-2} - 1 \end{pmatrix} \begin{pmatrix} p_{up} \\ p_{mid} \\ p_{down} \end{pmatrix} = \begin{pmatrix} 1 \\ S_{i+1,j'}^{-1} E_{i+1,j} - 1 \\ S_{i+1,j'}^{-2}\left(V_{i+1,j} + E_{i+1,j}^2\right) - 1 \end{pmatrix}$$

- Subtract (α+1) times the second line to the third to get:

$$(1-\alpha)(\alpha^{-2}-1)\, p_{down} = S_{i+1,j'}^{-2}\left(V_{i+1,j} + E_{i+1,j}^2\right) - 1 - (\alpha+1)\left(S_{i+1,j'}^{-1} E_{i+1,j} - 1\right)$$

- p_{down} has no dimension (good!), but is not guaranteed to be in [0, 1]

Probabilities (5)

$$p_{down} = \frac{S_{i+1,j'}^{-2}\left(V_{i+1,j} + E_{i+1,j}^2\right) - 1 - (\alpha+1)\left(S_{i+1,j'}^{-1} E_{i+1,j} - 1\right)}{(1-\alpha)(\alpha^{-2}-1)}$$

- When there is no dividend, $S_{i+1,j'} = E_{i+1,j}$ and we have:

$$p_{down} = \frac{S_{i+1,j'}^{-2} V_{i+1,j}}{(1-\alpha)(\alpha^{-2}-1)} = \frac{e^{\sigma^2 \Delta t} - 1}{(1-\alpha)(\alpha^{-2}-1)}$$

- Use the second row to solve p_{up}:

$$(\alpha - 1) p_{up} + (\alpha^{-1} - 1) p_{down} = S_{i+1,j'}^{-1} E_{i+1,j} - 1$$

- Without div: $p_{up} = \frac{p_{down}}{\alpha}$
- Use the sum to solve p_{mid}

European option pricing

- Build a lattice with N steps and $t_N = T$, the option maturity
- Then value the payoff on each node on T
  - On ITM nodes the payoff is for a call: $S_{N,j} - K$
  - On OTM nodes the payoff is equal to 0
- Propagate the net future value on t_{N-1} nodes:
  - $NFV_{N-1,j}$ = sum of probability x next node value x DF
  - DF is the discount factor = $\exp(-r\ \Delta t)$
- $NFV_{0,0}$ = NPV is the net present value of the option

24-Aug-24 21:50:02

American option pricing

- Same as European options except for the early exercise
- At any time before the maturity the option holder can exercise the option
- On each interim node the value is the maximum of two quantities:
  - Hold: value = discounted value like for European case
  - Exercise: value = $S_{i,j} - K$ for a call on node (i,j)

24-Aug-24 21:50:02

Assignment (1)

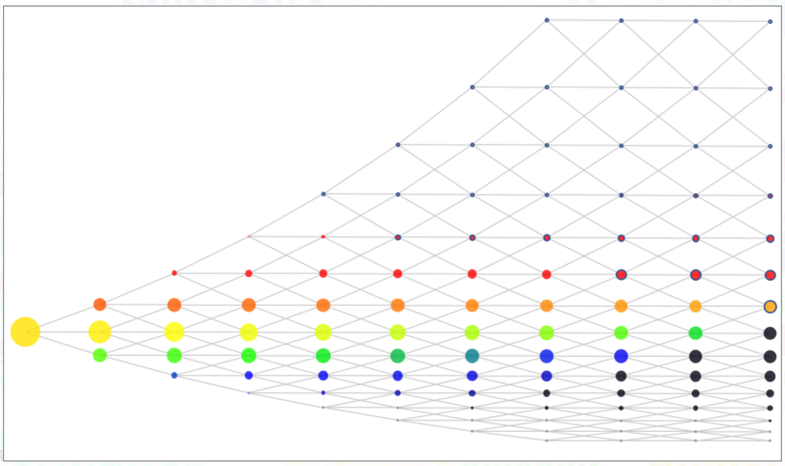
- Build a trinomial lattice and an option pricer
  - Start with few steps, no dividend, European exercise
  - Try to have as much code as possible in the classes
  - Check that the price converges to Black formula
  - Describe the gap as a function of the number of steps
  - Describe the gap as a function of the strike
- Add the American vs. European exercise
  - Study the difference of price American – European
  - Observe the effect of the interest rate (>0 or <0)
  - Describe the cases where the two prices are equal

Assignment (2)

- Add the dividend
  - With interest rates = 0, see the impact on the call and the put
- Other possible extensions:
  - Graph tree
    - Option value
    - Exercise frontier
  - Search the limit of number of steps (time, memory)
    - Try to remove nodes with very low probability
      - Describe the effect on run time for 1000 steps
    - Extend the number of steps
  - Set the number of steps from a user-defined precision

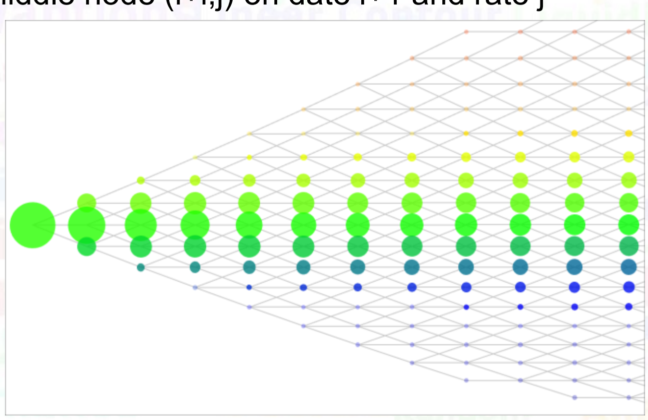
Example of tree with 10 steps

- 1Y ATM American call (vol=30%, int. rate=-1%)



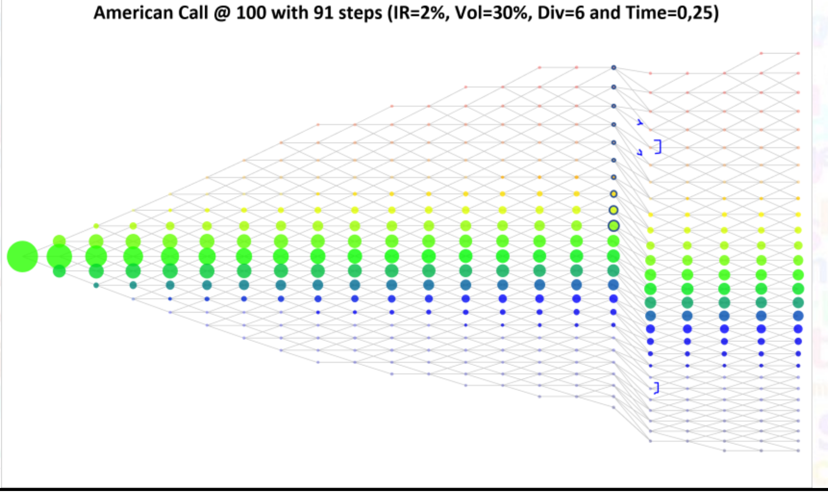
Effect of discrete dividends (1)

- In general, the middle nodes are aligned:
  - The node (i,j) on date i and rate j is connected to the middle node (i+i,j) on date i+1 and rate j



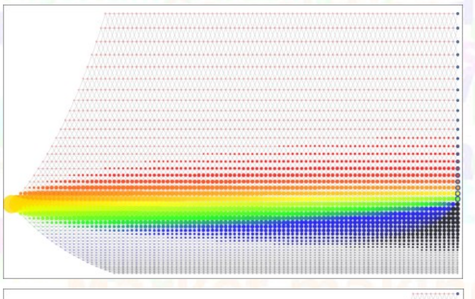
Effect of discrete dividends (2)

- But that doesn't hold with dividends:

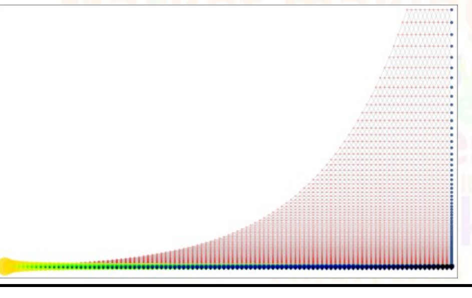


Example of tree with 100 steps

- With prices between 30 and 300:

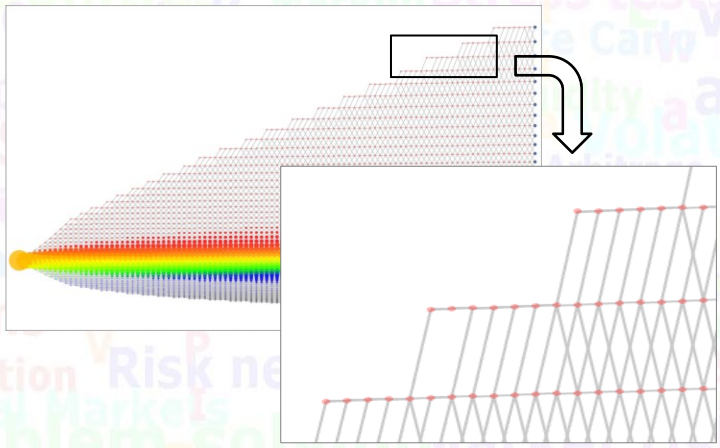


- …and between 1 and 10,000:



Tree with side connections cut

- Tree with mid connection below 1E-7:



Tree "pruning" (1)

- It consists in not allocating nodes which probability is too small
- A typical design consists in branching to the middle node only, with a probability = 1
- The "monomial" branching can be triggered by one of two designs:
  1. by the probability to reach a node, calculated from the root
  2. or by a number of standard deviations of the node from the middle one

Tree "pruning" (2)

- We can calculate the probability to reach a node by:
  - starting with the root node with probability = 100%
  - any other node has a probability equal to the sum of previous nodes connected to it x transition probability
    - Example: the top node has a probability equal to the top node on the previous date x transition probability
    - If the move up probability is, say, 0.18, the probability to be on the top node decreases like 0.18^i, with i the number of the column date. After 9 steps the top node probability is close to 2.10^{-7}.

24-Aug-24 21:50:02

Tree "pruning" (3)

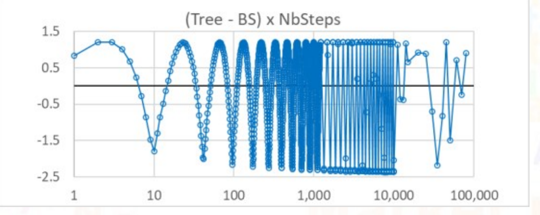
- We can also deduce a node probability by its number of standard deviations from the middle:
  - The process on date i is: $\frac{S_{t_i}}{S_{forward_i}} = e^{\sigma W_{i\,\Delta t} - \frac{1}{2}\sigma^2 \Delta t}$
  - $ln\left(\frac{S_{t_i}}{S_{forward_i}}\right)$ standard deviation = $\sigma\sqrt{i\;\Delta t}$
  - The relative space between nodes is $\sigma\sqrt{3\Delta t}$
  - If we allocate nodes over 4 standard deviations we need k nodes on either sides:

$k\sigma\sqrt{3\Delta t} \geq 4\sigma\sqrt{i\;\Delta t}$, which simplifies into: $k \geq 4\sqrt{\frac{i}{3}}$

24-Aug-24 21:50:02

Calculate number of steps (1)

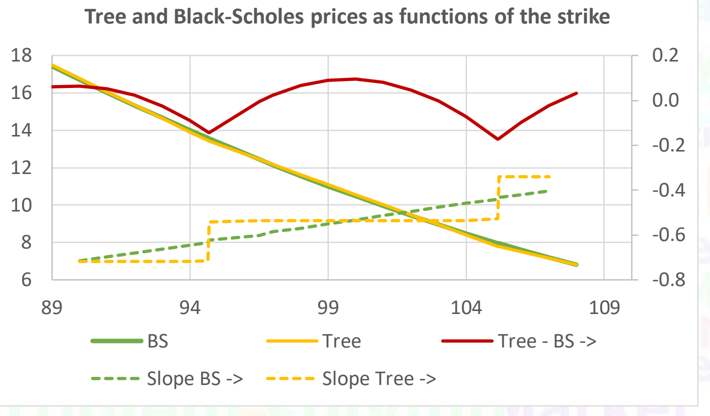
- (Tree – BS) x NbSteps gap lies in a tunnel



- => Gap = $f(S_0, \sigma, K, T, r)$ / NbSteps
- $f(S_0, \sigma, K, T, r)$ is a function of other parameters: vol, strike, maturity, etc.

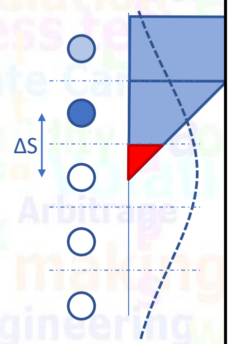
Calculate number of steps (2)

- Tree vs. Black Price function of K (10 steps):



Calculate number of steps (3)

- Idea: consider the ATM forward option
- Gap ≈ lost value of node where S = K
- Value of that node with BS:



Red triangle x density = $\frac{1}{2}\left(\frac{\Delta S}{2}\right)^2 \frac{1}{\sqrt{2\pi}\,\Sigma}$

$$\Delta S = Fwd(T)\ (\alpha - 1)$$

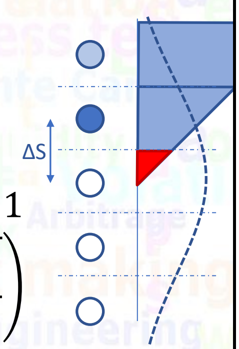
$$\Delta S = (S_0\ e^{rT} - D)\left(e^{r\Delta t + \sigma\sqrt{3\Delta t}} - 1\right)$$

- Σ is the standard deviation of the stock distribution on the maturity date T

$$\Sigma = S_0\ e^{rT}\sqrt{e^{\sigma^2 T} - 1}$$

Calculate number of steps (4)

- Gap function of Δt is:



$$Gap \approx \frac{3\ S_0}{8\sqrt{2\pi}} \frac{\left(e^{\sigma^2 \Delta t} - 1\right) e^{2r\Delta t}}{\sqrt{e^{\sigma^2 T} - 1}}$$

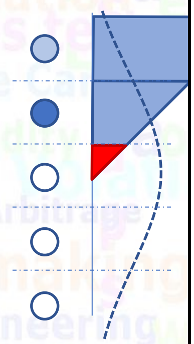
- The main dependency on Δt is in $e^{\sigma^2 \Delta t} - 1$

$$\Delta t \approx \frac{1}{\sigma^2} ln\left(1 + \frac{8\sqrt{2\pi}\ Gap\ \sqrt{e^{\sigma^2 T} - 1}}{3\ S_0}\right)$$

$$NbSteps = \frac{T}{\Delta t} \approx \frac{\sigma^2\ T}{ln\left(1 + \frac{8\sqrt{2\pi}\ Gap\ \sqrt{e^{\sigma^2 T} - 1}}{3\ S_0}\right)}$$

Calculate number of steps (5)

Remarks:



- $\frac{Gap}{S_0}$ is a relative precision on the stock price
- For small gaps, NbSteps = T / Δt is inversely proportional to the gap (and the gap to NbSteps)
- Simplified formulas:

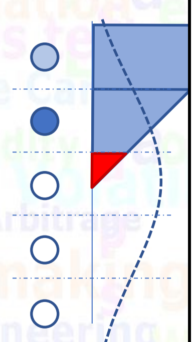
$$Gap \approx \frac{3\,S_0}{8\sqrt{2\pi}} \frac{\sigma^2 \Delta t}{\sqrt{e^{\sigma^2 T}-1}} \qquad \Delta t \approx \frac{8\sqrt{2\pi}}{3} \frac{Gap}{S_0} \frac{\sqrt{e^{\sigma^2 T}-1}}{\sigma^2}$$

$$NbSteps = \frac{T}{\Delta t} \approx \frac{3}{8\sqrt{2\pi}} \frac{S_0}{Gap} \frac{\sigma^2 T}{\sqrt{e^{\sigma^2 T}-1}}$$

24-Aug-24 21:50:02

Calculate number of steps (6)

Remarks:



- With T = 1 year, σ = 30%, we have:
  - $\frac{Gap}{S_0} \approx \frac{1}{N} \frac{3}{8\sqrt{2\pi}} \frac{0.3^2}{\sqrt{e^{0.3^2}-1}}$
  - N=1: $\frac{Gap}{S_0}$ = 4%
  - N=10: $\frac{Gap}{S_0}$ = 0.4%
  - N=100: $\frac{Gap}{S_0}$ = 0.04%
  - N=1000: $\frac{Gap}{S_0}$ = 0.004%

24-Aug-24 21:50:02

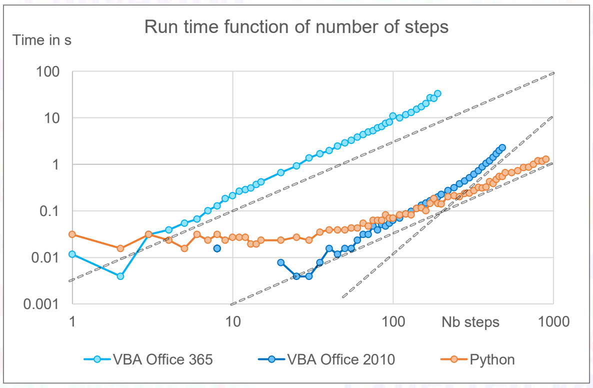
How to check prices?

- You can check several results:
  - European options without dividends compared to Black-Scholes prices
  - European calls with strike = 0, compared with the forward price on maturity, including with dividends
  - American options are harder to test, but the risk of error is lower (very little code is involved) and you can check the exercise frontier

27-Nov-19 21:50:02

Performance: VBA vs. Python

- VBA performance is very poor in 365 version:



Gap analysis: VBA vs. Python

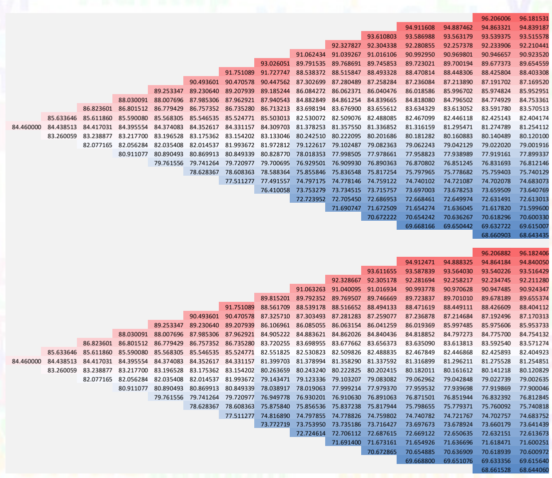
- Test on random parameters:
  - Draw a set of random parameters
  - Price with VBA and Python
  - Store set of parameters if:
    - The gap is larger than 1.10^{-13}
    - Or there is a negative probability

Start date	Start price	iterest rate	Volatility	Dividend	Div ex-date	Maturity	Type	Exercise	Strike	Nb steps	VBA	Py Rec	Py Backward	Gap 1	Gap 2	Run test
23-Aug-22	88.71	1.46%	30.95%	1.36	08-Dec-22	07-Apr-23	Put	European	70.40	46	1.882848	1.882848	1.882848	-3.11E-15	0	On
06-Aug-22	117.04	3.80%	44.15%	10.04	29-Apr-23	08-May-23	Put	American	122.97	81	25.482674	AssertionError: Err	NbStep=81	#VALUE!	#VALUE!	
08-Sep-22	86.50	11.64%	49.14%	5.25	29-May-23	13-Dec-23	Put	American	69.13	70	7.877375	AssertionError: Err	NbStep=70	#VALUE!	#VALUE!	
30-Nov-22	101.49	9.30%	3.91%	0.61	15-Oct-23	29-Oct-23	Call	American	73.13	94	34.065895	34.065895	34.065895	3.41E-13	0	
12-Aug-22	114.21	-4.60%	45.57%	6.17	11-Aug-23	01-Feb-24	Call	European	121.40	81	17.189573	AssertionError: Err	NbStep=81	#VALUE!	#VALUE!	
12-Jan-22	100.60	-2.07%	43.71%	2.21	01-Feb-22	04-Sep-22	Put	American	134.34	90	41.606851	41.606851	41.606851	4.19E-13	0	
19-Mar-22	128.08	-3.09%	40.63%	3.39	24-Aug-22	03-Apr-23	Put	European	199.60	69	85.186486	85.186486	85.186486	4.97E-13	0	
22-Feb-22	84.46	-1.97%	7.14%	2.91	27-Mar-22	29-Apr-22	Put	American	70.14	14	0.000000	0.000000	0.000000	-2.02E-08	0	
14-May-22	81.70	-3.97%	38.45%	1.86	01-May-23	13-Jun-23	Put	European	127.98	91	55.845025	55.845025	55.845025	4.05E-13	0	
28-Jan-22	156.32	10.98%	42.66%	3.16	22-Jul-22	28-Oct-22	Call	American	109.78	91	55.288951	55.288951	55.288951	-5.47E-13	0	
18-Feb-22	163.73	9.71%	10.77%	1.66	02-Apr-22	07-Dec-22	Call	European	108.88	75	61.348146	61.348146	61.348146	-5.47E-13	0	

Example of gap analysis

- Price differ by 2.10^{-8}, for a put very OTM
- Print underlying:
- What happens?

Market	Start date	22-Feb-22	
	Start price	84.46	€
	Interest rate	-1.97%	
	Volatility	7.14%	
	Dividend	2.91	€
	Div ex-date	27-Mar-22	
Trade	Maturity	29-Apr-22	
	Type	Put	
	Exercise	American	
	Strike	70.14	€
Model	Nb steps	14	



The dividend periods differ

- The dividend is not paid on the same period!

22-Feb 00:00	26-Feb 17:08	03-Mar 10:17	08-Mar 03:25	12-Mar 20:34	17-Mar 13:42	22-Mar 06:51	**27-Mar 00:00**	31-Mar 17:08	05-Apr 10:17	10-Apr 03:25	14-Apr 20:34	19-Apr 13:42	24-Apr 06:51	**29-Apr 00:00**
													96.206006	96.181531
											94.911608	94.887462	94.863321	94.839187
										93.610803	93.586988	93.563179	93.539375	93.515578
									92.327827	92.304338	92.280855	92.257378	92.233906	92.210441
								91.062434	91.039267	91.016106	90.992950	90.969801	90.946657	90.923520
							93.026051	**89.791535**	89.768691	89.745853	89.723021	89.700194	89.677373	89.654559
						91.751089	91.727747	**88.538372**	88.515847	88.493328	88.470814	88.448306	88.425804	88.403308
					90.493601	90.470578	90.447562	**87.302699**	87.280489	87.258284	87.236084	87.213890	87.191702	87.169520
				89.253347	89.230640	89.207939	89.185244	**86.084272**	86.062371	86.040476	86.018586	85.996702	85.974824	85.952951
			88.030091	88.007696	87.985306	87.962921	87.940543	**84.882849**	84.861254	84.839665	84.818080	84.796502	84.774929	84.753361
		86.823601	86.801512	86.779429	86.757352	86.735280	86.713213	**83.698194**	83.676900	83.655612	83.634329	83.613052	83.591780	83.570513
	85.633646	85.611860	85.590080	85.568305	85.546535	85.524771	85.503013	**82.530072**	82.509076	82.488085	82.467099	82.446118	82.425143	82.404174
84.460000	84.438513	84.417031	84.395554	84.374083	84.352617	84.331157	84.309703	**81.378253**	81.357550	81.336852	81.316159	81.295471	81.274789	81.254112
	83.260059	83.238877	83.217700	83.196528	83.175362	83.154202	83.133046	**80.242510**	80.222095	80.201686	80.181282	80.160883	80.140489	80.120100
		82.077165	82.056284	82.035408	82.014537	81.993672	81.972812	**79.122617**	79.102487	79.082363	79.062243	79.042129	79.022020	79.001916
			80.911077	80.890493	80.869913	80.849339	80.828770	**78.018353**	77.998505	77.978661	77.958823	77.938989	77.919161	77.899337
				79.761556	79.741264	79.720977	79.700695	**76.929501**	76.909930	76.890363	76.870802	76.851245	76.831693	76.812146
					78.628367	78.608363	78.588364	**75.855846**	75.836548	75.817254	75.797965	75.778682	75.759403	75.740129
						77.511277	77.491557	**74.797175**	74.778146	74.759122	74.740102	74.721087	74.702078	74.683073
							76.410058	**73.753279**	73.734515	73.715757	73.697003	73.678253	73.659509	73.640769
								72.723952	72.705450	72.686953	72.668461	72.649974	72.631491	72.613013
									71.690747	71.672509	71.654274	71.636045	71.617820	71.599600
										70.672222	70.654242	70.636267	70.618296	70.600330
											69.668166	69.650442	69.632722	69.615007
													68.660903	68.643435
													96.206882	96.182406
											94.912471	94.888325	94.864184	94.840050
										93.611655	93.587839	93.564030	93.540226	93.516429
									92.328667	92.305178	92.281694	92.258217	92.234745	92.211280
								91.063263	91.040095	91.016934	90.993778	90.970628	90.947485	90.924347
							89.815201	89.792352	89.769507	89.746669	89.723837	89.701010	89.678189	89.655374
						91.751089	**88.561709**	88.539178	88.516652	88.494133	88.471619	88.449111	88.426609	88.404112
					90.493601	90.470578	**87.325710**	87.303493	87.281283	87.259077	87.236878	87.214684	87.192496	87.170313
				89.253347	89.230640	89.207939	**86.106961**	86.085055	86.063154	86.041259	86.019369	85.997485	85.975606	85.953733
			88.030091	88.007696	87.985306	87.962921	**84.905222**	84.883621	84.862026	84.840436	84.818852	84.797273	84.775700	84.754132
		86.823601	86.801512	86.779429	86.757352	86.735280	**83.720255**	83.698955	83.677662	83.656373	83.635090	83.613813	83.592540	83.571274
	85.633646	85.611860	85.590080	85.568305	85.546535	85.524771	**82.551825**	82.530823	82.509826	82.488835	82.467849	82.446868	82.425893	82.404923
84.460000	84.438513	84.417031	84.395554	84.374083	84.352617	84.331157	**81.399703**	81.378994	81.358290	81.337592	81.316899	81.296211	81.275528	81.254851
	83.260059	83.238877	83.217700	83.196528	83.175362	83.154202	**80.263659**	80.243240	80.222825	80.202415	80.182011	80.161612	80.141218	80.120829
		82.077165	82.056284	82.035408	82.014537	81.993672	**79.143471**	79.123336	79.103207	79.083082	79.062962	79.042848	79.022739	79.002635
			80.911077	80.890493	80.869913	80.849339	**78.038917**	78.019063	77.999214	77.979370	77.959532	77.939698	77.919869	77.900046
				79.761556	79.741264	79.720977	**76.949778**	76.930201	76.910630	76.891063	76.871501	76.851944	76.832392	76.812845
					78.628367	78.608363	**75.875840**	75.856536	75.837238	75.817944	75.798655	75.779371	75.760092	75.740818
						77.511277	**74.816890**	74.797855	74.778826	74.759802	74.740782	74.721767	74.702757	74.683752
							73.772719	73.753950	73.735186	73.716427	73.697673	73.678924	73.660179	73.641439
								72.724614	72.706112	72.687615	72.669122	72.650635	72.632151	72.613673
									71.691400	71.673161	71.654926	71.636696	71.618471	71.600251
										70.672865	70.654885	70.636909	70.618939	70.600972
											69.668800	69.651076	69.633356	69.615640
													68.661528	68.644060

Equality between dates (1)

- Initial date + n x time step can seem different from the ex-dividend date
- Solution: define the equality with a tolerance

```
' returns True if the div falls between the current date (excluded) and the next one (included)
Public Function IsDivAfterThisDate() As Boolean
  Dim divDate As Date
  Let divDate = Me.TheTree.Market.DivExDate
  Let IsDivAfterThisDate = Not (Me.AreEqualDates(Me.ColumnDate, divDate)) _
    And Me.ColumnDate < divDate _
    And (divDate < Me.NextDate() Or Me.AreEqualDates(divDate, Me.NextDate()))
End Function

' returns True if the two dates are closer than a small fraction of time
Function AreEqualDates(ByVal d1 As Date, ByVal d2 As Date) As Boolean
  Let AreEqualDates = Abs(d1 - d2) < 1 / Me.TheTree.nbSteps / 10
End Function
```

Equality between dates (2)

- In Python, as a member in TruncNode:

```
class TruncNode(Node):
  # constructor with reference to the Node constructor
  def __init__(self, precNode, colDate, price, tree):
    Node.__init__(self, price, self, tree)
    self.precMid: TruncNode = precNode
    self.columnDate = colDate
    self.nextMid: TruncNode = None
    exDivDate = self.tree.mkt.divExDate
    # means self.columnDate < exDivDate <= self.nextDate()
    self.isDivInTheFollowingPeriod = not(self.areSameDates(self.columnDate, exDivDate)) \
      and self.columnDate < exDivDate \
      and (exDivDate < self.nextDate() or self.areSameDates(exDivDate, self.nextDate()))

  # this equality between dates avoids small numerical errors that make dates look different
  def areSameDates(self, d1, d2):
    return abs(d1 - d2) < datetime.timedelta(days=1) / self.tree.nbSteps / 10
```

Price precision: lower in Python

- The largest gaps between VBA and Python are around 1.10^{-11} € on stock prices around 100 €
- Observing the transition probabilities (without dividend), we can see that Python values are 1000 times more instable with low volatilities:

Start date	08-Jun-22	
Start price	112.53	€
Interest rate	2.00%	
Volatility	**0.44%**	
Dividend	0.00	€
Div ex-date	19-Jun-22	
Maturity	08-Jul-22	
Type	Call	
Exercise	European	
Strike	107.44	€
Nb steps	6	

VBA, down probabilities

		0.1667410088542450
	0.1667410088544540	0.1667410088542450
0.1667410088543840	0.1667410088543840	0.1667410088543840
	0.1667410088543860	0.1667410088543130
		0.1667410088542420

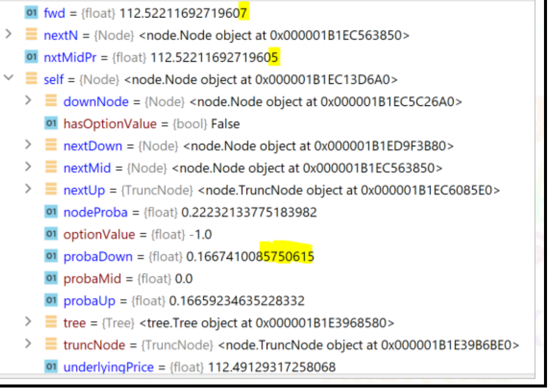
Python

		0.1667410087146660
	0.1667410088544570	0.1667410087146660
0.1667410088543950	0.1667410088543950	0.1667410088543950
	0.1667410088544570	0.1667410085750610
		0.1667410088542700

- Why?!

Analysis

- Underlying prices differ only by the last 2 digits
- The same for forward rates
- Variances are all exactly equal
- In the debugger, the only visible gap is between the **forward** and the value on the **next node**, just on the last digit
- Is it enough?



Detailed calculation

- The down probability has the following code:

```
self.probaDown = ((variance + fwd * fwd) / (nxtMidPr * nxtMidPr) – 1
- (a + 1) * (fwd / nxtMidPr - 1)) / ((1 - a) * (1 / (a * a) - 1))
```

$$p_{down} = \frac{S_{i+1,j'}^{-2}\left(V_{i+1,j} + E_{i+1,j}^{2}\right) - 1 - (\alpha + 1)\left(S_{i+1,j'}^{-1} E_{i+1,j} - 1\right)}{(1 - \alpha)(\alpha^{-2} - 1)}$$

- The numerator is 2.7 10^{-7} and the denominator 1.6 10^{-6}. Therefore a small error in the numerator is magnified in p_{down}.
- The solution is to use the simplified formula when there is no dividend (store 3 values in tree)

Other precision points

- It also helps to rephrase the math formula:

$$\frac{V_{i+1,j} + E_{i+1,j}^2}{S_{i+1,j'}^2} - 1$$

- It is less precise when the next node and forward values are close than this variation:

$$\frac{V_{i+1,j} + (E_{i+1,j} + S_{i+1,j'})(E_{i+1,j} - S_{i+1,j'})}{S_{i+1,j'}^2}$$

- Ditto for the other term:

$$\frac{E_{i+1,j}}{S_{i+1,j'}} - 1 = \frac{E_{i+1,j} - S_{i+1,j'}}{S_{i+1,j'}}$$

Organization

- For questions please use Moodle **Forum**.
- Please post your spreadsheet on Moodle before Sunday 3rd of November
- Please include your name in the files:
  - Folder_<name1>_<name2>…
    - Excel file
    - Memo: Word or pdf file
    - Python file 1,
    - Python file 2, etc.

24-Aug-24 21:50:02