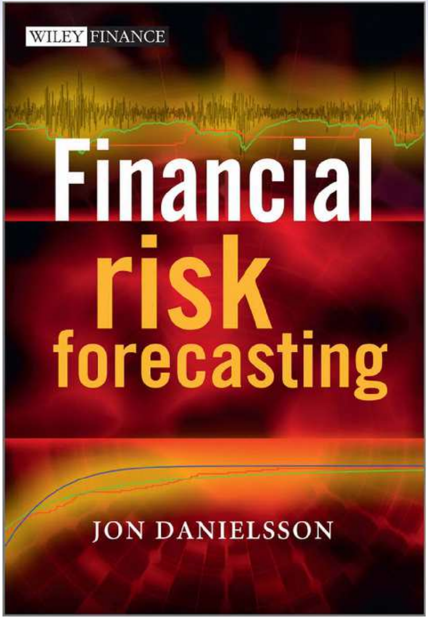
Financial Risk Forecasting
Chapter 7
Simulation Methods For VaR For Options And Bonds

Jon Danielsson ©2025
London School of Economics

To accompany
Financial Risk Forecasting
FinancialRiskForecasting.com

Published by Wiley 2011

Version 10.0, August 20254


WILEY FINANCE
Financial
risk
forecasting
JON DANIELSSON

Simulation Methods For VaR For Options And Bonds

Simulations in Practice

- Risk Management:
  - Monte Carlo simulation is used to estimate VaR and ES for complex portfolios
  - Captures nonlinearities and fat tails better than analytical methods
- Stress Testing:
  - Simulate extreme but plausible market scenarios
  - Assess impact on capital, liquidity, and solvency
- Regulatory Compliance:
  - Banks use simulation-based VaR for internal models under regulations
  - Backtesting frameworks rely on simulated distributions
- Derivative Pricing:
  - Path-dependent or exotic options often require Monte Carlo methods
  - Simulation integrates risk-neutral pricing with market-implied volatilities

Simulation Methods: Pros and Cons

Advantages

- Handles complex, nonlinear payoffs (e.g. options, credit derivatives)
- Easily incorporates fat tails, jumps, or other non-normal features
- Scales to large portfolios with mixed instruments
- Flexible for scenario and stress testing

Disadvantages

- Computationally intensive, especially with many paths/assets
- Sensitive to assumptions (e.g. distribution, model parameters)
- Requires careful random number control and convergence checking
- Harder to explain and audit than closed-form models

Why Simulation?

- Analytical methods (duration, delta) are limited:
  - Linear approximations
  - Often assume normality and small price changes
- Many real-world instruments (e.g. options) are highly non-linear
- Simulation allows us to:
  - Model realistic price dynamics
  - Capture full portfolio behaviour
  - Flexibly estimate risk measures like VaR

The Focus of This Chapter

- Chapter 6 demonstrates the limitations of analytical VaR methods for options and bonds
- The focus of this chapter is on simulation methods, sometimes called *Monte Carlo (MC) simulation*
  1. Pseudo random number generators (RNGs)
  2. Simulation pricing of options and bonds
  3. Simulation VaR for one asset and portfolios
  4. Issues in Monte Carlo estimation

The Simulation Idea

- Use a computer to replicate many possible future market outcomes
- Based on a model of price dynamics (e.g. normal returns)
- Run a large number of simulations to create a sample distribution
- Use this to estimate prices, payoffs, or risk measures like VaR

Calendar Time and Trading Time

- We use two different measures of time

  Calendar time (365/6 days) used for interest rate calculations

  Trading time ($\approx$ 250 days) used for risk calculations
- This is because we earn interest every day
- But calculate volatilities only from days (trading days) when stock exchanges open (Mondays to Fridays, excluding holidays)
- R will allow precise date calculations and has a database of dates when various exchanges are open
- Not needed here, but can be important in applications

Notation new to this Chapter

F	Futures price
s	Index for simulation, like the s^{th} simulated price
μ_a	Annual mean of returns
u	Uniformly distributed random number
S	Number of simulations
x^b	How many basic assets are held
x^o	How many options are held

Learning outcomes

1. Understand the basic issues in Monte Carlo simulation
2. Know how to price a bond with simulations
3. Know how to price an option with simulations
4. Be able to obtain risk forecasts with simulations based on a single underlying asset
5. Be able to obtain risk forecasts with simulations based on a portfolio of underlying assets
6. Know the basic strengths and weaknesses of simulation methods
7. Recognise the importance of setting the simulation size

Why Use Simulation for VaR?

- Analytical VaR methods often rely on linearity, normality, and small moves
- But real-world portfolios include non-linear instruments like bonds and options
- Simulation allows us to:
  - Capture non-linear payoffs
  - Use flexible return distributions
  - Model full portfolio dynamics

Random Numbers and Monte Carlo Simulations

Are Natural Phenomena Truly Random?

- Many physical systems appear random — radioactive decay, thermal noise, coin tosses
- But most are governed by deterministic laws (e.g. Newtonian mechanics, quantum probabilities)
- What we observe as randomness often reflects:
  - Limited precision in measurement
  - Sensitivity to initial conditions
  - Incomplete information about the system
- True mathematical randomness — sequences with no pattern — is extremely rare in nature
- That is why computers use *pseudo-random numbers*: deterministic sequences designed to look random

Why Random Numbers Matter

- Monte Carlo simulation depends on high-quality random numbers
- From a random number generator (RNG)
- Risk forecasts, pricing, and model testing rely on simulated randomness
- Poor RNGs lead to biased results, incorrect VaR, and flawed pricing
- We need to understand how RNGs work and how to check their quality

The Problem of Randomness

- The fundamental input in Monte Carlo (MC) analysis is a long sequence of random numbers (*RNs*)
- Creating a large sample of *high-quality* RNs is difficult
- It is impossible to obtain pure random numbers
  - There is no natural phenomena that is purely random
  - Computers are deterministic by definition
- Computer algorithm known as *pseudo random number generator* (RNG), creates outcomes that *appear* to be random even if they are deterministic

Pseudo Random Number Generators

- A particular of RN is generated by a function of a previous RN

$$u_{i+1} = h(u_i)$$

  where u_i is the i^{th} RN and $h(\cdot)$ is the RNG
- If RNs are truly random, it is essential that their unconditional distribution is IID *uniform*

Period of a Random Number Generator

Definition: Random number generators can only provide a fixed number of different random numbers, after which they repeat themselves. This fixed number is called a *period*.

- Symptoms of low-quality RNGs include:
  - Low period (RNG repeats itself quickly)
  - Serial dependence
  - Deviations from uniform distribution

A Classic RNG: Linear Congruential Generators

- No numerical algorithm generates truly random numbers
- The best-known RNGs are so-called *linear congruential generators* (LCGs), which link the i^{th} and $i+1^{\text{th}}$ integers in the sequence of RNs by

$$u_{i+1} = (a \times u_i + b) \bmod m$$

  where a is multiplier; b is increment; m is integer modulus and mod is the modulus function (remainder after division)
- The first RN in a sequence is called *seed* and is usually chosen by the user

Illustration: RNGs, Seed, Size and Period

- Think of the *RNG* as an ellipse where each point represents a RN and the number of RNs is finite
- The *seed* determines the starting point of the sequence of RNs and is set by the user
- Think of the sequence generated as a specific arc of the ellipse which depends on the chosen seed
- The *size* of the simulation determines the length of the sequence
- The *period* of the simulation is the number of RNs that the RNG is able to generate without repeating itself

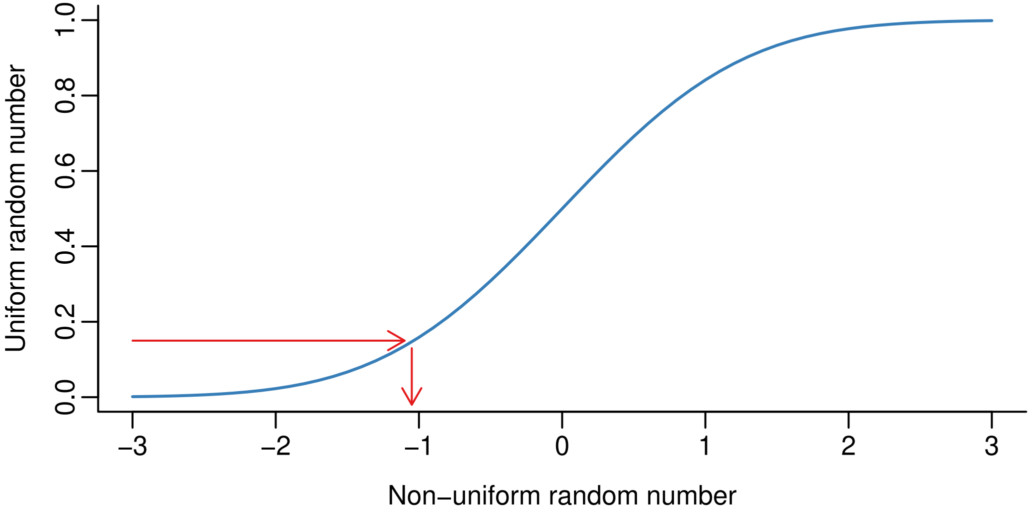
What we use

- Main flaw of LCGs is serial correlation, which cannot be easily eliminated
- More complicated RNGs which introduce non-linearities are generally preferred
- The default RNG in R is the *Mersenne twister*, which has a period of $2^{19,937} - 1 \approx 10^{6,002}$ (a Mersenne prime number)

Generating From Other Distributions

- Most RNGs generate uniform random numbers, usually scaled so that $(u) \in [0, 1]$
- However, most practical applications require RNs from a different distribution
- To obtain such RNs, we use *transformation methods* which convert uniform numbers into RNs from the distribution of interest
  - The inverse distribution is obvious candidate, but often slow and inaccurate

Normal Inverse Distribution



Why RNGs Matter

- Low-quality RNGs lead to biased or repeating simulations
- Bad transformations distort tail behaviour — critical for VaR
- Always use high-period, tested generators (e.g. Mersenne Twister)
- In R: default RNGs are robust — but always check sensitivity

Weak Randomness and Hacking

- Encryption relies on randomness to protect secrets:
  - Keys, nonces, and initialisation vectors
- If the random numbers are not truly random:
  - Hackers can guess or reconstruct your private keys
  - Encrypted data becomes easy to decrypt
  - Digital signatures can be forged
- Real-world attacks have exploited bad RNGs to steal data, forge identities, and compromise systems
- A weak RNG turns secure encryption into a locked door with the key taped to it

When Random Isn't Random

- OpenSSL Bug
  - A code cleanup disabled entropy gathering in OpenSSL's RNG
  - Only 32,768 possible keys — easily brute-forced
  - Affected SSH, SSL, and GPG keys for over two years
- Dual_EC_DRBG Backdoor (NIST, NSA)
  - NIST-standardised RNG with a suspected NSA backdoor
  - RNG appeared secure but allowed prediction of future values
  - Used in commercial cryptographic products before being withdrawn

When Randomness Goes Wrong — Real-World Consequences

- Therac-25 (1985–1987) — medical radiation machine
  - Poor timing logic and pseudo-random behaviour led to race conditions
  - Administered massive overdoses of radiation; at least six deaths
- Patriot missile failure (1991) — Gulf War
  - Time drift from rounding errors caused simulation-based tracking to lose accuracy
  - The missile system simulated target trajectories but assumed exact timing
  - A Scud missile hit a US barracks in Dhahran; 28 fatalities
- Volkswagen emissions scandal (2015)
  - Engine software detected test conditions by monitoring steering, speed, and timing patterns
  - When it detected a lab test, it simulated cleaner engine behaviour
  - In real driving, emissions were far higher — up to 40$\times$ above legal limits

Quantum Computing and the Future of Simulations

- Quantum computers use qubits instead of classical bits — allowing them to process vast numbers of states in parallel
- This could make many current encryption schemes obsolete
- Problems that would take classical computers millions of years could be solved in hours
- Most public-key encryption (like RSA and ECC) is especially vulnerable
- At the same time, quantum mechanics provides a new source of true randomness — useful for secure key generation
- The future of cryptography will depend on both defending against quantum attacks and using quantum tools for better security
- Many governments copy encrypted data now with a view to de-crypt it once quantum computing becomes practical

Random Numbers in R

```
runif(1)   0.704862
runif(1)   0.2267493
runif(1)   0.9921351
set.seed(999); runif(1)   0.3890714
set.seed(999); runif(1)   0.3890714
x=rnorm(n=5)
-0.2817402 -1.3125596  0.7951840  0.2700705 -0.2773064
x=rt(n=5,df=3)
-0.34910572 -0.34198996  0.39319432  0.07968276  0.46555150
plot(rnorm(n=1000),type='l')
plot(rt(n=1000,df=2),type='l')
```

Simulation Pricing of Bonds

Bond Pricing

- Price and risk of fixed income assets (for example, bonds) is based on market interest rates
- Using a model of the distribution of interest rates, we can simulate random yield curves and obtain the distribution of bond prices
- We map distribution of interest rates to the distribution of bond prices

Analytical Bond Pricing

- Denote ι_n as the annual interest rate in year n
- The present value of a bond is the present discounted value of its cash flows, c_n:

$$p = \sum_{n=1}^{N} \frac{c_n}{(1+\iota_n)^n}$$

Analytical Bond Pricing

- Suppose we have a bond with 10 years to expiration, 7% annual interest, \$10 par value, (that is, \$0.70 coupon) and current market rates

$$\{\iota_n\}_{n=1}^{10} = (5.00, 5.69, 6.09, 6.38, 6.61, 6.79, 7.07, 7.19, 7.30) \times 0.01$$

- The bond has a current value of \$9.91

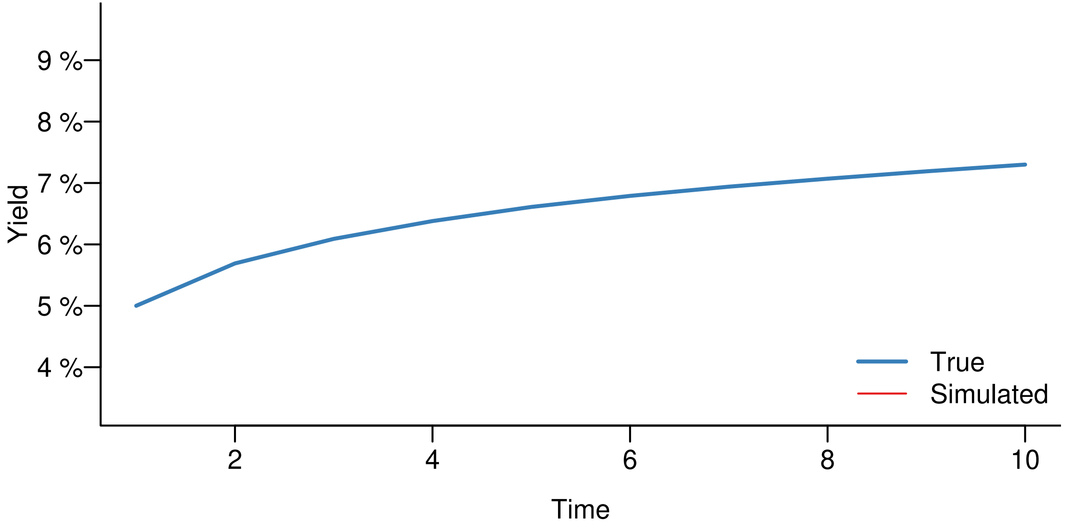
Simulated Bond Pricing

- Assume here that the yield curve can only shift up and down and will not change shape
- Shocks to interest rate (changes), ϵ_s

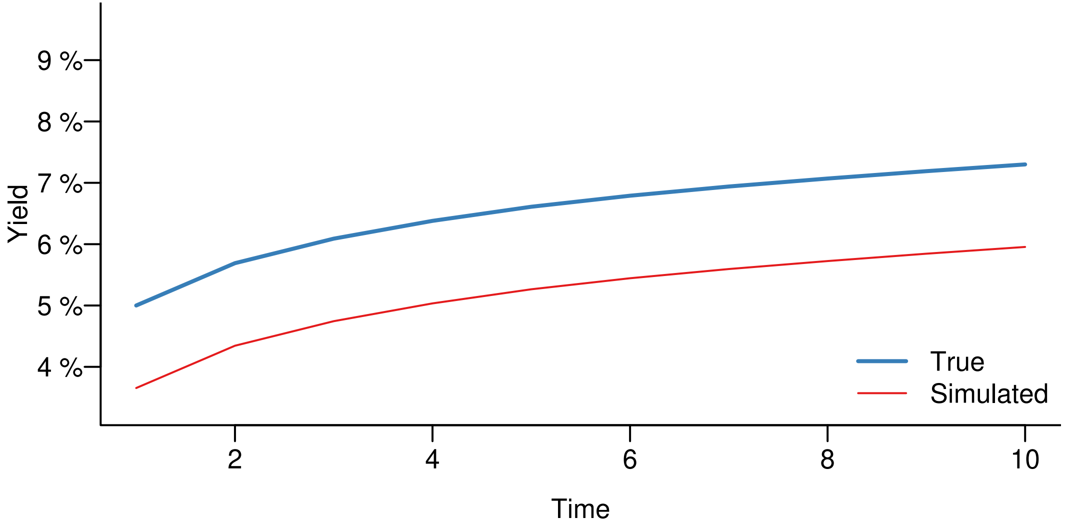
$$\epsilon_s \sim \mathcal{N}\left(0, \sigma_\iota^2\right)$$

- For the sake of demonstration, we set the number of simulations as $S = 8$, but note that accurate estimates require much more simulations

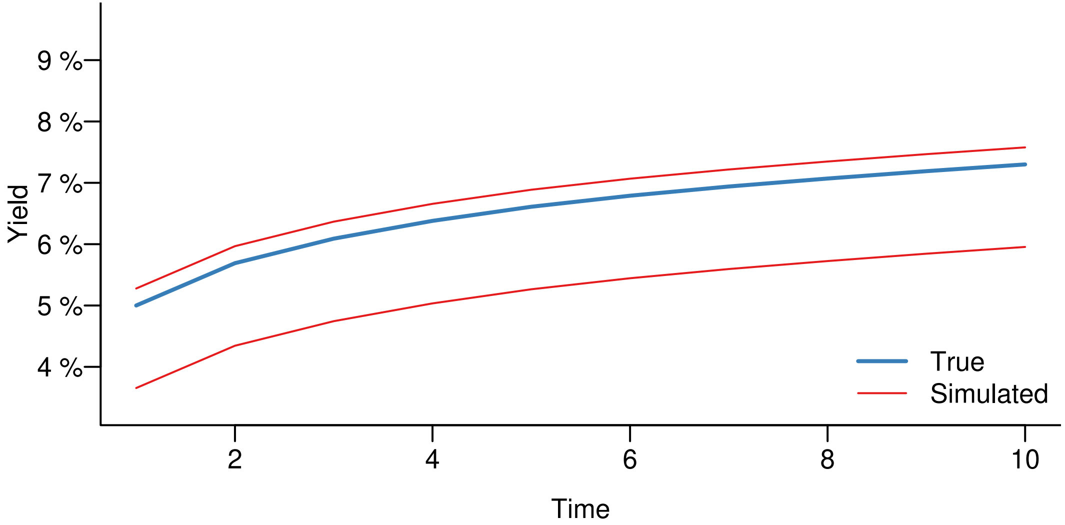
Eight Yield Curve Simulations



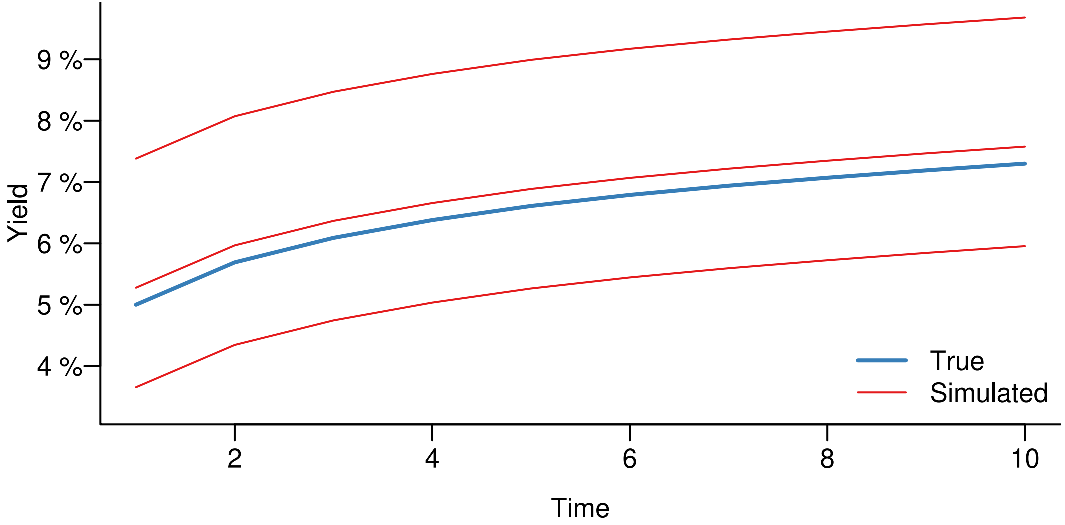
Eight Yield Curve Simulations



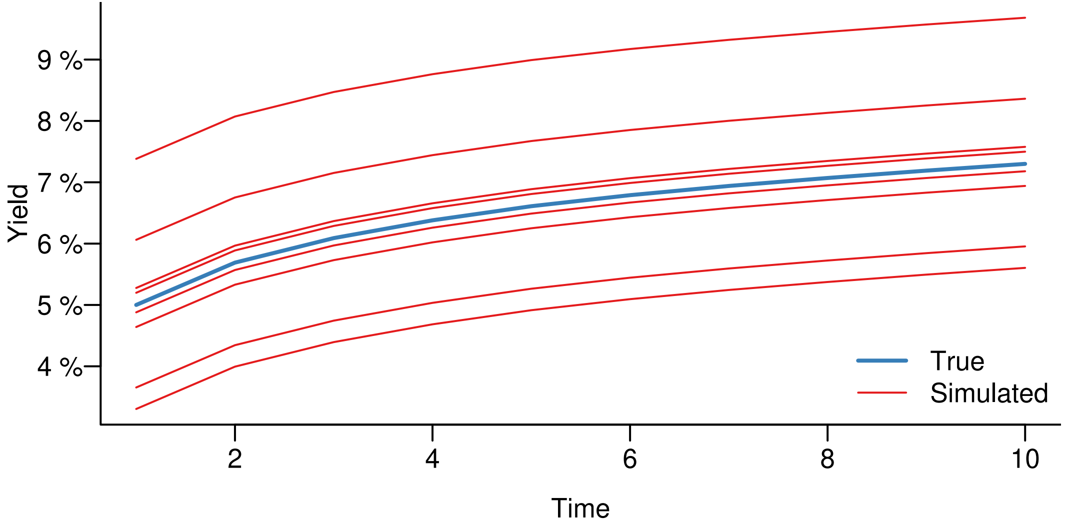
Eight Yield Curve Simulations



Eight Yield Curve Simulations



Eight Yield Curve Simulations



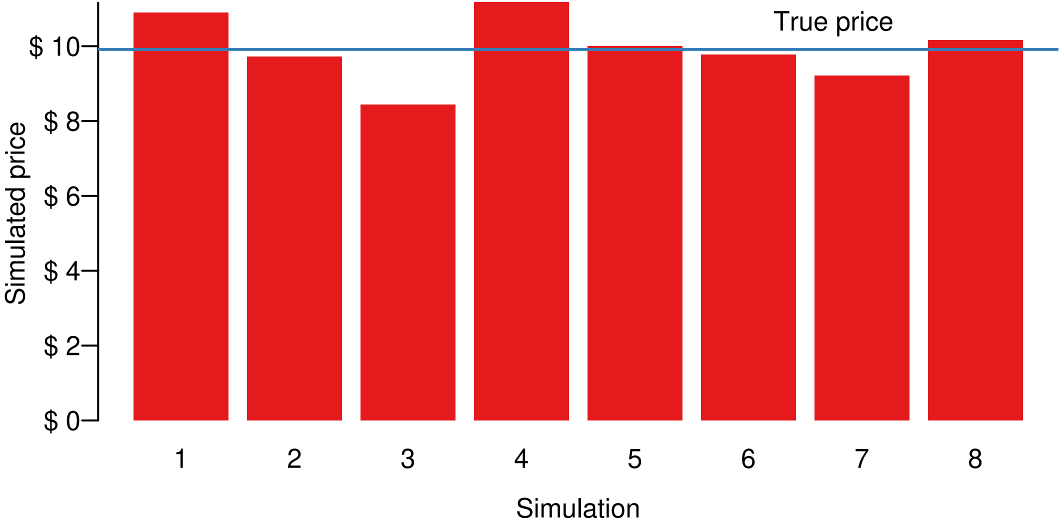
Simulated Bond Pricing

- The equation for the s^{th} simulated price, p_s, now becomes

$$p_s = \sum_{n=1}^{N} \frac{c_n}{(1 + r_n + \epsilon_s)^n}$$

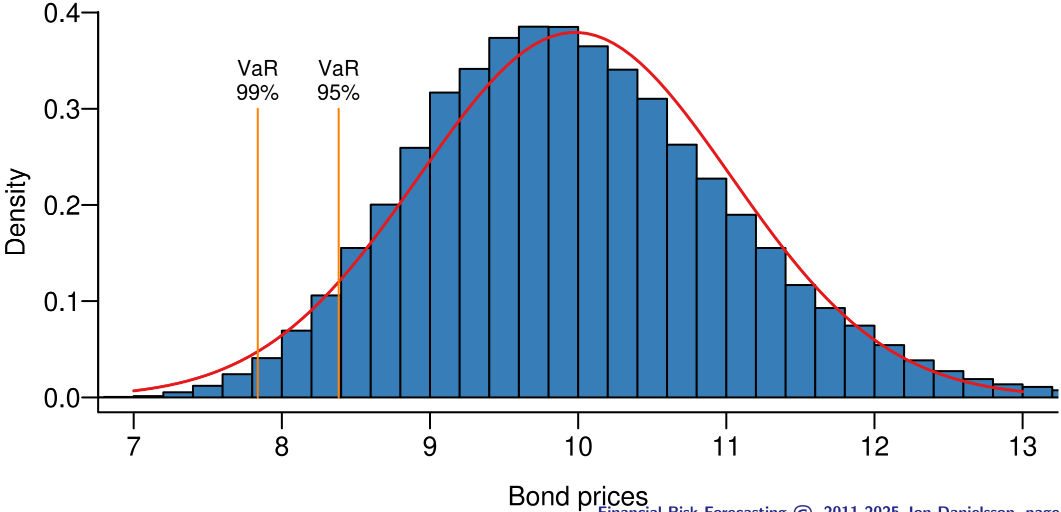
  where $r_n + \epsilon_s$ is the s^{th} simulated interest rate at year n
- Compare the eight bond prices obtained with $S = 8$ yield curve simulations with the distribution of bond prices when $S = 50,000$

Eight Bond Price Simulations



Density of Simulated Bond Prices: Normal Superimposed.

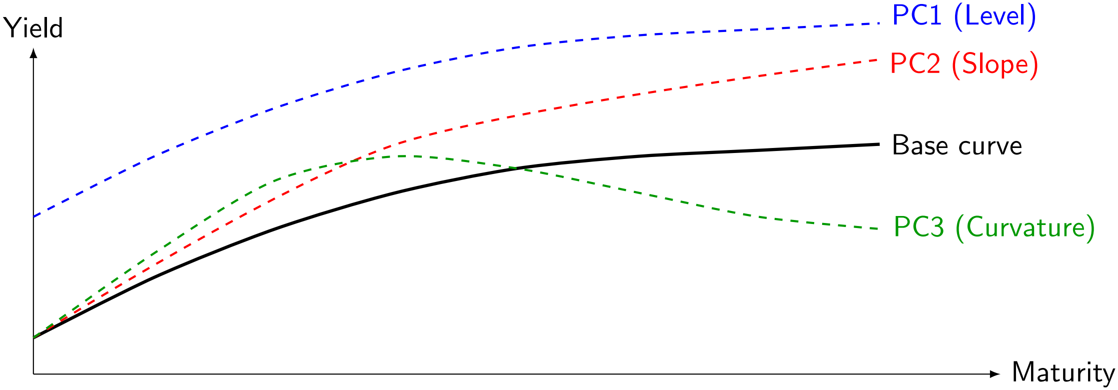
$$S = 50,000$$



Allowing Yield Curve to Change Shape

- Key assumptions:
  - Yield curve only shifts up and down
  - Distribution of interest rate changes is normal
- The assumptions may be unrealistic in practice but it is relatively straightforward to relax them
  - In practice it rotates and twists
  - Can use principal components (PCA)
- See next slide

How Principal Components Affect the Yield Curve



So

- Simulation allows yield curve shocks to propagate into bond price uncertainty
- Even simple assumptions yield realistic VaR bands
- Yield curve modelling matters — assumptions like shifts vs twists are key
- Higher simulation counts yield smoother, more reliable distributions

Simulation Pricing of Options

Simulation Approach

- The price of a derivative must be the expectation of its final pay-off under *risk-neutrality*
- Depends on the price movements of its underlying asset
- If sufficient number of price paths are simulated, we obtain an estimate of the true price

Option Pricing

- Get price of European options on non-dividend-paying stock where all Black-Scholes (BS) assumptions hold
- Two primitive assets in BS pricing model:
  - *Risk-free asset* with instantaneous and constant annual rate ι
  - Underlying stock, follows normally distributed random walk with drift ι (geometric Brownian motion in continuous time)
- The no-arbitrage *futures* price of stock for delivery at time Υ, is given by:

$$F_{\Upsilon} = p_{\tau} \times e^{\iota(\Upsilon - \tau)}$$

- Where $\Upsilon - \tau$ in number of years from now (τ) until Υ

Analytical Option Pricing

- Suppose we have a European call option with
  1. Current stock price $p_\tau = \$50$
  2. 20% annual volatility, σ_a
  3. 5% annual risk-free rate, ι
  4. Six months to expiration, $\Upsilon - \tau$
  5. $X = \$40$ strike price
- The price is \$11.0873

Simulated Option Pricing

- We simulate returns until expiration and use these values to calculate simulated futures prices
- With sufficient sample of futures prices, we can compute the set of pay-offs of the option
- The Monte Carlo option price is then given by the PV of the mean of these pay-offs

Simulating a Stock Price Using Continuous Time

- The simplest model of a stock price in continuous time is geometric Brownian motion
- Then, daily discrete returns are

$$y_\tau = \log p_\tau - \log p_{\tau-1/365}$$

- Note, this is the same as writing

$$y_t = \log p_t - \log p_{t-1}$$

- Where t is a day

Simulated Option Pricing

- The main complexity is due to expectation of a log-normal random number, that is, if
$$\epsilon \sim \mathcal{N}\left(\mu, \sigma^2\right)$$
- Then: the expected $\exp(\epsilon)$ is not $\exp(\mu)$ but
$$\mathrm{E}\left[\exp(\epsilon)\right] = \exp\left(\mu + \frac{1}{2}\sigma^2\right)$$
- We apply a *log-normal correction* (subtract $\frac{1}{2}\sigma^2$ from simulated stock return) to ensure that expectation of simulated returns is the same as theoretical value
- See density of $S = 10^6$ futures prices and option pay-offs using same values as in the example before

Therefore

- The simulated returns are

$$\epsilon_s \sim \mathcal{N}\left(0.5 \times r, 0.5 \times \sigma_a^2\right)$$

- Then the s^{th} simulation of the futures price at Υ is

$$F_{\Upsilon,s} = p_{\mathcal{T}} \times e^{\left(\epsilon_s - \frac{1}{2}0.5 \times \sigma_a^2\right)}$$

- And when strike is X, the simulated present value of the pay-off (for a call)

$$\max(F_{\Upsilon,s} - X, 0) \times e^{-0.5r}$$

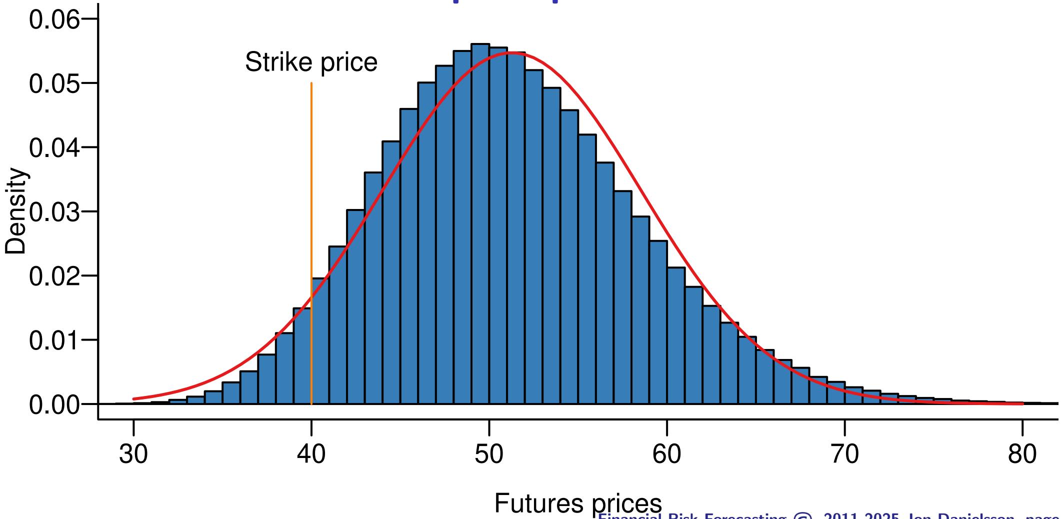
- And for put

$$\max(X - F_{\Upsilon,s}, 0) \times e^{-0.5r}$$

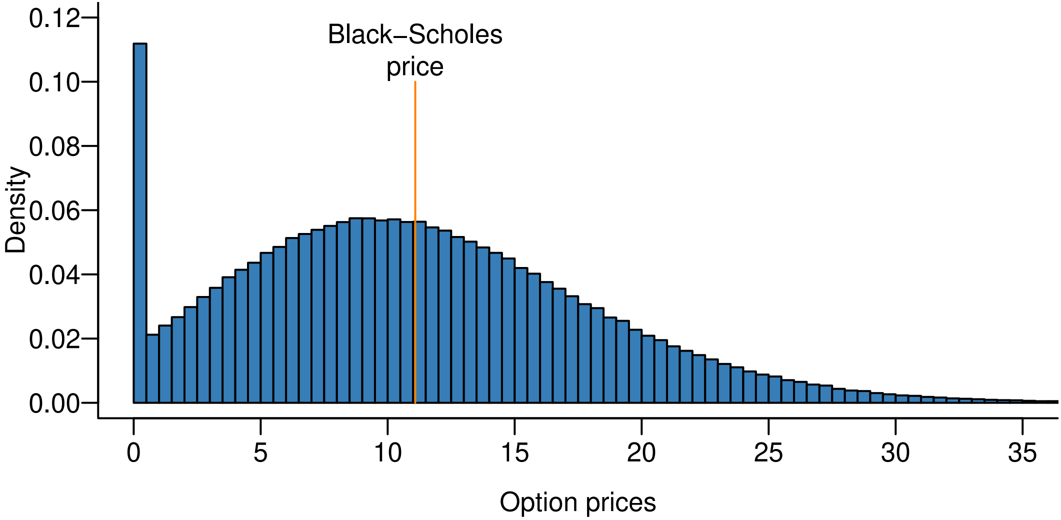
Visualising Simulated Results

- We now show the density of simulated futures prices and option payoffs
- These distributions are shaped by the log-normal nature of the underlying
- Option prices are always positive and skewed
- We compare simulated densities to the Black-Scholes benchmark

Density of Simulated Futures Prices: $S = 10^6$, normal superimposed



Density of Simulated Option Prices: $S = 10^6$



Simulated Option Pricing: Numerical Example

- Mean of simulated option prices is the MC price
  - In this case, R gives \$11.08709, close enough to the Black-Scholes price of \$11.0873
- Note the asymmetry in the density of simulated futures prices (result of log-normal distribution of prices)
- The VaR can be read off the previous graph, for example, 1% smallest value of distribution gives 99% VaR

Why Simulate If We Know the Answer?

- Simulation gives the same price as Black-Scholes in this simple case
- But it remains useful when:
  - No closed-form solution exists
  - Payoffs are path-dependent (e.g. Asian options)
  - Models include stochastic volatility or jumps
- Monte Carlo methods scale to realistic portfolios

Simulation Estimation of Option VaR

Monte Carlo VaR Workflow

1. Simulate future asset prices
2. Revalue portfolio under each scenario
3. Compute profit/loss for each scenario
4. Construct the empirical distribution of outcomes
5. Read off quantiles to obtain VaR

Notes

- We assume there are 365 calendar days in a year
- And 250 business days
- In what follows we use the notation for a day, t, to mean business days and use Υ and τ for calendar time in years
- Both t and τ refer to today
- We follow the Black-Sholes assumptions of IID normal returns

Simulating a Stock Price Using Arithmetic Returns

- Use arithmetic returns

$$r_{t+1} = \frac{p_{t+1}}{p_t} - 1$$

- That is,

$$p_{t+1} = p_t(r_{t+1} + 1)$$

- Note, no log-normal correction as p_{t+1} is normal if r_{t+1} is normal
- Also this is *not* the futures price $F_\Upsilon = p_\tau e^{\iota(\Upsilon - \tau)}$
- We are simulating spot prices using returns — not futures prices from risk-neutral expectations
- Futures pricing involves discounting under the risk-neutral measure, which we are not doing here

Simulation of VaR for One Asset

- Simulate the one-day return of an asset
- Apply analytical pricing formulas to simulated one-day-ahead price
- Obtain simulated profit or loss (q_s) as difference between tomorrow's simulated one-day-ahead values and today's known value
- Calculate Monte Carlo MC VaR from simulated profit or loss using the same approach as in historical simulation

Setup

- Consider an asset with price p_t and IID normal returns, with *one-day* volatility σ and *annual* risk-free rate r
- Number of units of basic assets held in a portfolio is denoted by x^b, while x^o indicates number of options held
- Note we simulate the $t+1$ price

We Will Go Through a Series of Ever More Complicated Examples

1. Simulation of VaR for one asset (no option)
2. Simulation of VaR for one option
3. Simulation of VaR for a portfolio of one option and one stock

1. VaR With One Basic Asset

Six-step procedure for obtaining MC VaR

1. Compute initial portfolio value:

$$\vartheta_t = x^b p_t$$

1. VaR With One Basic Asset

Six-step procedure for obtaining MC VaR

1. Compute initial portfolio value:

$$\vartheta_t = x^b p_t$$

2. Simulate S one-day returns

$$r_{t+1,s} \sim \mathcal{N}\left(0, \sigma^2\right), \ s = 1, ..., S$$

1. VaR With One Basic Asset

Six-step procedure for obtaining MC VaR

1. Compute initial portfolio value:

$$\vartheta_t = x^b p_t$$

2. Simulate S one-day returns

$$r_{t+1,s} \sim \mathcal{N}\left(0, \sigma^2\right), \quad s = 1, ..., S$$

3. Calculate the one-day-ahead s^{th} simulated price:

$$p_{t+1,s} = p_t(1 + r_{t+1,s})$$

1. VaR With One Basic Asset

Six-step procedure for obtaining MC VaR

4. Calculate the s^{th} simulated one-day-ahead value of the portfolio:

$$\vartheta_{t+1,s} = x^b p_{t+1,s}$$

1. VaR With One Basic Asset

Six-step procedure for obtaining MC VaR

4. Calculate the s^{th} simulated one-day-ahead value of the portfolio:

$$\vartheta_{t+1,s} = x^b p_{t+1,s}$$

5. The s^{th} simulated profit or loss is then:

$$q_{t+1,s} = \vartheta_{t+1,s} - \vartheta_t$$

1. VaR With One Basic Asset

Six-step procedure for obtaining MC VaR

4. Calculate the s^{th} simulated one-day-ahead value of the portfolio:

$$\vartheta_{t+1,s} = x^b p_{t+1,s}$$

5. The s^{th} simulated profit or loss is then:

$$q_{t+1,s} = \vartheta_{t+1,s} - \vartheta_t$$

6. VaR can be obtained directly from the vector of simulated profit or loss,

$$\{q_{t+1,s}\}_{s=1}^{S}$$

for example, VaR(0.01) is the 1% smallest value

2. VaR With an Option

Modified six-step procedure

- For options we need to modify the procedure
- Let $g(\cdot)$ denote the Black-Scholes equation and suppose we have x^o options
- We replace steps 1 and 4 and come up with the following procedure

2. VaR With an Option

Modified six-step procedure

1'. Initial portfolio is

$$\vartheta_t = x^o g\left(p_t, X, \Upsilon - \tau, \sqrt{250}\sigma, \iota\right)$$

2. VaR With an Option

Modified six-step procedure

1'. Initial portfolio is

$$\vartheta_t = x^o g\left(p_t, X, \Upsilon - \tau, \sqrt{250}\sigma, \iota\right)$$

2. Simulate S one-day returns

$$r_{t+1,s} \sim \mathcal{N}\left(0, \sigma^2\right), \; s = 1, ..., S$$

2. VaR With an Option

Modified six-step procedure

3. Calculate the one-day-ahead price:

$$p_{t+1,s} = p_t(1 + r_{t+1,s})$$

2. VaR With an Option

Modified six-step procedure

3. Calculate the one-day-ahead price:

$$p_{t+1,s} = p_t(1 + r_{t+1,s})$$

4'. The s^{th} simulated one-day-ahead value of the portfolio is

$$\vartheta_{t+1,s} = x^o g\left(p_{t+1,s}, X, \Upsilon - \tau - \frac{1}{365}, \sqrt{250}\sigma, \iota\right)$$

2. VaR With an Option

Modified six-step procedure

5. The s^{th} simulated profits or losses is then:

$$q_{t+1,s} = \vartheta_{t+1,s} - \vartheta_t$$

2. VaR With an Option

Modified six-step procedure

5. The s^{th} simulated profits or losses is then:

$$q_{t+1,s} = \vartheta_{t+1,s} - \vartheta_t$$

6. VaR can be obtained directly from vector of simulated profits or losses,

$$\{q_{t+1,s}\}_{s=1}^{S}$$

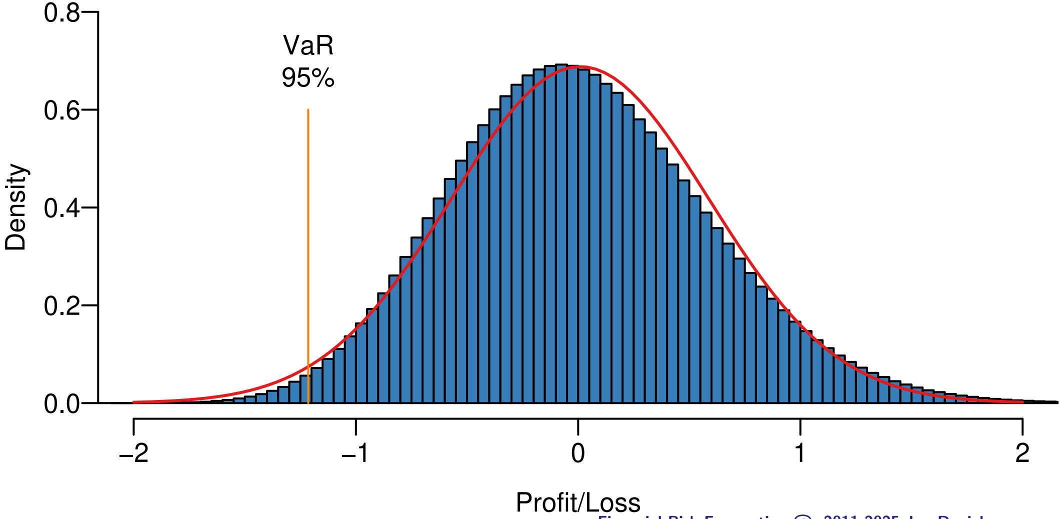
for example, VaR(0.01) is the 1% smallest value

2. MC VaR of Option

- One call option with strike price $X = 100$ and 3 months to expiry
- R gives VaR(0.01) of \$1.21

2. Density of Simulated profits or losses

Normal distribution superimposed



3. VaR With an Option and a Stock

Modified six-step procedure

- Now consider the case of a portfolio with both a stock and option(s) on the same stock
- Suppose we only have one type of option
- As in the case where we only had one option on a basic asset, we replace steps 1 and 4

3. VaR With an Option and a Stock

Modified six-step procedure

1”. Initial portfolio is

$$\vartheta_t = x^b p_t + x^o g\left(p_t, X, \Upsilon - \tau, \sqrt{250}\sigma, \iota\right)$$

3. VaR With an Option and a Stock

Modified six-step procedure

1”. Initial portfolio is

$$\vartheta_t = x^b p_t + x^o g\left(p_t, X, \Upsilon - \tau, \sqrt{250}\sigma, \iota\right)$$

2. Simulate S one-day returns

$$r_{t+1,s} \sim \mathcal{N}\left(0, \sigma^2\right), \; s = 1, ..., S$$

3. VaR With an Option and a Stock

Modified six-step procedure

3. Calculate the one-day-ahead price:

$$p_{t+1,s} = p_t(1 + r_{t+1,s})$$

3. VaR With an Option and a Stock

Modified six-step procedure

3. Calculate the one-day-ahead price:

$$p_{t+1,s} = p_t(1 + r_{t+1,s})$$

4". The s^{th} simulated one-day-ahead value of the portfolio is

$$\vartheta_{t+1,s} = x^b p_{t+1,s} + x^o g\left(p_{t+1,s}, X, \Upsilon - \tau - \frac{1}{365}, \sqrt{250}\sigma, \iota\right)$$

3. VaR With an Option and a Stock

Modified six-step procedure

5. The s^{th} simulated profits or losses is then:

$$q_{t+1,s} = \vartheta_{t+1,s} - \vartheta_t$$

3. VaR With an Option and a Stock

Modified six-step procedure

5. The s^{th} simulated profits or losses is then:

$$q_{t+1,s} = \vartheta_{t+1,s} - \vartheta_t$$

6. VaR can be obtained directly from the vector of simulated profits or losses,

$$\{q_{t+1,s}\}_{s=1}^{S}$$

for example, VaR(0.01) is the 1% smallest value

Simulation Pricing of Portfolio VaR

Simulation of Portfolio VaR

- Consider the multivariate case, that is, the case of more than one underlying asset
  - Main difference: we need to simulate *correlated returns* for all assets
  - Simulated one-day-ahead prices calculated as before and portfolio value obtained by summing up individual simulated asset holdings

Simulation of Portfolio VaR

- Suppose we have two non-derivative assets with daily return distribution

$$\mathcal{N}\left(0, \Sigma = \begin{pmatrix} 0.01 & 0.0005 \\ 0.0005 & 0.02 \end{pmatrix}\right)$$

- Let $\mathbf{x}^b$ be a vector of number of assets held

Notation

- The notation becomes cluttered for the multivariate case
- Now we have to denote variables by time period, asset and simulation
- We let $p_{t,k,s}$ denote the s^{th} simulated price of asset k at time t, that is:

$$p_{\text{time},\text{asset},\text{simulation}} = p_{t,k,s}$$

Portfolio VaR for Basic Assets

Six-step procedure for obtaining MC portfolio VaR

1. Compute initial portfolio value:

$$\vartheta_t = \sum_{k=1}^{K} x_k^b p_{t,k}$$

Portfolio VaR for Basic Assets

Six-step procedure for obtaining MC portfolio VaR

1. Compute initial portfolio value:

$$\vartheta_t = \sum_{k=1}^{K} x_k^b p_{t,k}$$

2. Simulate a vector of one-day returns from today to tomorrow

$$r_{t+1,s} \sim \mathcal{N}(0, \Sigma)$$

Portfolio VaR for Basic Assets

Six-step procedure for obtaining MC portfolio VaR

3. The s^{th} simulated one-day-ahead price of asset k is:

$$p_{t+1,k,s} = p_{t,k}\left(1 + r_{t+1,k,s}\right)$$

Portfolio VaR for Basic Assets

Six-step procedure for obtaining MC portfolio VaR

3. The s^{th} simulated one-day-ahead price of asset k is:

$$p_{t+1,k,s} = p_{t,k}\left(1 + r_{t+1,k,s}\right)$$

4. The s^{th} simulated one-day-ahead value of the portfolio is:

$$\vartheta_{t+1,s} = \sum_{k=1}^{K} x_k^b p_{t+1,k,s}$$

Portfolio VaR for Basic Assets

Six-step procedure for obtaining MC portfolio VaR

5. The s^{th} simulated profits or losses is then:

$$q_{t+1,s} = \vartheta_{t+1,s} - \vartheta_t$$

Portfolio VaR for Basic Assets

Six-step procedure for obtaining MC portfolio VaR

5. The s^{th} simulated profits or losses is then:

$$q_{t+1,s} = \vartheta_{t+1,s} - \vartheta_t$$

6. VaR can be obtained directly from the vector of simulated profits or losses,

$$\{q_{t+1,s}\}_{s=1}^{S}$$

as before

Portfolio VaR for Options

Modified six-step procedure

- For options, we need to modify steps 1 and 4 from the procedure outlined above
  - Similar to modifications for the univariate case before
- For simplicity suppose the portfolio has only one type of option type per stock

Portfolio VaR for Options

Modified six-step procedure

1'. Initial portfolio is

$$\vartheta_t = \sum_{k=1}^{K} \left(x_k^b p_{t,k} + x_k^o g\left(p_{t,k}, X_k, \Upsilon - \tau, \sqrt{250}\sigma_k, \iota \right) \right)$$

Portfolio VaR for Options

Modified six-step procedure

1'. Initial portfolio is

$$\vartheta_t = \sum_{k=1}^{K} \left(x_k^b p_{t,k} + x_k^o g \left(p_{t,k}, X_k, \Upsilon - \tau, \sqrt{250}\sigma_k, \iota \right) \right)$$

2. Simulate a vector of one-day returns from today to tomorrow

$$r_{t+1,s} \sim \mathcal{N}(0, \Sigma)$$

Portfolio VaR for Options

Modified six-step procedure

3. The s^{th} simulated one-day-ahead price of asset k is:

$$p_{t+1,k,s} = p_{t,k}\left(1 + r_{t+1,k,s}\right)$$

Portfolio VaR for Options

Modified six-step procedure

3. The s^{th} simulated one-day-ahead price of asset k is:

$$p_{t+1,k,s} = p_{t,k}\left(1 + r_{t+1,k,s}\right)$$

4'. The s^{th} simulated one-day-ahead value of the portfolio is

$$\vartheta_{t+1,s} = \sum_{k=1}^{K}\Bigg(x_k^b p_{t+1,k,s} + x_k^o g\left(p_{t+1,k,s}, X_k, \Upsilon - \tau - \frac{1}{365}, \sqrt{250}\sigma_k, \iota\right)\Bigg)$$

Portfolio VaR for Options

Modified six-step procedure

5. The s^{th} simulated profits or losses is then:

$$q_{t+1,s} = \vartheta_{t+1,s} - \vartheta_t$$

Portfolio VaR for Options

Modified six-step procedure

5. The s^{th} simulated profits or losses is then:

$$q_{t+1,s} = \vartheta_{t+1,s} - \vartheta_t$$

6. VaR can be obtained directly from vector of simulated profits or losses,

$$\{q_{t+1,s}\}_{s=1}^{S}$$

as before

Richer Versions

- We used simple examples to avoid cluttered notation, straightforward to allow for more complicated portfolios
  - Number of stocks and multiple options on each stock
  - American (or more exotic) options
  - Combination of fixed income assets with stocks and options
- Also, we could use other distributions (eg Student-t or even historical simulation)

Takeaways: Portfolio VaR by Simulation

- Monte Carlo VaR scales naturally from single assets to portfolios
- Correlated returns are handled via multivariate simulation
- Options require revaluation using pricing models (e.g. Black-Scholes)
- Simulation allows considerable flexibility — but at greater computational cost

Issues In Simulation Estimation

Why This Matters

- Monte Carlo simulation is powerful — but only if implemented carefully
- Poor randomness, bad transformations or too few runs can make results misleading
- This section highlights the most common pitfalls and how to manage them

Simulation Issues

- Several issues need to be addressed in all MC exercises, of which two are most important:
  1. Quality of RNG and transformation method
  2. Number of simulations

Quality of RNG

- MC simulation is not only dependent on quality of the underlying stochastic model, also depends on quality of the RNG used
- Low-quality generators give biased or inaccurate results
  - For example, a simulation size of 100 with period of 10 will repeat same calculation 10 times
- Complicated portfolios may demand large number of RNs and therefore high-quality RNGs

Quality of RNG Transformation

- Many transformation methods (from a uniform to a desired distribution) are only optimally tuned for the centre of the distribution
- This becomes particularly problematic when simulating extreme events
- Some transformation methods use linear approximations for extreme tails, which leads to extreme uniforms being incorrectly transformed
- E.g. inverse CDF approximations often use linear extrapolation in extreme tails, leading to underestimation of large risks

Choosing Simulation Size: Trade-offs

- Choosing appropriate number of simulations is important
  - Too few give inaccurate answers
  - Too many waste time and computer resources
- In special cases formal statistical tests provide guidance, but usually informal methods have to be relied upon
- It is sometimes stated that accuracy of simulations is related to inverse simulation size
  - This is based on assumption of linearity, which is not correct for the problems in this chapter

Practical Guidance for Simulation Size

- Best way is to simply increase number of simulations and see how MC estimate converges
- Rule of thumb: Sufficient simulation size when numbers have stopped changing up to three significant digits
- We can also compare *convergence of MC estimate* to the true (analytical) price

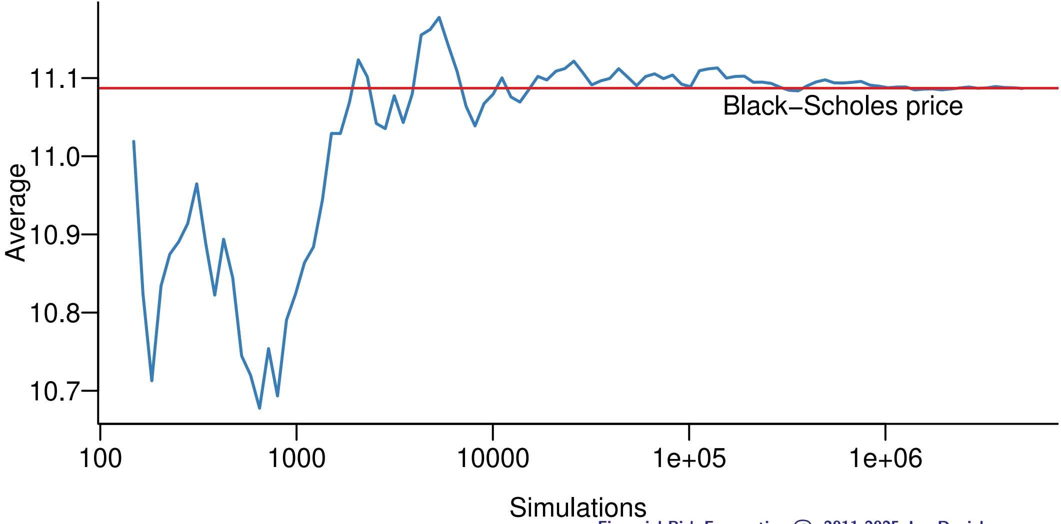
Convergence of MC Estimate

Comparison with analytical Black-Scholes price

- In a example on slide 44 we computed analytical call price of $11.0873 for a European option
- Now calculate MC estimates for different simulation sizes and compare the results with the true (analytical) price

Cumulative MC Estimates

Comparison with analytical Black-Scholes price



Convergence of MC Estimate

Comparison with analytical Black-Scholes price

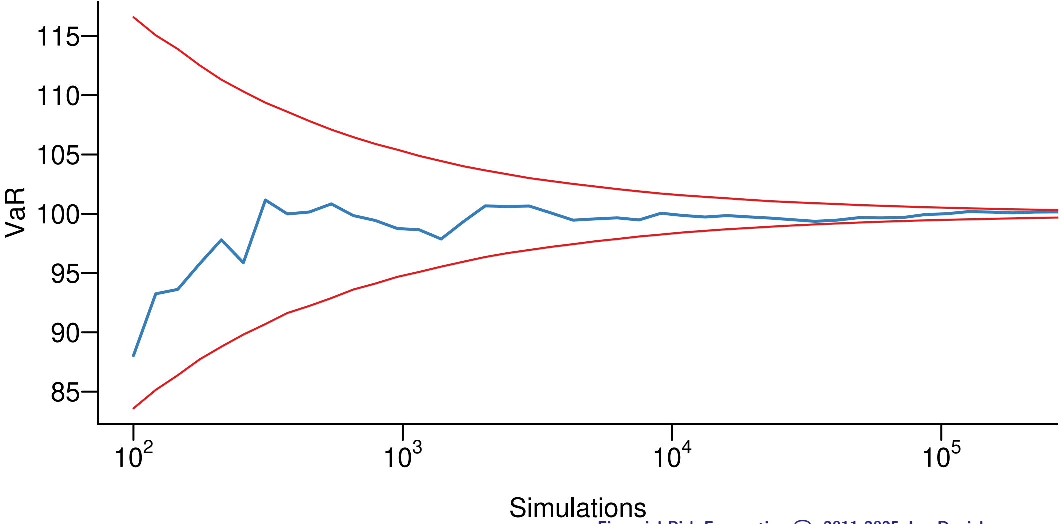
- Based on graph on previous slide, it seems to take about 5000 simulations to get three significant digits correct
- However, there are still fluctuations in the estimate for 5 million simulations

Convergence of MC VaR Estimates

- Look at the convergence of MC VaR estimates as the simulation size increases
- Graph MC VaR for a stock with daily volatility 1% along with ±99% confidence intervals

Convergence of MC VaR Estimates

With ±99% confidence intervals



Simulation VaR: Key Takeaways

- Simulation is flexible and widely applicable
- Especially useful for portfolios with options and non-linear risk
- Quality of inputs (RNGs, assumptions, pricing models) matters enormously
- More computationally expensive — but increasingly feasible
- A crucial tool for modern risk management