Financial Systems Design Part II

Derivatives Pricing with Numerical Methods

July 2020

Slido Event: #FSD1

SINGAPORE UNIVERSITY OF TECHNOLOGY AND DESIGN

Outline

- Introduction
- Derivatives in Banking
- Overview of Derivatives Pricing
  - PDE and Probability
  - Finite Difference and Monte Carlo
- Python for this course
  - Sparse matrices in Python
- Discretization of derivatives

Introduction

SINGAPORE UNIVERSITY OF
TECHNOLOGY AND DESIGN

Profile

- Lu Chung I (not a professor so just call me Lu)
- 12 years in Credit Suisse
- Head of Front Office Derivatives Risk Management for the Private Banking division
- Previously a member of the Structured Products Advisory team
- Alumni of NUS (Quant Finance) and NTU (MFE)

Introduction to Course

- This part of the course focuses on the design of valuation model (VM) algorithms for pricing derivatives
- Expectation post-course is not for you to be able to build your VM
- Gain understanding of the principles behind the VM and how it interacts with the rest of the bank
- Requires a certain level of proficiency in programming
- Exam will be about applying the techniques taught in this course

Introduction to Course

- In-Class exercises using Jupyter notebook to get comfortable with Python
- Skeleton of the code is provided with key portions left blank
- Homework also largely consists of assignments in a similar vein to the In-Class exercises but with more challenging problems

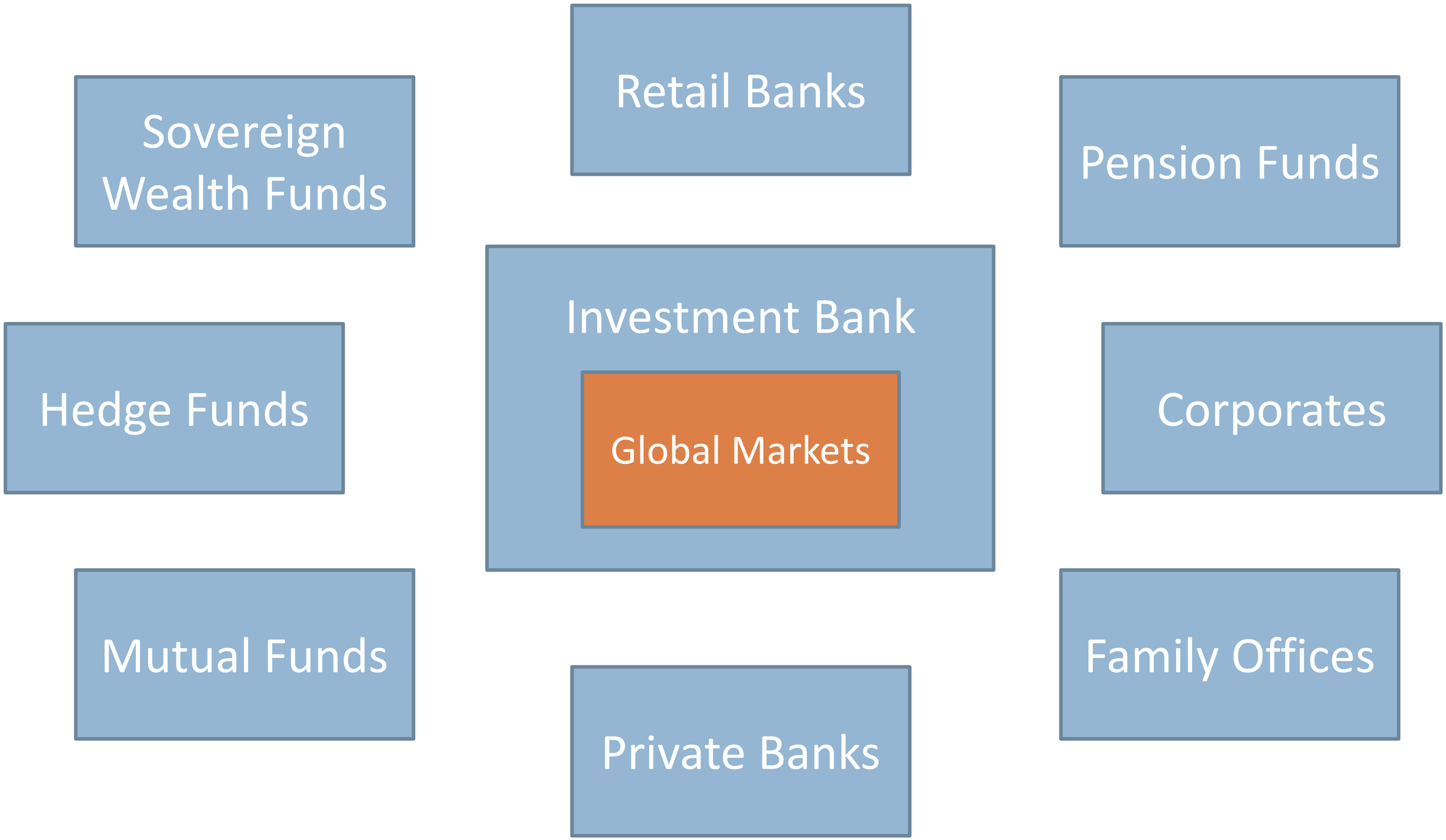
Derivatives in Banking

SINGAPORE UNIVERSITY OF TECHNOLOGY AND DESIGN

Derivatives in Banking

- Total notional value of all financial derivatives in existence in 2010 was USD 1.2 quadrillion
- Derivatives can be used for speculation or hedging
- They are typically transacted through an investment bank
- The cross of mathematics and finance was the spark for this explosion in derivatives starting with the Black Scholes model for pricing vanilla options

Derivatives in Banking



Derivatives in Banking

Global Markets

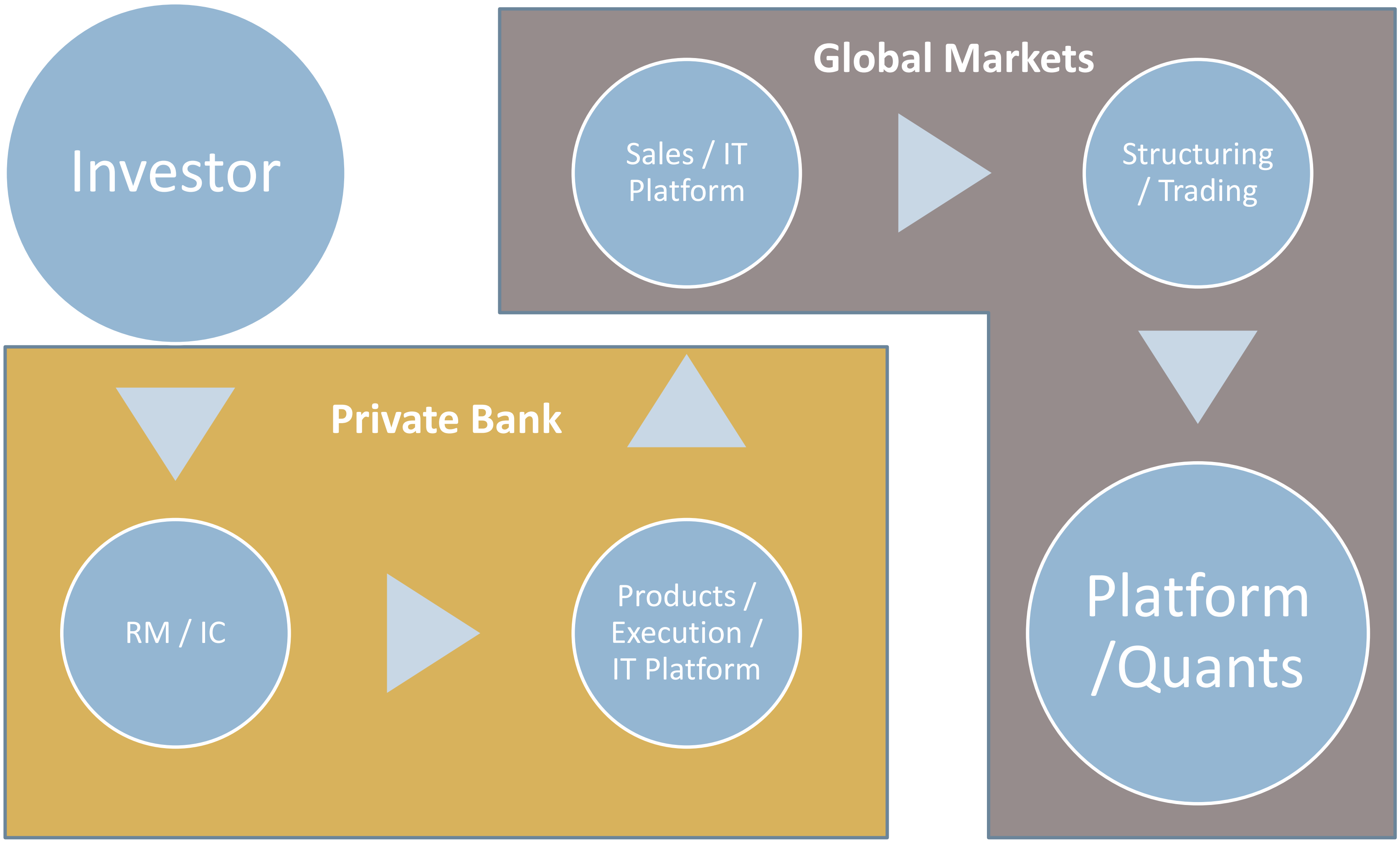
- Sales
- Quants
- Structuring
- Platform / IT
- Trading
- Support Functions

Derivatives in Banking

Private Bank

- Relationship Manager
- Investment Consultant
- Products
- Platform / IT
- Execution
- Support Functions

Derivatives in Banking



Derivatives in Banking

- In Private Banking, derivatives are often offered as a package in what is called a Structured Product
- A Structured Product is a combination of traditional financial instruments (typically a bond) & derivatives
- Example: Equity Linked Note (ELN) is a conventional name given to a specific Structured Product
- It is a combination of a long position in a bond + a short position and an equity European put option

Derivatives in Banking

- Typical terms highlighted in an ELN termsheet
  - Issuer: ABC Bank
  - Tenor: 6 months
  - Currency: USD
  - Underlying: Uber Technologies Inc
  - Strike: 90% of initial spot
  - Yield: ?% p.a.
  - Issue Price: 92%
  - Notional: USD 500,000
- An investor would pay the notional x issue price to buy this ELN i.e. USD 460,000

Derivatives in Banking

- What happens in 6 months for the buyer of this ELN
  - If Uber is at or above the strike, the issuer will redeem this ELN at USD 500,000
  - If Uber is below the strike, the issuer will deliver shares equal to notional / strike price
- 1st scenario: Buyer earns the yield of ?% p.a.
- 2nd scenario: Buyer sits on an unrealised loss as the shares were bought at the strike price but the value is now below the purchase price

Derivatives in Banking

- The two scenarios can be explained by looking at how each component of the ELN behaves
- The bond is priced at a discount to par and matures at par at maturity (assume a zero-coupon bond)
- The put option is priced at a positive premium and can expire worthless or be exercised
- 1st Scenario: Bond maturing at par and put expiring worthless i.e. buyer pockets the option premium
- 2nd Scenario: Bond matures at par but the proceeds get converted to stock as the put gets exercised

Derivatives in Banking

- Price of the ELN = Price of the bond minus the put option premium
- Prices of both components are required to price the ELN
- Bonds depend on the credit spread of the issuer
- How should we price the put option?

Overview of Derivative Pricing

SINGAPORE UNIVERSITY OF TECHNOLOGY AND DESIGN

Purpose of Pricing / Valuation

In the absence of an active market allowing for price discovery e.g. exchange traded derivatives, pricing/valuation models provide the following:

- Determine a fair price for over-the-counter (OTC) trades
- Valuation of positions and portfolios

Even with a market based price discovery mechanism, the models are still required for the following:

- Margining of a derivative position
- Determining a hedging strategy

What is Derivative Pricing?

- Finding a fair (arbitrage free) price for a contingent claim (derivative)
- The pricing mechanism also determines how you can replicate the contingent claim using primary market instruments e.g. the stock itself
- Mathematical models are used for the pricing mechanism
- The models typically make assumptions to simplify reality to solvable problems

Two Approaches to Derivative Pricing

1. Partial Differential Equations (PDEs) constructed by no arbitrage arguments with self financing portfolio (Approach by Black, Scholes and Merton)

2. Probability approach using the Martingale Measure (or Risk Neutral Measure) to calculate discounted Expected value of the contingent claim (payoff of the derivative)

PDE Approach

- Finite Difference Methods (FDM)
  1. Discretize continuous variables
  2. Start from known quantities of the variables (boundary conditions)
  3. Differential equations provide a map of how variables change with respect to each other (e.g. time, price of underlying , value of derivative)
  4. Evolve the known quantity in the direction required using the differential equations

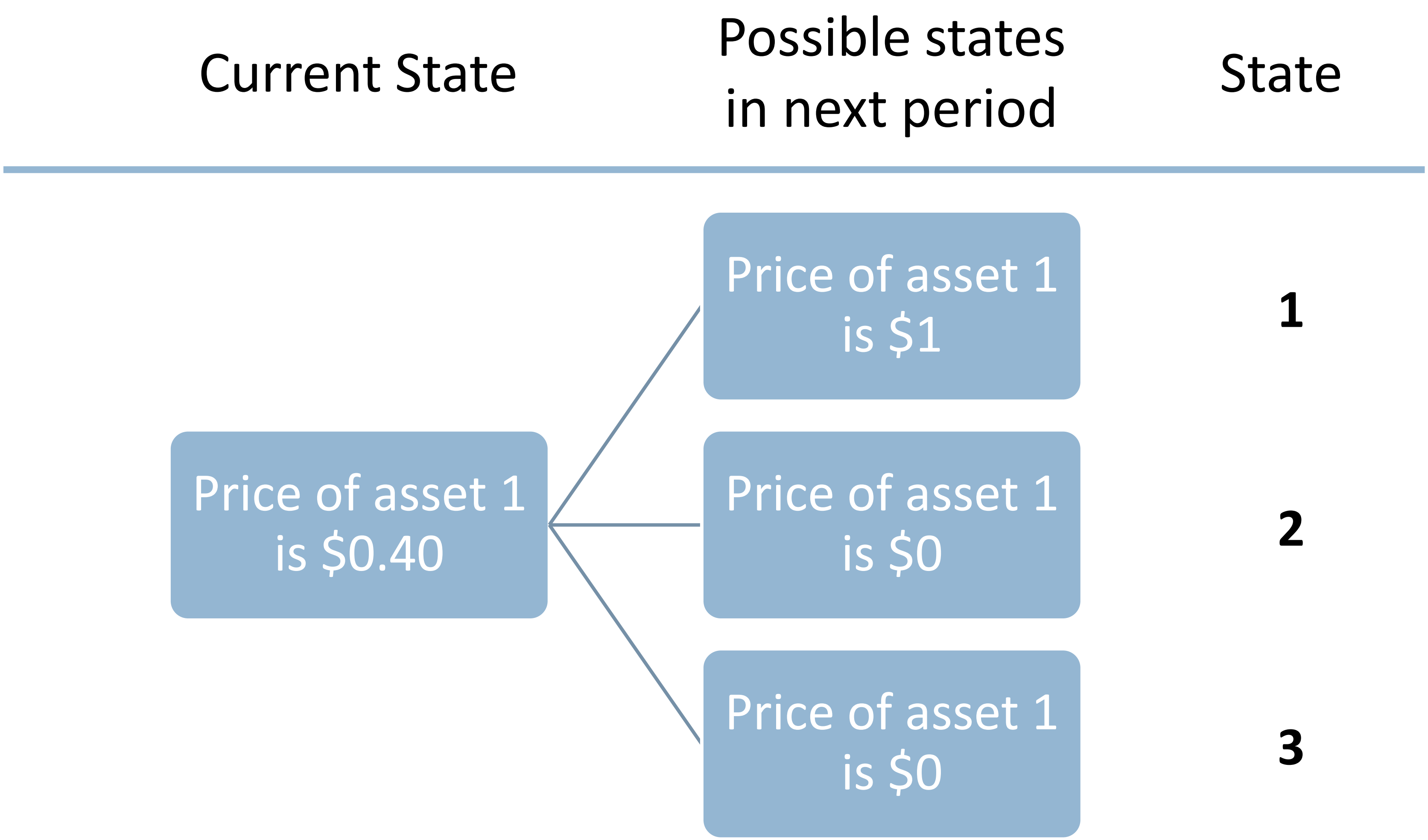
Probabilistic Approach

- Monte Carlo (MC) simulation
  1. Use random number generators to simulate asset price paths using Risk Neutral probabilities
  2. Compute final value of derivative based on the payoff of the derivative
  3. With a suitable number of simulations, compute the discounted expected value through averaging

Risk Neutral Measure

- Using Arrow securities (named after Kenneth Arrow) to understand risk neutral measure
- Start with simple case where only 3 financial assets exists in the market and interest rate is 0
- There are also only 3 states that the market can evolve into after one period
- Assume we know the current price of the assets and the value of these assets in each of these 3 states
- How do we use this information to price new assets?

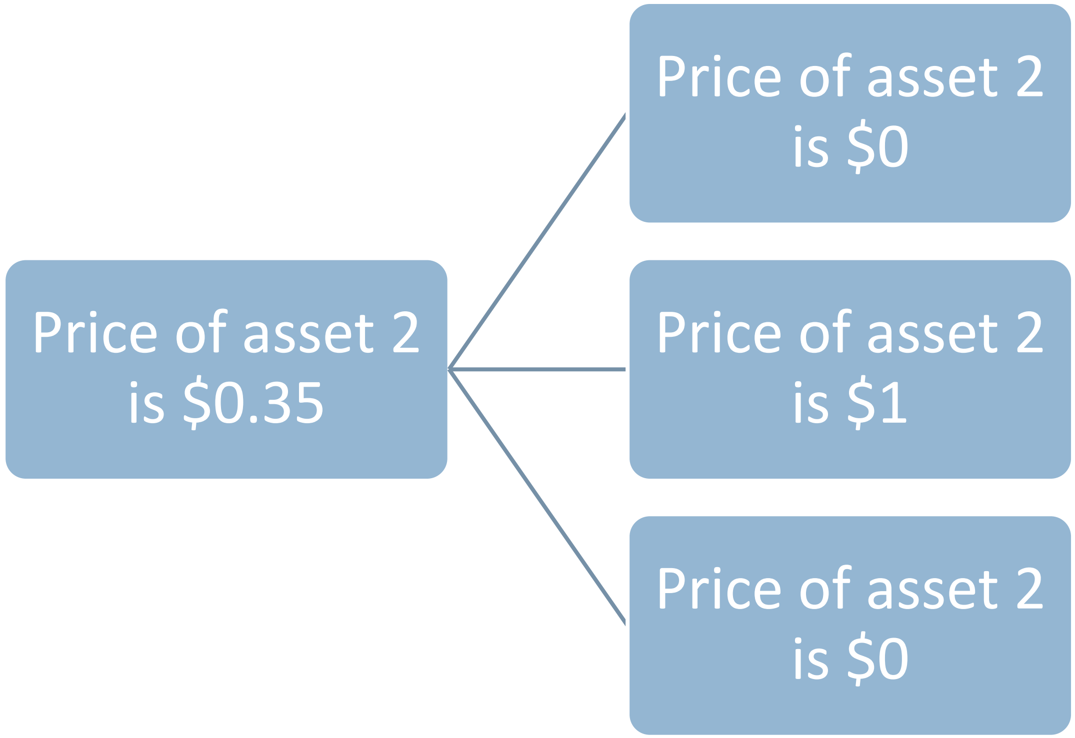
Risk Neutral Measure



Risk Neutral Measure

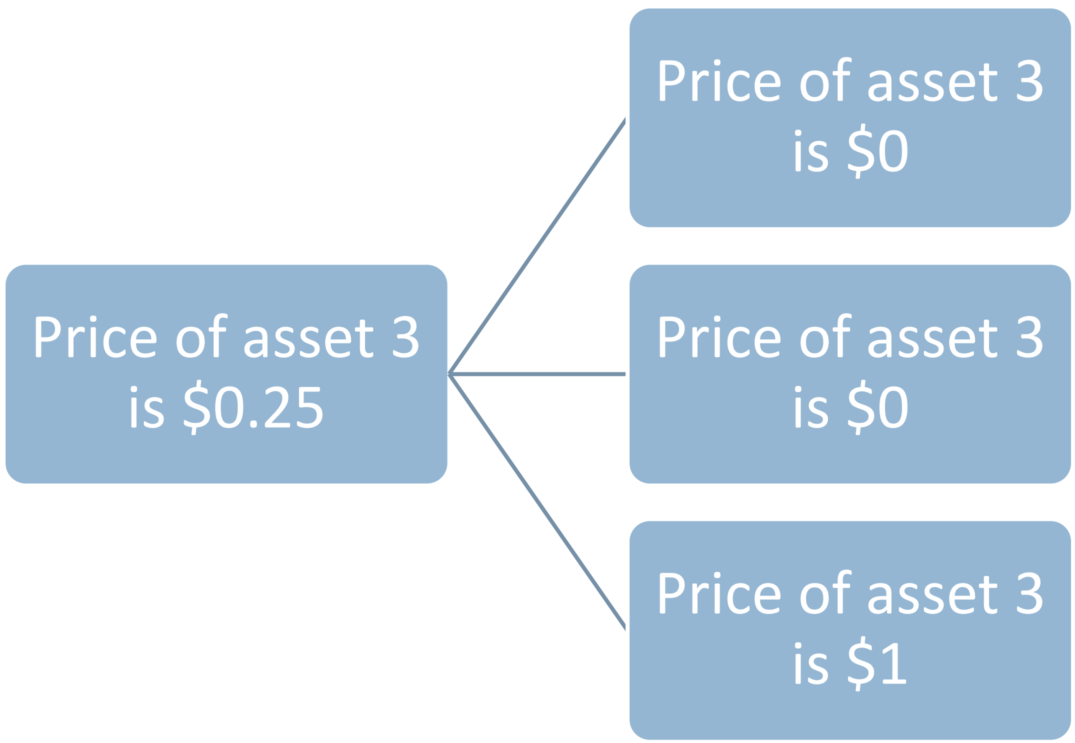
Current State

Possible states in next period



Risk Neutral Measure

Current State	Possible states in next period



Risk Neutral Measure

- Now the derivative (contingent claim) we want to price in this simplified world works like this:
  - In State 1, the price is \$2
  - In State 2, the price is \$4
  - In State 3, the price is \$10
- What should the price be in the current state?
  - \$4.50
  - \$4.70
  - \$5.00

Risk Neutral Measure

- What is a probability measure?
- Maps states (events) to a probablity
- Sums up to 1
- Additive for disjoint events i.e. P(A U B) = P(A) + P(B) if A and B are disjoint
- Check if the Arrow Securities fit this definition?

Risk Neutral Measure

- What if interest rates are not 0?
- This means the time value of money must be taken into account
- A dollar in the current state is not worth the same as a dollar in future states
- If interest rates are positive then is a dollar in the current state worth more or less than a dollar in future states?

Risk Neutral Measure

- Assume the interest rate in one period is 3%
- Then Arrow security prices for all states must add up to 1/1.03 (assuming single period compounding)
- This can be derived from the same arbitrage arguments
- Buying all Arrow securities guarantees a payout of $1 in the next period regardless of the final state
- This payout is riskless and therefore must be the same as the risk-free rate (assumptions?)

Risk Neutral Measure

- Is it still a probability measure now that it does not add up to 1?
- Can be fixed easily by adjusting the payouts in each state in the next period to \$1.03
- It is easier to price a derivative with the former as you typically know the payout at maturity
- Then you solve for the price currently by multiplying the current price of the Arrow security to the dollar payout of the derivative linked to that state

Pricing with Arrow Securities

	Price if interest rate was 0	Price if interest rate was 2%
Arrow Security 1	0.2	= 0.2 / 1.02
Arrow Security 2	0.3	= 0.3 / 1.02
Arrow Security 3	0.1	= 0.1 / 1.02
Arrow Security 4	0.2	= 0.2 / 1.02
Arrow Security 5	0.2	= 0.2 / 1.02

- Do you need to discount again when pricing?
- Use a known instrument e.g. deposit / bond
- Say the principal is \$$\frac{1}{1+r}$
- In every state of the next period, the value is \$1

Finding Solutions

- Solutions to mathematical problems can be solved analytically or numerically
- Analytical methods derive exact solutions with closed form equations
- Numerical methods involve algorithms to compute approximate solutions
- Numerical solutions are used because analytical solutions are not available
- It is a powerful technique with the computing power we now possess.

Errors with Numerical Methods

- Machine Errors
  - Precision of numerical representation
- Mathematical Errors
  - Consistency
  - Stability
  - Convergence

FDM vs MC

Criteria	FDM	MC
Dimensionality	Better in lower dimensions	Easy to go to higher dimensions
Path Dependency	Depends on the type of dependency	Generally deals well with path dependency
Computing Greeks	Excellent for computing Greeks	Not efficient
Decision Points	Excellent for implementation	Difficult to implement

Python

SINGAPORE UNIVERSITY OF
TECHNOLOGY AND DESIGN

Python

- It is increasingly being used in quantitative finance, largely thanks to its "scientific stack" of NumPy, SciPy, matplotlib, and pandas;
- Open source is naturally more commonly used versus a licensed software such as Matlab
- In the interest of time, we will make little or no use of Python features for “programming in the large,” e.g. object orientation

Python

- Anaconda will be the main platform for using Python https://www.anaconda.com/distribution/
  - IDE will be Spyder https://www.spyder-ide.org/
  - Jupyter Notebook is used for lectures, in-class exercises and assignments http://jupyter-notebook.readthedocs.io/en/latest/
  - Conda is Anaconda's built-in package manager http://conda.pydata.org/docs/

Sparse Matrices in Python

- Unavoidable for the problems we will be solving
  - Extracting rows or columns as nparrays will be possible; converting an entire sparse matrix to dense will not be
  - N.B. scipy.sparse.linalg and numpy.linalg have different functionality
- There are seven available formats; you must know them all and choose the most appropriate one(s) for what you are doing
- In general, use either the CSR or CSC format for your (linear algebra) *calculations*, and consider the others for *constructing* your initial matrix, i.e. construct in one format and then convert to another

In-Class Exercise

- Quick guide to using Markdown syntax in Jupyer Notebook
  - When in a cell, set the drop down box which is “Code” by default to “Markdown”
  - See example on basic styling syntax
  - Check Jupyter documentation for more functionalities https://jupyter-notebook.readthedocs.io/en/latest/examples/Notebook/Working%20With%20Markdown%20Cells.html

Machine Precision Consideration

- *Subtracting* numbers can give rise to large fractional errors when the numbers are similar
  - E.g. x = 1000000000000000, y = 1000000000000001.2345678901234; the computer can only represent y as 1000000000000001.2 and thus decides y-x is 1.2, which is off by almost 3%
- But subtracting similar numbers is exactly how we are attempting to approximate limits
- We want to make h small, but if we make it too small *relative to machine precision* we will get a large rounding error

In-Class Exercise

- Let's find out the machine precision by the following algorithm
  - Start with x = 1
  - Divide x by 2 until the machine sees x as equivalent to zero

In-Class Exercise

- Use cmath (complex math) functions to show that

$$(5 + i)^4(239 - i) = 2^2(13^4)(1 + i)$$

- Side note: Verifying this identity is tantamount to proving Machin's 1706 formula for π,

$$\pi/4 = 4 \cdot \arctan(1/5) - \arctan(1/239)$$

In-Class Exercise

- NumPy: Use it to find all the eigenvectors and eigenvalues of the 3x3 matrix
[[2 1 3] [1 2 3] [3 3 20]].

In-Class Exercise

- matplotlib #1:
  - Create a vector, called x, of points from 0 to 10, with a step size of 0.01
  - Create a new vector, y, that has $y(i) = x(i)*e^{x(i)}$
  - Plot x versus y

In-Class Exercise

- matplotlib #2:
  - Create a vector of 50000 random samples from a lognormal distribution whose underlying normal distribution has $\mu=0$ and $\sigma=1$.
  - Plot a histogram of it using 500 bins.

Discretizing Derivatives

SINGAPORE UNIVERSITY OF TECHNOLOGY AND DESIGN

PDE Approach

- Finite Difference Methods (FDM)
  1. Discretize continuous variables
  2. Start from known quantities of the variables (boundary conditions)
  3. Differential equations provide a map of how variables change with respect to each other (e.g. time, price of underlying , value of derivative)
  4. Evolve the known quantity in the direction required using the differential equations

First Derivative

- Recall the original definition of a derivative,

$$f'(x) = \lim_{h \to 0} \frac{f(x+h) - f(x)}{h}$$

- Mathematical analysis can compute this exactly e.g. $f(x) = x^2$ then $f'(x) = 2x$
- Computers deal with the discrete and not a continuum of numbers
- Discretize the derivative using small values of h

First Derivative

- If we know f at two neighboring points, we can approximate the derivative near them both:

$$f'(x) \approx \frac{f(x+h) - f(x)}{h}$$

- This is called *FORWARD* DIFFERENCE approximation
- The values $f(x)$ and $f(x+h)$ must be computed
- For smaller values of h this approximation should be increasingly accurate (if the function is "nice")

In Class Exercise

- Let $f(x) = x^3$
- Estimate the derivative at $x = 1$ for
  h = 0.1, 0.01, and 0.001
- We know the actual derivative is $3x^2$; how does the approximation error vary with h?

Approximation Error

- We find the error goes down in proportion to h
- We also could have approximated the derivative as

$$f'(x) \approx \frac{f(x) - f(x-h)}{h}$$

- This is a *BACKWARD* DIFFERENCE approximation
- Find the error of f' with respect to h using this second estimate

Approximation Error

- We can obtain a third approximation by averaging the two previous approximations:

$$f'(x) \approx \frac{1}{2}\frac{f(x+h)-f(x)}{h} + \frac{1}{2}\frac{f(x)-f(x-h)}{h}$$

$$f'(x) \approx \frac{f(x+h)-f(x-h)}{2h}$$

- This is called the *CENTRAL* DIFFERENCE approximation to the derivative

Approximation Error

- The error in the approximation can be analyzed using Taylor’s theorem (recall from Calculus)
- Get the Taylor’s series of terms in the numerator of the approximation and use it to find the series for the approximation

$$f'(x+h) = f(x) + hf'(x) + \tfrac{1}{2}h^2 f''(x) + \tfrac{1}{6}h^3 f'''(x) + \cdots$$

$$f'(x-h) = f(x) - hf'(x) + \tfrac{1}{2}h^2 f''(x) - \tfrac{1}{6}h^3 f'''(x) + \cdots$$

$$\frac{f'(x+h) - f(x-h)}{2h} = f'(x) + \tfrac{1}{6}h^2 f'''(x) + \cdots$$

- The first term is what we are approximating and the subsequent terms in the series is the error of the approximation

In-Class Exercise

- Use $f(x) = x^3$ with the central difference approximation to calculate the error at x = 1 for h = 0.1, 0.01, and 0.001
- How does the error depend on h?
- Now try $f(x) = x^2$ with both the forward and central difference approximations
- What happens to the error for the latter?

Order of the Approximation

- In general we have that

$$\frac{f(x+h) - f(x-h)}{2h} = f'^{(x)} + \tfrac{1}{6}h^2 f'''(x) + \cdots$$

- Therefore as h get smaller the error goes down in proportion to h^2
- We call this SECOND ORDER ACCURACY
- $f'''(x)$ and higher order terms equals zero for a quadratic equation hence error is zero in the example of $f(x) = x^2$

Taylor’s Theorem

- We can also use Taylor’s Theorem to find the different coefficients that gives the right approximation

- For example, we can use Taylor’s Theorem to show that the approximation below is also correct

$$f'(x) \approx \frac{-3f(x) + 4f(x+h) - f(x+2h)}{2h}$$

Taylor's Theorem

- Expanding the terms in the numerator

$$-3f(x) + 4f(x+h) - f(x+2h)$$

$$= -3f(x) + 4f(x) + 4hf'(x) + 2h^2 f''(x) - f(x) - 2hf'(x) - 2h^2 f''(x) + O(h^3)$$

$$= 2hf'(x) + O(h^3)$$

- Therefore, $\frac{-3f(x)+4f(x+h)-f(x+2h)}{2h} = f'(x) + O(h^2)$

- This is a higher order *FORWARD* DIFFERENCE approximation

In-Class Exercise

- Generate a vector x of 101 equally spaced points between -2 and 2
- Let $f(x) = x \exp(-x^2)$ in a vector y.
- Create a new vector z which approximates $f'(x)$ at all but the end points, using central differencing
- Plot the difference between the actual derivative and this approximation
- Repeat using 501 equally spaced points between -2 and 2

Method of Undetermined Coefficients

- The METHOD OF UNDETERMINED COEFFICIENTS is used to find different approximations to derivatives
- Let's say we want to approximate the derivative using $f(x)$, $f(x-h)$ and $f(x-2h)$
- We want to find a, b, and c such that

$$f'(x) = af(x) + bf(x-h) + cf(x-2h) + error(h)$$

Method of Undetermined Coefficients

- We want
  - $f'(x) = af(x) + bf(x-h) + cf(x-2h) + \ldots$
- There are 3 unknowns (a,b,c) so we need 3 equations to solve for them
- Expand $f(x-h)$ & $f(x-2h)$ with Taylor's series
- Then set the coefficients of the key terms as follows:
  - The coefficient for the $f(x)$ term must equate to 0
  - The coefficient for the $f'(x)$ term must equate to 1
  - The coefficient for the $f''(x)$ term must equate to 0

Second Derivative

- For the Black Scholes PDE, we will also need *second* derivatives:

$$f''(x) \approx \frac{f(x+h) - 2f(x) + f(x-h)}{h^2}$$

- Try verifying this using Taylor Series on your own

In-Class Exercise

- For $f(x) = x^3$, apply this approximation and compare the result to the actual second derivative

Using Matrices

- All these derivative approximations are just linear combinations of the function evaluated at different points
- If we have a function represented by a vector, to take the derivative we just multiply some points by some numbers and add them up
- This is the exact definition of matrix multiplication

Using Matrices

- This would be a matrix representation of discretizing the derivatives across a set of points along x
- Multiply the matrix out for each row to verify

$$\frac{1}{2h}\begin{bmatrix} -3 & 4 & -1 & 0 & 0 & 0 \\ -1 & 0 & 1 & 0 & 0 & 0 \\ 0 & -1 & 0 & 1 & 0 & 0 \\ 0 & 0 & -1 & 0 & 1 & 0 \\ 0 & 0 & 0 & -1 & 0 & 1 \\ 0 & 0 & 0 & 1 & -4 & 3 \end{bmatrix}\begin{bmatrix} f(x_1) \\ f(x_2) \\ f(x_3) \\ f(x_4) \\ f(x_5) \\ f(x_6) \end{bmatrix} = \begin{bmatrix} f'(x_1) \\ f'(x_2) \\ f'(x_3) \\ f'(x_4) \\ f'(x_5) \\ f'(x_6) \end{bmatrix}$$

- The x_n symbols are a shortcut e.g. $x_1 + h = x_2$ and so on

Using Matrices

In the matrix on the previous slide,

- the top row encodes a <u>forward</u> difference, and indeed it must
- the bottom row encodes a <u>backward</u> difference, and again it must
- the other intermediate rows can, and do, encode <u>central</u> differences

In-Class Exercise

- Create a vector of x values from -2 to 2 with a step size of 0.01, i.e. 401 values over 400 steps
- Create a vector of $y = x\,e^{-x^2}$
- Create a sparse matrix A just like the one three slides back using the sparse package in Scipy
  - https://docs.scipy.org/doc/scipy/reference/sparse.html
- Multiply A by y using the @ operator for matrix multiplication
- Plot the resulting estimated derivative vs. x
- Compare it to the actual derivative