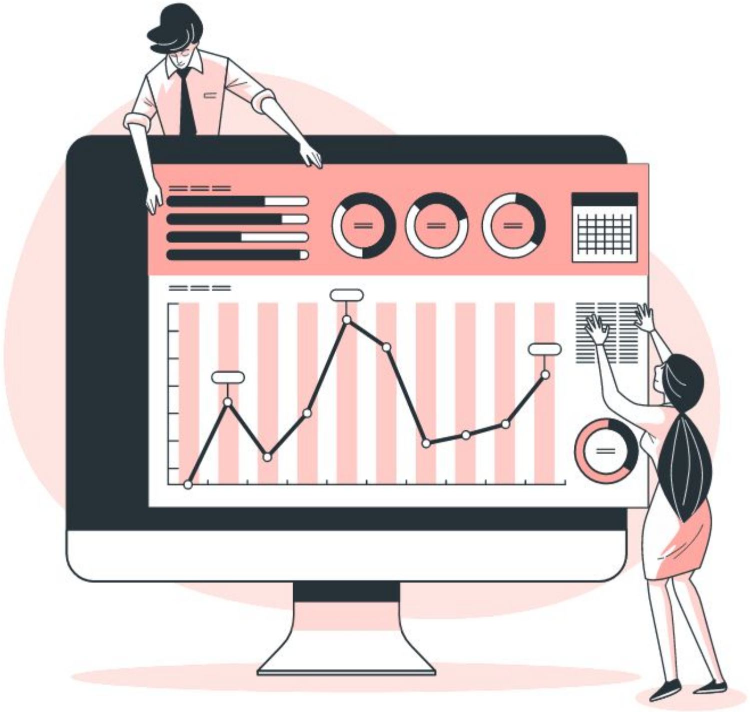
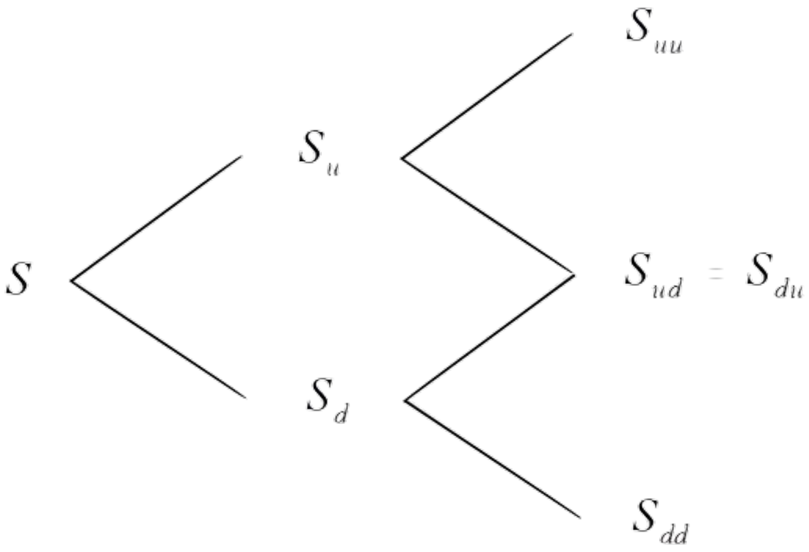
Binomial Tree Models

Pricing a European Option with Two Correlated Assets



Introduction



The members of this group are Lauren, Kobe, Michelle and Ayan

Introduction

This topic takes the *basis of Binomial Trees* that we've been learning about in class and applies it to *more complex situations*.

Introduction

We are curious about the impact of having *multiple correlated assets*, like numerous stocks or bonds, what this might look like in terms of diagrams and algorithms in comparison to the single asset case, as well as *what the outcome could look like in the real world*.

Table of contents

01 Objectives

What we want to accomplish with this research

02 Literature Review

Expansion on sources we found helpful

03 Theory and Algorithms

How to use the Binomial Tree model for 2+ assets

04 Applications

What this looks like in action - simulation and in the real world

05 Conclusion

Meanings, messages and more

01

Objectives

Main Objectives

1.
To provide a clear and complete explanation of the process for using a Binomial Tree model with multiple assets.

2.
Develop the single asset binomial case for multiple correlated assets

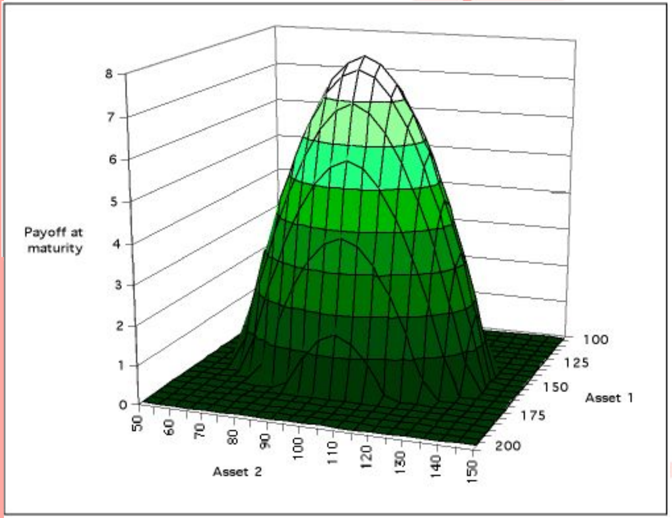
3.
Creating a Python code to evaluate option prices

02

Literature Review

Resources

ScienceDirect	Quantitative information highlighting multi-asset cases for both European and American price options
EspenHaug	Qualitative visual aids with representations in other topics



03

Theory and Algorithms

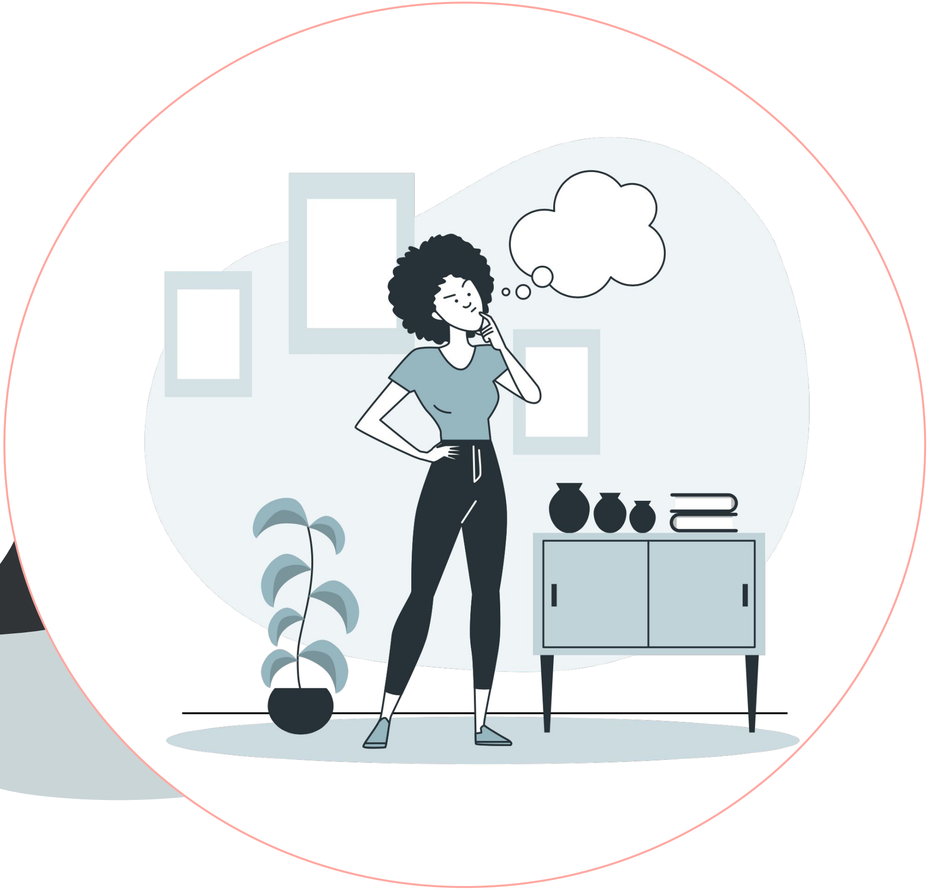
Basic Binomial Tree Model

- Each step in the tree has a probability of *p* to increase, and *1-p* to decrease
- In this model, the stock price cannot be stagnant over a step in the tree, it must increase or decrease
- Computation requires data such as the strike price, the period and the interest rate

But...

Real life data, especially stocks, are often very interconnected.

So what happens when you have to look at more than 1 asset at a time?



2 Asset Binomial Case

01

Inputs

02

Equations and Process

03

Outputs

2 Asset Binomial Case

01

Inputs

- T : the time to maturity of the assets
- N : the number of intervals/steps in $[0,T]$
- t_n : the time intervals from $0=t_0<t_1<...<t_N=T$
- r : the portfolio's interest rate
- X_1 : the spot price of the first asset
- X_2 : the spot price of the second asset
- K_1 : the strike price of asset 1
- K_2 :the strike price of asset 2
- σ_1 : the volatility of the first asset
- σ_2 : the volatility of the second asset
- ϱ : the correlation coefficient for the two assets

2 Asset Binomial Case

02

Equations and Process

$t_n = n\Delta t$ for $n=1,2,...,N$ and $\Delta t=T/N$:

Then we have the following cases for t, while h_i, $i=1,2$, denotes the increment in each coordinate direction:

- $t=0$: $x(0)=(x_1(0),x_2(0))=(x_1^0,x_2^0)$
- $t=t$:
    a. $(x_1^0+h_1,x_2^0+h_2)$ with probability p_{uu}
    b. $(x_1^0+h_1,x_2^0-h_2)$ with probability p_{ud}
    c. $(x_1^0-h_1,x_2^0-h_2)$ with probability p_{dd}
    d. $(x_1^0-h_1,x_2^0+h_2)$ with probability p_{du}

2 Asset Binomial Case

02

Equations and Process

Asset Price : $x(n)=(x_{1,j}^{n},x_{2,j}^{n})$ at (i,j) at time $t=n\Delta t$ can move to any of the following 4 cases:

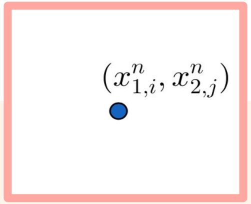
a. $(x_{1,i}^{n}+h_1,x_{2,j}^{n}+h_2)$ with probability p_{uu}
b. $(x_{1,i}^{n}+h_1,x_{2,j}^{n}-h_2)$ with probability p_{ud}
c. $(x_{1,i}^{n}-h_1,x_{2,j}^{n}-h_2)$ with probability p_{dd}
d. $(x_{1,i}^{n}-h_1,x_{2,j}^{n}+h_2)$ with probability p_{du}

What does this look like?

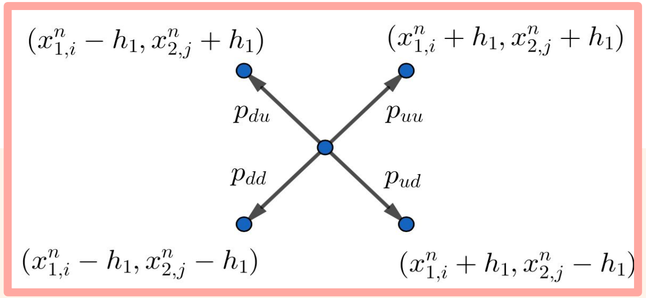
2 Asset Binomial Case

02

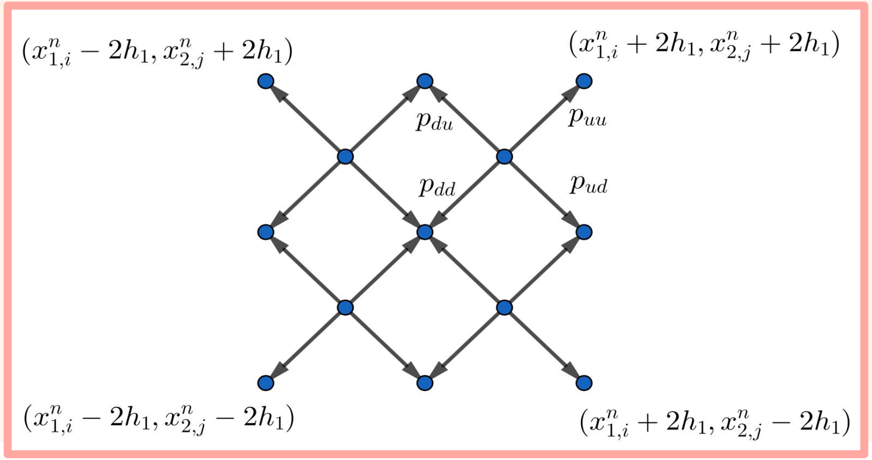
Equations and Process



$t=n\Delta t$



$t=(n+1)\Delta t$



$t=(n+2)\Delta t$

2 Asset Binomial Case

02

Equations and Process

A single increment in this type of continuous process has the following properties:

- Mean = 0
- Variance = $(\sigma_1^2 \Delta t, \sigma_2^2 \Delta t)$
- Covariance = $\varrho \sigma_1 \sigma_2 \Delta t$

Properties of probability also imply the following:

- $p_{uu} = p_{dd} = 1/4(1+\varrho)$, $p_{du} = p_{ud} = 1/4(1-\varrho)$,
- $h_1 = \sigma_1 \sqrt{\Delta t}$, $h_2 = \sigma_2 \sqrt{\Delta t}$

2 Asset Binomial Case

02

Equations and Process

Payoff: begin with an approximation $\overline{V}_{i,j}^{\,n}$ to $V_{i,j}^{\,n}$.

- Generate all possible values of the asset price X^N at maturity, when $T=N\Delta t$
- Compute the payoff at each of those values
  - $\overline{V}_{i,j}^{\,N} = \boldsymbol{\Lambda} X_{i,j}^{\,N}$ for $i, j = -N, -N + 2, \dots, N - 2, N$

$$\overline{V}_{i,j}^{\,N} = \boldsymbol{\Lambda} X_{i,j}^{\,N} = \begin{cases} \max\{S_2(T) - K_2, 0\}, & \text{if } S_1(T) > K_1 \\ 0, & \text{otherwise} \end{cases}$$

2 Asset Binomial Case

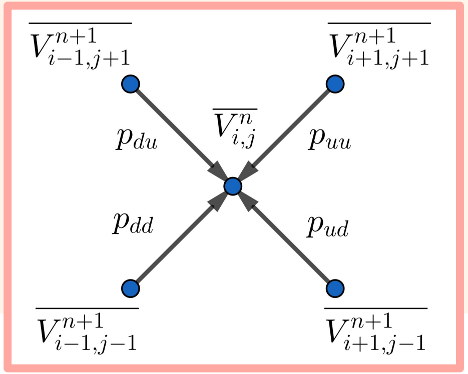
02

Equations and Process

- Discount the payoff for for $n = N - 1, \ldots, 0$ and

  $i, j = -n, -n + 2, \ldots, n - 2, n$

  - $\overline{V_{i,j}^{n}} = e^{-rt}[(\overline{V_{i+1,j+1}^{n+1}}p_{uu}) + (\overline{V_{i+1,j-1}^{n+1}}p_{ud}) + (\overline{V_{i-1,j+1}^{n+1}}p_{du}) + (\overline{V_{i-1,j-1}^{n+1}}p_{dd})]$
- Continue evaluating until you reach $\overline{V_{0,0}^{0}}$ at $t=0$

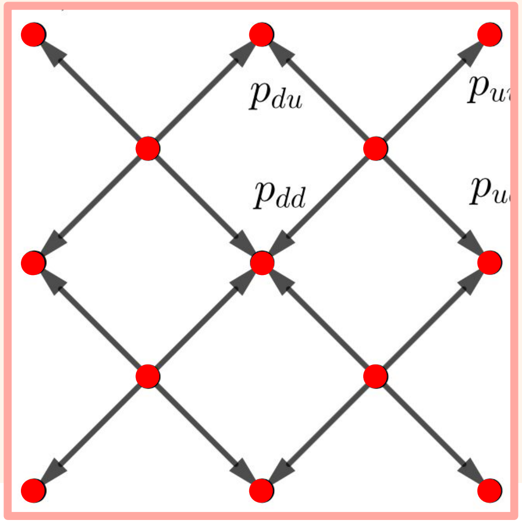


2 Asset Binomial Case

03

Outputs

Option Price : $\overline{V_{0,0}}^{0}$ at $t=0$



More than 2 Assets?

We've got it covered!

Variables

d : the number of assets
T : the time to maturity
N : number of intervals in $[0,T]$
ϱ, σ's, X's, K's, for each asset

Algorithms

$\sum_{k=1}^{2d} \varrho_k = 1$: probability sum

$P_k = 1/2^d \, (1 + \sum_{l,m=1}^{d} \delta_{lm} \varrho_{lm})$: probability calculation, with $l<m$

$h_k = \sigma_k \sqrt{\Delta t}$, all with $k=1,\ldots,d$

Here δ_{lm} is the sign function with the following conditions:

- $\delta_{lm} = 1$ if the l^{th} and m^{th} asset move in the same direction
- $\delta_{lm} = -1$ if the l^{th} and m^{th} asset move in opposite directions

04

Applications

Let’s test it out!

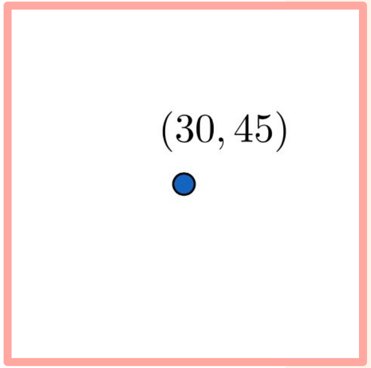
Here are our variables:

(2 years, 2)	15%	(30, 45)	(20, 50)	(15%, 12%)	0.37
Term, steps in [0,T]	Interest Rate	Spot Prices	Strike Prices	Volatility	Correlation Coefficient

In action...

First Increment, *t*=0

- Initial Asset Prices



Asset 1:

- X_1=\$30
- K_1=\$20
- σ_1=15%
- t=0,1,2

Asset 2:

- X_2=\$45
- K_2=\$40
- σ_2=12%
- t=0,1,2

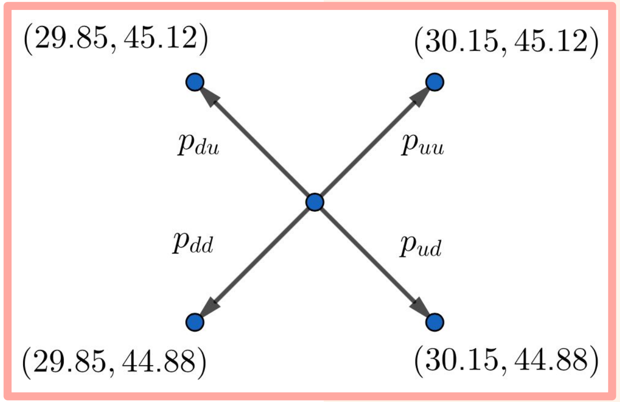
Probabilities

- $p_{uu} = p_{dd} = 1/4(1+0.37) = 0.3425$
- $p_{du} = p_{ud} = 1/4(1-0.37) = 0.1575$

$h_1 = 0.15$ $h_2 = 0.12$

In action...

Second Increment, *t=1*

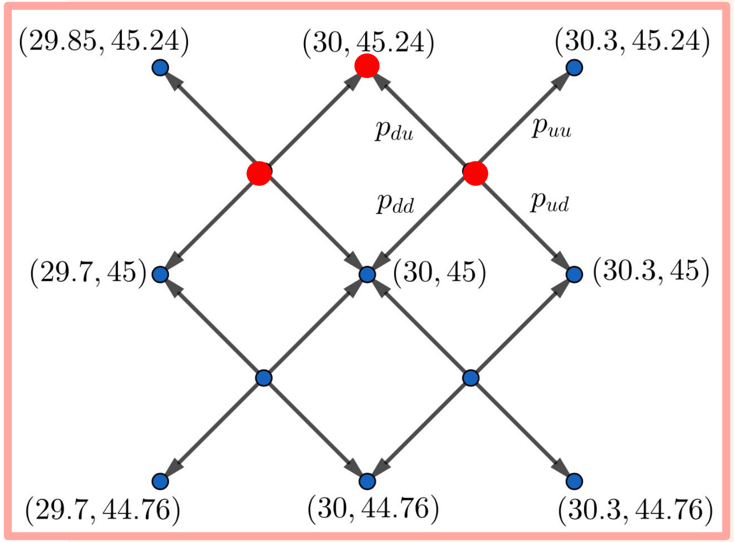


Cases:

- X_1 up, X_2 up
  - *(30+0.15,45+0.12)*
- X_1 up, X_2 down
  - *(30+0.15,45-0.12)*
- X_1 down, X_2 down
  - *(30-0.15,45-0.12)*
- X_1 down, X_2 up
  - *(30-0.15,45+0.12)*

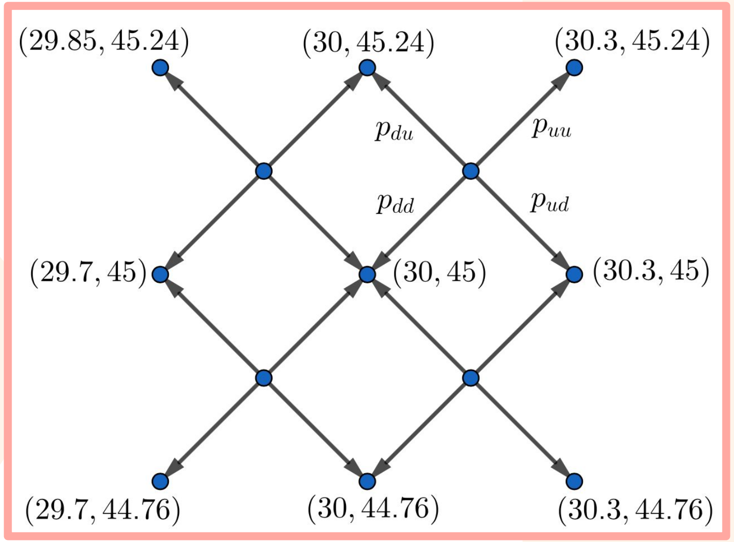
In action...

Third Increment, *t=2*



In action...

Option Prices



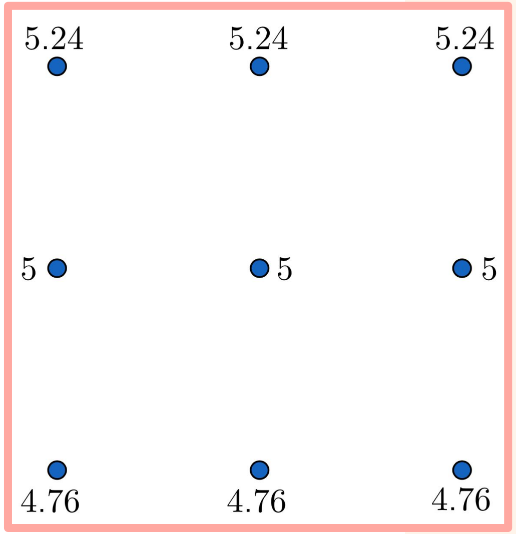
Payoff at $n=T=2$:

$$\overline{V_{i,j}}^{N} = \begin{cases} max\{S_2(T) - K_2, 0\}, & \text{if } S_1(T) > K_1 \\ 0, & \text{otherwise} \end{cases}$$

1. Check the condition "if $S_1(T) > K_1$"
   - Each node for $T=2$ satisfies the condition
2. Compute $max\{S_2(T) - K_2, 0\}$ for each node

In action...

Option Prices

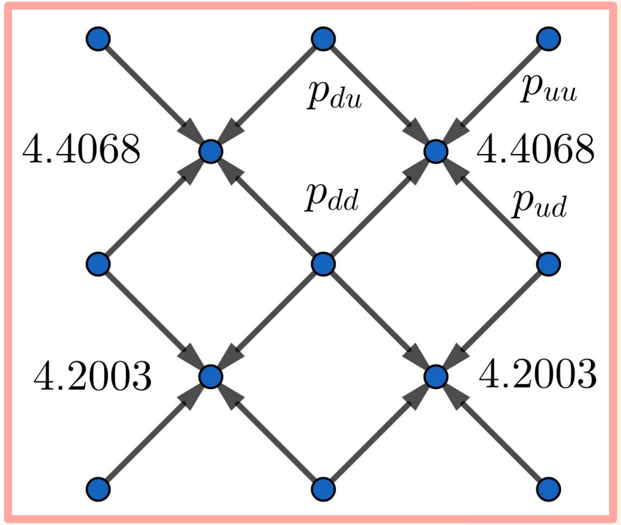


Payoff at $n=T=2$:

- $\overline{V}_{2,2}^{2} = max\{45.24-40,0\}=5.24$
- $\overline{V}_{2,0}^{2} = max\{45-40,0\}=5$
- $\overline{V}_{2,-2}^{2} = max\{44.76-40,0\}=4.76$
- $\overline{V}_{0,-2}^{2} = max\{44.76-40,0\}=4.76$
- $\overline{V}_{-2,-2}^{2} = max\{44.76-40,0\}=4.76$
- $\overline{V}_{-2,0}^{2} = max\{45-40,0\}=5$
- $\overline{V}_{-2,2}^{2} = max\{45.24-40,0\}=5.24$
- $\overline{V}_{0,2}^{2} = max\{45.24-40,0\}=5.24$
- $\overline{V}_{0,0}^{2} = max\{45-40,0\}=5$

In action...

Option Prices

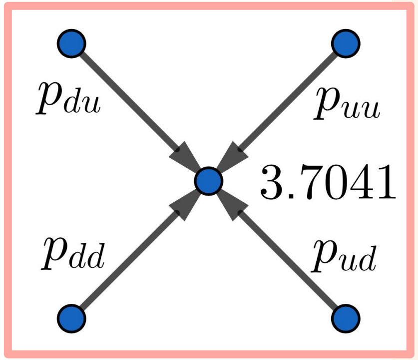


Payoff at *n=1*

- $\overline{V}_{1,1}^{1} = e^{-0.15*1}[(5.24)(0.3425) + (5)(0.1575) + (5)(0.3425) + (5.24)(0.1575)] = \mathbf{4.4068}$
- $\overline{V}_{1,-1}^{1} = e^{-0.15*1}[(5)(0.3425) + (4.76)(0.1575) + (4.76)(0.3425) + (5)(0.1575)] = \mathbf{4.2003}$
- $\overline{V}_{-1,-1}^{1} = V_{1,-1}^{1} = \mathbf{4.2003}$
- $\overline{V}_{-1,1}^{1} = V_{1,1}^{1} = \mathbf{4.4068}$

In action...

Option Prices



Payoff at *n*=0

- $\overline{V_{0,0}^{0}} = e^{-0.15*1}[(4.4068)(0.3425) + (4.2003)(0.1575) + (4.2003)(0.3425) + (4.4068)(0.1575)] = \boldsymbol{3.7041}$

Therefore our portfolio's option price is $3.70

Code Overview – Putting it into Action

```
#%% Given Values

T=2
N=2
dt=T/N
r=0.15

X1=30
X2=45
X_0=[X1,X2]

K1=20
K2=40
K=[K1,K2]

sigma1=0.15
sigma2=0.12
sigmas=[sigma1,sigma2]

rho=0.37
```

```
#%% Probabilities and Increments

p_uu=1/4*(1+rho)
p_dd=p_uu
p_du=1/4*(1-rho)
p_ud=p_du

h1=sigma1*math.sqrt(dt)
h2=sigma2*math.sqrt(dt)
h=[h1,h2]
```

The code is an edited version from https://odr.chalmers.se/bitstream/20.500.12380/255030/1/255030.pdf

Code Overview – Putting it into Action

```
#%% Asset Price

def asset_price(X_0, sigmas, N, T):
    X_T=np.zeros((N+1,N+1, 2))
    for column in range(0, N+1):
        for row in range(0, N+1):
            for element in range (0 , 2) :
                if element == 0:
                    X_T[row][column][element] = X_0[element] + h[element]*(-N+2*column)
                if element == 1:
                    X_T[row][column][element] = X_0[element] + h[ element ]*(N-2*row)
    return X_T
```

The code is an edited version from
https://odr.chalmers.se/bitstream/20.500.12380/255030/1/255030.pdf

Code Overview – Putting it into Action

```
#%% Payoff

def payoff( prices , r , sigmas , K, T) :
    [dim_row, dim_column, dim_element] = prices.shape
    payoff_T = np. zeros((dim_row , dim_column))
    for column in range(0, dim_column):
        for row in range(0, dim_row):
            X_1 = prices[row][column][0]
            if X_1>K[0]:
                X_2 = prices[row][column][1]
                payoff_T[row][column]=max(X_2-K[1], 0)
    return payoff_T
```

The code is an edited version from
https://odr.chalmers.se/bitstream/20.500.12380/255030/1/255030.pdf

Code Overview – Putting it into Action

```
#%% Initial Payoff

def initial_price(payoff, rho, r, N, T) :
    dt = T/N
    [ dim_row , dim_column ] = payoff.shape
    P_uu = P_dd = 1/4*(1+rho)
    P_ud = P_du = 1/4*(1-rho)
    while dim_row > 1:
        discounted_option_value = np.zeros((dim_row-1, dim_column-1))
        for row in range(0, dim_row-1):
            for column in range(0, dim_column-1):
                v_uu = payoff[row][column+1]
                v_dd = payoff[row+1][column]
                v_du = payoff[row][column]
                v_ud = payoff [row+1][column+1]
                discounted_option_value[row][column] = np.exp(-r*dt)*(
v_uu*P_uu+v_ud*P_ud+v_du*P_du+v_dd*P_dd )
        return initial_price(discounted_option_value , rho , r , N, T)
    else :
        return payoff [0][0]
```

The code is an edited version from
https://odr.chalmers.se/bitstream/20.500.12380/255030/1/255030.pdf

Code Overview – Putting it into Action

```python
#%% Evaluating the entire code

def binomial_two_asset(X_0, sigmas, r, rho, T , N, K):
    prices = asset_price(X_0, sigmas, N, T)
    payoff_T = payoff(prices, r, sigmas , K, T)
    option_price = initial_price( payoff_T, rho, r, N, T)
    return option_price

print("The option price of the two correlated assets in this")
print("European Call option portfolio is: ")
print("$","%.2f" % binomial_two_asset(X_0, sigmas, r, rho, T , N, K))
```

```
The option price of the two correlated assets in this
European Call option portfolio is:
$ 3.70
```

The code is an edited version from
https://odr.chalmers.se/bitstream/20.500.12380/255030/1/255030.pdf

Advantages and Limitations

2 Asset Advantage

Set of 6 equations, 6 unknowns to solve

3 Asset Advantage

Closed set of equations with a known number of unknowns to solve for

2 Asset Limitation

Only a rough approximation

3 Asset Limitation

Also only a rough approximation, more assumptions made

05

Conclusions

Thanks!

Questions?

slidesgo