

Referências Bibliográficas

- [1] ARAÚJO, R. O. **Avaliação de Opções Reais Através do Método dos Mínimos Quadrados de Monte Carlo**. Dissertação de Mestrado. Departamento de Engenharia Industrial, PUC-Rio, 2004.
- [2] ASQUITH, P. **Convertible Bonds Are Not Called Late**. Journal of Finance, v. 50, n. 4, 1995, p. 1275-1289
- [3] BARDHAN, I. et al. **Valuing Convertible Bonds as Derivatives**. Quantitative Strategies Research Notes, Goldman Sachs, novembro de 1994.
- [4] BAUMOL, W. J.; MALKIEL, B. G.; QUANDT, R. E. **The Valuation of Convertible Securities**. Quarterly Journal of Economics, v. 80, 1966, 48 - 59.
- [5] BLACK, F.; **The Pricing of Complex Options and Corporate Liabilities**. Graduate School of Business Administration, University of Chicago.
- [6] BLACK, F.; SCHOLES, M. **The Pricing of Options and Corporate Liabilities**. Journal of Political Economy, v. 81, 1973, p. 637-659.
- [7] Boletim Técnico de Debêntures, quinto bimestre de 2005. Disponível em: <<http://www.debentures.com.br>>. Acesso em: 15 jan. 2005.
- [8] BRENNAN, M. J.; SCHWARTZ, E. **Convertible Bonds: Valuation and Optimal Strategies for Call and Conversion**. The Journal of Finance, v. 32, 1977, p. 1699-1715.
- _____. **A Continuous Time Approach to the Pricing of Bonds**. Journal of Banking and Finance, v. 3, 1979, p. 133-155.
- _____. **Analyzing Convertible Securities**. Journal of Financial and Quantitative Analysis, v. 15, 1980, p. 907 - 929.
- [11] BROADIE, M.; GLASSERMAN, P. **Pricing American-Style Securities using Simulation**. Journal of Economic Dynamics and Control, 21, 1997, p. 1323-1352.
- [12] BROADIE, M.; GLASSERMAN, P.; JAIN, G. **Enhanced Monte Carlo Estimates for American Option Prices**. The Journal of Derivatives, 1997.
- [13] CHEN, H.Y. **A Model of Warrant Pricing in a Dynamic Market**. Journal of Finance, v. 25, 1970, p. 1041 - 1060.
- [14] COURTADON, G. **A More Accurate Finite Difference Approximation for Valuation of Options**. Journal of Financial and Quantitative Analysis, v. 17, n. 5, 1982, p. 697-703.

- [15] COX, J.; ROSS, S.; RUBINSTEIN, M. **Option Pricing: A Simplified Approach**. Journal of Financial Economics, n. 7, 1979.
- [16] CRAIG, M. L. **Convertible Debt: Valuation and Conversion in Complex Capital Structures**. Journal of Banking and Finance, v. 15, 1991, p. 665-682.
- [17] DATTA, S.; DATTA, M. I. **New Evidence on the Valuation Effects of Convertible Bond Calls**. The Journal of Financial and Quantitative Analysis, v. 31, n. 2, 1996, p. 295-307.
- [18] DIXIT, A.; PINDYCK, R. **Investment Under Uncertainty**. Princeton University Press, Princeton, New Jersey, 1994.
- [19] FROTA, A. E. F. **Avaliação de Opções Americanas Tradicionais e Complexas**. Dissertação de Mestrado, Departamento de Engenharia Industrial, PUC-Rio, 2003.
- [20] FU, M.C.; LAPRISE, S. B.; MADAN, D. B.; SU, Y.; WU, R. **Pricing American Options: A Comparison of Monte Carlo Simulation Approaches**. Working Paper, University of Maryland at College Park, 2000.
- [21] GRAND, D.; VORA, G.; WEEKS, D. **Simulation and Early-Exercise of Option Problem**. Journal of Financial Engineering, v. 5, n. 3, 1996, p.211-227.
- _____. **Path-Dependent Options: Extending the Monte Carlo Simulation Approach**. Management Science, v. 43, n. 11, 1997, p. 1589-1602.
- [23] GRIMWOOD, RUSSELL; HODGES, STEWART. **The Valuation of Convertible Bonds: A Study of Alternative Pricing Models**. [s.n.], 2002.
- [24] HULL, J. **Options, Futures, and Other Derivatives**. Prentice Hall, Inc., Englewood Cliffs, New Jersey, 2000.
- [25] HULL, J. C.; WHITE, A. **Valuing Derivatives Securities Using the Explicit Finite Difference Method**. Journal of Financial and Quantitative Analysis, v. 25, n. 1, 1990, p. 87 -100.
- [26] JENNINGS, E.H. **An Estimation of Convertible Bonds Premiums**. Journal of Financial and Quantitative Analysis, v. 9, 1974, p. 33-56.
- [27] JEVTIC, BRANKO Z.; KALIAKATSOS, K. A. **Valuation of a Liquid Yield Option Note: A New Approach**. Working Paper, University of Southampton, 1995.
- [28] JONATHAN, E. INGERSOLL, Jr. **A Contingent-Claims Valuation of Convertible Securities**. Journal of Financial Economics, n. 4, 1976, p. 289 -321.
- [29] LEWIS, C. M. **Convertible Debt: Valuation and Conversion in Complex Capital Structures**. Journal of Banking and Finance, v. 15, 1991, p. 665 - 682.
- [30] LONGSTAFF, F. A.; SCHWARTZ, E. S. **Valuing American Options By Simulation: A Simple Least-Square Approach**. The Review of Financial Studies, v. 14, n. 1, 2001, p. 113-147
- _____. **A Simple Approach to Valuing Risky Fixed and Floating Rate Debt**. Journal of Finance, v. 50, n. 3, 1995, p. 789 - 819.

- [32] LOUIS, H E.; JEREMY C. GOH. **Is Convertible Bond Call Really Bad News?** The Journal of Business, v. 74, n. 2, 2001, p. 459-476
- [33] MARINS, JAQUELINE T. M. **Ensaio sobre a Aplicação de Técnicas de Redução de Variância em Simulação de Monte Carlo para Avaliação de Opções e de Risco de Crédito.** Tese de Doutorado, Instituto Coppead de Administração, UFRJ, 2006.
- [34] MAYERS, D. **Why Firms Issue Convertible Bonds: The Matching of Financial and Real Investment Options.** Journal of Financial Economics, v. 47, 1998, p. 83-102.
- [35] MCCONNEL, J.; SCHWARTZ, E. **LYON Taming.** The Journal of Finance, v. 41, n. 3, 1986, p. 561-577.
- _____. **The Origin of LYONs: A Case Study in Financial Innovation.** Journal of Applied Corporate Finance, v. 4, 1992, p. 40-47
- [37] MERTON, R. **The Theory of Rational Option Pricing.** Bell Journal of Economics and Management Science, v. 4, 1973, p. 141-83.
- [38] MODIGLIANI, F.; MILLER, M. H. **The Cost of Capital, Corporation Finance, and The Theory of Investment.** American Economy Review, v. 48, 1958.
- [39] NASCIMENTO, A. F. **Avaliação de Investimentos em Tecnologia da Informação: Uma Perspectiva de Opções Reais.** Dissertação de Mestrado, Departamento de Engenharia Industrial, PUC-Rio, 2005.
- [40] POENSGEN, O. H. **The Valuation of Convertible Bonds.** Industrial Management Review, v. 7, 1965, p. 83 - 98.
- [41] RAMOS, P. B. **Um Estudo dos Impactos no Mercado Acionário Decorrentes das Emissões de Debêntures e Um Caso Prático de Avaliação de Títulos Conversíveis.** Tese de Doutorado, Instituto Coppead de Administração, UFRJ, 2005.
- [42] Relatório de Inflação do Banco Central do Brasil, Dezembro de 2005, v. 7, nº 4. Disponível em: <<http://www.bacen.gov.br>>. Acesso em: 20 jan. 2005.
- [43] ROSENGRES, E. S. **Defaults of Original Issues High-Yield Convertible Bonds.** The Journal of Finance. v. 48, n. 1, 1993, p. 345-362
- [44] SAÁ-REQUEIJO, J.; SANTA-CLARA, P. **Bond Pricing with Default Risk.** Working Paper, Anderson Graduate School of Management, 1999.
- [45] SALIBY, E. **Descriptive Sampling: A Better Approach to Monte Carlo Simulation.** Journal of the Operational Research Society, v. 41, nº 12, 1990, p. 1133-1142.
- [46] SAMUELSON, P.A. **Rational Theory of Warrant Pricing.** Industrial Management Review, v. 6, 1965, p. 13 - 31
- [47] SARKAR, S. **Probability of Call and Likelihood of the Call Feature in a Corporate Bond."** Journal of Banking and Finance, v. 25, 2001, p.505-533
- [48] SHARPE, W. F.; ALEXANDER, G. J.; BAILEY, J. V. **Investments.** Prentice-Hall, 1999, 6th edition

- [49] SMITH, G.D., **Numerical Solution of Partial Differential Equations: Finite Difference Methods**. Oxford University Press, 1985.
- [50] TSANG, W. W.; MARSAGLIA, G. **The Ziggurat Method for Generating Random Variables**. Working paper. National Science Foundation.
- [51] TSIVERIOTIS, K.; FERNANDES, C. **Valuing Convertible Bonds with Credit Risk**. Journal of fixed Income, 1998. p. 95 – 102.
- [52] VOSE, DAVID. **Quantitative Risk Analysis: A Guide to Monte Carlo Simulation Modelling**. John Wiley & Sons, 1996.
- [53] WALTER, J.E.; QUE, A.V. **The Valuation of Convertible Bonds**. Journal of Finance, v. 28, 1973, p. 713 - 732.
- [54] WEIL, R.; SEGALL, J.E.; GREEN, D. JR. **Premiums on Convertible Bonds**. Journal of Finance, v. 23, 1968, p. 445 - 463.
- [55] WILMOTT, P.; HOWISON, S.; DEWYNNE, J. **The Mathematics of Financial Derivatives. A Student Introduction**. Cambridge University Press, 1996.

Apêndice 1

Derivação do Lema de Itô

Dado um derivativo qualquer, G , função das variáveis x e t , a expansão de Taylor para dG é:

$$dG = \frac{\partial G}{\partial x} dx + \frac{\partial G}{\partial t} dt + \frac{1}{2} \frac{\partial^2 G}{\partial x^2} dx^2 + \frac{1}{2} \frac{\partial^2 G}{\partial t^2} dt^2 + \frac{\partial^2 G}{\partial x \partial t} dxdt + \dots \quad (30)$$

Supondo que a variável x segue o processo de ITÔ:

$$dx = a(x,t)dt + b(x,t)dZ \quad (31)$$

Onde dZ é um processo de Wiener: $dZ = \eta\sqrt{dt}$, sendo $\eta \sim N(0,1)$

$$\text{Pode-se escrever: } dx = a dt + b dZ \quad (32)$$

Elevado a eq. 31 ao quadrado:

$$dx^2 = a^2 dt^2 + 2abdt\eta\sqrt{dt} + b^2 dZ^2 \quad (33)$$

Considerando qualquer potência de dt maior que um, igual a zero:

$$dx^2 = b^2 dZ^2 = b^2 \eta^2 dt \quad (34)$$

Encontrando o valor esperado:

$$E(dx^2) = E(b^2 dZ^2) = b^2 E(dZ^2) = b^2 E(\eta^2 dt) = b^2 dt E(\eta^2) \quad (35)$$

Sendo $\eta \sim N(0,1)$:

$$\text{Var}(\eta) = E(\eta^2) - [E(\eta)]^2 = 1 \quad (36)$$

Como $E(\eta) = 0 \therefore E(\eta^2) = 1$

Então: $E(dx^2) = b^2 dt$

Assim:

$$\text{Var}(dx^2) = \text{Var}(b^2 dZ^2) = \text{Var}(b^2 dt \eta^2) = b^4 dt^2 \text{Var}(\eta^2) = 0 \quad (37)$$

Como $\text{Var}(dx^2) = 0$, $E(dx^2) = dx^2 = b^2 dt$

$$\text{Logo: } dG = \frac{\partial G}{\partial x} (adt + bdZ) + \frac{\partial G}{\partial t} dt + \frac{1}{2} \frac{\partial^2 G}{\partial x^2} b^2 dt \quad (38)$$

Apêndice 2

Derivação da Equação Diferencial Satisfeita por um Derivativo de uma Ação que Paga Dividendo Contínuo

Sendo L , o derivativo em questão, e S o preço do ativo base:

$$L = G(S, t) \quad (39)$$

montando uma carteira sem risco:

$\phi = -L + nS \longrightarrow$ posição vendida do título e posição comprada em n ações

$$d\phi = -dL + ndS \quad (40)$$

Assumindo o processo de ITÔ:

$$dS = (\mu S - D)dt + \sigma S dZ \quad (41)$$

Onde D é a taxa de dividendos.

Ganho esperado = ganho de capital + dividendo

$$d\phi = -\left[\frac{\partial L}{\partial S} dS + \frac{\partial L}{\partial t} dt + \frac{1}{2} \frac{\partial^2 L}{\partial S^2} \sigma^2 S^2 dt \right] + ndS + nqSdt \quad (42)$$

Onde $D = q$ = taxa de dividendos

Para que o termo aleatório dZ desapareça: $n = \frac{\partial L}{\partial S}$

Assim, substituindo $n = \frac{\partial L}{\partial S}$ na eq. 42:

$$d\phi = - \left\{ \left[\frac{\partial L}{\partial S} (\mu S - D) + \frac{\partial L}{\partial t} + \frac{1}{2} \frac{\partial^2 L}{\partial S^2} \sigma^2 S^2 \right] dt + \frac{\partial L}{\partial S} \sigma S dZ \right\} + \frac{\partial L}{\partial S} [(\mu S - D)dt + \sigma S dZ] + \frac{\partial L}{\partial S} q S dt \quad (43)$$

Simplificando:

$$d\phi = \left[-\frac{\partial L}{\partial t} - \frac{1}{2} \frac{\partial^2 L}{\partial S^2} \sigma^2 S^2 \right] dt + \frac{\partial L}{\partial S} q S dt \quad (44)$$

Como o portfólio é sem risco, o ganho requerido é: $\left[r \left(-L + \frac{\partial L}{\partial S} S \right) \right] dt$

Onde r é a taxa de juros sem risco.

Igualando $d\phi$ com o ganho requerido:

$$\left[-\frac{\partial L}{\partial t} - \frac{1}{2} \frac{\partial^2 L}{\partial S^2} \sigma^2 S^2 \right] dt + \frac{\partial L}{\partial S} q S dt = r \left(-L + \frac{\partial L}{\partial S} S \right) dt \quad (45)$$

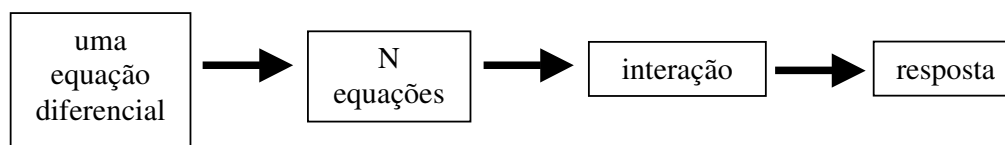
Organizando a eq. 45:

$$\frac{1}{2} \sigma^2 S^2 \frac{\partial^2 L}{\partial S^2} + (r - q) \frac{\partial L}{\partial S} S + \frac{\partial L}{\partial t} - rL = 0 \rightarrow \begin{array}{l} \text{Equação diferencial} \\ \text{satisfeita por um} \\ \text{derivativo de uma ação} \\ \text{que paga dividendos} \\ \text{contínuos} \end{array} \quad (46)$$

Apêndice 3

O Método de Diferenças Finitas Implícito

Converte-se a equação diferencial em um conjunto de equações de diferença que são resolvidas iterativamente.



$$\text{Exemplo: } \frac{1}{2} \sigma^2 S^2 \frac{\partial^2 f}{\partial S^2} + rS \frac{\partial f}{\partial S} + \frac{\partial f}{\partial t} - rf = 0 \quad (47)$$

variáveis que definem f : S e t

- discretizando t ($0 \rightarrow T$):

É escolhida uma quantidade de instantes, igualmente espaçados entre o tempo atual, zero, e o vencimento do título, T .

Para $N+1$ instantes ($0, \Delta t, 2\Delta t, 3\Delta t, \dots, T$):

$$\Delta t = \frac{T}{N}$$

- discretizando S ($0 \rightarrow S_{\text{máx}}$):

É escolhida uma quantidade de preços para o ativo objeto igualmente espaçados, onde o $S_{\text{máx}}$ é determinado de forma que se possa definir o valor do título em qualquer instante, caso o ativo objeto atinja este valor .

Para $M+1$ preços do ativo objeto ($0, \Delta S, 2\Delta S, 3\Delta S, \dots, S_{\text{máx}}$) -

$$\Delta S = \frac{S_{\text{máx}}}{M}$$

Representando graficamente, tem-se uma grade com $(M+1)(N+1)$ pontos, cada um representando um valor de S e t . Observa-se que os valores possíveis dessas variáveis estão discretizados em um número finito de pontos, sendo o ponto (i,j) o correspondente ao instante $i\Delta t$ e ao preço da ação $j\Delta S$.

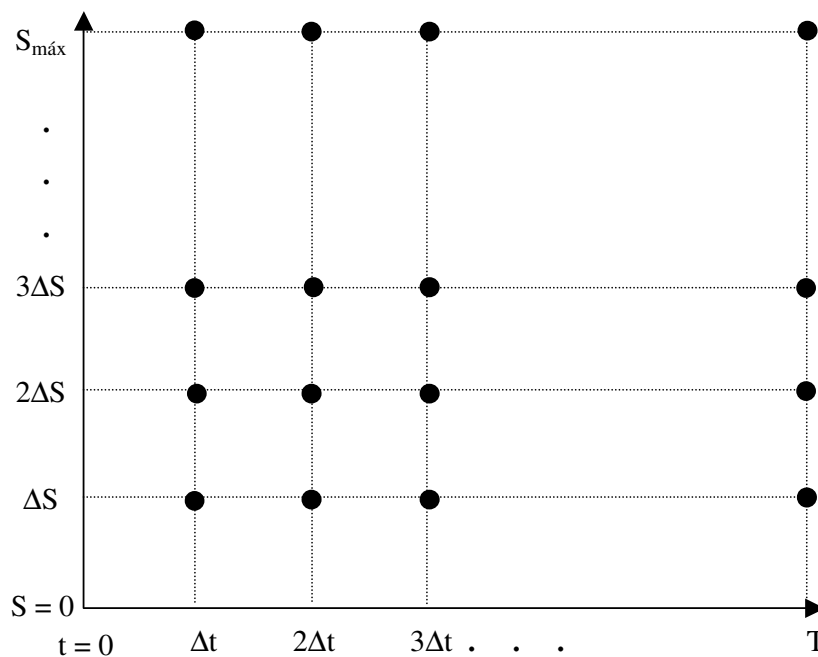


Figura 27 – Malha preço da ação Vs. instante de tempo

Utilizando a variável $f_{i,j}$ para denotar o valor de um título qualquer no ponto (i,j):

Para um ponto (i,j) da grade, $\frac{\partial f}{\partial S}$ pode ser aproximado como:

$$\frac{\partial f}{\partial S} = \frac{f_{i,j+1} - f_{i,j}}{\Delta S} \longrightarrow \text{Aproximação de diferença avançada} \quad (48)$$

ou,

$$\frac{\partial f}{\partial S} = \frac{f_{i,j} - f_{i,j-1}}{\Delta S} \longrightarrow \text{Aproximação de diferença reversa} \quad (49)$$

Pode-se usar uma aproximação mais simétrica, calculando a média das duas, conforme sugerido por Hull [24]:

$$\frac{\partial f}{\partial S} = \frac{f_{i,j+1} - f_{i,j-1}}{2\Delta S} \quad (50)$$

Para $\frac{\partial f}{\partial t}$, usa-se a aproximação de diferença avançada, de modo que o valor, no instante $i\Delta t$, esteja relacionado ao valor no instante $(i+1)\Delta t$:

$$\frac{\partial f}{\partial t} = \frac{f_{i+1,j} - f_{i,j}}{\Delta t} \quad (51)$$

A aproximação de diferença reversa para $\frac{\partial f}{\partial S}$ no ponto (i,j+1) é:

$$\frac{\partial f}{\partial S} = \frac{f_{i,j+1} - f_{i,j}}{\Delta S} \quad (52)$$

Logo, no ponto (i,j):

$$\frac{\partial^2 f}{\partial S^2} = \frac{\left(\frac{f_{i,j+1} - f_{i,j}}{\Delta S} - \frac{f_{i,j} - f_{i,j-1}}{\Delta S} \right)}{\Delta S} = \frac{f_{i,j+1} + f_{i,j-1} - 2f_{i,j}}{\Delta S^2} \quad (53)$$

Substituindo as equações (51), (52) e (53) na eq. (47), tem-se:

$$\left(\frac{f_{i+1,j} - f_{i,j}}{\Delta t}\right) + rj\Delta S\left(\frac{f_{i,j+1} - f_{i,j-1}}{2\Delta S}\right) + \frac{1}{2}\sigma^2(j^2\Delta S^2)\left(\frac{f_{i,j+1} + f_{i,j-1} - 2f_{i,j}}{\Delta S^2}\right) - rf_{i,j} = 0 \quad (54)$$

Simplificando:

$$\left(\frac{f_{i+1,j} - f_{i,j}}{\Delta t}\right) + rj\left(\frac{f_{i,j+1} - f_{i,j-1}}{2}\right) + \frac{1}{2}\sigma^2 j^2 (f_{i,j+1} + f_{i,j-1} - 2f_{i,j}) - rf_{i,j} = 0 \quad (55)$$

Multiplicando por $2\Delta t$:

$$2f_{i+1,j} - 2f_{i,j} + rj\Delta t f_{i,j+1} - rj\Delta t f_{i,j-1} + \sigma^2 j^2 \Delta t f_{i,j+1} + \sigma^2 j^2 \Delta t f_{i,j-1} - 2f_{i,j}\sigma^2 j^2 \Delta t - 2r\Delta t f_{i,j} = 0 \quad (56)$$

Colocando em evidência:

$$2f_{i+1,j} - (2 + 2\sigma^2 j^2 \Delta t + 2r\Delta t)f_{i,j} + (rj\Delta t + \sigma^2 j^2 \Delta t)f_{i,j+1} + (\sigma^2 j^2 \Delta t - rj\Delta t)f_{i,j-1} = 0 \quad (57)$$

Multiplicando por $-1/2$:

$$-f_{i+1,j} + \underbrace{(1 + \sigma^2 j^2 \Delta t + r\Delta t)}_{b_j} f_{i,j} + \underbrace{\left(-\frac{1}{2}rj\Delta t - \frac{1}{2}\sigma^2 j^2 \Delta t\right)}_{c_j} f_{i,j+1} + \underbrace{\left(\frac{1}{2}rj\Delta t - \frac{1}{2}\sigma^2 j^2 \Delta t\right)}_{a_j} f_{i,j-1} = 0$$

Ou seja:

$$a_j f_{i,j-1} + b_j f_{i,j} + c_j f_{i,j+1} - f_{i+1,j} = 0 \quad (58)$$

Condições de Contorno:

Por exemplo, para uma opção americana de venda, as condições de contorno são:

O valor da opção de venda no instante T é $\max [X - S_T, 0]$, onde S_T é o preço do ativo objeto no instante T. Assim:

$$f_{N,j} = \max[X - jS, 0] \quad j = 0, 1, 2, 3 \dots M \quad (59)$$

O valor da opção de venda, quando o preço da ação é zero, é X. Logo:

$$f_{i,0} = X \quad i = 0, 1, 2, 3 \dots N \quad (60)$$

O valor da opção de venda tende a zero conforme o preço da ação tende a infinito. Pode-se utilizar a aproximação:

$$f_{i,M} = 0 \quad i = 0, 1, 2, 3 \dots N \quad (61)$$

Graficamente tem-se:

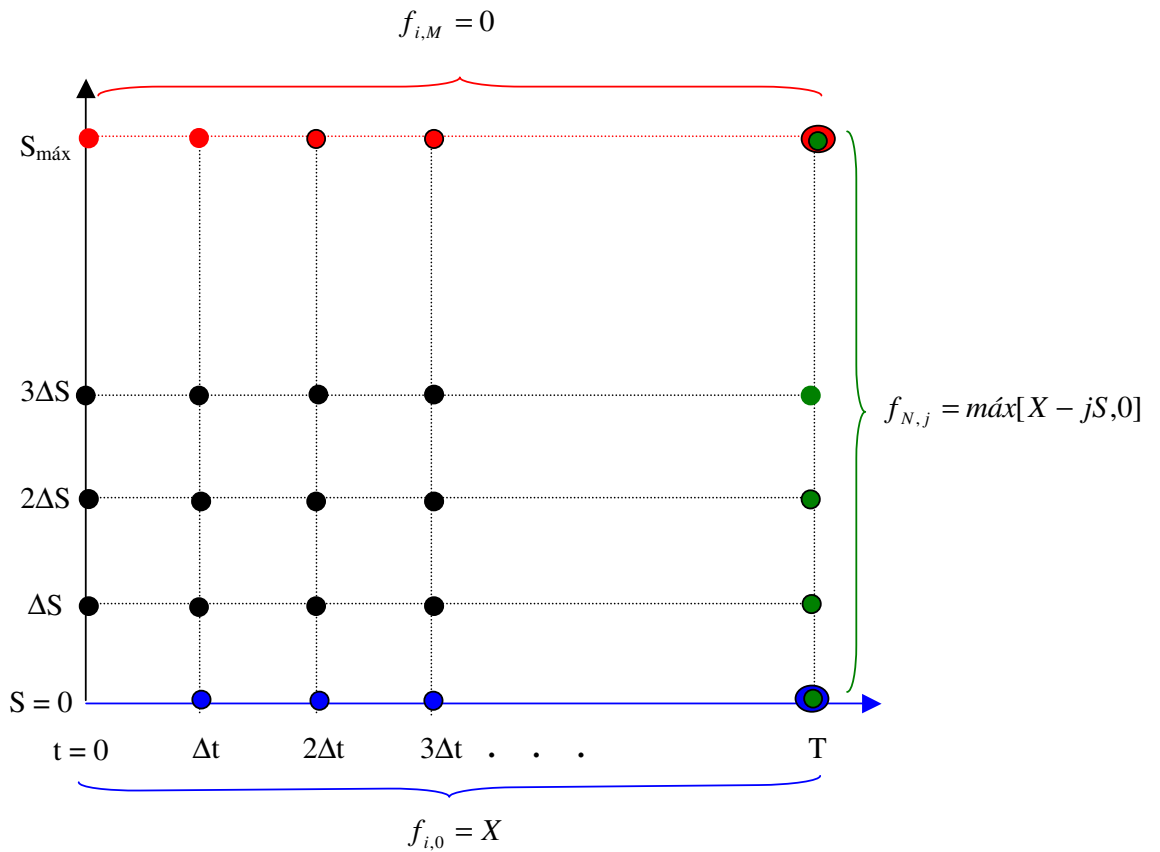


Figura 28 – Condições de contorno na malha preço da ação Vs. instante de tempo

Estando definidos os valores da opção de venda nas três extremidades da grade, resta utilizar a eq. (54) para chegar-se ao valor de f em todos os outros pontos. Abordando primeiramente os pontos correspondentes ao instante $T - \Delta t$, ou seja, com $i = N - 1$:

$$a_j f_{N-1,j-1} + b_j f_{N-1,j} + c_j f_{N-1,j+1} = f_{N,j} \quad (62)$$

Como j vai de 0 a M , para $j = 0$ e $j = M$, a eq. (58) não pode ser escrita, pois nestes casos não existiriam os termos $f_{N-1,j-1}$ e $f_{N-1,j+1}$ respectivamente. Assim, tem-se:

$$\text{para } j = 1: a_1 f_{N-1,0} + b_1 f_{N-1,1} + c_1 f_{N-1,2} = f_{N,1}$$

$$\text{para } j = 2: a_2 f_{N-1,1} + b_2 f_{N-1,2} + c_2 f_{N-1,3} = f_{N,2}$$

$$\text{para } j = 3: a_3 f_{N-1,2} + b_3 f_{N-1,3} + c_3 f_{N-1,4} = f_{N,3}$$

.

.

.

$$\text{para } j = M-1: a_{M-1} f_{N-1, M-2} + b_{M-1} f_{N-1, M-1} + c_{M-1} f_{N-1, M} = f_{N, M-1}$$

Como estão definidos nas condições de contorno os valores de $f_{N-1,0}$, $f_{N-1,M}$ e $(f_{N,1}, f_{N,2}, f_{N,3}, \dots f_{N,M-1})$, tem-se então M-1 equações e M-1 incógnitas.

Depois de determinados todos os valores possíveis da opção de venda para aquele determinado “i”, deve-se compará-los com o valor $X-j\Delta S$. Este valor representa a estratégia ótima, ou seja, se $f_{N-1,j} < X-j\Delta S$, o exercício antecipado no instante $T-\Delta t$ é ideal, com $f_{N-1,j}$ igual a $X-j\Delta S$. Os nós correspondentes $T-2\Delta t$ são então tratados de forma semelhante, e assim por diante.

Apêndice 4

Código do Modelo por Diferenças Finitas (Matlab)

```

global N;          % Numero de dias uteis
global dt;         % Incremento de tempo
global dS;         % Incremento preco do ativo subscrito
global M;          % Numero de valores "discretos" de S
global Sigma;      % volatilidade anual
global Rf;         % Taxa livre de risco
global F;          % Face value
global dy;         % Dividend Yield
global d;          % Dividend Rate
global Cr;         % Conversion ratio
global t0;         % Numero de dias entre a emissão e a avaliação
(negativo ou zero - avl = emi)
global g;          % Growth rate
global Call        % Vetor de N linhas e 2 colunas
global Put         % Vetor de "?" linhas e 2 colunas
global PutAux      % Vetor de "?" linhas (datas _ dias úteis)

L=sparse(zeros(N+1,M+1));

a=sparse(zeros(N,M-1));

b=sparse(zeros(N,M-1));

c=sparse(zeros(N,M-1));

% CC no Vencimento

for j = 0 : M

    L(N+1,j+1) = max(Cr*dS*j,F);

end

% CC se S=0

for i = 0 : N

    L(i+1,1) = 0;

end

% CC se S=Smax

for i = 0 : N

    L(i+1,M+1) = Cr * dS * M;

```



```

end

% Modelagem do Lyon

for cont = 0 : (N-1)

    i = (N - 1) - cont;

    i

    for j = 1 : (M - 1)

        Const1 = Sigma^2 * j^2 * (dt/252);

        Const2 = 0.5 * Rf * j * (dt/252);

        Const3 = 0.5 * dy * j * (dt/252);

        Const4 = 0.5 * 1/dS * d * exp(g * ((i * dt - t0)/252));

        a(i+1,j) = -0.5 * Const1 + Const2 - Const3 - Const4;

        b(i+1,j) = Const1 + Rf * (dt/252) + 1;

        c(i+1,j) = -0.5 * Const1 - Const2 + Const3 + Const4;

    end

% Elaboracao da Matriz A

    MatA = sparse(zeros(M-1,M-1));

    MatA(1,1) = b(i+1,1);

    MatA(1,2) = c(i+1,1);

    for j = 2 : (M - 2)

        MatA(j,j-1) = a(i+1,j);

        MatA(j,j) = b(i+1,j);

        MatA(j,j+1) = c(i+1,j);

    end

    MatA(M-1,M-2) = a(i+1,M-1);

    MatA(M-1,M-1) = b(i+1,M-1);

% Elaboracao da Matriz C

    MatC(1,1) = L((i+1)+1,2) - a(i+1,1) * L(i+1,1);

    for j = 2 : (M - 2)

        MatC(j,1) = L((i+1)+1,j+1);

    end

```

```

    MatC(M-1,1) = L((i+1)+1,(M-1)+1) - c(i+1,M-1) * L(i+1,M+1);

% Obtencao da Matriz de Resultados

    MatB = inv(MatA) * MatC;

    for j = 1 : (M - 1)

        L(i+1,j+1)=MatB(j,1);

    end

% Bond Holder exercendo conversibilidade do ativo

    for j = 0 : M

        if (L(i+1,j+1) < Cr*dS*j)

            L(i+1,j+1) = Cr*dS*j;

        end

    end

% Issuer exercendo a opção de compra CALL

    for j = 0 : M

        if (L(i+1,j+1) > Call(i+1,2))

            L(i+1,j+1) = max(Cr*dS*j,Call(i+1,2));

        end

    end

% Bond holder exercendo a opção de venda PUT

    MaximoLin=length(PutAux);

    ScheduleCorreto=0;

    Data = i;

    for aux = 1 : MaximoLin

        Valor = PutAux(aux);

        if (Valor == Data)

            ScheduleCorreto=aux;

        end

    end

    if (ScheduleCorreto > 0)

        for j = 0 : M

```

```

        if ( (L(i+1,j+1) < Put(ScheduleCorreto,2)) & (j > 0) )
            L(i+1,j+1) = Put(ScheduleCorreto,2);
        end
    end
end

% Exportando resultados para arquivo

fid = fopen('malha.csv','w');
for i = 1 : N+1
    fprintf(fid,'%4d,',i-1);

    for j = 1 : M+1
        fprintf(fid,'%8.4f,',L(i,j));
    end

    fprintf(fid,'\n');
end

fclose(fid);

```

Apêndice 5

Código do Modelo Grand, Vora e Weeks (Matlab): Determinação das Cuvas de Gatilho

```

global NumSim % Numero de simulacoes
global NumSimMean % Numero de simulacoes para calculo do valor do
título
global N;      % Numero de dias uteis
global S0;     % Valor inicial do ativo subjacente
global Sigma;  % Volatilidade anual
%global dt;    % Incremento de tempo
global Rf;     % Taxa livre de risco
global F;      % Face value
global dy;     % Dividend Yield
%global g;     % Growth rate
%global d;     % Dividend Rate
global Cr;     % Conversion ratio
global StrCall; % Vetor de N linhas e 2 colunas
global StrPut;  % Vetor de "?" linhas e 2 colunas

% Inicializacao de constantes
dt = 1/252;
time = N + 1;
S_MAX = 10^6;
S_MIN = 0;
dS = 0.25;

% vetor -> Call
[NumData,NumValues] = size(StrCall);

IsCall = zeros(time,1);

CallValue = zeros(time,1);

for i = 1 : NumData
    IsCall(StrCall(i,1)+1,1)=1;
    CallValue(StrCall(i,1)+1,1)=StrCall(i,2);
end

% vetor -> Put
[NumData,NumValues] = size(StrPut);

IsPut = zeros(time,1);
PutValue = zeros(time,1);

for i = 1 : NumData
    IsPut(StrPut(i,1)+1,1)=1;
    PutValue(StrPut(i,1)+1,1)=StrPut(i,2);
end
end

```

```

% Inicializacao das curvas de gatilho
CG_Conv = zeros(time,1);
CG_Call = zeros(time,1);
CG_Put = zeros(time,1);

% Determinacao do preco de partida no vencimento
CG_Conv(time) = F/Cr;

if (IsCall(time) == 1)

    CG_Call(time) = F/Cr;

else

    CG_Call(time) = S_MAX;

    % Adotamos o valor da referencia como sendo igual ao gatilho
da
    % conversao, pois este eh checado primeiro
    CG_Call_REF = CG_Conv(time);

end

if (CG_Put(time) == 1)

    CG_Put(time) = max(F,CallValue(time))/Cr;

else

    CG_Put(time) = S_MIN;

    % Adotamos o valor da referencia como sendo igual ao gatilho
da
    % call, pois esta eh checada primeiro
    if (IsCall(time) == 1)

        CG_PutREF = CG_Call(time);

    else

        CG_PutREF = CG_Conv(time);

    end

end

end

% Backwardation
for cont = (time-1):-1:1

    display('Instante de Tempo -> ');
    cont

    % Conversao
    display('Inicio do calculo da curva de gatilho da conversao');

    CG_Conv(cont) = CG_Conv(cont+1) - dS;

    ACHOU_CV = false;

    SEGURANCA = 0;

```

```

while ~ACHOU_CV

    %SEGURANCA = SEGURANCA + 1

    CG_Conv(cont) = CG_Conv(cont) + dS;

    % Imprimindo na tela para acompanhar processamento
    %CG_Conv(cont)

    VP = [];

    for i = 1 : NumSimMean

        % Armazenando os sorteio (variavel aleatoria + imagem)
        sorteios = [];

        for k = 1 : (time - cont)

            sorteios(k,1) = normrnd(0,1);

            sorteios(k,2) = (-1)*sorteios(k,1);

        end

        % Primeiro caminho

        Data_1 = cont;

        Exercicio_1 = false;

        tempo_1 = 0;

        Fluxo_1 = 0;

        S_Value_1 = CG_Conv(cont);

        while ( (Data_1 < time) & (~Exercicio_1) )

            Data_1 = Data_1 + 1;

            tempo_1 = tempo_1 + 1;

            % Simulando S
            epsilon = sorteios(tempo_1,1);

            S_Value_1 = S_Value_1*exp((Rf - dy - Sigma^2/2)*dt
+ Sigma*epsilon*sqrt(dt));

            % Condições de contorno
            if ( S_Value_1 > CG_Conv(Data_1) )

                Fluxo_1 = S_Value_1 * Cr;

                Exercicio_1 = true;

            elseif ( S_Value_1 > CG_Call(Data_1) )

                Fluxo_1 = max(CallValue(Data_1), S_Value_1 *

Cr);

                Exercicio_1 = true;

```

```

elseif ( S_Value_1 < CG_Put(Data_1) )

    Fluxo_1 = PutValue(Data_1);

    Exercicio_1 = true;

elseif (Data_1 == time)

    Fluxo_1 = max(Cr*S_Value_1,F);

end

end

% Segundo caminho

Data_2 = cont;

Exercicio_2 = false;

tempo_2 = 0;

Fluxo_2 = 0;

S_Value_2 = CG_Conv(cont);

while ( (Data_2 < time) & (~Exercicio_2) )

    Data_2 = Data_2 + 1;

    tempo_2 = tempo_2 + 1;

    % Simulando S
    epsilon = sorteios(tempo_2,2);

    S_Value_2 = S_Value_2*exp((Rf - dy - Sigma^2/2)*dt
+ Sigma*epsilon*sqrt(dt));

    % Condições de contorno
    if ( S_Value_2 > CG_Conv(Data_2) )

        Fluxo_2 = S_Value_2 * Cr;

        Exercicio_2 = true;

    elseif ( S_Value_2 > CG_Call(Data_2) )

        Fluxo_2 = max(CallValue(Data_2), S_Value_2 *

Cr);

        Exercicio_2 = true;

    elseif ( S_Value_2 < CG_Put(Data_2) )

        Fluxo_2 = PutValue(Data_2);

        Exercicio_2 = true;

```

```

elseif (Data_2 == time)

    Fluxo_2 = max(Cr*S_Value_2,F);

end

end

    VP(i,1) = (Fluxo_1 * exp(-1 * Rf * tempo_1 * dt) +
Fluxo_2 * exp(-1 * Rf * tempo_2 * dt))/2;

end

if ( mean(VP) < CG_Conv(cont)*Cr )

    ACHOU_CV = true;

end

%if ( 300 < SEGURANCA )

%    ACHOU_CV = true;

%end
end

% Call
display('Inicio do calculo da curva de gatilho da call');

if (IsCall(cont) ~= 1)

    CG_Call(cont) = S_MAX;

    CG_Call_REF = CG_Call_REF;

else

    if (IsCall(cont+1) ~= 1)

        CG_Call(cont) = CG_Call_REF + dS;

    else

        CG_Call(cont) = CG_Call(cont+1) + dS;

    end

    ACHOU_CALL = false;

    SEGURANCA = 0;

    while ~ACHOU_CALL

        %SEGURANCA = SEGURANCA + 1

        CG_Call(cont) = CG_Call(cont) - dS;

        % Imprimindo na tela para acompanhar processamento
        %CG_Call(cont)

```



```

VP = [];

for i = 1 : NumSimMean

    % Armazenando os sorteio (variavel aleatoria +
imagem)
    sorteios = [];

    for k = 1 : (time - cont)

        sorteios(k,1) = normrnd(0,1);

        sorteios(k,2) = (-1)*sorteios(k,1);

    end

    % Primeiro caminho

    Data_1 = cont;

    Exercicio_1 = false;

    tempo_1 = 0;

    Fluxo_1 = 0;

    S_Value_1 = CG_Call(cont);

    while ( (Data_1 < time) & (~Exercicio_1) )

        Data_1 = Data_1 + 1;

        tempo_1 = tempo_1 + 1;

        % Simulando S
        epsilon = sorteios(tempo_1,1);

        S_Value_1 = S_Value_1*exp((Rf - dy -
Sigma^2/2)*dt + Sigma*epsilon*sqrt(dt));

        % Condições de contorno
        if ( S_Value_1 > CG_Conv(Data_1) )

            Fluxo_1 = S_Value_1 * Cr;

            Exercicio_1 = true;

        elseif ( S_Value_1 > CG_Call(Data_1) )

            Fluxo_1 = max(CallValue(Data_1), S_Value_1
* Cr);

            Exercicio_1 = true;

        elseif ( S_Value_1 < CG_Put(Data_1) )

            Fluxo_1 = PutValue(Data_1);

            Exercicio_1 = true;

```

```

elseif (Data_1 == time)

    Fluxo_1 = max(Cr*S_Value_1,F);

end

end

% Segundo caminho

Data_2 = cont;

Exercicio_2 = false;

tempo_2 = 0;

Fluxo_2 = 0;

S_Value_2 = CG_Call(cont);

while ( (Data_2 < time) & (~Exercicio_2) )

    Data_2 = Data_2 + 1;

    tempo_2 = tempo_2 + 1;

    % Simulando S
    epsilon = sorteios(tempo_2,2);

    S_Value_2 = S_Value_2*exp((Rf - dy -
Sigma^2/2)*dt + Sigma*epsilon*sqrt(dt));

    % Condições de contorno
    if ( S_Value_2 > CG_Conv(Data_2) )

        Fluxo_2 = S_Value_2 * Cr;

        Exercicio_2 = true;

    elseif ( S_Value_2 > CG_Call(Data_2) )

        Fluxo_2 = max(CallValue(Data_2), S_Value_2
* Cr);

        Exercicio_2 = true;

    elseif ( S_Value_2 < CG_Put(Data_2) )

        Fluxo_2 = PutValue(Data_2);

        Exercicio_2 = true;

    elseif (Data_2 == time)

        Fluxo_2 = max(Cr*S_Value_2,F);

    end

end

end

```

```

        VP(i,1) = (Fluxo_1 * exp(-1 * Rf * tempo_1 * dt) +
Fluxo_2 * exp(-1 * Rf * tempo_2 * dt))/2;

        end

        if ( mean(VP) < CallValue(cont) )

            ACHOU_CALL = true;

        end

        CG_Call_REF = CG_Call(cont);

        %if ( 300 < SEGURANCA )

            %    ACHOU_CALL = true;

        %end

    end

end

% Put
display('Inicio do calculo da curva de gatilho da put');

if (IsPut(cont) ~= 1)

    CG_Put(cont) = 0;

    CG_PutREF = CG_PutREF;

else

    if (IsPut(cont+1) ~= 1)

        CG_Put(cont) = CG_PutREF + dS;

    else

        CG_Put(cont) = CG_Put(cont+1) + dS;

    end

    end

    ACHOU_PUT = false;

    SEGURANCA = 0;

    while ~ACHOU_PUT

        %SEGURANCA = SEGURANCA + 1

        CG_Put(cont) = CG_Put(cont) - dS;

        % Imprimindo na tela para acompanhar processamento
        %CG_Put(cont)

        VP = [];

```

```

for i = 1 : NumSimMean
    % Armazenando os sorteio (variavel aleatoria +
    imagem)
    sorteios = [];
    for k = 1 : (time - cont)
        sorteios(k,1) = normrnd(0,1);
        sorteios(k,2) = (-1)*sorteios(k,1);
    end
    % Primeiro caminho
    Data_1 = cont;
    Exercicio_1 = false;
    tempo_1 = 0;
    Fluxo_1 = 0;
    S_Value_1 = CG_Put(cont);
    while ( (Data_1 < time) & (~Exercicio_1) )
        Data_1 = Data_1 + 1;
        tempo_1 = tempo_1 + 1;
        % Simulando S
        epslon = sorteios(tempo_1,1);
        S_Value_1 = S_Value_1*exp((Rf - dy -
        Sigma^2/2)*dt + Sigma*epslon*sqrt(dt));
        % Condições de contorno
        if ( S_Value_1 > CG_Conv(Data_1) )
            Fluxo_1 = S_Value_1 * Cr;
            Exercicio_1 = true;
        elseif ( S_Value_1 > CG_Call(Data_1) )
            Fluxo_1 = max(CallValue(Data_1), S_Value_1
            * Cr);
            Exercicio_1 = true;
        elseif ( S_Value_1 < CG_Put(Data_1) )
            Fluxo_1 = PutValue(Data_1);
            Exercicio_1 = true;
        elseif (Data_1 == time)
            Fluxo_1 = max(Cr*S_Value_1,F);

```

```

        end

    end

    % Segundo caminho

    Data_2 = cont;

    Exercicio_2 = false;

    tempo_2 = 0;

    Fluxo_2 = 0;

    S_Value_2 = CG_Put(cont);

    while ( (Data_2 < time) & (~Exercicio_2) )

        Data_2 = Data_2 + 1;

        tempo_2 = tempo_2 + 1;

        % Simulando S
        epslon = sorteios(tempo_2,2);

        S_Value_2 = S_Value_2*exp((Rf - dy -
Sigma^2/2)*dt + Sigma*epslon*sqrt(dt));
        % Condições de contorno
        if ( S_Value_2 > CG_Conv(Data_2) )
            Fluxo_2 = S_Value_2 * Cr;

            Exercicio_2 = true;

        elseif ( S_Value_2 > CG_Call(Data_2) )

            Fluxo_2 = max(CallValue(Data_2), S_Value_2
* Cr);

            Exercicio_2 = true;

        elseif ( S_Value_2 < CG_Put(Data_2) )

            Fluxo_2 = PutValue(Data_2);

            Exercicio_2 = true;

        elseif (Data_2 == time)

            Fluxo_2 = max(Cr*S_Value_2,F);

        end

    end

    VP(i,1) = (Fluxo_1 * exp(-1 * Rf * tempo_1 * dt) +
Fluxo_2 * exp(-1 * Rf * tempo_2 * dt))/2;

end

```

```

        if ( mean(VP) < PutValue(cont) )

            ACHOU_PUT = true;

        end

        % Apenas para garantir nao-negatividade
        if ( CG_Put(cont) < 0 )

            ACHOU_PUT = true;

            CG_Put(cont) = 0;

        end

        CG_PutREF = CG_Put(cont);

        %if ( 300 < SEGURANCA )

            %    ACHOU_PUT = true;

        %end

    end

end

end

```

Apêndice 6

Código do Modelo Grand, Vora e Weeks (Matlab): Apreçamento

```

global NumSim % Numero de simulacoes
global NumSimMean % Numero de simulacoes para calculo do valor do
título
global N;      % Numero de dias uteis
global S0;     % Valor inicial do ativo subjacente
global Sigma;  % Volatilidade anual
%global dt;    % Incremento de tempo
global Rf;     % Taxa livre de risco
global F;      % Face value
global dy;     % Dividend Yield
%global g;     % Growth rate
%global d;     % Dividend Rate
global Cr;     % Conversion ratio
global StrCall; % Vetor de N linhas e 2 colunas
global StrPut;  % Vetor de "?" linhas e 2 colunas
global CPDate;  % Data de validade da soft call protection
global CPValue; % Valor limite da acao durante a soft call
protection

global CG_Conv;
global CG_Call;
global CG_Put;

% Inicializacao de constantes
dt = 1/252;
time = N + 1;

% vetor -> Call
[NumData,NumValues] = size(StrCall);

IsCall = zeros(time,1);
CallValue = zeros(time,1);

for i = 1 : NumData

    IsCall(StrCall(i,1)+1,1)=1;
    CallValue(StrCall(i,1)+1,1)=StrCall(i,2);

end

% vetor -> Put
[NumData,NumValues] = size(StrPut);

IsPut = zeros(time,1);
PutValue = zeros(time,1);

for i = 1 : NumData

    IsPut(StrPut(i,1)+1,1)=1;

```

```

        PutValue(StrPut(i,1)+1,1)=StrPut(i,2);

end

for i = 1 : NumSim

    % Armazenando os sorteio (variavel aleatoria + imagem)
    sorteios = [];

    for k = 1 : time

        sorteios(k,1) = normrnd(0,1);

        sorteios(k,2) = (-1)*sorteios(k,1);

    end

    disp('Simulacao ->');

    i

    % Primeiro caminho

    Data_1 = 0;

    Exercicio_1 = false;

    tempo_1 = 0;

    Fluxo_1 = 0;

    S_Value_1 = S0;

    while ( (Data_1 < time) & (~Exercicio_1) )

        Data_1 = Data_1 + 1;

        tempo_1 = tempo_1 + 1;

        % Simulando S
        epslon = sorteios(tempo_1,1);

        S_Value_1 = S_Value_1*exp((Rf - dy - Sigma^2/2)*dt +
Sigma*epslon*sqrt(dt));

        % Condições de contorno
        if ( S_Value_1 > CG_Conv(Data_1) )

            Fluxo_1 = S_Value_1 * Cr;

            Exercicio_1 = true;

        elseif ( ( S_Value_1 > CG_Call(Data_1) ) & ( ( Data_1 >
CPDate + 1) | ( S_Value_1 >= CPValue ) ) )

            Fluxo_1 = max(CallValue(Data_1), S_Value_1 * Cr);

            Exercicio_1 = true;

        elseif ( S_Value_1 < CG_Put(Data_1) )

```



```

        Fluxo_1 = PutValue(Data_1);

        Exercicio_1 = true;

    elseif (Data_1 == time)

        Fluxo_1 = max(Cr*S_Value_1,F);

    end

end

% Segundo caminho

Data_2 = 0;

Exercicio_2 = false;

tempo_2 = 0;

Fluxo_2 = 0;

S_Value_2 = S0;

while ( (Data_2 < time) & (~Exercicio_2) )

    Data_2 = Data_2 + 1;

    tempo_2 = tempo_2 + 1;

    % Simulando S
    epslon = sorteios(tempo_2,2);

    S_Value_2 = S_Value_2*exp((Rf - dy - Sigma^2/2)*dt +
Sigma*epslon*sqrt(dt));

    % Condições de contorno
    if ( S_Value_2 > CG_Conv(Data_2) )

        Fluxo_2 = S_Value_2 * Cr;

        Exercicio_2 = true;

    elseif ( ( S_Value_2 > CG_Call(Data_2) ) & ( ( Data_2 >
CPDate + 1) | ( S_Value_2 >= CPValue ) ) )

        Fluxo_2 = max(CallValue(Data_2), S_Value_2 * Cr);

        Exercicio_2 = true;

    elseif ( S_Value_2 < CG_Put(Data_2) )

        Fluxo_2 = PutValue(Data_2);

        Exercicio_2 = true;

    elseif (Data_2 == time)

        Fluxo_2 = max(Cr*S_Value_2,F);

    end

```

```
end

    VP(i,1) = (Fluxo_1 * exp(-1 * Rf * tempo_1 * dt) + Fluxo_2 *
exp(-1 * Rf * tempo_2 * dt))/2;

end

Lyon_Value = mean(VP);

Lyon_Value
```

Apêndice 7

Código do Modelo por Mínimos Quadrados de Monte Carlo (Matlab): Apreçamento

```

global NumSim % Numero de simulacoes
global N;      % Numero de dias uteis
global S0;     % Valor inicial do ativo subjacente
global Sigma;  % Volatilidade anual
%global dt;    % Incremento de tempo
global Rf;     % Taxa livre de risco
global F;      % Face value
global dy;     % Dividend Yield
%global g;     % Growth rate
%global d;     % Dividend Rate
global Cr;     % Conversion ratio
global StrCall; % Vetor de N linhas e 2 colunas
global StrPut;  % Vetor de "?" linhas e 2 colunas

global CPDate; % Data de validade da soft call protection
global CPValue; % Valor limite da acao durante a soft call
protection

dt = 1/252;

% Simulacao dos precos do ativo subjacente
Simulacao_1 = [];

Simulacao_2 = [];

for i = 1 : NumSim

    Simulacao_1(i,1) = S0;

    Simulacao_2(i,1) = S0;

end

disp('Inicio da simulacao');

for i = 1 : NumSim

    disp('Simulacao -> ');
    i

    for j = 2 : (N+1)

        epslon_1 = normrnd(0,1);

        epslon_2 = (-1) * epslon_1;

```

```

        Simulacao_1(i,j) = Simulacao_1(i,j-1)*exp((Rf - dy -
Sigma^2/2)*dt + Sigma*epslon_1*sqrt(dt));

        Simulacao_2(i,j) = Simulacao_2(i,j-1)*exp((Rf - dy -
Sigma^2/2)*dt + Sigma*epslon_2*sqrt(dt));

    end

end

disp('Fim da simulacao');

% Tamanho da matriz com os precos do ativo subjacente
[sim,time] = size(Simulacao_1);

% vetor -> Call
[NumData,NumValues] = size(StrCall);

IsCall = zeros(time,1);
CallValue = zeros(time,1);

for i = 1 : NumData

    IsCall(StrCall(i,1)+1,1)=1;
    CallValue(StrCall(i,1)+1,1)=StrCall(i,2);
end

% vetor -> Put
[NumData,NumValues] = size(StrPut);

IsPut = zeros(time,1);
PutValue = zeros(time,1);

for i = 1 : NumData

    IsPut(StrPut(i,1)+1,1)=1;
    PutValue(StrPut(i,1)+1,1)=StrPut(i,2);
end

% Inicializacao -> Maturity Condition

CashFlow_1 = zeros(sim,time);

CashFlow_2 = zeros(sim,time);

for i = 1 : sim

    CashFlow_1(i,time) = max(Cr*Simulacao_1(i,time),F);

    CashFlow_2(i,time) = max(Cr*Simulacao_2(i,time),F);

end

disp('Inicio da recursividade');

% Calculo recursivo

VP_1 = zeros(sim,1);

```

```

VP_2 = zeros(sim,1);

for col = (time-1):-1:1
    disp('Tempo -> ');
    col

    for row = 1 : sim

        VP_1(row,1) = exp(-1 * Rf * dt) * (VP_1(row,1) +
        CashFlow_1(row,col+1));

        VP_2(row,1) = exp(-1 * Rf * dt) * (VP_2(row,1) +
        CashFlow_2(row,col+1));

    end

    if (col > 1)

        % Calculando funcao de continuacao

        continuacao_1 = regression(VP_1(:,1), Simulacao_1(:,col));
        continuacao_2 = regression(VP_2(:,1), Simulacao_2(:,col));

        for row = 1 : sim

            % Primeiro caminho

            if (Cr*Simulacao_1(row,col) > continuacao_1(row,1))

                for i = (col+1) : time

                    CashFlow_1(row,i) = 0;

                end

                CashFlow_1(row,col) = Cr*Simulacao_1(row,col);

                VP_1(row,1) = 0;

                continuacao_1(row,1) = Cr*Simulacao_1(row,col);

            elseif (IsCall(col,1) == 1) & (CallValue(col,1) <
            continuacao_1(row,1)) & ( ( col > CPDate + 1 ) | (
            Simulacao_1(row,col) >= CPValue ) )

                for i = (col+1) : time

                    CashFlow_1(row,i) = 0;

                end

```

```

        CashFlow_1(row,col) =
max(CallValue(col,1),Cr*Simulacao_1(row,col));

        VP_1(row,1) = 0;

        continuacao_1(row,1) =
max(CallValue(col,1),Cr*Simulacao_1(row,col));

        elseif (IsPut(col,1) == 1) & (PutValue(col,1) >
continuacao_1(row,1))

            for i = (col+1) : time

                CashFlow_1(row,i) = 0;

            end

            CashFlow_1(row,col) = PutValue(col,1);

            VP_1(row,1) = 0;

            continuacao_1(row,1) = PutValue(col,1);

        else

            CashFlow_1(row,col) = 0;

        end

        % Segundo caminho

        if (Cr*Simulacao_2(row,col) > continuacao_2(row,1))

            for i = (col+1) : time

                CashFlow_2(row,i) = 0;

            end

            CashFlow_2(row,col) = Cr*Simulacao_2(row,col);

            VP_2(row,1) = 0;

            continuacao_2(row,1) = Cr*Simulacao_2(row,col);

        elseif (IsCall(col,1) == 1) & (CallValue(col,1) <
continuacao_2(row,1)) & ( ( col > CPDate + 1 ) | (
Simulacao_2(row,col) >= CPValue ) )

            for i = (col+1) : time

                CashFlow_2(row,i) = 0;

            end

            CashFlow_2(row,col) =
max(CallValue(col,1),Cr*Simulacao_2(row,col));

```

```

        VP_2(row,1) = 0;

        continuacao_2(row,1) =
max(CallValue(col,1),Cr*Simulacao_2(row,col));

        elseif (IsPut(col,1) == 1) & (PutValue(col,1) >
continuacao_2(row,1))

            for i = (col+1) : time

                CashFlow_2(row,i) = 0;

            end

            CashFlow_2(row,col) = PutValue(col,1);

            VP_2(row,1) = 0;

            continuacao_2(row,1) = PutValue(col,1);

        else

            CashFlow_2(row,col) = 0;

        end

    end

end

% Calculo do valor do titulo
VP_Total = [];

VP_Total = (VP_1 + VP_2)/2;

% Calculo do valor do titulo
Lyon_Value = mean(VP_Total);

if (Cr*S0 > Lyon_Value)

    Lyon_Value = Cr*S0;

elseif (IsPut(1,1) == 1) & (PutValue(1,1) > Lyon_Value);

    Lyon_Value = PutValue(1,1);

elseif (IsCall(1,1) == 1) & (CallValue(1,1) < Lyon_Value);

    Lyon_Value = max(CallValue(1,1),Cr*S0);

end
Lyon_Value

%regressão

function [result] = regression(Y, X)

```

```
varInteresse = [];  
varExplicativa = [];  
varInteresse = Y;  
varExplicativa = X;  
varExplicativaFinal = [ones(size(varExplicativa)) varExplicativa  
varExplicativa.^2];  
coeficientes = regress(varInteresse, varExplicativaFinal);  
result = varExplicativaFinal * coeficientes;
```


Anexo 1

Circular BACEN nº 2.199, de 16 de julho de 1992.

Estabelece condições para emissão e colocação, no exterior, de títulos conversíveis em ações de empresas e instituições sediadas no País.

Comunicamos que a Diretoria do Banco Central do Brasil, em sessão realizada em 15/07/92, tendo em vista as disposições do art. 10, inciso VI, da Lei nº 4.595, de 31/12/64, e do art. 3º da Lei nº 4.131, de 03/08/62, decidiu:

Art. 1º - Dependerá de prévia autorização do Banco Central do Brasil a emissão e colocação, no exterior, de títulos conversíveis em ações, por parte de empresas e instituições sediadas no País, observadas as disposições legais pertinentes.

Art. 2º - Para os efeitos desta Circular, entende-se por:

- a) títulos conversíveis em ações: aqueles emitidos por instituição sediada no País, colocados no exterior e que representem direitos sobre ações de sua própria emissão ou de outra instituição sediada no País;
- b) " warrants" : opções de compra de ações, colocados no exterior, por instituições sediadas no País;
- c) instituição emissora: instituição ou empresa sediada no País, emissora dos títulos conversíveis ou detentora das ações que constituirão o lastro dos " warrants" ;
- d) investidor: pessoa física ou jurídica sediada no exterior, detentora dos títulos conversíveis ou dos " warrants" ;
- e) agente dos investidores: instituição sediada no exterior, representante dos investidores;
- f) data da conversão ou do exercício da opção de compra: aquela em que o investidor ou seu agente liquida a operação de compra das ações, conforme definido na cláusula de conversão ou nos termos dos " warrants" .

Art. 3º - Para os fins do disposto nesta Circular os direitos decorrentes dos "warrants" se equiparam aos da cláusula de conversibilidade dos títulos conversíveis.

Art. 4º - Apenas ações representativas do capital de companhias abertas registradas na Comissão de Valores Mobiliários e negociadas em Bolsas de Valores no País poderão constituir lastro para a emissão e colocação dos títulos de que trata esta Circular.

Parágrafo único - Até a data da conversão ou do exercício da opção de compra pelos detentores dos "warrants", a distribuição de dividendos e o exercício de subscrição constituirão direitos da instituição emissora.

Art. 5º - Para fins de registro do investimento decorrente da conversão dos títulos ou do exercício dos "warrants", o investidor ou seu agente deverá se enquadrar nas disposições de qualquer dos Anexos à Resolução nº 1.289, de 20/03/87.

Art. 6º - O ingresso das divisas correspondentes à colocação dos títulos conversíveis dar-se-á sob a modalidade de empréstimo externo, observadas as disposições legais e regulamentares em vigor.

Art. 7º - Fica o Departamento de Capitais Estrangeiros (FIRCE) autorizado a expedir as normas complementares e adotar as medidas julgadas necessárias à execução do disposto nesta Circular.

Art. 8º - Esta Circular entra em vigor na data de sua publicação.

Art. 9º - Fica revogada a Circular nº 1.969, de 06/06/91.

ARMINIO FRAGA NETO

Diretor