

UNIVERSIDADE FEDERAL FLUMINENSE

ISRAEL CALHEIROS DA SILVA  
RENATO JERONIMO G. DE MATTOS

Implementação de Modelos de Derivativos por meio  
de fórmula fechada e árvore

NITERÓI

2006

UNIVERSIDADE FEDERAL FLUMINENSE

ISRAEL CALHEIROS DA SILVA  
RENATO JERONIMO G. DE MATTOS

# Implementação de Modelos de Derivativos por meio de fórmula fechada e árvore

Estudo do Mercado de Opções e Implementação da solução da equação diferencial de Black-Scholes pelo método de fórmula fechada e árvore.

Orientador:  
Dsc. Inhaúma Neves Ferraz

NITERÓI

2006

# Implementação de Modelos de Derivativos por meio de fórmula fechada e árvore

Israel Calheiros da Silva  
Renato Jeronimo G. de Mattos

Projeto de final de curso apresentado ao Curso de Graduação em Ciência da Computação da Universidade Federal Fluminense como requisito para obtenção da Graduação).

Aprovada por:

---

Prof. D.Sc. Inhaúma Neves Ferraz/ IC-UFF

---

Prof. D.Sc. Marco Antonio Silva Ramos/ IC-UFF

---

Prof. D.Sc. Carlos José Guimarães Cova/ ADM-UFF

Niterói, 14 de Dezembro de 2006.

*Dedico esse trabalho a minha família, Jared Calheiros da Silva e Lúcia Calheiros da Silva e a minha namorada, Jaqueline Mayworm Grünevald. I*

*Israel Calheiros da Silva*

*Dedico esse trabalho a todas as pessoas que me ajudaram direta ou indiretamente na minha formação, não só acadêmica como também profissional.*

*Renato Jeronimo G. de Mattos*

# Agradecimentos

Aos professores Inhaúma Neves Ferraz, Marco Antonio Silva Ramos e Carlos José Guimarães Cova que proporcionaram grande ajuda na confecção deste trabalho.

A todos os professores com quem convivi e de quem tive o privilégio de ser aluno no curso de Graduação em Ciência da Computação da Universidade Federal Fluminense.

*Israel Calheiros da Silva*

Aos meus pais, Sérgio Roberto de Mattos e Gleise Godinho de Mattos, pelo grande esforço que tiveram para me oferecer a oportunidade de cursar uma universidade e, como mais importante, o apoio, incentivo e amizade.

Ao meu professor Orientador Inhaúma Neves Ferraz, não somente pelo seu conhecimento acadêmico, mas principalmente pela experiência compartilhada nas suas aulas e nas nossas conversas.

Ao meu professor Roberto Gouveia pela orientação ao longo de todo o curso.

Aos amigos que, ao longo do curso, foram fundamentais para minha formação e principalmente ao meu amigo Israel Calheiros pelo ótimo trabalho que realizou neste estudo juntamente comigo.

E em especial para o meu amor Rachel Luciana pela compreensão, incentivo e amizade que teve ao longo dos meus 7 anos de vida universitária.

*Renato Jeronimo G. de Mattos*

# Resumo

Estudo do Mercado de Opções e implementação da solução da equação diferencial de Black-Scholes pelo método de fórmula fechada e árvore. A abordagem adotada seguirá estes passos:

1. Modelagem do comportamento estocástico de ativos utilizando o movimento browniano geométrico;
2. Utilização do Lema de Itô para chegar ao comportamento estocástico geral de derivativos;
3. Criação de uma carteira, ou portfólio, de risco zero que deve render taxa livre de risco por arbitragem chegando à equação diferencial geral de derivativos;
4. De cada contrato extrair as condições de contorno para a solução da equação diferencial;
5. Efetuar a busca de soluções por meio de fórmula fechada, árvore, método de Monte Carlo ou risk neutral valuation;

# Abstract

A Financial derivative, for example an option, is a contract defined in terms of some underlying asset which already exists on the market, such as a stock. The concept of option comes from the negotiable right of buying or selling an asset with a predetermined future price. The option pricing, which consists in determining the premium (price), is a crucial problem the option theory tackles with. In an attempt to solve this problem, one of the most used models in Finance was developed, the Black and Scholes model. However, this method is inefficient in pricing options where there is dependence of the way followed by the asset price. This work aimed to implement the lattice based models (binomial and trinomial) and verify their utilization in pricing european, american options.



# Palavras-chave

1. Equação Diferencial de Black-Scholes
2. Árvore Binomial
3. Precificação de opção
4. Monte Carlo
5. Modelos Lattices
6. Derivativos
7. risco

# Glossário

UFF	:	Universidade Federal Fluminense;
IC	:	Instituto de Computação;
BOVESPA	:	Bolsa de valores de São Paulo;
CFTC	:	Commodity Futures Trading Commission;
BMF	:	Bolsa de Mercadorias e Futuros;
CVM	:	Comissão de Valores Mobiliários;
CBLC	:	Companhia Brasileira de Liquidação e Custódia;
CLC	:	Câmara de Liquidação de Custódia;

# Sumário

<b>Lista de Figuras</b>	<b>xii</b>
<b>Lista de Tabelas</b>	<b>xiii</b>
<b>1 Introdução aos Mercados Futuros</b>	<b>1</b>
1.1 História dos Mercados Futuros . . . . .	2
1.2 Derivativos . . . . .	3
1.3 Ação . . . . .	6
1.4 Participantes . . . . .	6
1.5 Operadores . . . . .	7
1.6 Mercado de Balcão e Bolsa de Valores . . . . .	7
1.7 Bolsa de Mercadorias e Futuros . . . . .	8
1.8 Pregão . . . . .	9
1.9 Home broker e After Market . . . . .	9
1.10 Caixa de Liquidação ou Câmara de Compensação das Bolsas . . . . .	10
1.11 Convergência entre os preços futuros e Spot . . . . .	10
1.12 Determinação dos Preços a Termo e Futuros . . . . .	11
<b>2 Opções</b>	<b>13</b>
2.1 Preço a vista (Spot Price) . . . . .	14
2.2 Preço de entrega ou Exercício (Strike price) . . . . .	15
2.3 Data de maturidade, entrega ou vencimento (Expiration date) . . . . .	16
2.4 Vencimento . . . . .	16

---

2.5	Volatilidade (Volatility) . . . . .	16
2.6	Dividendos (Dividend Yield) . . . . .	18
2.7	Autorização da Negociação das Séries de Opções . . . . .	18
<b>3</b>	<b>Black-Scholes</b>	<b>19</b>
3.1	Movimento Browniano . . . . .	19
3.2	Equação Diferencial de Black-Scholes . . . . .	20
3.2.1	Hipóteses assumidas . . . . .	23
<b>4</b>	<b>Árvore Binomial</b>	<b>25</b>
4.1	Modelo Binomial de um passo . . . . .	25
4.2	Árvore de Cox-Ross-Rubinstein . . . . .	27
4.3	Opção do tipo Americano . . . . .	28
4.4	Árvore de Preços de Ações . . . . .	28
4.5	Voltando com Backwardation . . . . .	28
4.6	Opção do Tipo Europeu . . . . .	29
<b>5</b>	<b>Monte Carlo</b>	<b>30</b>
5.1	Metodologia de Monte Carlo para precificação de opções . . . . .	31
5.2	Geração das Cadeias de Markov para Simulação de Monte Carlo . . . . .	32
<b>6</b>	<b>Análise e Metodologia</b>	<b>33</b>
<b>7</b>	<b>Implementação</b>	<b>37</b>
7.1	Modelo Lattice . . . . .	37
7.2	Modelo Black-Scholes . . . . .	38
7.3	BinomialTree.h . . . . .	39
7.4	BinomialTree.cpp . . . . .	45
7.5	BlackScholes.h . . . . .	66

---

7.6	BlackScholes.cpp . . . . .	69
7.7	MonteCarlo.h . . . . .	73
7.8	MonteCarlo.cpp . . . . .	75
7.9	Statistic.h . . . . .	79
7.10	Statistic.cpp . . . . .	81
7.11	MersenneTwister.h . . . . .	83
7.12	MersenneTwister.cpp . . . . .	86
<b>8</b>	<b>Testes</b>	<b>94</b>
<b>9</b>	<b>Implantação</b>	<b>95</b>
<b>10</b>	<b>Conclusão</b>	<b>96</b>
	<b>Apêndice A</b>	<b>97</b>
A.1	Lemma de Itô . . . . .	97
A.2	Distribuição Normal . . . . .	97
A.3	Distribuição Lognormal . . . . .	98
A.4	Recursivo.cpp . . . . .	99
	<b>Índice Remissivo</b>	<b>101</b>
	<b>Referências</b>	<b>102</b>

# Lista de Figuras

1.1	Possibilidades acerca do Preço de uma ação depois de 3 meses . . . . .	11
4.1	Possibilidades acerca do Preço de uma ação depois de 3 meses . . . . .	25
4.2	Árvore Binomial de 4 passos . . . . .	27

# Lista de Tabelas

2.1	Opções de Compra da Telebrás PN . . . . .	18
2.2	Opções de Venda da Telebrás PN . . . . .	18

# Capítulo 1

## Introdução aos Mercados Futuros

Em anos recentes, os mercados futuros e opções vêm obtendo crescente importância. Atingiu-se um estágio no qual é essencial, para todos os profissionais de finanças, o entendimento do funcionamento e da utilização desses mercados e dos fatores que determinam os preços dos produtos neles negociados. O conceito de opção nasce como um direito negociável de compra ou venda de um ativo a um preço futuro pré-estabelecido. O cálculo de risco, popularmente conhecido como prêmio de uma opção é vital na teoria de risco aplicada a Finanças. Um dos métodos mais utilizados é o modelo de Black-Scholes entretanto, este modelo torna-se ineficiente quando consideramos alguma dependência do caminho percorrido ou exercício antecipado. Para este tipo de problema torna-se necessário utilizar o modelo de árvore ou *Lattice* ou Monte Carlo.



## 1.1 História dos Mercados Futuros

If I have seen farther than others, it is because I was standing on the shoulders of giants.

- *Isaac Newton*

Os mercados futuros remontam à Idade Média. Foram originalmente desenvolvidos para satisfazer as necessidades de produtores e comerciantes de produtos agrícolas. [1] Considere a posição de um produtor, em abril de determinado ano, que está para colher certa quantidade de grãos em junho.

Existe incerteza acerca do preço que receberá pela sua mercadoria. Em anos de escassez, possivelmente obterá preços relativamente altos - especialmente se a venda não for realizada de forma imediata. Por outro lado, em anos de excesso de oferta, o grão provavelmente será vendido a preços muito baixos. O produtor e sua família estão claramente expostos a grandes riscos.

Considere, agora, uma companhia que necessite do grão. Assim como o produtor, esta também está exposta ao risco de preço. Em alguns anos, o excesso de oferta pode criar preços favoráveis; em outros, a escassez pode fazer que os preços sejam exorbitantes.

Em outras palavras, é sensato, para esses dois agentes de mercado, a negociação de um tipo de contrato futuro, que proporcionará um meio de eliminar o risco ao qual estão expostos por conta da incerteza do preço futuro do grão.

Diante dessa situação, faz bastante sentido, para o fazendeiro e a companhia, marcarem um encontro em abril (ou até mais cedo) e acertarem o preço para a produção que será colhida em junho.

No entanto, deve-se indagar o que ocorre com o abastecimento de grãos da companhia durante o restante do ano. Terminada a época da colheita, o produto será estocado até a próxima estação. Ao decidir pelo armazenamento, a companhia não incorre em risco de preço, mas arcará com os custos da estocagem. Se o produtor ou qualquer outra pessoa armazenar o grão, tanto a companhia quanto o armazenador assumirão os riscos associados ao preço futuro desse ativo, novamente surgindo espaço para que os contratos futuros entrem em cena.

## 1.2 Derivativos

Onde está a vida que eu perdi vivendo? Onde está a sabedoria que eu perdi no conhecimento?

- *T.S. Eliot*

Entender derivativos, mais que conhecer sua mecânica operacional, é dominar o apreçamento e a avaliação do risco desses instrumentos.

Um **derivativo** (derivative) é um instrumento cujo preço depende ou é derivado do preço de um ativo subjacente. Pode ser utilizado com a finalidade de obter ganhos especulativos ou como proteção contra oscilações dos preços de seus produtos e de seus investimentos em ativos financeiros. Um derivativo também pode ser visto como um contrato entre duas partes que especifica condições - em particular, datas e os valores resultantes das variáveis relacionadas - sobre os quais pagamentos, ou payoffs, são para serem feitos entre as partes. [2]

Três tipos de derivativos muito conhecidos dos quais poderíamos citar são os contratos futuros, de opções e a termos .

Os contratos futuros e a termo são acordos de compra ou venda de um ativo em determinado período futuro por um preço pré-estabelecido. Os **contrato futuros** (Future Contracts) são negociados em bolsa de valores<sup>1</sup> , sendo necessário que esta defina cuidadosamente a natureza precisa do que esta sendo negociado, os procedimentos que serão seguidos e as normas que governarão o mercado enquanto os contratos a termo (Forward Contracts) são transacionados em mercados de balcão<sup>2</sup> , diretamente via telefone por dois indivíduos relativamente sofisticados. Como resultado, não há necessidade de padronização do produto nem de extenso conjunto de regras e procedimentos.

Os mercados de liquidação futura estão regulamentados pela Instrução 283, de 10/07/1998 da Comissão de Valores Mobiliários(CVM).

A grande parte negociada dos contratos futuros não resulta na entrega do ativo objeto (**underlying asset**) em seu vencimento. No mercado futuro de commodities<sup>3</sup> , por exemplo, calcula-se estimativamente que apenas 2% das operações são liquidados pela entrega efetiva do bem negociado. A razão disso é que a maioria dos investidores prefere

<sup>1</sup>O papel das bolsas é explicado na seção 1.7

<sup>2</sup>Consulte a seção 1.7 para mais informação

<sup>3</sup>O termo commodity aplica-se àquelas mercadorias cujo preço é determinado em bolsas de mercadorias

encerrar suas posições antes do período de entrega especificado no contrato. Entregar ou receber sob os termos de um contrato futuro é freqüentemente, inconveniente e, em alguns casos, bastante dispendioso. Isso pode ocorrer mesmo para o hedger<sup>4</sup> que deseja comprar ou vender o ativo subjacente ao contrato futuro. Esse hedger em geral prefere fechar sua posição no mercado futuro e depois comprar ou vender o ativo pelo caminho a que está acostumado. Fechar uma posição requer a realização de uma operação de natureza oposta a posição original, ou seja, as posições encerradas, antes que o período de entrega seja alcançado. Entretanto, é a possibilidade de entrega final que governa a determinação dos preços futuros ao preço **spot**<sup>5</sup>.

Há, geralmente, uma única data de entrega para o contrato a termo enquanto que para cada contrato futuro, há um período dentro do qual as entregas podem ser realizadas, bem como procedimentos de liquidação bem definidos.

A especificação do contrato é uma atividade importante para uma bolsa de futuros. As duas partes envolvidas em um negócio com contrato futuro devem saber o que pode ser entregue, onde a entrega pode ser realizada (praça) e quando pode acontecer. Precisam saber também detalhes relativos ao horário de negociação, forma de cotação, oscilações máximas diárias de preços permitidas etc.. Nos EUA, novos contratos devem ser aprovados pela **Commodity Futures Trading Commission (CFTC)** antes que sua negociação se inicie. No Brasil, o mercado de futuros é realizado através da Bolsa de Mercadorias e Futuros BM&F.

Existem dois tipos de opções: opções de compra (calls) e opções de venda (puts).

A call confere ao seu detentor o direito<sup>6</sup> de comprar o ativo em determinada data a certo preço, nas condições estabelecidas pela BOVESPA. Este comprador é dito que assumiu **long position**.

A put garante ao titular o direito de vender o ativo por certo preço em determinada data, nas condições estabelecidas pela BOVESPA. Este vendedor é dito que assumiu **short position**.

Por esses direitos e em função das obrigações (e riscos) assumidos no lançamento, o titular<sup>7</sup> de uma opção paga uma determinada quantia (**prêmio**) ao lançador da opção<sup>8</sup>,

---

<sup>4</sup>Hedgers são investidores que entram no mercado futuro para estabilizar sua estrutura de lucro e reduzir seu risco comercial. Consulte 1.4

<sup>5</sup>Mercado a vista ou disponível. É o preço de entrega quase imediata

<sup>6</sup>Este direito não pressupõe obrigação. Se não for do interesse do titular ele não será exercido

<sup>7</sup>O titular de uma opção é quem compra a opção

<sup>8</sup>O lançador da opção é quem vende a opção

podendo exercê-las até a data de vencimento (no caso de opções do estilo americano) ou na data de vencimento (no caso de opções do estilo europeu), ou revendê-los no mercado.

O **lançador de uma opção de compra** é uma pessoa que, por intermédio de seu corretor, vende uma opção de compra no pregão, assumindo assim, perante a Bolsa, a obrigação de vender as ações-objeto a que se refere a opção, após o recebimento de uma comunicação de que sua posição foi exercida. Ele entregará a totalidade das ações-objeto mediante o pagamento do preço de exercício

O **lançador** que, por intermédio de seu corretor, vende uma **opção de venda** no pregão, assume perante a Bolsa a obrigação de comprar as ações-objeto, caso sua posição seja designada para o atendimento de uma operação de exercício. Nesse caso, ele pagará o preço de exercício, recebendo as ações-objeto. O titular tem sempre os direitos (comprador) enquanto que o lançador tem obrigações.

Perceba que existem quatro pessoas envolvidas no contrato de opções.

Duas para um contrato de opção de compra

- Lançador de uma Opção de Compra;
- Titular da Opção de Compra ou comprador da Opção de Compra;

E outras duas para um contrato de opção de venda

- Lançador de uma Opção de Venda;
- Titular da Opção de Venda ou Comprador da Opção de Venda;

O valor dos preços de exercícios são estabelecidos pela BOVESPA.

O prêmio, ou preço da opção, é negociado entre o comprador e o lançador, por meio de seus representantes no pregão da Bolsa. Ele reflete fatores como oferta e demanda, o prazo da vigência da opção, a diferença entre o preço de exercício e preço a vista do ativo. A volatilidade do preço, bem como outras características. Nos capítulos que se seguem daremos uma explicação detalhada sobre cada um desses fatores que influem no preço final da opção.

Os lançadores podem ser considerados especuladores e os compradores como hedgers porque tentam diminuir os riscos associados a mudança de preço (hedgers) ou obter lucros desta mudança (especuladores).

## 1.3 Ação

Os valores mobiliários, representados por ações e debêntures, são emitidos pelas Sociedades Anônimas de acordo com a aprovação prévia da CVM Comissão de Valores Mobiliários.

## 1.4 Participantes

Os três principais participantes desses mercados podem ser identificados como: hedgers, especuladores e arbitradores.

- Os hedgers são agentes negociadores que enfrentam o risco de preço de um ativo e, assim, buscam tais mercados para reduzir ou eliminar essa exposição.
- Os especuladores desejam apostar nas oscilações futuras do preço do ativo. Existem ainda três tipos de especuladores:
  - Scalpers. Procuram observar tendências a curtíssimos prazos e buscam obter lucro em pequenas variações de preços;
  - Day traders. Procuram manter posições por períodos menores do que um dia pois não querem arriscar correr o risco da variação de preço de um dia para o outro;
  - Position Traders. Mantém suas posições por períodos de tempos mais longos.

Os contratos futuros, a termo e de opções podem proporcionar a alavancagem de suas posições, sendo possível elevar tanto o ganho como a perda do investimento.

- Os arbitradores realizam negócios para aproveitar a discrepância entre os preços de dois mercados diferentes. Se, por exemplo, perceberem que o preço futuro do ativo não estiver em sintonia com o seu preço a vista, posições serão tomadas nos dois mercados a fim de travar lucro.

Quando se fala em investimento em mercado financeiro, duas metodologias podem ser utilizadas:

- **Escola Gráfica e Técnica** Admite que os investidores possuem um grau de informação razoavelmente uniforme, e tende a repetir atitudes do passado. Este tipo

de análise tenta buscar padrões de comportamento aos quais possam extrair informações que possam ajudar na análise. O fato de você estar num pregão todos os dias é um efeito auto-recorrente (um efeito não-neutro). Um dos melhores argumentos em defesa da análise técnica é que você não está desconectado do passado. Entretanto se o volume negociado for pequeno esta hipótese é desprezada. Ela é fundamental para Marketing time<sup>9</sup>

- **Escola Fundamentalista** Tem como objetivo reduzir a imporenabilidade do valor de um ativo. Buscando, por meio de projeções individuais de futuro e comparações, atingir um valor justo para o objetivo da avaliação. Ela é fundamental para o stock picking, ou seja, a escolha da empresa cuja ação deva ser adquirida e, como consequência, a determinação de qual deve ser vendida num determinado tempo (Eduardo Fortuna) Baseia-se no fato de que as informações do mercado são assimétricas. Para análise fundamentalista, compreender a dinâmica do processo é mais importante do que tentar compreender um quadro de algo que não é estático. Ela não nega a auto-recorrência, ela aceita com ressalvas. O problema que ela aborda é que o mercado é assimétrico, dinâmico.

## 1.5 Operadores

Existem dois tipos de operadores executando negócio: corretores (brokers) e os operadores especiais (locals). Os corretores seguem as instruções dos seus clientes e cobram comissão por isso. Os locals operam para si próprios.

## 1.6 Mercado de Balcão e Bolsa de Valores

Nem todos os negócios são realizados no recinto das bolsas. Outro segmento, conhecido como mercado de balcão é importante alternativa. Trata-se de um conjunto de dealers<sup>10</sup>, conectados por telefones e redes computadorizadas, que não se encontram fisicamente. Os negócios são feitos por telefone entre duas instituições financeiras ou entre uma instituição financeira e um de seus clientes corporativos. É comum que tais empresas atuem como market makers para a maioria dos instrumentos negociados. Isso significa que estão sempre preparados para cotar os preços de compra e de venda. As conversas telefônicas no mercado de balcão são, em geral, gravadas. Se houver controvérsia sobre o que foi

---

<sup>9</sup>Tipos de investimentos de curtíssimo prazo

<sup>10</sup>Agentes que intermediam a negociação no mercado mobiliário

acordado, utilizam-se as gravações a fim de resolver a questão. O número de negócios, em regra, é muito maior em relação ao realizado em bolsa. A grande vantagem consiste no fato de os termos do contrato não precisarem ter a mesma padronização que o contrato negociado em bolsa. Seus participantes podem negociar, livremente, os itens do contrato. A desvantagem é que, usualmente, há o risco de crédito (ou seja, existe pequeno risco de que o contrato não seja honrado). As bolsas se organizaram de maneira a eliminar esse tipo de problema criando margens de segurança. Margens são aspectos importantes dos mercados futuros. Um investidor mantém uma conta de margem com seu corretor através de um depósito prévio como garantia margem inicial que pode ser em dinheiro, ouro, carta de fiança bancária ou título da dívida pública federal. São uma espécie de caução que o investidor tem de depositar para operar no mercado. Ele deve ser suficiente para cobrir o valor de certo número de ajustes diários, ou seja, a volatilidade do preço em pregão do ativo específico (Eduardo Fortuna). A conta é ajustada diariamente para refletir ganhos ou perdas e, de tempos em tempos, o corretor pode requerer que essa conta seja nivelada se movimentos adversos de preços ocorrerem. O corretor pode ser membro da clearing<sup>11</sup> ou manter uma conta com um membro da clearinghouse. Cada membro da clearing deve manter uma conta de margem com a clearinghouse. O saldo dessa conta é ajustado diariamente para refletir os ganhos e as perdas nas posições pelas quais o membro de compensação é responsável. As informações sobre os preços futuros são capturadas de forma sistemática pelas bolsas e, em fração de segundos, estão disponíveis para os investidores em todo o mundo.

## 1.7 Bolsa de Mercadorias e Futuros

O objetivo da BM&F segundo Eduardo Fortuna, é organizar, operacionalizar, e desenvolver um mercado de derivativos livre e transparente. Um mercado que proporcione segurança a seus agentes.

Com este objetivo, ele disponibiliza as seguintes modalidades de operação:

- Contratos Futuros
- Contratos a Termo
- Opções sobre disponível

---

<sup>11</sup>Sistema que compatibiliza as posições compradas com as vendidas, de forma a garantir o fiel cumprimento de obrigações contratuais assumidas em mercados organizados

- Opções sobre futuro
- Opções Flexíveis
- Disponíveis (à vista)
- Swaps
- Swaps com ajustes diários

## 1.8 Pregão

É o recinto onde se reúne os operadores da bolsa de valores para executar fisicamente as ordens de compra e venda dadas pelos compradores e vendedores de ações as suas corretoras. Entretanto, hoje, é muito comum utilizar a informática para fazer este tipo de operação. O conceito de home broker<sup>12</sup> está cada vez mais difundido e hoje, da própria casa é possível um investidor fazer lançamentos na bolsa.

## 1.9 Home broker e After Market

Com as facilidades criadas pela informática, foram implantados em 1999 no Brasil, o conceito de home Broker e After Market

No primeiro os clientes passam a operar eletronicamente pela Internet, através de sites das corretoras de valores. No segundo, estende-se o horário do pregão eletronicamente que passa funcionar até as 22 horas. Esse tipo de negociação é chamado de After Market. Entretanto, o governo estabeleceu algumas regras, dentre as quais:

- Só pode ser negociada ações no mercado à vista ou de opções
- O preço das ordens de compra e venda só pode variar até 2%, em relação ao valor de fechamento das ações no horário normal da Bovespa
- Todas as variações de preço no dia seguinte são calculadas com base no último preço registrado no funcionamento do pregão regular
- O índice bovespa continua valendo, apenas, para variação da bolsa no pregão regular, as operações de After Market são divulgadas no dia seguinte

---

<sup>12</sup>Home brokers são empresas que disponibilizam pela internet o acesso a investimentos em bolsa mediante ao pagamento de uma corretagem



## 1.10 Caixa de Liquidação ou Câmara de Compensação das Bolsas

A clearing ou caixa de liquidação é o sistema elaborado pelas bolsas para garantir o fiel cumprimento de todos os negócios nelas realizados. A clearing pode ser tanto um departamento interno da instituição quanto uma organização independente, controlada ou não pela bolsa a qual está ligada.

Seus serviços vão desde o registro das operações e controle das posições, a compensação dos ajustes diários e liquidação física e financeira dos negócios, até a administração das garantias vinculadas às operações.

No Bovespa, funciona, após cisão patrimonial do Bovespa em 02/98, a Companhia Brasileira de Liquidação e custódia CBLC.

A CBLC se caracterizou no sentido de construir-se na empresa depositária central do mercado no Brasil, objetivo alcançado quando assumiu o controle da câmara de Liquidação e Custódia CLC, da Bolsa de Valores do Rio de Janeiro.

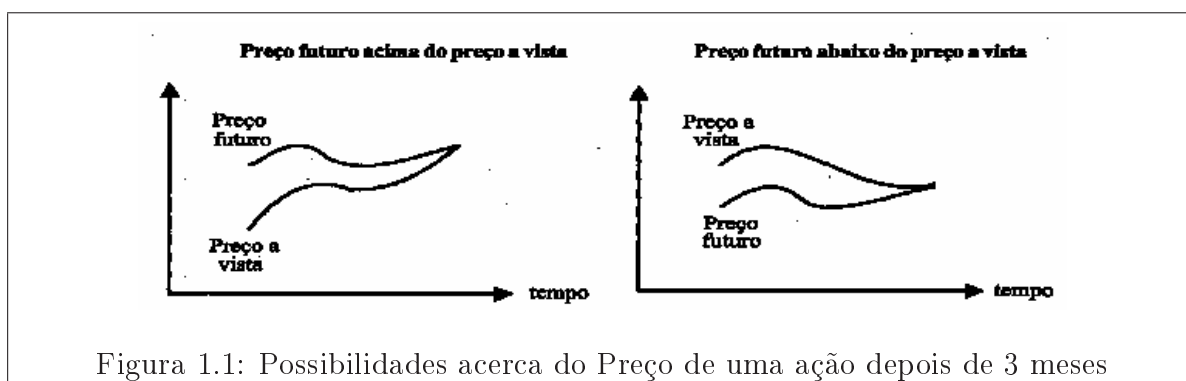
Constituiu-se juridicamente como sociedade anônima, tendo como objetivo social a prestação de serviço de compensação e liquidação física e financeira de operações realizadas nos mercados à vista e a prazo da Bovespa e de outros mercados, bem como a operacionalização dos sistemas de custódia de títulos e valores mobiliários em geral. (FORTUNA, Eduardo. Mercado Financeiro produtos e serviços. 16 ed Qualitymark, Rio de Janeiro, 2006 p.565)

## 1.11 Convergência entre os preços futuros e Spot

Quando o mês de entrega de um contrato futuro se aproxima, o preço converge para o preço à vista do ativo subjacente. Quando o período de entrega é alcançado, os preços futuros são iguais, ou muito próximos, ao preço a vista. Isto acontece devido à oportunidades de arbitragem. A figura demonstra este comportamento

Quando o preço futuro está abaixo do preço futuro esperado, a situação é conhecida como normal **backwardation** ; quando o preço futuro esta acima do preço futuro esperado a situação a conhecida como **contango**.

Para assegurar a convergência do preço futuro de um contrato e o preço em Spot, os



preços do ativo-objeto<sup>13</sup> são fixados no último dia

## 1.12 Determinação dos Preços a Termo e Futuros

Em quase todas as situações, o preço futuro do contrato para determinada data de entrega pode ser considerado igual ao preço a termo com data de entrega semelhante.

Pode ser demonstrado, em teoria, que os dois preços devem ser rigorosamente iguais na hipótese de taxas de juro perfeitamente previsíveis.

Para efeito de entendimento dos preços futuros (ou a termo), será conveniente dividir os contratos futuros em duas categorias: aqueles referenciados em ativos que são carregados com o objetivo de investimento por significativo número de investidores e aqueles referenciados em ativos que são mantidos com o objetivo de consumo. No caso de ativos de investimento, foram consideradas três situações diferentes:

- O ativo não gera rendimento ou renda;
- O ativo gera renda de valor conhecido em dólares;
- O ativo gera rendimento conhecido (percentual do preço do ativo).

Os resultados mostram a obtenção de preços futuros para índices de ações, moedas, ouro e prata. Custos de armazenagem podem ser considerados renda negativa. No caso de ativos de consumo, não é possível obter os preços futuros como uma função do preço spot e outras variáveis observáveis. Aqui, o parâmetro conhecido como convenience yield torna-se importante, pois mede em que extensão os detentores de commodity acreditam

<sup>13</sup> Ativo que o titular tem o direito de comprar e o lançador, a obrigação de vender (no caso de opções de compra) ou que o titular tem o direito de vender e o lançador, a obrigação de comprar (no caso de opções de venda)

que a propriedade do ativo físico crie benefícios que não são gerados por contrato futuro. Tais benefícios incluem a possibilidade de auferir lucros da escassez local temporária ou da capacidade de manter o processo de produção funcionando.

No caso de ativos de consumo, o preço futuro é maior que o preço spot em um montante que reflete o custo de carregamento abatido do convenience yield.

# Capítulo 2

## Opções

O desenvolvimento teórico da Teoria das Opções Financeiras tem como marco principal o trabalho apresentado por Fisher Black e Miron Scholes, que desenvolveu uma fórmula analítica para avaliação de opção de compra européia . A principal contribuição deste trabalho talvez não tenha sido a fórmula em si, mas sim a metodologia proposta. A formação de uma carteira dinâmica independente das preferências ao risco de seu detentor, permitiu que fosse utilizado uma taxa de juros livre de risco para descontar as remunerações futuras da opção.

Em um trabalho posterior, [52], alguns conceitos apresentados por Black e Scholes foram generalizados. Por exemplo, uma opção de compra Européia cujo ativo objeto pagava dividendos foi avaliada, foi mostrado que uma opção de compra Americana sobre um ativo objeto que não paga dividendos tem o mesmo valor que a opção Européia compra sobre o mesmo ativo objeto, ou seja, não é ótimo exercer antecipadamente uma opção de compra Americana quando o ativo objeto não paga dividendos. Em 1997, os pesquisadores Scholes e Merton<sup>1</sup> receberam o prêmio Nobel de Economia, pelo desenvolvimento desse modelo de precificação de opções.

Como vimos, opções podem ser do tipo Europeu ou Americano. No estilo Europeu, as opções só podem ser exercidas no dia do vencimento enquanto que as opções do estilo americano pode ser exercida a qualquer dia.

Opções são negociadas no mesmo horário de compra e venda do ativo no mercado a vista. É o Bovespa quem abre ou fecha séries de opções. No Brasil, opções são negociadas segundo o modelo americano\*.

O investidor deve estar atendo aos riscos de mercado. Apesar de opções proporcionar

---

<sup>1</sup>Fisher Black já havia falecido

maior alavancagem. É possível que se perca todo o capital investido. Uma opção de compra deverá só deverá ser exercida quando o preço de mercado for superior ao preço de exercício da opção; para opção de venda, o contrário. Apenas o valor intrínseco não justifica o exercício porque deve ser levado em conta o valor atual do prêmio pago pela opção. Só quando a soma desses dois valores for maior que o preço de exercício é que o investidor estará ganhando. No Brasil, A medida provisória N. 206 de 06/08/04 regulamentou que estão livres de cobrança de impostos operações de exercício de opção e operações realizadas no mercado de balcão, com intermediação, tendo por objeto ações, ouro ativo financeiro e outros valores mobiliários, bem como as operações realizadas em mercados de liquidação futura fora da bolsa. Os ganhos líquidos auferidos em operações realizadas em bolsas de valores, de futuros, e assemelhadas, inclusive nas operações de day-trade, permanecem sujeitos à legislação vigente e são tributados segundo as seguintes alíquotas de imposto de renda:

- 20% no caso de operação de day-trade
- 15% nas demais hipóteses

Estão isentos de imposto de renda sobre o ganho de capital os ganhos líquidos auferidos por pessoa física em operações no mercado à vista de ações nas bolsas de valores, cujo o valor das alienações realizadas a cada mês seja inferior a R\$ 20000.

## 2.1 Preço a vista (Spot Price)

Os preços a vista podem ser obtidos diretamente no site do bovespa. É o preço do ativo-objeto. O sistema desenvolvido para este projeto final obtém dados diretamente do site do BOVESPA, BMF e banco central. Estes dados também são disponibilizados diariamente pelo site do bovespa. Pode ser usado também o preço de fechamento do pregão do ativo no dia anterior. Existem diversos sites que oferecem dados de preço do ativo gratuitamente. Segue uma lista:

<http://finances.yahoo.com.br>

<http://bovespa.com.br>

Esses dados também podem ser obtido através de jornais em circulação. O mais utilizado é a gazeta mercantil, jornal do comércio e valor econômico.

Algumas pessoas utilizam ferramentas de análise.

Economática

Metastock

Grafix

## 2.2 Preço de entrega ou Exercício (Strike price)

O preço de entrega é o preço no qual o titular de uma opção poderá comprar ou vender na data de vencimento. Esse preço é estabelecido e divulgado diariamente pelo Bovespa.

Nos casos de opções de compra, a data de exercício representa uma possível despesa para o titular e uma possível receita para o lançador. Já nas opções de venda; a recíproca é verdadeira.

A fim de propiciar mais geração de liquidez no mercado, a bolsa define série de opções. (Eduardo Fortuna) Cada série se caracteriza por corresponder a um tipo de ação (mesma companhia emissora, espécie, classe e forma), uma determinada data de vencimento e um preço de exercício previamente fixado. A ação que se refere é chamada de ação-objeto. As séries de opções podem ser identificadas pelo símbolo do ativo associado a uma letra.

Opção		Vencimento
Compra	Venda	
A	M	Janeiro
B	N	Fevereiro
C	O	Março
D	P	Abril
E	Q	Maio
F	R	Junho
G	S	Julho
H	T	Agosto
I	U	Setembro
J	V	Outubro
K	X	Novembro
L	Y	Dezembro

Quando se faz uma operação inversa à anteriormente realizada, ou seja, se está comprando, vende; ou se está vendendo, compra em mesma quantidade é dito que a posição está encerrada.

## 2.3 Data de maturidade, entrega ou vencimento (Expiration date)

É a data limite na qual a opção poderá ser exercida.

O exercício da opção de estilo americano pode ser efetuado a qualquer momento até a data da maturidade. O exercício da opção do estilo Europeu só pode ser exercido na data de vencimento. Sendo assim, é muito mais simples fazer estimativas no modelo de apreçamento Europeu.

## 2.4 Vencimento

As datas de vencimentos das séries são definidas pelas bolsas e variam de acordo com o ativo-objeto.[3, p.509]

No mercado de ações, os vencimentos ocorrem na segunda-feira mais próxima do dia 15, nos meses pares. As datas são fixadas pela bolsa em que as opções são negociadas. Há, portanto, vencimentos nos meses de fevereiro, abril, junho, agosto, outubro, dezembro.

No mercado de dólar os vencimentos acontecem no primeiro dia útil de cada mês; no mercado de ouro os vencimentos ocorrem na terceira sexta-feira dos meses ímpares. O número de vencimentos até os quais poderão ser negociados contratos são definidos pela bolsa, após a prévia autorização da Comissão de Valores Mobiliários a partir da **Lei no 10.303/91** a CVM passou a ser responsável pela autorização e fiscalização de todos os contratos derivativos. Por exemplo, no mercado de dólar é admitida a negociação (simultânea) de contratos até os vinte e quatro vencimentos mais próximos.

## 2.5 Volatilidade (Volatility)

É a medida de incerteza sobre os retornos proporcionados pela ação. A volatilidade pode ser calculada através do desvio padrão dos rendimentos obtidos de um determinado derivativo, também chamado de volatilidade histórica. Os dados dos rendimentos podem ser calculados diariamente com o preço de fechamento do pregão sobre um período de tempo de 3 meses. Períodos maiores podem ser usados mas podem não refletir com veracidade a situação do mercado. Hull afirma que a volatilidade do preço de uma ação pode ser definida como sendo o desvio-padrão da variação do retorno proporcionado pela ação em um ano medido em taxas logarítmicas, sendo expresso com base em capitalização

contínua.

A volatilidade pode ser estimada a partir de dados históricos com o cálculo do desvio padrão dos rendimentos. Entretanto para valor de input da fórmula de Black-Scholes devemos usar a volatilidade implícita que pode ser calculada a partir do método de Back-Scholes usando o método da bisseção. Em aplicações comerciais é melhor usar métodos mais eficientes tais como Newton-Raphson.

A escolha do valor apropriado para o número de passos não é simples. Um valor alto costuma produzir resultados melhores mas se houver muita variação da volatilidade os resultados obtidos podem não ser satisfatórios. Um procedimento que parece funcionar razoavelmente é usar os preços de fechamento diários dos mais recentes 90 a 180 dias.

Uma questão importante reside na avaliação dos dias. Devem ser usados dias úteis (cerca de 252 dias) ou todos os dias do ano (365 dias) ao se estimar o parâmetro da volatilidade? De fato, esta questão ainda não foi plenamente respondida. As duas formas podem ser usadas.

Alguns analistas defendem a hipótese que a volatilidade do preço de uma ação é motivada pela chegada aleatória de novas informações acerca dos retornos futuros de uma ação.

Fama e French testaram esta hipótese avaliando diariamente por um longo período de tempo a variância dos retornos dos preços de fechamento de um dia e o fechamento do dia seguinte quando não há dias de negociação entre eles e a variância do retorno de sexta para segunda.

Se fossem equivalentes, a variância de segunda deveria ser três vezes maior que a variância entre dois dias consecutivos. Nos estudos, Fama encontrou um resultado 22% maior; French encontrou 19%.

Esses resultados demonstram que a volatilidade é muito maior quando a bolsa está em funcionamento.

Portanto, para produzir resultados melhores, o número de dias em negociação em um ano é 252 dias. Este é o enfoque utilizado pelos operadores e que será utilizado também neste projeto Final.



## 2.6 Dividendos (Dividend Yield)

Se uma ação distribui dividendos ou qualquer outro provento. O valor líquido será deduzido do preço de exercício da série

Exemplo: Um derivativo com preço de exercício R\$ 100,00 e ocorre a distribuição de dividendos no valor de R\$ 20,00 , o preço de exercício é alterado para R\$ 80,00

## 2.7 Autorização da Negociação das Séries de Opções

Para que se inicie a negociação das opções é necessário que a CVM e a bolsa, onde as opções serão negociadas, previamente autorizem essa negociação[3]. Não é possível haver negociação de opções em determinada bolsa, sem sua autorização. Vejamos, a seguir, um exemplo de autorização à negociação de oito séries de opções da Telebrás PN

Tabela 2.1: Opções de Compra da Telebrás PN

Série	Vencimento	Preço de exercício
OTC 12	13/04/98	110.00
OTC 14	13/04/98	115.00
OTC 16	13/04/98	120.00
OTC 18	13/04/98	130.00
OTC 20	13/04/98	140.00

Tabela 2.2: Opções de Venda da Telebrás PN

Série	Preço de Exercício	Vencimento
OTC 40	115.00	13/04/98
OTC 42	120.00	13/04/98
OTC 44	130.00	13/04/98

Podemos observar que foram definidos pela bolsa a ação-objeto, a data de vencimento, o preço de exercício e a sigla de cada série.[3, p.508]

Uma vez autorizada a negociação das séries pela bolsa, como essa negociação se inicia no pregão? Para que haja negociação de uma série em pregão, é necessário que uma corretora (ou um home broker) apregoe uma ordem de compra e uma outra corretora apregoe uma ordem de venda. Quando as duas corretoras concordarem com o preço da negociação e fecharem o negócio, terá havido negociação com aquela série. A partir deste momento, o comitente (cliente da corretora ou corretora, se ela estiver operando por conta própria) que tiver comprado a série estará abrindo uma posição titular , ao passo que a corretora que tiver vendido a série estará abrindo a posição lançadora.

# Capítulo 3

## Black-Scholes

### 3.1 Movimento Browniano

[4] "Na primeira metade do século 19, o botânico inglês Robert Brown observou que, se um objeto flutuante muito pequeno (ele utilizou um grão de pólen) fosse colocado em um recipiente com água em repouso, ele se movimentaria aleatoriamente, sem que fosse possível prever em que direção ou em qual sentido o objeto se deslocaria. Observou, ainda que, se fosse utilizado um objeto flutuante de tamanho razoável, como um pequeno pedaço de madeira ou um pequeno pedaço de borracha ele permaneceria parado. Diversos pesquisadores estudaram, ao longo dos anos seguintes, o fenômeno observado por Brown e conseguiram achar explicações lógicas para as observações que, aparentemente não faziam sentido. Os pesquisadores concluíram que as diferenças de comportamento dos objetos se devem ao fato de que, quando se utiliza um objeto maior, os choques de moléculas recebidos pelo objeto tende a ser próximos em termos percentuais, o que faz com que o objeto permaneça parado. Quando se utiliza um objeto muito pequeno, há probabilidade razoável de que os choques de moléculas recebidos por um lado do objeto sejam, em termos percentuais, significativamente superiores ou inferiores aos choques recebidos pelo lado oposto, gerando deslocamento. Estas tendências são confirmadas pela Lei dos Grandes Números, segundo a qual, quando se aumenta o número de repetições de um experimento aleatório, a frequência relativa observada de um determinado resultado tende para a probabilidade de ocorrência do resultado. Portanto, se os objetos tem a mesma esperança de recebimento de choques por área de superfície, por dois lados opostos, com determinada variância, a probabilidade de que os choques de um lado superem os choques de outro em termos percentuais tende para zero à medida que se aumentam as áreas de superfície dos objetos. É importante ressaltar que mesmo que os objetos tendem a ter objetos

*aleatórios, a esperança de movimento é igual a zero. (considerando-se que a água esteja em repouso), já que há igual probabilidade de o objeto mover-se em qualquer direção ou sentido. Podemos fazer um paralelo entre a variação aleatória no preço de um ativo que possua determinada esperança de variação (em geral os ativos possuem uma esperança de retorno positiva) e o movimento aleatório do grão de pólen. Se considerarmos que a água sobre a qual o objeto está flutuando tem um leve deslocamento em determinado sentido, passamos a ter uma esperança de deslocamento do objeto (que é análoga a esperança de variação do preço do ativo) igual ao deslocamento da água, permanecendo, entretanto, uma aleatoriedade em seu movimento. Ao longo de muitas décadas, uma grande gama de pesquisa e de modelos de física foram desenvolvidos, com objeto de estudar movimentos aleatórios. Os pesquisadores da área de Finanças se utilizaram desses estudos para avaliar e modelar os movimentos aleatórios de preços."*

## 3.2 Equação Diferencial de Black-Scholes

O Modelo de Movimento Browniano pode ser estendido e utilizado como modelo de indeterminação de preço de ação. Em 1900, Bachelier, em seu famoso trabalho intitulado 'la theorie de la speculation' propõe o movimento Browniano como modelo de flutuação dos preços de um ativo. Suponha que seja o preço no tempo  $t$  de um ativo que não gera dividendos. Assumimos, por simplicidade que

$S_0^0$  Uma carteira de investimento de valor R\$1,00

$S_t^0 = e^{rt}$  representa a taxa de juros contínuo sobre um tempo  $t$

Nós modelamos utilizando um processo de difusão.

$$dS_t = rS_t dt + b(S_t, t)S_t dW_t \quad (3.1)$$

$S_t$  é o preço de um ativo no tempo  $t$ ;

$m$  é o coeficiente acumulado, que neste caso é o retorno esperado do ativo;

$b(S_t, t)$  representa algum processo de difusão. Onde  $b(0, t) = 0$  (Se um ativo atingir 0 ele nunca mais sairá dessa região);

$Z_t$  é um processo Wiener.  $N(0, 1)$ ;

A equação acima pode ser convertida facilmente em:

$$dS_t = rS_t dt + \sigma S_t dW_t \quad (3.2)$$

ou

$$dS = rS dt + \sigma S dW \quad (3.3)$$

Onde  $m$  e  $\sigma$  são as taxas constantes de juros e volatilidade.

Embora pareça a primeira vista que, a utilização de volatilidade constante possa ser irrealista, veremos que ele conduz a um modelo bem robusto [5, p. 6]. A suposição de volatilidade constante é razoável, num instante inicial, para uma primeira aproximação. Isso significa que a variancia do retorno percentual num período curto de tempo  $,dt$ , é a mesma, sem levar em consideração o preço de um ativo. A equação 3.3 pode ser integrada na forma fechada.

Vamos introduzir uma nova variável  $x$ :

$$x = f(S) = \ln(s) \quad (3.4)$$

Utilizando o Lemma de Itô, sabemos que qualquer função  $f$  de  $S$  conduz a um processo de difusão:

$$df = \left( \frac{df}{dt} + mS \frac{df}{dS} + \sigma^2 S^2 \frac{d^2}{dS^2} \right) dt + \sigma \frac{df}{dS} dz \quad (3.5)$$

Neste caso uma função logarítmica:

$$dx = \left( m - \frac{\sigma^2}{2} \right) dt + \sigma dz \quad (3.6)$$

ou

$$dx = \mu dt + \sigma dz$$

onde  $\mu = m - \frac{\sigma^2}{2}$

Então, o o log do preço de um ativo que segue o Movimento Geométrico Browniano com coeficiente acumulado<sup>1</sup>  $\mu = m - \frac{\sigma^2}{2}$

e coeficiente de difusão  $\sigma$ . Esta Equação Diferencia pode ser integrada que nos leva a:

---

<sup>1</sup>drift rate

$$x_T = x + \left(m - \frac{\sigma^2}{2}\right)\tau + \sigma\sqrt{\tau}\varepsilon_T \quad (3.7)$$

Onde nós fizemos uso do fato de que:

$$dz = \varepsilon\sqrt{dt}, T - t \quad (3.8)$$

e  $\varepsilon$  é o desvio padrão *normal*. Então, desde que  $x = \ln S$ , então:

$$\ln \frac{S_T}{S} = \left(m - \frac{\sigma^2}{2}\right)\tau + \sigma\sqrt{\tau}\varepsilon_T \quad (3.9)$$

ou

$$S_T = S\left\{\left(m - \frac{\sigma^2}{2}\right)\tau + \sigma\sqrt{\tau}\varepsilon_T\right\} \quad (3.10)$$

Esta é a fórmula fechada da solução da equação diferencial.

Nós podemos agora encontrar a função probabilidade densidade, dado que  $x$  em  $t$  é normalmente distribuído:

$$x_T \sim N\left(x + \left(m - \frac{\sigma^2}{2}\right)(T - t), \sigma\sqrt{T - t}\right) \quad (3.11)$$

ou

$$p(x_T, T|x, t)dx_T = \frac{1}{\sqrt{2\pi\sigma^2\tau}} \exp\left\{-\frac{(x_T - x - \mu\tau)^2}{2\sigma^2\tau}\right\} dx_T \quad (3.12)$$

Onde  $\mu = m - \sigma^2/2$

$$\ln S_T \sim N(\ln S + \mu(T - t), \sigma\sqrt{T - t}) \quad (3.13)$$

ou

$$p(x_T, T|x, t)dx_T = \frac{1}{\sqrt{2\pi\sigma^2\tau}} \exp\left\{-\frac{\left(\frac{S_T}{S} - \mu\tau\right)^2}{2\sigma^2\tau}\right\} \frac{dS_T}{dS} \quad (3.14)$$

Observe que  $dx_T = \frac{dS_T}{dS}$

Agora temos um modelo de precificação de opções sobre ações. Considere um derivativo  $f = F(S_T)$  no tempo T e que segue um movimento browniano geométrico. Considere ainda que  $\Pi$  contendo a posição short em uma opção e uma posição vendida em  $\Delta$  ações. Logo:

$$\Pi = \Delta S - f \quad (3.15)$$

Do lema de Itô, a mudança no valor do portfolio é dada por:

$$d\Pi = \Delta S - df = \Delta(mSdt + \sigma Sdz) - \left(\frac{df}{dt} + mS\frac{df}{dS} + \sigma^2 S^2 \frac{d^2 f}{dS^2}\right)dt - \sigma \frac{df}{dS} dz \quad (3.16)$$

Considere:

$$\Delta = \frac{df}{dS} \quad (3.17)$$

Esta escolha faz o portfolio instantaneamente livre de risco para um período infinitesimal  $dt$

$$d\Pi = r\Pi dt \Rightarrow \frac{d}{dt} dt = dt \quad (3.18)$$

Logo, juntando 3.18, 3.17 e 3.16, temos:

$$\frac{\partial f}{\partial t} + \frac{1}{2}\sigma^2 S^2 \frac{\partial^2 f}{\partial S^2} + r\Pi = 0 \quad (3.19)$$

ou

$$\frac{1}{2}\sigma^2 S^2 \frac{\partial^2 f}{\partial S^2} + rS \frac{\partial f}{\partial S} - rf = \frac{\partial f}{\partial t} \quad (3.20)$$

Conseqüentemente, o preço  $f$  de uma opção deve satisfazer esta equação diferencial como consequência da hipótese de não-arbitragem e volatilidade constante. **Esta é a equação de Black-Scholes**, do ponto de vista matemático, é uma equação de difusão (Heat Equation). [5, p.10]

### 3.2.1 Hipóteses assumidas

O Modelo de Black-Scholes, conforme podemos observar na seção anterior, admite as seguintes hipóteses:

1. O preço da ação possui densidade probabilidade lognormal, e, conseqüentemente, a função densidade da probabilidade do retorno médio medidos em taxas logarítmicas

é normal

2. O retorno da ação é igual a taxa de juro, e a volatilidade do retorno é constante até o vencimento das opções
3. É permitida a venda a descoberto de todo o tipo de título. Esta hipótese é análoga à de ser possível alugar um ativo a custo zero e vendê-lo no mercado à vista
4. Não existem custo de transação ou impostos. Os títulos são perfeitamente divisíveis
5. Não há dividendos durante a vida da opção
6. Não existem oportunidades de arbitragem sem risco. Esta hipótese equivale a admitir que o mercado é eficiente
7. A negociação sobre títulos é contínua

# Capítulo 4

## Árvore Binomial

Podemos discretizar a equação de Black-Scholes para tentar encontrar a solução. Uma técnica muito comum é utilizar o Modelo de apreçamento *lattice* ou por árvore binomial.

### 4.1 Modelo Binomial de um passo

Considere este exemplo. O preço de uma ação R\$ 20, ao final de três meses, será US\$ 18 ou R\$22. Considere que um investidor deseja lançar uma opção de compra(CALL) Européia no mercado que dá direito a comprar uma ação por R\$21.

Ao final de três meses, duas situações são possíveis: Observe, que depois de 3 meses os preços podem ser R\$18 ou R\$22 mas o preço de entrega, R\$ 21, não muda. O Titular só irá exercer a opção se for vantajoso para ele. Isto significa dizer que o preço da ação deve ser maior que R\$21 na data de entrega para que opção seja exercida. Da mesma forma, se for menor que R\$21 ela não será exercida. O lançador, assume os riscos e está submetido a vontade do titular deste ativo-objeto. Após 3 meses, o titular pode exercer ou não a

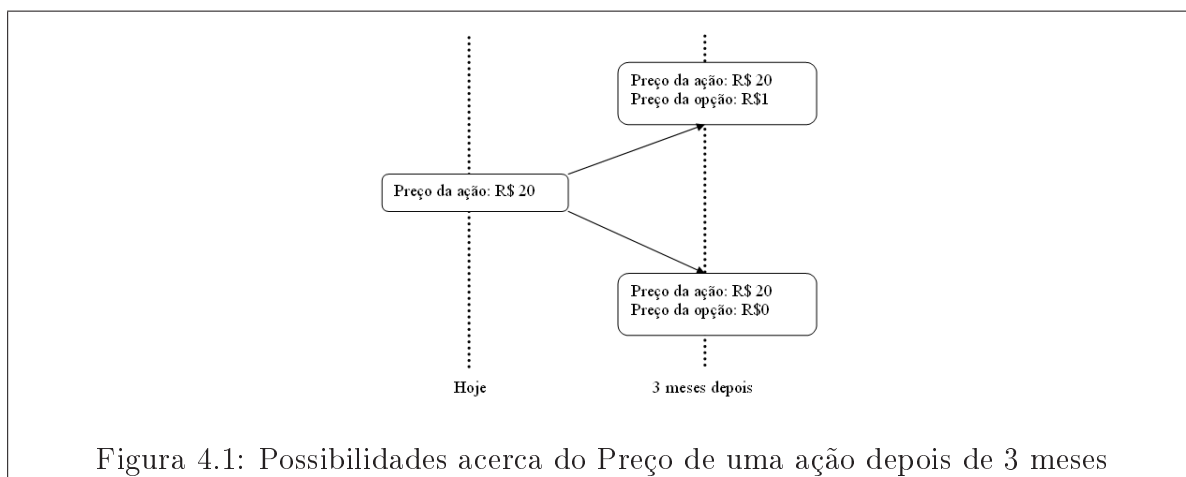


Figura 4.1: Possibilidades acerca do Preço de uma ação depois de 3 meses



opção, se ele exercer, o lançador tem que vender o ativo-objeto pelo preço estabelecido, que no exemplo foi R\$21. Perceba que esta situação oferece a possibilidade de cobertura de risco através de estratégias opostas.

Quantas ações são necessárias para cobrir o risco de lançar 1 opção de compra<sup>1</sup> ?

Considere  $\Delta$  o número de ações, logo:

$$22\Delta - 1 = 18\Delta$$

ou

$\Delta = 0.25$  Logo, para cobrir o risco da opção é necessário comprar 0.25 ações no mercado a vista.<sup>2</sup>

O que nós acabamos de fazer foi hedgear o risco. Não importa se a ação subir para 22 ou descer para 18, o lançador estará seguro que o risco com o exercício da opção estará coberto.

Observe:

Se o preço subir para R\$ 22, o valor do portfólio será:

$$22x0.25 - 1 \Rightarrow 4,5 \quad (4.1)$$

ou

$$18x0.25 \Rightarrow 4,5 \quad (4.2)$$

O valor do portfólio será sempre 4,5 independente do movimento da ação

Portfólios livres de risco devem na ausência de arbitragem render a taxa de juros livre de risco. Suponha que o valor da taxa de juros livre de risco seja 10% ao ano. O valor presente do portfólio será:

$$4.5 \exp -0.10x3/12 = 4.3670 \quad (4.3)$$

Isso significa que o lançador deve vender a opção de compra por:

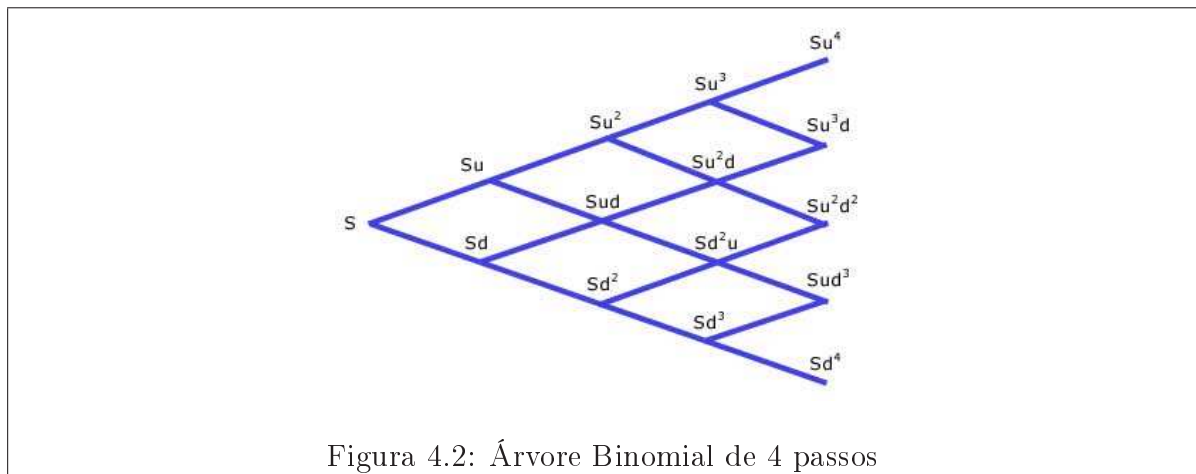
$$5 - f = 4.3670f = 0.633 \quad (4.4)$$

A opção deve ser vendida por 0.633 para atingir o break even. A condição que se elimina o risco.

Se o valor da opção for maior que 0.633, o portfólio custaria menos que 4.367 e renderia mais que a taxa de juros livre de risco. Se o valor da opção fosse menor que 0.633, ficar short no portfólio proporcionaria um meio de tomar dinheiro emprestado a uma taxa

<sup>1</sup>Perceba também que lançadores de opção de compra estão expostos a um risco diferente dos lançadores de Opções de Venda. O maior risco que o lançador de uma opção de compra tem é o preço do ativo-objeto enquanto que o risco do lançador de uma opção de venda é desconhecido

<sup>2</sup>Ou seja, para eliminar a fração, compre 4 opções por ação



menor que a taxa de juros livre de risco.

O exemplo acima pode ser generalizado para  $n$  passos de forma a criar uma árvore recombinante.

Existe várias formas de se construir uma árvore binomial para calcular o preço de uma opção. A mais utilizada é a de Cox-Ross-Rubinstein, que veremos a seguir.

## 4.2 Árvore de Cox-Ross-Rubinstein

Pode ser utilizada para calcular put do tipo americano. Ela discretiza o tempo em intervalos constantes denominado passo. Quanto mais passos mais preciso é o cálculo, entretanto; evidências mostram que uma árvore não oferece resultados muito melhores que 200 passos num intervalo de até um ano. Portanto, considere 200 passos um número satisfatório a quantidade de passos dessa árvore. A quantidade de passos tem ligação com a altura da árvore. Na figura anterior, vemos que a distância do topo a base é 4. Portanto a árvore anterior é de quatro passos.

Cabe apenas uma observação. Numa situação incomum em que os passos são muito grandes a ponto de

$$\sigma < |(r - q)\sqrt{\delta t}| \quad (4.5)$$

o modelo de Cox-Ross-Rubinstein dá probabilidades negativas. Uma solução seria utilizar uma árvore de Jarrow-Rudd que possui probabilidades iguais a 0.5. A desvantagem é que o cálculo das letras gregas não é exato porque os valores do ativo subjacente nas datas  $\delta t$  e  $2\delta t$  não estão mais centrados em  $S_0$  [1, p.462].

## 4.3 Opção do tipo Americano

Primeiramente calculamos algumas variáveis que serão usada em nosso problema

$$u = e^{\sigma \sqrt{\delta t}} \quad (4.6)$$

$$d = u^{-1} \quad (4.7)$$

$$p = \frac{a - d}{u - d} \quad (4.8)$$

Muitas vezes,  $a$  é denominado como taxa de crescimento (grow rate) e pode ser calculado da seguinte forma:

$$a = e^{(r-q)\delta t} \quad (4.9)$$

Para calcular uma opção PUT do tipo americano sobre uma ação que não paga dividendos dividimos em  $N$  subintervalos o tempo e calculamos para cada intervalo observando o seguinte algoritmo:

1. Calculamos os valores dos preços da ação multiplicando o valor da ação na subida por  $u$  e na descida por  $d$ . A figura 4.2 representa os valores do ativo-objeto no tempo  $t$ .
2. Calculamos o valor do preço da opção utilizando Backwardation. Fórmula 4.10

## 4.4 Árvore de Preços de Ações

A figura 4.2 mostra uma árvore completa dos preços de uma ação quando o modelo binomial é utilizado. No instante inicial  $S_0$  o preço da ação é conhecido. No momento  $\delta t$  há duas possibilidades de preços  $S_0u$  e  $S_0d$ , no instante 3 há 3 possibilidades  $S_0u^2$ ,  $S_0$  e  $S_0d^2$ , e assim consecutivamente.

## 4.5 Voltando com Backwardation

Após calcularmos os preços do início da árvore para a base. Fazemos o processo inverso, da base para o início calculando para cada nó o preço da opção. Este processo é conhecido como backwardation.

$$f_{i,j} = \{X - S_0u^i d^{i-j}, e^{-r\delta t}[pf_{up} + (1-p)f_{down}]\} \quad (4.10)$$

No segundo passo calculamos  $f_{up}$  e  $f_{down}$  da mesma forma.

O algoritmo prossegue até chegar na base da árvore. Na data de expiração, ou seja na base da árvore, calculamos da seguinte forma:

$$f_{N,j} = \text{maximo}(X - S_0 u^i d^{i-j}, 0) \quad (4.11)$$

Desta forma conseguimos calcular o preço justo de uma opção de venda americana.

## 4.6 Opção do Tipo Europeu

Para calcular uma opção do tipo Europeu basta modificar a fórmula 4.10

$$f_{i,j} = e^{-r\delta t} [p f_{up} + (1 - p) f_{down}] \quad (4.12)$$

O resto é semelhante. Não precisa de modificação.

# Capítulo 5

## Monte Carlo

A metodologia da Simulação Monte Carlo pode ser utilizada para estimar as distribuições de probabilidade das opções e esta estimativa possibilita o cálculo dos prêmios justos, de forma análoga aos modelos binomial e de BlackScholes. A simulação de Monte Carlo também permite o cálculo do *Value at Risk* das opções.[3, p.895].

O Modelo Binomial e a metodologia da Simulação de Monte Carlo, comparativamente ao Modelo de BlackScholes, apresentam maior grau de liberdade para adoção de diversas hipóteses que influenciam a precificação das opções, por exemplo:

- O modelo Binomial permite precificar *calls* e *puts* do tipo americano. A metodologia de Monte Carlo também permite precificar *calls* e *puts* do tipo americano a partir da simulação de árvores de preços, mas pode apresentar restrições do ponto de vista operacional quando o número de períodos é muito elevado. O Modelo de BlackScholes precifica *calls* do tipo americano apenas quando estas têm prêmios iguais às do tipo Europeu;
- Permite precificar *calls* e *puts* sobre títulos de renda fixa prefixados e sobre títulos de renda fixa pós-fixados;
- Possibilitam admitir diferentes tipos de probabilidades de haver a alta ou a baixa no preço do ativo-objeto;
- Possibilitam considerar que a volatilidade do ativo-objeto seja diferenciada ao longo do tempo;
- Permite considerar que as taxas de juros do mercado de renda fixa sejam variáveis ao longo do tempo, o que possibilita incluir a estrutura das taxas de juros a termo;

- Tornam possível considerar que a esperança de retorno do ativo-objeto seja variável ao longo do tempo, possibilitando incluir a trajetória prevista para o ativo objeto pelos mercados a termo e futuros;
- Permite o cálculo dos prêmios justos de opções sobre ativos-objetos que paguem dividendos como um valor monetário constante;
- Permite calcular a cotação justa de estratégias conjuntas que envolvam mais de uma opção (ou que envolvam mais de uma opção e o ativo-objeto), por meio de apenas um processo de precificação. Por exemplo, o modelo Binomial e a metodologia de simulação de Monte Carlo permite calcular a cotação justa (diferença líquida entre os prêmios) de uma *butterfly*. O Modelo de BlackScholes necessitaria calcular o prêmio justo de cada opção individualmente, para que fosse possível calcular a diferença líquida justa entre os prêmios, que seria a cotação justa do *butterfly*.
- Possibilita admitir que a volatilidade do ativo-objeto varie de forma aleatória ao longo dos períodos (simulação com volatilidade estocástica). Neste caso é necessário admitir uma volatilidade para a volatilidade, sendo possível considerar ainda, uma correlação entre a volatilidade e o preço do ativo-objeto[3, p.896]
- Permite estimar as distribuições de probabilidade das estratégias conjuntas que envolvam mais de uma opção ou que envolvam mais de uma opção e o ativo-objeto.

A grande vantagem do Modelo de BlackScholes é por ser uma fórmula de aplicação direta.

## 5.1 Metodologia de Monte Carlo para precificação de opções

O Método de precificação de opções do Modelo Binomial utiliza a função de probabilidade dos possíveis preços do ativo-objeto na data de vencimento, para precificar opções do tipo europeu, as funções de probabilidade dos possíveis preços do ativo-objeto até a data de vencimento, para precificar as opções do tipo americano.

De forma análoga, a metodologia de Simulação de Monte Carlo utiliza os milhares de preços do ativo-objeto na data de vencimento, obtidos por meio dos milhares de simulações, para precificar opções do tipo europeu.

Conhecendo-se as características do ativo-objeto efetuam-se milhares de simulações dos

seus possíveis preços. Para calcular o prêmio justo de uma opção do tipo Europeu basta calcular o valor presente da média aritmética dos valores intrínsecos na data de vencimento da opção.

Para precificar opções do tipo americano, é necessário simular milhares de árvores de preços, sendo que os preços de cada árvore são obtidos de forma aleatória, de acordo com os valores previamente estabelecidos.

Devido a este fato pode-se perceber que, enquanto nas árvores binomiais tradicionais o número possíveis de preços do ativo-objeto aumenta linearmente com o tempo, enquanto que nas árvores de preços simuladas o número de possíveis preços do ativo-objeto aumenta exponencialmente com o tempo, no caso de opções do tipo americano. Isto pode representar um custo operacional para a utilização de árvores de preços simuladas. Se a opção for do tipo americano torna-se necessário avaliar se é vantajoso ou não o exercício antecipado, ao longo de milhares de árvores de preços simuladas, ou seja, é necessário obter milhares de preços de calls e de puts do tipo americano. O prêmio justo será a média aritmética dos prêmios obtidos nas milhares de simulações.

## 5.2 Geração das Cadeias de Markov para Simulação de Monte Carlo

Os caminhos desenvolvidos na árvore denominam-se cadeias de Markov <sup>1</sup> Se observarmos o código perceberemos a existência de uma variável denominada *probability*. Esta variável pode ser interpretada como a probabilidade de subida e  $1 - probability$  como a probabilidade de descida. Através do algoritmo Mersenne Twister geramos centenas de milhares de séries aleatórias e caminhamos do topo para a base da árvore obtendo o valor intrínseco da opção. Segundo a Lei dos Grandes Números efetuamos milhares de simulações, os prêmios justos obtidos a partir das simulações irão convergir para o prêmio justo obtido com a utilização do modelo binomial, quando este considera um número muito elevado de períodos.

---

<sup>1</sup>Cadeia de Markov é um caminho aleatório sem memória

# Capítulo 6

## Análise e Metodologia

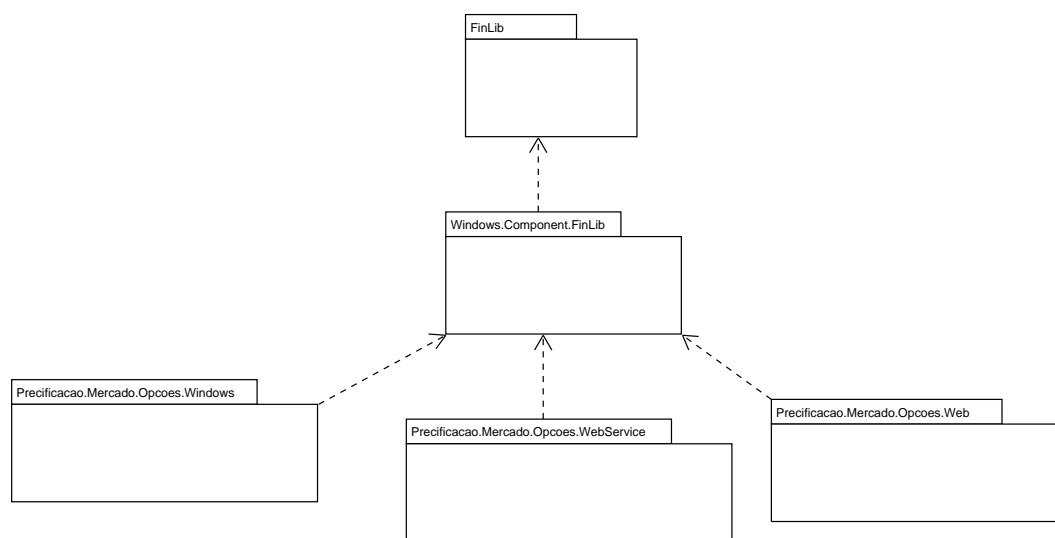


Diagrama de Packages

Primeiramente foi realizada uma pesquisa sobre o assunto, a fim de adquirir material e conhecimento necessários para a realização do projeto. Os métodos foram implementados em linguagem C++ e possui um front-end em C#. O Compilador utilizado foi o Microsoft Visual Studio .NET 2005 Framework 2.0

Duas funcionalidades são especialmente importante, cálculo do prêmio sobre uma ação que não gera dividendos do tipo americano e europeu. Estes modelos podem ser calculados para opção de compra ou de venda.



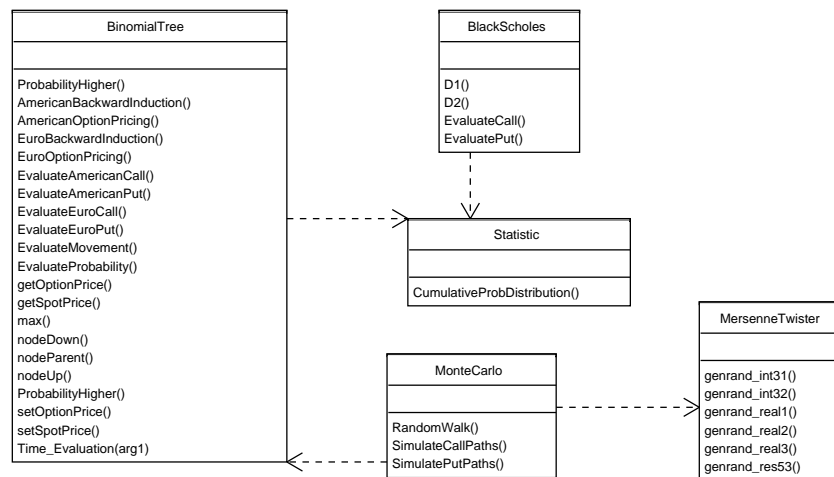
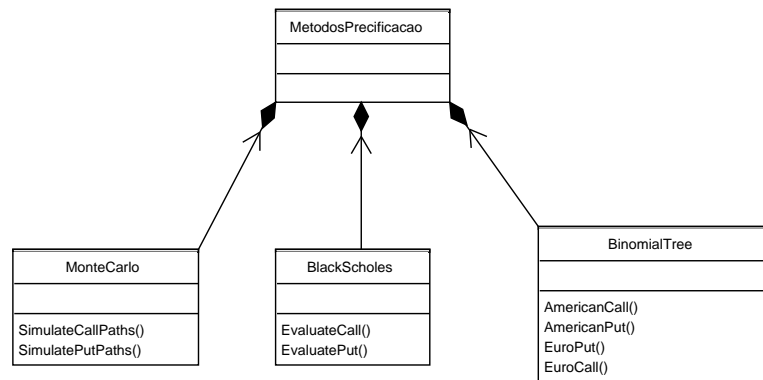


Diagrama de Classe - Módulo FinLib



O primeiro Modelo implementado foi o de Black-Scholes sobre ações que não geram dividendos. A razão disso é que Black-Scholes serve de referencial para outros métodos na medida que ele fornece soluções ótimas para ações do tipo Europeu. Para implementação da árvore Binomial foi utilizado a estrutura de dados vetor pela flexibilidade e desempenho no percurso e nos cálculos. O tamanho da árvore Binomial pode ser calculado pelo número de passos:

$$tamanho = ((passos + 1) * (passos + 2))/2; \quad (6.1)$$

A vantagem de utilizar um vetor e não uma matriz como muitas implementações disponíveis na internet utilizam está no fato de que uma matriz precisaria de uma quantidade bem maior de memória, num total de:

$$tamanho = passos^2 \quad (6.2)$$

Portanto seriam desperdiçados um total de:

$$totaldesperdiado = (passos^2)/2 - 1.5passos - 1 \quad (6.3)$$

Na verdade, o armazenamento não é necessário para o cálculo da árvore Binomial. Uma abordagem que utiliza estrutura de dados recursivas com memória igual ao tamanho do números de passos poderia ser utilizada para fornecer o resultado.<sup>1</sup> Entretanto, a desvantagem é que não teríamos disponível massa de dados para o cálculo do método de Monte Carlo ou qualquer outro método que aperfeiçoe o resultado calculado. Portanto, várias execuções do mesmo método poderiam ser utilizadas. Como hoje qualquer máquina dispõe de grandes recursos no que diz respeito a memória. Isto não se torna um problema sensível. A dificuldade reside mesmo em aplicações móveis tais como aparelhos celulares ou palmtops. Nestes tipos de hardware poderia ser empregado um algoritmo que utiliza uma quantidade menor de memória em detrimento do desempenho. Métodos que não gerem árvore recombinantes devem sempre ser evitados pela razão da densidade da árvore. Uma árvore não-recombinante possui densidade  $2^n$  que inviabilizaria o cálculo até mesmo em computadores mais potentes devido a complexidade logarítmica. Existem no total 5 classes principais, os nomes já oferecem informação suficiente sobre sua finalidade:

BlackScholes

BinomialTree

MonteCarlo

Statistic

MersenneTwister

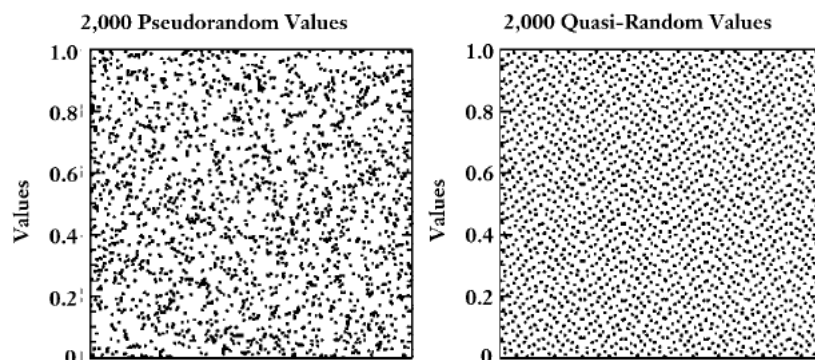
O código que implementa o algoritmo do Mersenne Twister foi obtido do site oficial do autor e encontra-se sobre livre distribuição. É notadamente um dos melhores algoritmos para geração de séries aleatórias. Foi desenvolvido em 1997 por Makoto Matsumoto e Takuji Nishimura que se baseia numa matriz linear de recorrência sobre um campo binário finito. Ele oferece qualidade muito alta na geração de números pseudo-aleatórios e foi modelado especialmente para retificar problemas que haviam em outros algoritmos. Seu nome é derivado do fato de que o tamanho do período é um número primo de Mersene. Há quatro variantes do algoritmo, diferindo apenas no tamanho da palavra utilizada. Algumas vantagens que poderiam ser citadas:

- Ele foi desenvolvido para ter um período colossal de  $2^{19937} - 1$  (Os criadores do algoritmo provaram esta propriedade). Na prática, a maioria das aplicações não requerem um número tão alto de combinações

---

<sup>1</sup>O apêndice mostra uma implementação utilizando recursividade

- Ele possui uma distribuição de alta ordem (Baixa granularidade)



Seqüência quasi-random ou geradas com o algoritmo Mersenne Twister apresentam melhores distribuições em comparação com as seqüências pseudo-randômica, repare que nestas últimas existe a formação de "ilhas". Em aplicações comerciais não use o random do C ou o aleatório do Excell pois estes produzem seqüências pseudo-randômicas.

- Rapidez
- Aprovado em diversos testes estatísticos

Foi também implementado diversos webservices que permite que a aplicação seja usada através da web. Um Webservice é um conjunto de funções que se comunicam através da Internet, utilizando o protocolo SOAP em formato XML. Estas funções recebem o nome de WebMethods. Um Webservice permite que dois programas se comuniquem de uma maneira tecnicamente muito semelhante à invocação de páginas Web. Pode ser utilizada na integração de sistemas e na comunicação entre aplicações diferentes. Com esta tecnologia é possível que novas aplicações possam interagir com aquelas que já existem e que sistemas desenvolvidos em plataformas diferentes sejam compatíveis.

srvBlackScholes	srvMonteCarlo	srvBinomialTree
EvaluatePut() EvaluateCall()	SimulateCallPaths() SimulatePutPaths()	AmericanCall() AmericanPut() EuroCall() EuroPut()

Webservices disponíveis no projeto FinLib

A Modelagem de Black-Scholes possui a vantagem de ser uma fórmula fechada e pode ser usada como benchmark para comparação de convergência dos métodos analíticos para opções do tipo Europeu.

# Capítulo 7

## Implementação

O programa FinLib é uma Biblioteca de Classes Financeira, resultado do projeto de final de curso de Israel Calheiros e Renato Mattos, alunos da graduação em ciência da computação da Universidade Federal Fluminense.

- Core em C++
- FrontEnd C#

O Core é um DLL gerado pelo Visual Studio .Net 2005. Ele foi todo implementado em C++ visto que esta linguagem combina desempenho na geração de código e fácil manutenibilidade.

### 7.1 Modelo Lattice

Para Modelagem da árvore Binomial foi utilizado um vetor e uma função que mapeia um dado nó da árvore no vetor. Algumas implementações disponíveis na internet utilizam matrizes pela simplicidade e desempenho, entretanto se a árvore binomial for grande, isto pode tornar-se oneroso para o sistema. O motivo consiste que enquanto uma árvore Binomial de  $N$  passos precisa de  $N * (N - 1) / 2$  nós uma árvore binomial implementada com matrizes vai necessitar de  $2^n$  que pode tornar-se computavelmente inviável para um número maior que 30. Uma abordagem que utiliza uma combinação das duas técnicas utilizando recursividade está sendo testada e parece ser uma boa solução para dispositivos móveis ou palmtops. A ideia consiste em fazer uma descida recursiva e criar um vetor para armazenar os resultados já calculados. A vantagem dessa abordagem é que o vetor só precisa ter no máximo um tamanho igual a  $N-1$  e permite aumentar os passos para

um número tão grande como o vetor. Entretanto esta abordagem não é eficiente para utilização combinada da técnica de Monte Carlo.

## 7.2 Modelo Black-Scholes

A implementação de Black-Scholes pode ser encontrada no arquivo BlackScholes.cpp e possui duas funções que realizam a precificação de uma opção do tipo europeu (EvaluateEuroCall e EvaluateEuroPut). Essas duas funções fornecem o resultado da fórmula fechada de Black-Scholes. É indiscutível que é o modelo mais usado para precificação de opção do tipo Europeu. A fórmula fechada possui boa precisão e a distribuição normal é feita através de uma classe que oferece uma aproximação muito boa da distribuição probabilidade logarítmica. Ela utiliza a classe Statistic para o cálculo da função probabilidade distributiva acumulativa e oferece de forma imediata a solução ótima supondo um mundo risk neutral. Para opções que gerem dividendos pode ser descontado do valor presente do portfólio o valor bruto dos rendimentos produzidos pelo dividendo sob um ativo subjacente. Também é possível encontrar uma boa aproximação para opções do tipo americano utilizando o procedimento de aproximação sugerido por Black [1, p.308].

## 7.3 BinomialTree.h

```
//*****
// C++ HEADER $RCSfile: BinomialTree.h,v $
//
// $Header: d:_repositorio/FinLib/Include/BinomialTree.h,v 1.1 2006/09/11 23:41:04
israel Exp $
//
// Description:
//
//
// Contents:
//
//
// *****

// Copyright (c) 2006 by Israel Calheiros *
// Home page: www.israelcalheiros.com E-mail: isracalheiros@yahoo.com.br *
// You are free to distribute and modify this file, as long as you do not *
// remove this copyright notice clearly label modified versions as being modified.*
// This software has NO WARRANTY. Use at your risk. *
//*****

#ifdef __BINOMIAL_TREE_H
#define __BINOMIAL_TREE_H
#include <iostream>
#include "definitions.h"

typedef struct SNODE {
```

```
double spotPrice;

double optionPrice;

} tnode;

using std::ostream;

using std::istream;

class BinomialTree{

    friend ostream &operator<<( ostream &, const BinomialTree &b);

public:

    bool BinomialTree::ProbabilityHigher( double value);

    ~BinomialTree() ;

    virtual double EvaluateEuroCall(double price, double strikePrice, double days,
double rate, double vol, double dividend=0, double ex_dividend_date=0);

    virtual double EvaluateEuroPut(double price, double strikePrice, double days,
double rate, double vol, double dividend=0 ,double ex_dividend_date=0);

    virtual double EvaluateAmericanPut ( double price, double strikePrice, double
days, double rate, double vol , double dividend=0, double ex_dividend_date=0);

    virtual double EvaluateAmericanCall ( double price, double strikePrice, double
days, double rate, double vol , double dividend=0, double ex_dividend_date=0);

    void setOptionPrice( const long int node, const double value);

    double getOptionPrice( const long int node ) const;

    double getTime() const { return T; };
```

```
    double getStrike() const { return strike;}

    double getRate() const { return rate;}

    double getVolatility() const { return volatility; }

    double getDividendYield() const { return dividend;}

    int getOptionType() const { return option_type;}

    void setSpotPrice( const long int node, const double value) const;

    double getSpotPrice( const long int node ) const;

    BinomialTree( const int steps=100, int days_by_year=365 );

    long int nodeParent( const long int node ) const;

    long int nodeUp( const long int node ) const;

    long int nodeDown( const long int node) const;

    virtual void EvaluateMovement();

    virtual int EvaluateProbability();

protected:

    tnode* pnode;

    long int size;

    long int height;

    double up;

    double down;
```



```
    double dt;

    double rate;

    double probability;

double volatility;

double dividend;

double dividend_date;

double T;

double price;

double strike;

double a;

int option_type;

private:

    int* exercise;

//int hasNegativeProbability();

    double EuroOptionPricing( double price, double strikePrice, double days,
double rate, double vol, double dividend=0.0, double ex_dividend_date=0.0 );

    double AmericanOptionPricing( double price, double strikePrice, double days,
double rate, double vol, double dividend=0.0, double ex_dividend_date=0.0 );

    inline void Time_Evaluation();

    void EuroBackwardInduction();

    void AmericanBackwardInduction();

    inline int max( double a, double b);
```

```
};

#endif

//-----
// Modification Log
//
// $Log: BinomialTree.h,v $
// Revision 1.1  2006/09/11 23:41:04  israel
// *** empty log message ***
//
// Revision 1.8  2006/09/11 04:20:32  israel
// *** empty log message ***
//
// Revision 1.7  2006/08/30 12:15:41  israel
// no message
//
// Revision 1.6  2006/08/26 12:43:42  israel
// no message
//
// Revision 1.5  2006/08/26 01:58:08  israel
// separacao das letras gregas do modelo de arvore binomial
//
// Revision 1.4  2006/08/25 14:36:25  israel
// *** empty log message ***
//
// Revision 1.3  2006/08/25 14:27:44  israel
// laioute e padronizacao do codigo
```

---

```
//
```

```
//
```

```
//-----
```

## 7.4 BinomialTree.cpp

```

//*****
// C++ Implementation: $RCSfile: BinomialTree.cpp,v $
//
// $Header: d:_repositorio/FinLib/src/BinomialTree.cpp,v 1.1 2006/09/11 23:41:23
israel Exp $
//
// Description:
// -----
// |           Cox-Ross-Rubinstein Binomial Model
// -----
// This source implements Cox-Ross-Rubinstein Binomial Tree.
// The stock price can move "up" with probability "p" and "down" with
// probability "1-p" (Hull, page 204). We know that stocks price grows
// on average at risk-free rate so:
// option_price = option_price_UP * ( probability_UP) + option_price_DOWN
// ( probability_DOWN)
//
//           ...
//           /
//           option_price_up
//           /
// option_price           ..
//           /
//           option_price_down
//           ...
//

```

```

// We assume that: ( up * down = 1 )
//
// up = exponential( volatility * square_root( dt ) )
// down = 1 / up;
//
// We discount this value "option_price" back to get the present value
// present_value = option_price * exp( - rate * dt )
//
// The objective of the analysis is to calculate the option price
// at the initial node of the tree. This can be done by repeatedly
// applying the principles established earlier in the chapter. (Hull, page 206)
//
// -----
// |                SOURCE                |
// -----
//
// This algorithm intends to reproduce this Binomial Tree.
// The Tree was simulated using array. This save computing time and memory.
//
//
//          ( 7 )
//          /
//          ( 4 )
//          /
//
//          ( 2 )    ( 8 )
//          /        /

```

```
// ( 1 )    ( 5 )
//          /

//      ( 3 )    ( 9 )
//          /

//          ( 6 )
//

//          ( 10 )
//

// Array was represented like:
//
// [ 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10]
//
// Each position has two components:
// - spotPrice (in that time)
// - optionPrice
// The evaluation of the option price has two steps
// 1) Evaluate Up and down price evaluation on that specific time
// 2) Backward Induction with the option price
//
// References:
// John Hull, Options, Futures & Others Derivatives, 4. Ed
// John Hull, Mercados Futuros e de Opções, 4. Ed.
// Justin London, Modeling Derivatives in C++ - Wiley finances series
//
// Contents:
//
```

```
//  
//  
// *****  
  
// Copyright (c) 2006 by Israel Calheiros *  
// Home page: www.israelcalheiros.com E-mail: isracalheiros@yahoo.com.br *  
// You are free to distribute and modify this file, as long as you do not *  
// remove this copyright notice clearly label modified versions as being modified.*  
// This software has NO WARRANTY. Use at your risk. *  
//*****  
  
#include <math.h>  
#include <time.h>  
#include <stdlib.h>  
#include <stdio.h>  
#include "BinomialTree.h"  
#include "Statistic.h"  
  
//-----  
// EvaluateMovement()  
// Exeption: N/A  
// Description: Up and down movements  
//-----  
  
void BinomialTree::EvaluateMovement(){  
  
up = exp(volatility * sqrt( dt ));  
  
down = 1/up;
```

```

}

//-----
// EvaluateProbability()
// Exeption: N/A
// Description:
//-----

int BinomialTree::EvaluateProbability(){
if ( ( dividend_date == 0 ) && ( dividend > 0 ) ) {
a = exp( ( rate - dividend ) * dt );
} else a = exp( rate * dt );

probability = ( a - down)/( up - down );

// Hull nao altera os valores de subida e descida com dividendos com rendimento
conhecido

// Wiley subtrai dividend_yield da taxa livre de juros para calculo de "a", isso altera
// sensivelmente as probabilidades. Descomente abaixo para seguir a linha de Wiley
( e comente acima )

// probability = ( exp( ( rate - dividend ) * dt ) - down)/( up - down ); // Wiley

return true;
}

//-----
// ProbabilityHigher( IN value)
// value - Evaluate if the probability is higher than value
// Exeption: N/A
// Description:

```



```

//-----
bool BinomialTree::ProbabilityHigher( double value){
if ( probability > value ) { return true; }
else return false;
}

//-----
// nodeParent( IN node )
// node - number of the parent node
// Exeption: N/A
// Description:
//-----
ulong BinomialTree::nodeParent( const ulong node ) const {
ulong n;

n = node - ( long int ) ( ceil( ( -1 + sqrt( double(1 + 8* ( node-1 ) ) ) )/2 ));

return n;
}

//-----
// nodeUp( IN node )
// node - upper node on tree
// Exeption: N/A
// Description:
//-----
ulong BinomialTree::nodeUp( const ulong node ) const {
ulong n;

```

```

n = node + ( long int ) ( ceil( ( -1 + sqrt( double(1 + 8* node ) ) )/2 ) );

if ( n > size ) return -node;

return n;

}

//-----
// nodeDown( const ulong node)
// node - down node on Tree
// Exeption: N/a
// Description:
//-----

ulong BinomialTree::nodeDown( const ulong node) const {
ulong n;

n = node + ( long int ) ( ceil( ( -1 + sqrt( double(1 + 8* node ) ) )/2 ) ) + 1;

if ( n > size ) return -node;

return n;

}

BinomialTree::~BinomialTree( ){
delete( exercise );
delete( pNode );
}

```

```
//-----  
// BinomialTree( IN steps )  
// Exeption: N/A  
// Description:  
//-----  
BinomialTree::BinomialTree( const int steps, int days_by_year )  
{  
    height = steps + 1;  
  
    size = ( (steps+1)*(steps+2) ) / 2;  
  
    pnode = new SNODE[ size + 1 ];  
  
    dt=0;  
    probability = 0;  
    rate=0;  
    volatility= 0;  
    dividend =0;  
    dividend_date=0;  
    exercise = NULL;  
}  
  
//-----  
// getSpotPrice(const along node)  
// Exeption: N/A  
// Description:  
//-----  
double BinomialTree::getSpotPrice(const along node) const
```

```

{
if ( node <= size ) {
return pnode[ node ].spotPrice;
} else return -1;
}

//-----
// setSpotPrice(IN node, IN value)
// node - Node that you want put the value
// value - Spot Price Value
// Exeption: N/A
// Description:
//-----

void BinomialTree::setSpotPrice(const ulong node, const double value) const
{
pnode[ node ].spotPrice = value;
}

//-----
// getPrice(IN node)
// node - Number of the Node on Tree
// Exeption: N/A
// Description:
//-----

double BinomialTree::getPrice(const ulong node) const
{
if ( node <= size ) {

```

```
    return pnode[ node ].optionPrice;
} else return -1;
}

//-----
// setOptionPrice(IN node, IN value)
// Exeption: N/A
// Description:
//-----

void BinomialTree::setOptionPrice(const ulong node, const double value)
{
pnode[ node ].optionPrice = value;
}

//-----
// max( IN a, IN b)
// Exeption: N/A
// Description:
//-----

inline int BinomialTree::max( double a, double b){
if ( ( a - b ) > 0 ) {
return 0;
} else return 1;
}

//-----
// Time_Evaluation( )
```

```

// Exeption: N/A

// Description:

//-----

inline void BinomialTree::Time_Evaluation( ){

// Inicializa os valores possiveis do ativo durante intervalo de tempo "time"

long int i,j;

long int sum;

double value;

double _div;

    sum = 0;

if ( ( dividend_date > 0 ) && ( dividend > 0 ) ) {

// Com dividendos conhecido em valor

// __S = price - dividend * exp( -rate * dividend_date);

for ( i=1; i <= height ; i++ ){

for ( j=1; j <= i ; j++ ){

// _div = dividend * exp( -rate * dividend_date);

    _div = price - dividend * exp( -rate * ( dividend_date ) );

if ( ( i - 1 ) * dt >= dividend_date ){

    _div = price - dividend * exp( -rate * ( dividend_date - i*dt ) );

    pnode[ sum + j ].spotPrice = _div * pow( double(down),int(j - 1) ) * pow( double(up), int(i-j) );

    } else {

        pnode[ sum + j ].spotPrice = _div * pow( double(down),int(j - 1) ) * pow( double(up), int(i-j) ) + dividend * exp( - rate * ( dividend_date - (i-1) * dt ) );

    }

    }

}
}

```

```

sum +=i;

}

pnode[1].spotPrice = price;

for ( j=height; j > 0 ; j-- ){

if ( option_type == OPTION_CALL ) {
value =  pnode[ sum - j + 1 ].spotPrice - strike;
} else {
value =  strike - pnode[ sum - j + 1 ].spotPrice;
}

//pnode[ sum - j + 1 ].optionPrice = value;
if ( ( value > 0 ) ) { pnode[ sum - j + 1 ].optionPrice = value; }
else pnode[ sum - j + 1 ].optionPrice = 0;
} //for

} else {

for ( i=1; i <= height ; i++ ){

for ( j=1; j <= i ; j++ ){

pnode[ sum + j ].spotPrice = price * pow( double(down),int(j - 1) ) * pow( double(up), int(i-j) );

}

sum +=i;

}

pnode[1].spotPrice = price;

for ( j=height; j > 0 ; j-- ){

```

```

if ( option_type == OPTION_CALL ) {
value =  pnode[ sum - j + 1 ].spotPrice - strike;
} else {
value =  strike - pnode[ sum - j + 1 ].spotPrice;
}

//pnode[ sum - j + 1 ].optionPrice = value;

if ( ( value > 0 ) ) { pnode[ sum - j + 1 ].optionPrice = value; }
else pnode[ sum - j + 1 ].optionPrice = 0;
} //for
}

//-----
// AmericanBackwardInduction( )
// Exeption: N/A
// Description:
//-----

void BinomialTree::AmericanBackwardInduction( ){
// backward induction

long int sum;

    double excPrice;

    double optPrice;

sum = size - height;

for ( long int i= height - 1; i >= 1 ; i-- ){

for ( long int j = sum; j > ( sum - i ) ; j-- ){

```



```
    optPrice = ( probability * ( pnode[ j + i ].optionPrice) + ( 1- probability)*(
pnode[ j + i + 1].optionPrice) ) * exp( -rate * dt );
```

```
    if ( option_type == OPTION_PUT ) {
excPrice = strike - pnode[ j ].spotPrice;
    } else {
excPrice = pnode[ j ].spotPrice - strike;
    }

    exercise[ j ] = max ( optPrice, excPrice );

    if ( exercise[ j ] ) {
pnode[ j ].optionPrice = excPrice;
    }
    else {
pnode[ j ].optionPrice = optPrice;
    }
}

sum = sum - i;
}
}
```

```
//-----
// EuroBackwardInduction()
// Exeption: N/A
// Description:
```

```

//-----
void BinomialTree::EuroBackwardInduction(){
    ulong sum;

    double continuous_compound; // continuously compounded rate

    sum = size - height;

    continuous_compound = exp( -rate * dt );

    for ( long int i= height - 1; i >= 1 ; i-- ){

        for ( long int j = sum; j > ( sum - i ) ; j-- ){

            pnode[ j ].optionPrice = ( probability * ( pnode[ j + i ].optionPrice) + ( 1- probabilit-
ity)*( pnode[ j + i + 1].optionPrice) ) * continuous_compound ;

        }

        sum = sum - i;

    }

}

//-----
// EuroOptionPricing( IN price, IN strikePrice, IN T, IN rate, IN vol, IN dividend,
IN expiration_date )
// Exeption: N/A
// Description:
//-----

double BinomialTree::EuroOptionPricing( double price, double strikePrice, dou-
ble T, double rate, double vol,double dividend, double ex_didivend_date ){

```

```
    ulong sum;

    this->dividend = dividend;

    this->dividend_date = ex_dividend_date;

    sum=0;

    this->T = T;

    dt = T / ( height - 1 );

    this->price=price;

    strike = strikePrice;

        this->rate = rate;

    volatility = vol;

    EvaluateMovement();

    EvaluateProbability();

        Time_Evaluation( );

        EuroBackwardInduction();

    return pnode[ 1 ].optionPrice;

}

//-----
// AmericanOptionPricing( IN price, IN strikePrice, IN T, IN rate, IN vol, IN dividend, IN expiration_date )
```

```
// Exeption: N/A
// Description:
//-----

double BinomialTree::AmericanOptionPricing( double price, double strikePrice,
double T, double rate, double vol, double dividend, double ex_dividend_date ){
    ulong sum;

    if ( exercise == NULL){

    exercise = new int[size+1];

    }

    this->dividend = dividend;

    dividend_date = ex_dividend_date;

    this->T = T;

    sum=0;

    dt = T / ( height - 1 );

    this->price=price;

    strike = strikePrice;

    this->rate = rate;

    volatility = vol;

    EvaluateMovement();

    EvaluateProbability();
```

```

    Time_Evaluation( );

    AmericanBackwardInduction( );

    return pnode[ 1 ].optionPrice;

}

/*int BinomialTree::hasNegativeProbability(){
if ( dividend_date == 0.0 ) {
if ( volatility < fabs( rate - dividend ) ){
throw ""
}

} else {
return volatility < fabs( rate ) * sqrt( dt );
}
}
*/

//-----

// EvaluateEuroCall(IN price, IN strikePrice, IN T, IN rate, IN vol, IN dividend, IN
expiration_date)

// Exeption: N/A

// Description:

//-----

    double BinomialTree::EvaluateEuroCall(double price, double strikePrice, double
T, double rate, double vol, double dividend, double ex_dividend_date)
    {
        option_type = OPTION_CALL;

        return EuroOptionPricing( price, strikePrice, T, rate, vol, dividend, ex_dividend_date);
    }
}

```

```

    }

    //-----
    // EvaluateEuroPut(IN price, IN strikePrice, IN T, IN rate, IN vol, IN dividend, IN
    expiration_date)
    // Exeption: N/A
    // Description:
    //-----

    double BinomialTree::EvaluateEuroPut(double price, double strikePrice, double
    T, double rate, double vol, double dividend, double ex_dividend_date)
    {

    option_type = OPTION_PUT;

    return EuroOptionPricing( price, strikePrice, T, rate, vol, dividend, ex_dividend_date);

    }

    //-----
    // EvaluateAmericanPut(IN price, IN strikePrice, IN T, IN rate, IN vol, IN dividend,
    IN expiration_date)
    // Exeption: N/A
    // Description:
    //-----

    double BinomialTree::EvaluateAmericanPut(double price, double strikePrice, double
    T, double rate, double vol, double dividend, double ex_dividend_date)
    {

    option_type = OPTION_PUT;

    return AmericanOptionPricing( price, strikePrice, T, rate, vol, dividend, ex_dividend_date);

    }

```

```

//-----
// EvaluateAmericanCall(IN price, IN strikePrice, IN T, IN rate, IN vol, IN dividend,
IN expiration_date)
// Exeption: N/A
// Description:
//-----

double BinomialTree::EvaluateAmericanCall(double price, double strikePrice, double
double T, double rate, double vol, double dividend, double ex_dividend_date)
{

option_type = OPTION_CALL;

    return AmericanOptionPricing( price, strikePrice,T,rate,vol,dividend,ex_dividend_date);

}

//-----
// Modification Log
//
// $Log: BinomialTree.cpp,v $
// Revision 1.1 2006/09/11 23:41:23 israel
// *** empty log message ***
//
// Revision 1.13 2006/09/11 15:25:15 israel
// *** empty log message ***
//
// Revision 1.12 2006/09/11 13:14:45 israel

```

```
// *** empty log message ***  
  
//  
// Revision 1.11 2006/09/11 04:20:15 israel  
// *** empty log message ***  
  
//  
// Revision 1.10 2006/08/30 23:42:10 israel  
  
// no message  
  
//  
// Revision 1.9 2006/08/30 12:15:36 israel  
  
// no message  
  
//  
// Revision 1.8 2006/08/26 12:43:49 israel  
  
// no message  
  
//  
// Revision 1.7 2006/08/26 01:58:07 israel  
  
// separacao das letras gregas do modelo de arvore binomial  
  
//  
// Revision 1.6 2006/08/25 14:36:25 israel  
  
// *** empty log message ***  
  
//  
//  
//-----
```



## 7.5 BlackScholes.h

```

//*****
// C++ HEADER $RCSfile: BlackScholes.h,v $
//
// $Header: d:_repositorio/FinLib/Include/BlackScholes.h,v 1.1 2006/09/11 23:41:04
israel Exp $
//
// Description:
//
//
// Contents:
//
//
// *****

// Copyright (c) 2006 by Israel Calheiros *
// Home page: www.israelcalheiros.com E-mail: isracalheiros@yahoo.com.br *
// You are free to distribute and modify this file, as long as you do not *
// remove this copyright notice clearly label modified versions as being modified.*
// This software has NO WARRANTY. Use at your risk. *
//*****

#ifdef __BLACKSCHOLES_H
#define __BLACKSCHOLES_H

class BlackScholes
{
public:

```

```
virtual double EvaluateCall( double spot, double strike, double T, double rate,
double volatility);
```

```
virtual double EvaluatePut( double spot, double strike, double T, double rate,
double volatility);
```

```
double D1( double spot, double strike, double T, double rate, double volatility
);
```

```
double D2( double spot, double strike, double T, double rate, double volatility
);
```

```
BlackScholes();
```

```
virtual ~BlackScholes();
```

```
};
```

```
#endif
```

```
//-----
```

```
// Modification Log
```

```
//
```

```
// $Log: BlackScholes.h,v $
```

```
// Revision 1.1 2006/09/11 23:41:04 israel
```

```
// *** empty log message ***
```

```
//
```

```
// Revision 1.3 2006/08/25 14:36:25 israel
```

```
// *** empty log message ***
```

```
//
```

```
// Revision 1.2 2006/08/25 14:27:44 israel
```

```
// laioute e padronizacao do codigo
```

```
//
```

```
//
```

```
//-----
```

## 7.6 BlackScholes.cpp

```
//*****  
  
//  
// C++ Implementation: $RCSfile: BlackScholes.cpp,v $  
//  
// $Header: d:_repositorio/FinLib/src/BlackScholes.cpp,v 1.1 2006/09/11 23:41:23  
israel Exp $  
  
//  
// Description:  
//  
//  
// Contents:  
//  
//  
//*****  
  
// Copyright (c) 2006 by Israel Calheiros *  
// Home page: www.israelcalheiros.com E-mail: isracalheiros@yahoo.com.br *  
// You are free to distribute and modify this file, as long as you do not *  
// remove this copyright notice clearly label modified versions as being modified.*  
// This software has NO WARRANTY. Use at your risk. *  
//*****  
  
#include <math.h>  
  
#include "definitions.h"  
  
#include "BlackScholes.h"  
  
#include "statistic.h"
```

```
BlackScholes::BlackScholes()
{
}

BlackScholes::~BlackScholes()
{
}

double BlackScholes::D1( double spot, double strike, double T, double rate,
double volatility ){
    double d1;

    d1= ( log( spot / strike ) + ( rate + ( pow( volatility, 2 ) / 2 ) ) * T ) / ( volatility
* sqrt( T ) ) ;

    return d1;
}

double BlackScholes::D2( double spot, double strike, double T, double rate,
double volatility ){
    double d2;

    d2 = ( log( spot / strike ) + ( rate - ( pow( volatility, 2 ) / 2 ) ) * T ) / ( volatility *
sqrt( T ) ) ;

    return d2;
}

double BlackScholes::EvaluatePut( double spot, double strike, double T, double
rate, double volatility )
```

```
{  
// [0001]  
double put, d1, d2;  
  
d1 = D1( spot, strike, T, rate, volatility);  
  
d2 = D2( spot, strike, T, rate, volatility);  
  
put = strike* exp( - rate*T)*( Statistic::CumulativeProbDistribution ( -d2 ) ) - spot  
* ( Statistic::CumulativeProbDistribution( -d1 ) ) ;  
  
return put;  
}  
  
double BlackScholes::EvaluateCall(double spot, double strike, double T, double  
rate, double volatility)  
{  
// [0002]  
double call, d1, d2;  
  
d1 = D1( spot, strike, T, rate, volatility);  
  
d2 = D2( spot, strike, T, rate, volatility);  
  
call = spot * ( Statistic::CumulativeProbDistribution ( d1 ) ) - strike* exp(- rate*T)*(  
Statistic::CumulativeProbDistribution ( d2 ) ) ;  
  
return call;  
}  
  
//-----  
// Modification Log
```

```
//  
// $Log: BlackScholes.cpp,v $  
// Revision 1.1 2006/09/11 23:41:23 israel  
// *** empty log message ***  
//  
// Revision 1.2 2006/08/25 14:36:25 israel  
// *** empty log message ***  
//  
//  
//-----
```

## 7.7 MonteCarlo.h

```
//*****
// C++ HEADER $RCSfile: MonteCarlo.h,v $
//
// $Header: d:_repositorio/FinLib/Include/MonteCarlo.h,v 1.1 2006/09/11 23:41:04
israel Exp $
//
// Description:
//
//
// Contents:
//
//
// *****

// Copyright (c) 2006 by Israel Calheiros *
// Home page: www.israelcalheiros.com E-mail: isracalheiros@yahoo.com.br *
// You are free to distribute and modify this file, as long as you do not *
// remove this copyright notice clearly label modified versions as being modified.*
// This software has NO WARRANTY. Use at your risk. *
//*****

#ifdef __MONTE_CARLO_H
#define __MONTE_CARLO_H

class MonteCarlo: public BinomialTree {

public:

double average;
```



```
MonteCarlo( unsigned int depth, unsigned long steps);

double SimulateCallPaths (double price, double strikePrice, double T, double
rate, double vol, int option_type );

double SimulatePutPaths (double price, double strikePrice, double T, double
rate, double vol, int option_type );

private:

    unsigned long m_steps;

inline void RandomWalk();

};

#endif

//-----
// Modification Log
//
// $Log: MonteCarlo.h,v $
// Revision 1.1 2006/09/11 23:41:04 israel
// *** empty log message ***
//
// Revision 1.3 2006/08/25 14:36:25 israel
// *** empty log message ***
//
// Revision 1.2 2006/08/25 14:27:44 israel
// laioue e padronizacao do codigo
//
//
//-----
```

## 7.8 MonteCarlo.cpp

```
//*****  
  
//  
// C++ Implementation: $RCSfile: MonteCarlo.cpp,v $  
//  
// $Header: d:_repositorio/FinLib/src/MonteCarlo.cpp,v 1.1 2006/09/11 23:41:23  
israel Exp $  
  
//  
// Description:  
//  
//  
// Contents:  
//  
//  
//*****  
  
// Copyright (c) 2006 by Israel Calheiros *  
// Home page: www.israelcalheiros.com E-mail: isracalheiros@yahoo.com.br *  
// You are free to distribute and modify this file, as long as you do not *  
// remove this copyright notice clearly label modified versions as being modified.*  
// This software has NO WARRANTY. Use at your risk. *  
//*****  
  
#include "BinomialTree.h"  
  
#include "MersenneTwister.h"  
  
#include "MonteCarlo.h"  
  
#include <stdlib.h>
```

```
#include <time.h>
```

```
#include <math.h>
```

```
MonteCarlo::MonteCarlo( unsigned int depth, unsigned long steps ) : Binomial-  
Tree( depth ){
```

```
    m_steps=steps;
```

```
}
```

```
inline void MonteCarlo::RandomWalk(){
```

```
double random_seed;
```

```
int node;
```

```
double r;
```

```
average = 0;
```

```
srand( (unsigned)time( NULL ) );
```

```
random_seed = rand() + 1;
```

```
MersenneTwister mTwister( random_seed );
```

```
for ( unsigned long i = 0 ; i < m_steps ; i++){
```

```
node = 1;
```

```
while ( node > 0 ) {
```

```
r = mTwister.genrand_real1();
```

```
if ( ProbabilityHigher( r ) ){
```

```
node = nodeUp( node );
```

```
} else {
```

```
node = nodeDown( node );
```

```
}
```

```
    }//while

    average = average + getOptionPrice( -node );

    }//for

    average /= m_steps;

}

    double MonteCarlo::SimulateCallPaths (double price, double strikePrice, double
T, double rate, double vol, int option_type ){

    BinomialTree::EvaluateEuroCall( price, strikePrice, T, rate, vol );

    RandomWalk();

    average = average * exp(-rate * T );

    return average;

}

    double MonteCarlo::SimulatePutPaths (double price, double strikePrice, double
T, double rate, double vol, int option_type ){

    BinomialTree::EvaluateEuroPut( price, strikePrice, T, rate, vol );

    RandomWalk();

    average = average * exp(-rate * T );

    return average;

}
```

```
//-----  
// Modification Log  
//  
// $Log: MonteCarlo.cpp,v $  
// Revision 1.1 2006/09/11 23:41:23 israel  
// *** empty log message ***  
//  
// Revision 1.3 2006/08/26 00:24:17 israel  
// no message  
//  
// Revision 1.2 2006/08/25 14:36:25 israel  
// *** empty log message ***  
//  
//  
//-----
```

## 7.9 Statistic.h

```

//*****

// C++ HEADER $RCSfile: Statistic.h,v $

//

// $Header: d:_repositorio/FinLib/Include/Statistic.h,v 1.1 2006/09/11 23:41:04 is-
rael Exp $

//

// Description:

//

//

// Contents:

//

//

// *****

// Copyright (c) 2006 by Israel Calheiros *

// Home page: www.israelcalheiros.com E-mail: isracalheiros@yahoo.com.br *

// You are free to distribute and modify this file, as long as you do not *

// remove this copyright notice clearly label modified versions as being modified.*

// This software has NO WARRANTY. Use at your risk. *

//*****

#ifdef __STATISTICS_H

#define __STATISTICS_H

class Statistic

{

public:

```

```
static double CumulativeProbDistribution( double x );
Statistic();
virtual ~Statistic();

};

#endif

//-----
// Modification Log
//
// $Log: Statistic.h,v $
// Revision 1.1  2006/09/11 23:41:04  israel
// *** empty log message ***
//
// Revision 1.3  2006/08/25 14:36:25  israel
// *** empty log message ***
//
// Revision 1.2  2006/08/25 14:27:44  israel
// laioute e padronizacao do codigo
//
//
//-----
```

## 7.10 Statistic.cpp

```
#include <math.h>

#include "Statistic.h"
#include "definitions.h"

#define N2( x ) ( 1 / (sqrt(2* PI) ) * exp ( - pow( x,2) / 2.0 ) )

#define K( x ) ( 1.0 / ( 1 + g * x ) )

#define N( x ) ( 1 - N2( x )*( a1* ( K( x ) ) + a2* ( pow( K( x ), 2) ) + a3* (
pow( K( x ), 3) ) + a4* ( pow( K( x ), 4) ) + a5* ( pow( K( x ), 5) ) ) )

const double g = 0.2316419;
const double a1 = 0.319381530;
const double a2 = -0.356563782;
const double a3 = 1.781477937;
const double a4 = -1.821255978;
const double a5 = 1.330274429;

Statistic::Statistic()
{
}

Statistic::~Statistic()
{
}

double Statistic::CumulativeProbDistribution( double x )
```



```
{  
  
double c;  
  
    if ( x >= 0 ) {  
        c = N ( x );  
    } else {  
        c = 1 - N( -x );  
    }  
  
return c;  
}  
  
//-----  
// Modification Log  
//  
// $Log: Statistic.cpp,v $  
// Revision 1.1  2006/09/11 23:41:23  israel  
// *** empty log message ***  
//  
// Revision 1.3  2006/08/30 00:55:08  israel  
// Correcao Bug na distribuicao cumulativa  
//  
// Revision 1.2  2006/08/25 14:36:25  israel  
// *** empty log message ***  
//  
//  
//-----
```

## 7.11 MersenneTwister.h

```
//*****
// C++ HEADER $RCSfile: MersenneTwister.h,v $
//
// $Header: d:_repositorio/FinLib/Include/MersenneTwister.h,v 1.1 2006/09/11 23:41:04
israel Exp $
//
// Description:
//
//
// Contents:
//
//
// *****

// Copyright (c) 2006 by Israel Calheiros *
// Home page: www.israelcalheiros.com E-mail: isracalheiros@yahoo.com.br *
// You are free to distribute and modify this file, as long as you do not *
// remove this copyright notice clearly label modified versions as being modified.*
// This software has NO WARRANTY. Use at your risk. *
//*****

#ifdef __MERSENNE_TWISTER_H
#define __MERSENNE_TWISTER_H

class MersenneTwister {

public:
MersenneTwister(unsigned long s);
```

```
MersenneTwister(unsigned long init_key[], int key_length);

unsigned long genrand_int32(void);

long genrand_int31(void);

double genrand_real1(void);

double genrand_real2(void);

double genrand_real3(void);

double genrand_res53(void);

private:

};

#ifdef

//-----

// Modification Log

//

// $Log: MersenneTwister.h,v $

// Revision 1.1 2006/09/11 23:41:04 israel

// *** empty log message ***

//

// Revision 1.3 2006/08/25 14:36:25 israel

// *** empty log message ***

//

// Revision 1.2 2006/08/25 14:27:44 israel

// laioue e padronizacao do codigo
```

```
//
```

```
//
```

```
//-----
```

## 7.12 MersenneTwister.cpp

```
//*****
//
// C++ Implementation: $RCSfile: MersenneTwister.cpp,v $
//
// $Header: d:_repositorio/FinLib/src/MersenneTwister.cpp,v 1.1 2006/09/11 23:41:23
israel Exp $
//
// Description:
//
//
// Contents:
//
//
//*****

// Copyright (c) 2006 by Israel Calheiros *
// Home page: www.israelcalheiros.com E-mail: isracalheiros@yahoo.com.br *
// You are free to distribute and modify this file, as long as you do not *
// remove this copyright notice clearly label modified versions as being modified.*
// This software has NO WARRANTY. Use at your risk. *
//*****

/*
A C-program for MT19937, with initialization improved 2002/1/26.
Coded by Takuji Nishimura and Makoto Matsumoto.

Before using, initialize the state by using init_genrand(seed)
```

or `init_by_array(init_key, key_length)`.

Copyright (C) 1997 - 2002, Makoto Matsumoto and Takuji Nishimura,

All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
3. The names of its contributors may not be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS

"AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT

LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR

A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR

CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL,

EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO,

PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE,  
 DATA, OR  
 PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON  
 ANY THEORY OF  
 LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (IN-  
 CLUDING  
 NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE  
 OF THIS  
 SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Any feedback is very welcome.

<http://www.math.sci.hiroshima-u.ac.jp/~m-mat/MT/emt.html>

email: m-mat @ math.sci.hiroshima-u.ac.jp (remove space)

\*/

**#include "MersenneTwister.h"**

**#include <stdio.h>**

*/\* Period parameters \*/*

**#define N 624**

**#define M 397**

**#define MATRIX\_A 0x9908b0dfUL** */\* constant vector a \*/*

**#define UPPER\_MASK 0x80000000UL** */\* most significant w-r bits \*/*

**#define LOWER\_MASK 0x7fffffffUL** */\* least significant r bits \*/*

**static unsigned long mt[N];** */\* the array for the state vector \*/*

**static int mti=N+1;** */\* mti==N+1 means mt[N] is not initialized \*/*

*/\* initializes mt[N] with a seed \*/*

```

MersenneTwister::MersenneTwister(unsigned long s)
{
    mt[0]= s & 0xffffffffUL;
    for (mti=1; mti<N; mti++) {
        mt[mti] =
(1812433253UL * (mt[mti-1] ^ (mt[mti-1] >> 30)) + mti);
        /* See Knuth TAOCP Vol2. 3rd Ed. P.106 for multiplier. */
        /* In the previous versions, MSBs of the seed affect */
        /* only MSBs of the array mt[]. */
        /* 2002/01/09 modified by Makoto Matsumoto */
        mt[mti] &= 0xffffffffUL;
        /* for >32 bit machines */
    }
}

/* initialize by an array with array-length */
/* init_key is the array for initializing keys */
/* key_length is its length */
/* slight change for C++, 2004/2/26 */
MersenneTwister::MersenneTwister(unsigned long init_key[], int key_length)
{
    int i, j, k;
    MersenneTwister(19650218UL);
    i=1; j=0;
    k = (N>key_length ? N : key_length);
    for (; k; k--) {
        mt[i] = (mt[i] ^ ((mt[i-1] ^ (mt[i-1] >> 30)) * 1664525UL))

```



```

        + init_key[j] + j; /* non linear */
    mt[i] ^= 0xffffffffUL; /* for WORDSIZE > 32 machines */
    i++; j++;
    if (i >= N) { mt[0] = mt[N-1]; i=1; }
    if (j >= key_length) j=0;
}

for (k=N-1; k; k--) {
    mt[i] = (mt[i] ^ ((mt[i-1] ^ (mt[i-1] >> 30)) * 1566083941UL))
        - i; /* non linear */
    mt[i] ^= 0xffffffffUL; /* for WORDSIZE > 32 machines */
    i++;
    if (i >= N) { mt[0] = mt[N-1]; i=1; }
}

mt[0] = 0x80000000UL; /* MSB is 1; assuring non-zero initial array */
}

/* generates a random number on [0,0xffffffff]-interval */
unsigned long MersenneTwister::genrand_int32(void)
{
    unsigned long y;

    static unsigned long mag01[2]={0x0UL, MATRIX_A};
    /* mag01[x] = x * MATRIX_A for x=0,1 */

    if (mti >= N) { /* generate N words at one time */
        int kk;

        if (mti == N+1) /* if init_genrand() has not been called, */
            MersenneTwister(5489UL); /* a default initial seed is used */

```

```

    for (kk=0;kk<N-M;kk++) {
        y = (mt[kk]&UPPER_MASK)|(mt[kk+1]&LOWER_MASK);
        mt[kk] = mt[kk+M] ^ (y >> 1) ^ mag01[y & 0x1UL];
    }
    for (;kk<N-1;kk++) {
        y = (mt[kk]&UPPER_MASK)|(mt[kk+1]&LOWER_MASK);
        mt[kk] = mt[kk+(M-N)] ^ (y >> 1) ^ mag01[y & 0x1UL];
    }
    y = (mt[N-1]&UPPER_MASK)|(mt[0]&LOWER_MASK);
    mt[N-1] = mt[M-1] ^ (y >> 1) ^ mag01[y & 0x1UL];

    mti = 0;
}

y = mt[mti++];

/* Tempering */
y ^= (y >> 11);
y ^= (y << 7) & 0x9d2c5680UL;
y ^= (y << 15) & 0xefc60000UL;
y ^= (y >> 18);

return y;
}

/* generates a random number on [0,0x7fffffff]-interval */
long MersenneTwister::genrand_int31(void)
{
    return (long)(genrand_int32()>>1);
}

```

```
}

/* generates a random number on [0,1]-real-interval */
double MersenneTwister::genrand_real1(void)
{
    return genrand_int32()*(1.0/4294967295.0);
    /* divided by 2^32-1 */
}

/* generates a random number on [0,1)-real-interval */
double MersenneTwister::genrand_real2(void)
{
    return genrand_int32()*(1.0/4294967296.0);
    /* divided by 2^32 */
}

/* generates a random number on (0,1)-real-interval */
double MersenneTwister::genrand_real3(void)
{
    return (((double)genrand_int32()) + 0.5)*(1.0/4294967296.0);
    /* divided by 2^32 */
}

/* generates a random number on [0,1) with 53-bit resolution*/
double MersenneTwister::genrand_res53(void)
{
    unsigned long a=genrand_int32()>>5, b=genrand_int32()>>6;
    return(a*67108864.0+b)*(1.0/9007199254740992.0);
}
```

```
/* These real versions are due to Isaku Wada, 2002/01/09 added */  
  
//-----  
// Modification Log  
//  
// $Log: MersenneTwister.cpp,v $  
// Revision 1.1 2006/09/11 23:41:23 israel  
// *** empty log message ***  
//  
// Revision 1.2 2006/08/25 14:36:25 israel  
// *** empty log message ***  
//  
//  
//-----
```

# Capítulo 8

## Testes

O plano de teste encontram-se disponível na pasta Teste. Ele foi feito a partir da comparação com a fórmula fechada e com outros programas e os resultados foram excelentes. Ele conseguiu passar por todos os ciclos de teste sem apresentar qualquer inconsistência com pelo menos 95% de tolerância para 2000 passos no caos da árvore binomial americana.

Os resultados do sistema FinLib foram comparados com os seguintes sistemas:

Foram utilizados 3 máquinas para realização dos testes:

1. Processador Intel Dual Core 1.66 ram RAM 1GB
2. Processador AMD 64 bits com RAM 512 MB
3. Processador AMD Athlon 64 bits RAM com 512 MB

E os softwares utilizados para comparação ds testes são:

- MatLab 5
- Maple 10
- Planilha do Excell anexo ao livro [1]
- website <http://www.tradingtoday.com/black-scholes>
- website <http://www.hoadley.net/>

Estes resultados fazem parte do documento de evidências e encontram-se disponíveis para consulta na planilha de casos de teste.

# Capítulo 9

## Implantação

O Sistema FinLib foi disponibilizado gratuitamente desde 01/11/2006 na internet através do endereço:

<http://www.israelcalheiros.com>

# Capítulo 10

## Conclusão

O Método de Black-Scholes pode ser empregado para calcular opções da modalidade Europeia e CALL americana porque oferece a melhor aproximação para o valor justo de uma opção. Para calcular a PUT americana é comum empregar modelos *lattices*, ou seja, modelos que usem árvore Binomial. Para refinar os resultados da árvore Binomial podemos empregar o Método de Monte Carlo.

Características dos Modelos de Precificação	Modelo de Black-Scholes	Modelo Binomial	Metodologia de Monte Carlo
Aplicabilidade	é de aplicação direta	necessita da construção de árvore Binomiais. Softwares tornam a sua aplicação direta	necessita da execução de milhares de simulações. Software facilitam a sua execução.
Distribuições de Probabilidades Consideradas para o Ativo-Objeto	apenas a distribuição de probabilidade lognormal	há flexibilidade para a definição das características da distribuição de probabilidade	Há flexibilidade para a definição das características da distribuição de probabilidade
Outputs	prêmios justos; letras gregas; efeito sorriso das volatilidades implícitas	prêmios justos; funções de probabilidades das opções(árvores binomiais das opções)	prêmios justos; estimativa das distribuições de probabilidade das opções (por meio de simulações possíveis de prêmios)
Opções que podem ser precificadas	calls e puts do tipo europeu sobre ações, moedas, futuros, etc.	calls e puts do tipo Europeu e americano sobre os ativos-objetos do Modelo de BlackScholes: sobre ativos que pagam dividendos como o valor monetário constante e sobre ativos com volatilidades variáveis(como PU's de títulos prefixados, etc.)	calls e puts do tipo europeu e americano sobre os ativos-objetos do modelo binomial; sobre os ativos com volatilidades estocásticas, etc.
Flexibilidade	há flexibilidade razoável	há grande flexibilidade	há flexibilidade ainda maior do que no modelo binomial

# APÊNDICE A

## A.1 Lemma de Itô

Seja  $x(t)$  um itô ( ou um processo Wiener) <sup>1</sup> . Que nos leva:

$$dx(t) = a(x, t)dt + b(x, t)dW_t \quad (\text{A.1})$$

onde  $W_t$  é um processo Wiener, e deixe  $f(x, t)$  ser uma função com derivada segunda contínua.

Então  $f(x(t), t)$  é também um processo de itô, e

$$df(x(t), t) = (a(x, t)\frac{\partial f}{\partial x} + \frac{\partial f}{\partial t}b(x, t)^2\frac{\partial^2 f}{\partial x^2})dt + b(x, t)\frac{\partial f}{\partial x}dW_t \quad (\text{A.2})$$

Se considerarmos a equação dos processos estocásticos visto em 3.3 Então concluímos que  $a(x) = m\sigma$  e  $b(x) = \sigma S$

Logo, o itô ficaria

$$df = (m\sigma\frac{\partial f}{\partial S} + \frac{\partial f}{\partial t} + \sigma^2 S^2\frac{\partial^2 f}{\partial S^2})dt + m\sigma\frac{\partial f}{\partial S}dZ \quad (\text{A.3})$$

## A.2 Distribuição Normal

Segundo [3, p.764], a função densidade probabilidade sobre uma variável aleatória:

$$f(t_i) = \frac{1}{\sigma\sqrt{2\Pi}}e^{-1/2\frac{(r_t-\mu)^2}{\sigma^2}} \quad (\text{A.4})$$

Função densidade normal do retorno medido em taxas logarítmicas:

---

<sup>1</sup>Essa demonstração informal foi retirada de [6] e serve de modelo para um entendimento mais imediato visto que o procedimento de prova é mais complexo e envolve alguns conceitos mais avançados de estatística.



## A.3 Distribuição Lognormal

Pode ser demonstrado que a hipótese do caminho aleatório faz que o preço da ação, em qualquer momento futuro, tenha distribuição lognormal [1] Segundo [3, p.765], Admitindo que  $PV_t$  é uma variável aleatória contínua e positiva. Se  $PV_t = e^y$  e  $Y$  possui distribuição normal com média  $\mu$  e desvio padrão  $\sigma$ ,  $PV_t$  tem distribuição normal, com função densidade probabilidade normal:

$$f(PV_t) = 1/2 \sqrt{2} e^{-1/2 \frac{(\ln PV_t - \mu)^2}{\sigma^2}} PV_t^{-1} \sigma^{-1} \frac{1}{\sqrt{\pi}} \quad (\text{A.5})$$

Enquanto a variável com distribuição pode assumir valores positivos e negativos, a variável com distribuição lognormal só pode assumir valores positivo. A distribuição normal é simétrica enquanto que a distribuição lognormal é desviada para o lado com a média, a mediana e moda diferentes.

Pode-se demonstrar que a média e a moda  $LnS_t$ , são iguais a:

$$\ln S_0 + (\mu - 1/2 \sigma^2) T \quad (\text{A.6})$$

e o desvio padrão:

$$\sigma \sqrt{T} \quad (\text{A.7})$$

onde  $S_0$  é preço corrente da ação. O resultado fica:

$$LnS_t \sim \phi[\ln S_0 + (\mu - 1/2 \sigma^2) T, \sigma \sqrt{T}] \quad (\text{A.8})$$

Os dois parâmetros que explicam o comportamento do preço de uma ação, quando se admite que o preço da ação em data futura possui uma distribuição lognormal, são:

- retorno esperado da ação
- a volatilidade do retorno da ação

O valor esperado ou o valor médio de  $S_T$  é dado por:  $E(S_t) = S_0 e^{\mu T}$  A variância de  $S_T$ ,  $var(S_T)$  é dado por:

$$var(S_T) = S^2 e^{2\mu T} (e^{\sigma^2 T} - 1) \quad (\text{A.9})$$

## A.4 Recursivo.cpp

```

double BinomialRecTree::EuroPutOptionPricing ( double Price, long x, long y ){
    double optionPrice;

    if ( x + y <= steps - 1 ){
        optionPrice = exp( - rate * delta_t ) * ( p * EuroPutOptionPricing( Price * up,
x+1 , y ) + (1-p)* EuroPutOptionPricing( Price * down, x , y+1 ) );
        return optionPrice;

    } else {
        if ( x == steps ) {
            MinValue = max( strike - Price, 0);
            return MinValue;
        } else if ( y == steps){
            MaxValue = max( strike - Price, 0);
            return MaxValue;
        }

        return max( strike - Price, 0);
    }
}

```

```

double BinomialRecTree::EuroCallOptionPricing ( double Price, long x, long y ){
    double optionPrice;

    if ( x + y <= steps - 1 ){
        optionPrice = exp( - rate * delta_t ) * ( p * EuroCallOptionPricing( Price *
up, x+1 , y ) + (1-p)* EuroCallOptionPricing( Price * down, x , y+1 ) );
    }
}

```

```
    return optionPrice;

} else {

    if ( x == steps ) {

        MaxValue = max( Price - strike, 0);

        return MaxValue;

    } else if ( y == steps){

        MinValue = max( Price - strike, 0);

        return MinValue;

    }

    return max( Price - strike, 0);

}

}
```

# Índice Remissivo

- backwardation, 10
- bolsa
  - mercadorias e futuros, 4
- call, 4
- contango, 10
- contrato
  - a termo, 3
  - futuro, 3
- derivativo, 3
- escola
  - fundamentalista, 7
  - gráfica e técnica, 6
- lançador
  - opção de compra, 5
  - opção de venda, 5
- Lei
  - nº 10.303/91, 16
- opção
  - compra, *veja* call
  - venda, *veja* put
- positiocrn
  - short, *veja* call
- position
  - long, *veja* put
- prêmio, 4
- put, 4
- underlying asset, 3

# Referências

- [1] HULL, J. C. *Fundamentos dos Mercados Futuros e de Opções*. [S.l.]: BMF Brasil, 2005.
- [2] RUBINSTEIN, M. *Rubinstein on Derivatives*. [S.l.]: Berkeley: Risk Books, 1999.
- [3] MARTINS, A. *Mercados Derivativos e Análise de Risco Vol2*. [S.l.]: AMS Editora, 2004.
- [4] MARTINS, A. *Mercados Derivativos e Análise de Risco*. [S.l.]: AMS, 2004.
- [5] LONDON, J. *Modeling Derivatives in C++*. [S.l.]: Wiley Finance, 2004.
- [6] WIKIPEDIA. *Itô' Lemma*. <http://en.wikipedia.org/>, Setembro 2006. GNU Free Documentation License.