

## Referências Bibliográficas

ABRAMOWITZ, W.; STEGUN, I. **Handbook of Mathematical Functions with Formulas, Graphs and Mathematical Tables**. New York: Dover Publications, 1972.

BARRAQUAND, J.; PUDET, T. **Pricing Of American Path-Dependent Contingent Claims**. Mathematical Finance, v.6, n.1, 1996, pp.17-51.

BLACK, F.; SCHOLES, M. **The Pricing Of Options And Corporate Liabilities**, Journal of Political Economy 81, 1973, pp.637-659.

BILGER, R. **Valuing American-Asian Options with the Longstaff-Schwartz Algorithm**. Master's thesis in Mathematical Finance, Kellogg College, University of Oxford, UK, 2003.

BOYLE, P. **Options: A Monte Carlo Approach**. Journal of Financial Economics, v.4, n.3, 1977, pp.323-338.

BOYLE, P.; BOYLE, F. **Derivatives: The Tools that Changed Finance**. North Yorkshire: Risk Books, 2001.

BOYLE, P. P., EMANUEL, D. **The Pricing of Options on the Generalized Mean**. University of British Columbia, Vancouver, BC, Canada, Working paper, 1980.

BROADIE, M.; DETEMPLE, J.B. **Option Pricing: Valuation Models and Applications**. Management Science, v.50, n.9, 2004, pp.1145-1177.

BROTHERTON-RATCLIFFE, R. **Monte Carlo Motoring**. Dec. 1994. Risk Magazine. In Over the Rainbow: Developments in Exotic Options and Complex Swaps. pp. 335-338.

BURBAN, P. A. C. **Apreçamento de Opções Exóticas: Uma Abordagem pela Simulação de Monte Carlo**. Dissertação de Mestrado, Departamento de Engenharia Industrial, PUC-Rio, 2008.

CARRIERE, J. F. **Valuation of the early-exercise price for options using simulations and nonparametric regression.** Insurance: Math. Economics, v.19, 1996, pp.19-30.

CERRATO, M.; CHEUNG, K. K. **Valuing American Style Derivatives by Least Squares Methods.** Department of Economics, Finance and International Business. London Metropolitan University. working paper, 2006.

CHAUDHARY, S. K. **American Options and the LSM Algorithm: Quasi-Random Sequences and Brownian Bridges.** Journal of Computational Finance, v.8, n.4, 2005.

CLEMENT, E.; LAMBERTON, D.; PROTTER, P. **An Analysis of the Longstaff-Schwartz Algorithm for American Option Pricing.** Finance Stochastics, v.6, n.4, 2002, pp.449-471.

FROTA, A. E. F. **Avaliação de Opções Tradicionais e Complexas.** Dissertação de Mestrado, Departamento de Engenharia Industrial, PUC-Rio, 2003.

GLASSERMAN, P. **Monte Carlo Methods in Financial Engineering.** New York: Springer, 2004.

GRANT, D.; VORA, G.; WEEKS, D. **Path-Dependent Options: Extending the Monte Carlo Simulation Approach.** Management Science, v.43, n.11, 1997, pp.1589-1602.

JALÓN, J.; RODRÍGUEZ, J.; BRAZÁLEZ, A. **Aprenda Matlab 6.1 como si estuviera en primero.** Escuela Técnica Superior de Ingenieros Industriales, Universidad Politécnica de Madrid, Manual Técnico, 2001.

JARROW, R. (edited by) **Over the Rainbow: Developments in Exotic Options and Complex Swaps.** London: Risk Publications, 1995.

JOY, C.; BOYLE, P.; TAN, K.S. **Quasi-Monte Carlo Methods in Numerical Finance, Management Science**, v. 42, n. 6, 1996, pp. 926-938.

KEMNA, A.G.Z.; VORST, A.C.F. **A Pricing Method for Options Based on Average Asset Values,** Journal of Banking and Finance, v.14, 1990, pp.113-129.

LIMA, E.L. **Espaços Métricos.** 4th ed. Rio de Janeiro: IMPA, 2007.

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, pp. 113-147.

MATLAB PROGRAMMING. **Financial Toolbox User's Guide; Hierarquia de Operações MATLAB 6.** Disponíveis em [www.freedownloads.com](http://www.freedownloads.com)

MCDONALD, Robert L. **Derivative Markets.** 2nd ed. Boston: Addison Wesley, c2006.

MORENO, M.; NAVAS, J.F. **On the Robustness of Least-Squares Monte Carlo (LSM) for Pricing American Derivatives.** Review of Derivatives Research, v.6, 2003, pp. 107-128.

MORO, B. **The Full Monte.** Feb. 1995. Risk Magazine. In Over the Rainbow: Developments in Exotic Options and Complex Swaps. pp. 331-333.

STENTOFT, L. **Convergence of the Least Squares Monte Carlo Approach to American Option Valuation.** Management Science, v.50, 2004, pp. 1193-1203.

TILLEY, J.A. **Valuing American Options in a Path Simulation Model.** Transactions of the Society of Actuaries, v.45, 1993, pp. 83-104.

TSITSIKLIS, J.; VAN ROY, B. **Optimal Stopping of Markov Processes: Hilberts Pace Theory, Approximation Algorithms, And An Application To Pricing High-Dimensional Financial Derivatives.** IEEE Trans. Automatic Control v.44,1999, pp.1840-1851.

TURNBULL, S.; WAKEMAN, L. **A Quick Algorithm For Pricing European Average Options.** Journal of Financial and Quantitative Analysis, v.26, 1991, pp.377-389.

WILMOTT, P.; DEWYNNE, J.; HOWISON, J. **Option Pricing: Mathematical Models and Computation.** Oxford Financial Press, Oxford, 1993.

ZVAN, R.; FORSYTH, P. A.; VETZAL, K. R. **Robust Numerical Methods for PDE Models of Asian Options.** Journal of Computational Finance, v.1, n.2, 1998, pp.39–78.

## Apêndice A

### Detalhando Categorias de Classificação Básica

Seja,

$t_i \equiv$  tempo  $t$  no instante  $i$ , em que  $i = 1, 2, \dots, n$ . Onde  $t_n = T$ , expiração da opção.

$S(t_i) = S_i \equiv$  preço do ativo básico observado no instante  $i$ .

$\overline{S}_{(i)} = \overline{S}_i \equiv$  média computada no instante  $i$ .

$K \equiv$  preço de exercício (ou *strike price*).

$W_i \equiv$  peso dado ao  $i$ -ésimo preço do ativo base observado.

$$\text{payoff}^{Call} \equiv \max[S(t_i) - K, 0] = (S(t_i) - K)^+ \quad (\text{A.1})$$

$$\text{payoff}^{Put} \equiv \max[K - S(t_i), 0] = (K - S(t_i))^+ \quad (\text{A.2})$$

a) **Quanto ao estilo** - Ou seja, se a opção asiática é *européia* ou *americana*:

- Estilo Europeu:

$$\text{payoff}_{Europeia}^{Call} = \max[S(t_n) - K, 0] = (S(t_n) - K)^+ \quad (\text{A.3})$$

$$\text{payoff}_{Europeia}^{Put} = \max[K - S(t_n), 0] = (K - S(t_n))^+ \quad (\text{A.4})$$

- Estilo Americano:

$$\text{payoff}_{Americana}^{Call} = \max[S(t_i) - K, 0] = (S(t_i) - K)^+, \quad \forall i = 1, \dots, n. \quad (\text{A.5})$$

$$\text{payoff}_{Americana}^{Put} = \max[K - S(t_i), 0] = (K - S(t_i))^+, \quad \forall i = 1, \dots, n. \quad (\text{A.6})$$

b) **Quanto ao tipo de média utilizada** – Opção asiática baseada em *média aritmética* ou *média geométrica*:

- Média Aritmética:

$$\overline{S}_{(n)} = \frac{S(t_1) + S(t_2) + \dots + S(t_n)}{n} = \frac{\sum_{i=1}^n S(t_i)}{n} \equiv \overline{A}_{(n)} \quad (\text{A.7})$$

- Média Geométrica:

$$\overline{S_{(n)}} = [S(t_1) \cdot S(t_2) \cdot \dots \cdot S(t_n)]^{1/n} = \left[ \prod_{i=1}^n S(t_i) \right]^{1/n} \equiv \overline{G_{(n)}} \quad (\text{A.8})$$

Ou alternativamente,

$$\overline{S_{(n)}} = \left[ \prod_{i=1}^n S(t_i) \right]^{1/n} \Rightarrow \ln(\overline{S_{(n)}}) = \ln \left\{ \left[ \prod_{i=1}^n S(t_i) \right]^{1/n} \right\} \Rightarrow \quad (\text{A.9})$$

$$\Rightarrow \ln(\overline{S_{(n)}}) = \frac{1}{n} \sum_{i=1}^n \ln[S(t_i)] \Rightarrow \overline{S_{(n)}} = \exp \left\{ \frac{1}{n} \sum_{i=1}^n \ln[S(t_i)] \right\}$$

- c) **Quanto à ponderação dos termos que compõem a média – Não flexível ou Flexível:**

1) *Não Flexível:*

i. *Média aritmética*

$$\overline{S_{(n)}} = \frac{W_1 S(t_1) + W_2 S(t_2) + \dots + W_n S(t_n)}{n} = \frac{\sum_{i=1}^n W_i S(t_i)}{n}, \quad (\text{A.10})$$

onde  $W_1 = W_2 = \dots = W_n$

ii. *Média geométrica*

$$\overline{S_{(n)}} = \left\{ [S(t_1)]^{W_1} \cdot [S(t_2)]^{W_2} \cdot \dots \cdot [S(t_n)]^{W_n} \right\}^{1/n} = \left\{ \prod_{i=1}^n [S(t_i)]^{W_i} \right\}^{1/n} = \quad (\text{A.11})$$

$$= \exp \left\{ \frac{1}{n} \sum_{i=1}^n W_i \ln[S(t_i)] \right\}, \text{ onde } W_1 = W_2 = \dots = W_n$$

2) *Flexível:*i. *Média aritmética*

$$\overline{S}_{(n)} = \frac{W_1 S(t_1) + W_2 S(t_2) + \dots + W_n S(t_n)}{n} = \frac{\sum_{i=1}^n W_i S(t_i)}{n}, \quad (\text{A.12})$$

onde  $W_1 \neq W_2 \neq \dots \neq W_n$

ii. *Média geométrica*

$$\overline{S}_{(n)} = \left\{ [S(t_1)]^{W_1} \cdot [S(t_2)]^{W_2} \cdot \dots \cdot [S(t_n)]^{W_n} \right\}^{1/n} = \left\{ \prod_{i=1}^n [S(t_i)]^{W_i} \right\}^{1/n}, \quad (\text{A.13})$$

onde  $W_1 \neq W_2 \neq \dots \neq W_n$

d) **Quanto ao tipo de amostragem** – *discreta ou contínua:*1) *Amostragem discreta*i. *Média Aritmética*

## • Não Flexível:

$$\overline{S}_{(n)} = \frac{W_1 S(t_1) + W_2 S(t_2) + \dots + W_n S(t_n)}{n} = \frac{\sum_{i=1}^n W_i S(t_i)}{n}, \quad (\text{A.14})$$

onde  $W_1 = W_2 = \dots = W_n$

## • Flexível:

$$\overline{S}_{(n)} = \frac{W_1 S(t_1) + W_2 S(t_2) + \dots + W_n S(t_n)}{n} = \frac{\sum_{i=1}^n W_i S(t_i)}{n}, \quad (\text{A.15})$$

onde  $W_1 \neq W_2 \neq \dots \neq W_n$

ii. *Média Geométrica*

- Não Flexível:

$$\begin{aligned}\overline{S_{(n)}} &= \left\{ [S(t_1)]^{W_1} \cdot [S(t_2)]^{W_2} \cdots [S(t_n)]^{W_n} \right\}^{\frac{1}{n}} = \left\{ \prod_{i=1}^n [S(t_i)]^{W_i} \right\}^{\frac{1}{n}} = \\ &= \exp \left\{ \frac{1}{n} \sum_{i=1}^n W_i \ln [S(t_i)] \right\}, \text{ onde } W_1 = W_2 = \dots = W_n\end{aligned}\quad (\text{A.16})$$

- Flexível:

$$\overline{S_{(n)}} = \left\{ [S(t_1)]^{W_1} \cdot [S(t_2)]^{W_2} \cdots [S(t_n)]^{W_n} \right\}^{\frac{1}{n}} = \left\{ \prod_{i=1}^n [S(t_i)]^{W_i} \right\}^{\frac{1}{n}}, \quad (\text{A.17})$$

onde  $W_1 \neq W_2 \neq \dots \neq W_n$

2) *Amostragem contínua*

i. *Média Aritmética*

- Não Flexível:

$$\overline{S_{(n)}} = \int_{t_1}^{t_n} W_i S(t_i) dt_i, \text{ onde } W_1 = W_2 = \dots = W_n \quad (\text{A.18})$$

- Flexível:

$$\overline{S_{(n)}} = \int_{t_1}^{t_n} W_i S(t_i) dt_i, \text{ onde } W_1 \neq W_2 \neq \dots \neq W_n \quad (\text{A.19})$$

ii. *Média Geométrica*

- Não Flexível:

$$\overline{S_{(n)}} = \exp \left\{ \frac{1}{n} \int_{t_1}^{t_n} W_i \ln [S(t_i)] dt_i \right\}, \text{ onde } W_1 = W_2 = \dots = W_n \quad (\text{A.20})$$

- Flexível:

$$\overline{S}_{(n)} = \exp \left\{ \frac{1}{n} \int_{t_1}^{t_n} W_i \ln [S(t_i)] dt_i \right\}, \text{ onde } W_1 \neq W_2 \neq \dots \neq W_n \quad (\text{A.21})$$

e) **Quanto ao lugar da média** – Opção Asiática *floating strike* ou *fixed strike*:

1) *Floating Strike*<sup>11</sup>

$$\text{payoff}_{\text{Floating Strike}}^{\text{Call}} = \max \left[ S(t_i) - \overline{S}_{(i)}, 0 \right] = \left( S(t_i) - \overline{S}_{(i)} \right)^+ \quad (\text{A.22})$$

$$\text{payoff}_{\text{Floating Strike}}^{\text{Put}} = \max \left[ \overline{S}_{(i)} - S(t_i), 0 \right] = \left( \overline{S}_{(i)} - S(t_i) \right)^+ \quad (\text{A.23})$$

2) *Fixed Strike*<sup>12</sup>

$$\text{payoff}_{\text{Fixed Strike}}^{\text{Call}} = \max \left[ \overline{S}_{(i)} - K, 0 \right] = \left( \overline{S}_{(i)} - K \right)^+ \quad (\text{A.24})$$

$$\text{payoff}_{\text{Fixed Strike}}^{\text{Put}} = \max \left[ K - \overline{S}_{(i)}, 0 \right] = \left( K - \overline{S}_{(i)} \right)^+ \quad (\text{A.25})$$

<sup>11</sup> Alternativamente denominada, *Floating Average*.

<sup>12</sup> Alternativamente denominada, *Average Price*.



## Apêndice B

### Derivando a Equação Diferencial Parcial de uma Opção Asiática

1. Seja,  $H(S, \bar{S}, t)$  uma opção asiática em função do preço de ativo básico ( $S$ ), da média de preços do ativo base ( $\bar{S}$ ) e do instante de tempo ( $t$ ).
2. Considera-se que o ativo básico não paga dividendos e que seu preço ( $S$ ) segue um movimento geométrico browniano denotado por:

$$dS = S\mu dt + S\sigma dZ \quad (\text{B.1})$$

Onde, o incremento de Wiener,  $dZ \sim N(0, dt)$ , pode ser reescrito como  $dZ = \varepsilon\sqrt{dt}$ , em que  $\varepsilon \sim N(0,1)$ .

3. A média de preços ( $\bar{S}$ ) é definida de acordo com a opção asiática estudada. Desta forma, definamos a média aritmética contínua ( $A$ ) e a média geométrica contínua ( $G$ ):

$$A = \frac{1}{t} \int_0^t S(\tau) d\tau \quad (\text{B.2})$$

$$G = \frac{1}{t} \int_0^t \ln(S(\tau)) d\tau \quad (\text{B.3})$$

4. Introduzamos o lema de Ito dado por:

$$dF = \frac{\partial F}{\partial t} dt + \sum_i \frac{\partial F}{\partial x_i} dx_i + \frac{1}{2} \sum_i \sum_j \frac{\partial^2 F}{\partial x_i \partial x_j} dx_i dx_j \quad (\text{B.4})$$

Aplicando o lema de Ito em (B.2) e (B.3), temos:

$$dA = -\frac{1}{t^2} \left[ \int_0^t S(\tau) d\tau \right] dt + \frac{1}{t} S(t) dt = \frac{1}{t} (S - A) dt \quad (\text{B.5})$$

$$dG = -\frac{1}{t^2} \left[ \int_0^t \text{Ln}(S(\tau)) d\tau \right] dt + \frac{1}{t} \text{Ln}(S(t)) dt = \frac{1}{t} (\text{Ln}(S(t)) - G) dt \quad (\text{B.6})$$

5. Montemos uma carteira ( $\Psi$ ) composta de uma unidade do derivativo asiático ( $H$ ) e  $n$  unidades do ativo básico ( $S$ ).

$$\Psi = H + nS \quad (\text{B.7})$$

$$d\Psi = dH + n dS \quad (\text{B.8})$$

Encontra-se  $dH$  aplicando o lema de Ito (1.4), logo:

$$dH = \frac{\partial H}{\partial t} dt + \frac{\partial H}{\partial \bar{S}} d\bar{S} + \frac{\partial H}{\partial S} dS + \frac{1}{2} \left[ \frac{\partial^2 H}{\partial S^2} (dS)^2 + \frac{\partial^2 H}{\partial \bar{S}^2} (d\bar{S})^2 + \frac{\partial^2 H}{\partial S \partial \bar{S}} dS d\bar{S} + \frac{\partial^2 H}{\partial \bar{S} \partial S} d\bar{S} dS \right]$$

Como,  $\forall \alpha > 1, (dt)^\alpha = 0 \Rightarrow (d\bar{S})^2 = 0; d\bar{S} dS = 0; (dS)^2 = \sigma^2 S^2 dt$ ,

portanto,

$$dH = \frac{\partial H}{\partial t} dt + \frac{\partial H}{\partial \bar{S}} d\bar{S} + \frac{\partial H}{\partial S} dS + \frac{1}{2} \frac{\partial^2 H}{\partial S^2} \sigma^2 S^2 dt \quad (\text{B.9})$$

6. Considerando que  $\frac{\partial H}{\partial S} \equiv H_s$ ,  $\frac{\partial^2 H}{\partial S^2} \equiv H_{ss}$ ,  $\frac{\partial H}{\partial t} \equiv H_t$ ,  $\frac{\partial H}{\partial \bar{S}} \equiv H_{\bar{S}}$  e substituindo

(B.9) e (B.1) em (B.8) tem-se:

$$d\Psi = dH + n dS = \left[ H_t dt + H_{\bar{S}} d\bar{S} + H_s dS + \frac{1}{2} H_{ss} \sigma^2 S^2 dt \right] + n [S \mu dt + S \sigma dZ] \Rightarrow$$

$$d\Psi = \left[ H_t dt + H_{\bar{S}} d\bar{S} + H_S (S\mu dt + S\sigma dZ) + \frac{1}{2} H_{SS} \sigma^2 S^2 dt \right] + n[S\mu dt + S\sigma dZ] \Rightarrow$$

$$d\Psi = \left[ H_t dt + H_{\bar{S}} d\bar{S} + H_S S\mu dt + \frac{1}{2} H_{SS} \sigma^2 S^2 dt + nS\mu dt \right] + H_S S\sigma dZ + nS\sigma dZ \Rightarrow$$

Eliminaremos o componente de risco da carteira ( $\Psi$ ) se mantivermos  $n = -H_S$  unidades do ativo básico, assim:

$$d\Psi = H_t dt + H_{\bar{S}} d\bar{S} + H_S S\mu dt + \frac{1}{2} H_{SS} \sigma^2 S^2 dt + nS\mu dt \Rightarrow$$

$$d\Psi = H_t dt + H_{\bar{S}} d\bar{S} + H_S S\mu dt + \frac{1}{2} H_{SS} \sigma^2 S^2 dt - H_S S\mu dt \Rightarrow$$

$$d\Psi = H_t dt + H_{\bar{S}} d\bar{S} + \frac{1}{2} H_{SS} \sigma^2 S^2 dt \quad (\text{B.10})$$

7. Na ausência de oportunidades de arbitragem a carteira sem risco ( $\Psi$ ) é remunerada à taxa livre de risco ( $r$ ), conforme descrito abaixo:

$$d\Psi = r \Psi dt \quad (\text{B.11})$$

8. Igualando (B.11) à (B.10) e substituindo (B.7):

$$H_t dt + H_{\bar{S}} d\bar{S} + \frac{1}{2} H_{SS} \sigma^2 S^2 dt = r \Psi dt \Rightarrow H_t dt + H_{\bar{S}} d\bar{S} + \frac{1}{2} H_{SS} \sigma^2 S^2 dt - r \Psi dt = 0 \Rightarrow$$

$$H_t dt + H_{\bar{S}} d\bar{S} + \frac{1}{2} H_{SS} \sigma^2 S^2 dt - rH dt + rH_S S dt = 0 \quad (\text{B.12})$$

9. Por fim, substituindo (B.5) e (B.6) em (B.12) obtemos, respectivamente, a Equação Diferencial de uma Opção Asiática baseada em Média Aritmética:

$$H_t + \frac{1}{t}(S - A)H_A + \frac{1}{2}H_{SS} \sigma^2 S^2 + rH_S S - rH = 0 \quad (\text{B.13})$$

E a Equação Diferencial de uma Opção Asiática baseada em Média Geométrica:

$$H_t + \frac{1}{t}(\ln(S(t)) - G)H_G + \frac{1}{2}H_{SS} \sigma^2 S^2 + rH_S S - rH = 0 \quad (\text{B.14})$$

Note que tanto (B.13) quanto (B.14) não possuem termo de difusão na direção de suas respectivas médias, A e G. Nesse sentido, tanto (B.13) quanto (B.14) são degeneradas dado que só possuem termo de convecção na direção da média. Conforme ressaltam Zvan, Vetzal & Forsyth (1998), a utilização de técnicas de diferenças finitas tradicionais sem refinamentos sucessivos na direção da média, graças a degeneração encontrada, implica em soluções que não estão livres de oscilações espúrias.

## Apêndice C

### Procedimento Operacional do MQMC

#### C.1 - Discretização do preço do ativo básico

Assumindo a hipótese de neutralidade ao risco, o comportamento do preço do ativo básico ( $S$ ) é descrito por:

$$dS = S(r - y)dt + S\sigma dZ \quad (C.1)$$

Em que  $y$  é o *dividend yield*.

Utilizemos a relação  $W = \ln(S)$  na qual aplica-se o lema de Ito,

$$dW = \frac{\partial W}{\partial t} dt + \frac{\partial W}{\partial S} dS + \frac{1}{2} \frac{\partial^2 W}{\partial S^2} (dS)^2 \quad (C.2)$$

Em que:  $\frac{\partial W}{\partial t} = 0$ ;  $\frac{\partial W}{\partial S} = \frac{1}{S}$ ;  $\frac{\partial^2 W}{\partial S^2} = -\frac{1}{S^2}$ ;  $(dS)^2 = \sigma^2 S^2 dt$ .

$$dW = \frac{1}{S} dS + \frac{1}{2} \left( -\frac{1}{S^2} \right) \sigma^2 S^2 dt \quad (C.3)$$

Substituindo (C.1) em (C.3) e realizando as simplificações devidas tem-se:

$$dW = \left( r - y - \frac{1}{2} \sigma^2 \right) dt + \sigma dz \quad (C.4)$$

Recapitulando o apêndice anterior, pode-se reescrever  $dz = \varepsilon \sqrt{dt}$ , em que  $\varepsilon \sim N(0,1)$ , portanto:

$$dW = \left( r - y - \frac{1}{2} \sigma^2 \right) dt + \sigma \varepsilon \sqrt{dt} \quad (C.5)$$

Da relação  $W = \ln(S)$  temos que:

$$W = \ln(S) \Rightarrow dW = d(\ln S) = \ln(S_{t+1}) - \ln(S_t) = \ln\left(\frac{S_{t+1}}{S_t}\right) \quad (\text{C.6})$$

Ademais, dado que o intervalo infinitesimal  $dt$  pode ser aproximado discretamente por  $\Delta t \rightarrow 0$  e o resultado (C.6), então (C.5) é reescrita como,

$$\ln\left(\frac{S_{t+1}}{S_t}\right) = \left(r - y - \frac{1}{2}\sigma^2\right)\Delta t + \sigma\varepsilon\sqrt{\Delta t} \quad (\text{C.7})$$

$$S_{t+1} = S_t \cdot \exp\left\{\left(r - y - \frac{1}{2}\sigma^2\right)\Delta t + \sigma\varepsilon\sqrt{\Delta t}\right\} \quad (\text{C.8})$$

Por fim, a discretização do preço do ativo básico é dada por (C.8).

## C.2 – Regressão de Mínimos Quadrados (procedimento operacional)

Seja  $H(\mathbf{S}, \bar{\mathbf{S}}, t)$  um derivativo em função das variáveis de estado  $\mathbf{S}$ , preço do ativo base,  $\bar{\mathbf{S}}$ , média de preços do ativo base e do tempo,  $t$ . O tempo é observado discretamente e dividido em intervalos igualmente espaçados sendo denotado por  $t_j \equiv j \cdot \Delta t$ ,  $j = 1, \dots, n$ .

Definamos, portanto, as matrizes que se seguem:

$$\mathbf{S} = \begin{bmatrix} S_{1,1} & S_{1,2} & \cdots & S_{1,n} \\ S_{2,1} & S_{2,2} & \cdots & S_{2,n} \\ \vdots & \vdots & \ddots & \vdots \\ S_{m,1} & S_{m,2} & \cdots & S_{m,n} \end{bmatrix}, \quad \bar{\mathbf{S}} = \begin{bmatrix} \bar{S}_{1,1} & \bar{S}_{1,2} & \cdots & \bar{S}_{1,n} \\ \bar{S}_{2,1} & \bar{S}_{2,2} & \cdots & \bar{S}_{2,n} \\ \vdots & \vdots & \ddots & \vdots \\ \bar{S}_{m,1} & \bar{S}_{m,2} & \cdots & \bar{S}_{m,n} \end{bmatrix} \text{ onde } S_{i,j} \text{ e } \bar{S}_{i,j}$$

representam, respectivamente, o preço do ativo e a média de preço observados na  $i$ -ésima trajetória simulada e no  $j$ -ésimo instante de tempo. Para efeito de notação, considera-se como instante inicial,  $t=1$  e não o instante  $t=0$ , como na notação usual.

A matriz de preços é constituída a partir da simulação de  $m$  trajetórias utilizando-se, para tanto, da Simulação de Monte Carlo combinada com a técnica de variáveis antitéticas, ou seja, substitui-se reiteradas vezes em

$s_{i,j+1} = s_{i,j} e^{\left(r-y-\frac{\sigma^2}{2}\right)\Delta t + \sigma \varepsilon \sqrt{\Delta t}}$  a variável  $\varepsilon \sim N(0,1)$  sorteada  $n-1$  vezes de modo a determinar a trajetória  $i$ .

Em se tratando de variáveis antitéticas, toda a vez em que  $\varepsilon \sim N(0,1)$  é sorteada para determinar o *path*  $i$ ,  $i = 1, 2, \dots, \frac{m}{2}$ , computa-se seu simétrico,  $-\varepsilon \sim N(0,1)$ , de modo a determinar a trajetória  $i + \frac{m}{2}$ ,  $i + \frac{m}{2} = 1 + \frac{m}{2}, 2 + \frac{m}{2}, \dots, m$ , este procedimento diminui pela metade o número de trajetórias geradas.

A seguir, a matriz de médias é definida de acordo com tipo de média utilizada. De tal forma que  $\forall i = 1, \dots, m$  e  $\forall j = 1, \dots, n$ ,  $\bar{s}_{i,j} \equiv \frac{\sum_{k=1}^j s_{i,k}}{j}$ , se estivermos tratando de uma opção asiática baseada em média aritmética, ou então  $\bar{s}_{i,j} \equiv \left[ \prod_{k=1}^j s_{i,k} \right]^{1/j}$ , se estivermos lidando com uma opção asiática baseada em média geométrica.

As matrizes de *input* da regressão são concernentes às bases polinomiais ortogonais escritas em função das variáveis de estado definidas anteriormente, em que  $\Phi_n(x)$  é a base polinomial de grau  $n$  em função de uma variável de estado  $x$ . Portanto, seja:

$$\Phi_1(S) \equiv \mathbf{B}_{m,n} = \begin{bmatrix} b_{1,1} & b_{1,2} & \cdots & b_{1,n} \\ b_{2,1} & b_{2,2} & \cdots & b_{2,n} \\ \vdots & \vdots & & \vdots \\ b_{m,1} & b_{m,2} & \cdots & b_{m,n} \end{bmatrix} = [\mathbf{b}_1, \mathbf{b}_2, \dots, \mathbf{b}_n]$$

Analogamente,

$$\begin{aligned}\Phi_2(\mathbf{S}) &\equiv \mathbf{C}_{m,n} = [\mathbf{c}_1, \mathbf{c}_2, \dots, \mathbf{c}_n]; \quad \Phi_1(\bar{\mathbf{S}}) \equiv \mathbf{D}_{m,n} = [\mathbf{d}_1, \mathbf{d}_2, \dots, \mathbf{d}_n]; \\ \Phi_2(\bar{\mathbf{S}}) &\equiv \mathbf{E}_{m,n} = [\mathbf{e}_1, \mathbf{e}_2, \dots, \mathbf{e}_n]; \quad \Phi_1(\mathbf{S}) \odot \Phi_1(\bar{\mathbf{S}}) \equiv \mathbf{Z}_{m,n} = [\mathbf{z}_1, \mathbf{z}_2, \dots, \mathbf{z}_n]; \\ \Phi_2(\mathbf{S}) \odot \Phi_1(\bar{\mathbf{S}}) &\equiv \mathbf{G}_{m,n} = [\mathbf{g}_1, \mathbf{g}_2, \dots, \mathbf{g}_n]; \quad \Phi_1(\mathbf{S}) \odot \Phi_2(\bar{\mathbf{S}}) \equiv \mathbf{H}_{m,n} = [\mathbf{h}_1, \mathbf{h}_2, \dots, \mathbf{h}_n].\end{aligned}$$

Assim, a cada instante de tempo  $j$  em que há a possibilidade de exercício antecipado, utiliza-se a equação que se segue para aproximar diretamente a função valor de continuação:

$$F_j = \alpha_0 + \alpha_1 \mathbf{b}_j + \alpha_2 \mathbf{c}_j + \alpha_3 \mathbf{d}_j + \alpha_4 \mathbf{e}_j + \alpha_5 \mathbf{z}_j + \alpha_6 \mathbf{g}_j + \alpha_7 \mathbf{h}_j$$

A regressão de mínimos quadrados a empreendida a cada instante tempo  $j$  é,  $F_j = \mathbf{X}\beta_j + u_j$ , em que  $\beta_j^T = [\alpha_0, \alpha_1, \alpha_2, \alpha_3, \alpha_4, \alpha_5, \alpha_6, \alpha_7]$  e  $\mathbf{X} = [\mathbf{1}, \mathbf{b}_j, \mathbf{c}_j, \mathbf{d}_j, \mathbf{e}_j, \mathbf{z}_j, \mathbf{g}_j, \mathbf{h}_j]$ .

A fim de simplificar a notação o subscrito  $j$  será suprimido, o que não obsta o fato de estarmos executando a regressão de mínimos quadrados.

Sendo assim, seja,

$$\begin{aligned}\min_{\beta} u^T u &\equiv \xi = (\mathbf{F} - \mathbf{X}\beta)^T (\mathbf{F} - \mathbf{X}\beta) = (\mathbf{F}^T - \beta^T \mathbf{X}^T) (\mathbf{F} - \mathbf{X}\beta) = \\ &= \mathbf{F}^T \mathbf{F} - \mathbf{F}^T \mathbf{X}\beta - \beta^T \mathbf{X}^T \mathbf{F} + \beta^T \mathbf{X}^T \mathbf{X}\beta\end{aligned}$$

Portanto,

$$\frac{\partial \xi}{\partial \beta} = -2\mathbf{X}^T \mathbf{F} + 2\mathbf{X}^T \mathbf{X}\beta = 0 \Rightarrow (\mathbf{X}^T \mathbf{X})\beta = \mathbf{X}^T \mathbf{F} \Rightarrow \beta = (\mathbf{X}^T \mathbf{X})^{-1} (\mathbf{X}^T \mathbf{F})$$

Deve-se ressaltar que, é necessário que a matriz  $(\mathbf{X}^T \mathbf{X})$  seja não singular, i.e. tenha inversa, para que o vetor de coeficientes seja determinado.

Uma vez determinado o vetor de coeficientes, pode-se computar o valor de continuação substituindo-se os valores das respectivas variáveis de estado do instante  $j$  na regressão. Dado que o valor intrínseco é determinado pela função de *payoff* logo o detentor da opção pode decidir entre exercer ou não a mesma. Isto faz com que se possa determinar no instante  $j$  a decisão ótima em cada uma das trajetórias condicional ao fato de não ter havido anteriormente o exercício da opção. Este procedimento é repetido a todo instante em que o exercício antecipado é possível, de modo a aproximar o valor de continuação.



## Apêndice D

### Desenvolvimento dos Formatos dos Polinômios Usados nas Simulações

#### ***Polinômio de Legendre – Escrito através da Fórmula de Rodrigues***

Partindo da definição da Fórmula de Rodrigues,

$$f_n(x) = \frac{1}{a_n \cdot \rho(x)} \cdot \frac{\partial^n}{\partial x^n} [\rho(x) \cdot (g(x))^n], \text{ e que considerando seus termos para o}$$

polinômio de Legendre, ou seja,  $a_n = (-1)^n \cdot 2^n \cdot n!$ ;  $\rho(x) = 1$ ; e  $g(x) = 1 - x^2$ ,

$$\text{tem-se que: Se } n = 1 \Rightarrow P_1(x) = \frac{1}{(-2)}(-2x) = x$$

$$\text{Se } n = 2 \Rightarrow P_2(x) = \frac{1}{8}(12x^2 - 4) = \frac{3x^2 - 1}{2}$$

#### ***Polinômio de Hermite A – Escrito através da Lei de Recorrência***

Partindo da Lei de Recorrência,

$$a_{n+1} \cdot f_{n+1}(x) = (a_n + b_n \cdot x) \cdot f_n(x) - a_{n-1} \cdot f_{n-1}(x), \text{ e considerando seus termos para o}$$

polinômio de Hermite A, ou seja,  $a_{n+1} = 1$ ;  $a_n = 0$ ;  $b_n = 2$ ;  $a_{n-1} = 2 \cdot n$ ;  $f_0(x) = 1$  e

$$f_1(x) = 2 \cdot x, \text{ tem-se que: Se } n = 1 \Rightarrow H_1(x) = 2 \cdot x$$

$$\text{Se } n = 2 \Rightarrow H_2(x) = 4x^2 - 2$$

#### ***Polinômios de Potência e de Laguerre – Escritos através da Forma Explícita***

Partindo da definição da Forma Explícita,  $f_n = d_n \sum_{m=0}^N c_m \cdot g_m(x)$ , e

considerando seus termos para o polinômio de Potência,  $W_n(x)$ , ou seja,  $N = 0$

;  $d_n = 1$ ;  $c_m = 1$ ;  $g_m(x) = x^n$  e os termos referentes ao polinômio de Laguerre,

$L_n(x)$ , os quais são,  $N = n$ ;  $d_n = 1$ ;  $c_m = \frac{(-1)^m}{m!} \cdot \binom{n}{n-m}$ ;  $g_m(x) = x^m$ , então, tem-

$$\text{se: Se } n = 1 \Rightarrow W_1(x) = x; L_1(x) = (1 - x)$$

$$\text{Se } n = 2 \Rightarrow W_2(x) = x^2; L_2(x) = 1 - 2x + \frac{1}{2}x^2$$

## Apêndice E

### Rotinas Implementadas em MATLAB 6.5

#### E.1 Regressão Base de Potência

```
function [vector] = regress_mod_131109(A,B,D,C,grau)
%
Y = [];
Y = A;
%
X = [];
X = B;
%
M = [];
M = D;
%
Z = [];
Z = C;
%
W = [];
n = length(Y);
YY = [];
XX = [];
MM = [];
i = 1;
for j = 1:n
    if Z(j)==0;
        W(j,1)=0;
    else
        YY(i,1)=Y(j);
        XX(i,1)=X(j);
        MM(i,1)=M(j);
        W(j,1)=1;
        i = i+1;
    end
end
%
if (grau==1)
    matrixx = [ones(size(XX)) XX MM XX.*MM];
    matrix = [ones(size(X)) X M X.*M];
elseif (grau==2)
    matrixx = [ones(size(XX)) XX MM XX.*XX MM.*MM XX.*MM XX.*MM.*MM
    XX.*XX.*MM];
    matrix = [ones(size(X)) X M X.*X M.*M X.*M X.*M.*M X.*X.*M];
end
%
coef = matrixx\YY;
vector = (matrix*coef).*W;
```

#### E.2 Amerasian Fixed Strike Aritmética - Base de Potência

```
function [put_value,call_value] =
Amerasian_Arit_Fixed_Strike(s,k,T,r,v,y,lapso,NumSim,DivTemp,grau)
% painel de controle
s = 100;      % preco inicial ativo basico
k = 100;      % strike price
T = 1;        % período de vida da opcao
r = 0.05;     % riskless rate
v = 0.2;      % volatilidade
y = 0;        % dividend
```

```

lapso = 3; % variavel lapso tempo
NumSim = 100; % Numero de simulacoes
DivTemp = 12; % particoes de tempo
grau = 2; % se grau 1 >> cte + 3; se grau 2 >> cte + 7
%
dt = T/DivTemp;
prices = zeros(NumSim,DivTemp);
for i = 1:NumSim;
    prices(i,1) = s;
end
for j = 2:DivTemp
    for i = 1:(NumSim/2)
        eps = normrnd(0,1);
        prices(i,j) = prices(i,j-1)*exp((r-y-((v^2)/2))*dt +
            v*eps*sqrt(dt));
        prices(i+(NumSim/2), j) = prices(i+(NumSim/2), j-1)*exp((r-y-
            ((v^2)/2))*dt - v*eps*sqrt(dt));
    end
end
media = zeros(NumSim,DivTemp);
for j = DivTemp:(-1):1
    for i = 1:NumSim
        media(i,j) = (sum(prices(i,1:j)))/j;
    end
end
option_put = max(k-media,0);
continue_value = option_put(:,DivTemp);
for j = (DivTemp-1):(-1):lapso
    temp = regress_mod_131109(exp(-r*dt)*continue_value,
        prices(:,j),media(:,j),option_pu
        t(:,j),grau);
    for i = 1:NumSim
        if (option_put(i,j)>temp(i))
            continue_value(i) = option_put(i,j);
        else
            continue_value(i) = exp(-r*dt)*continue_value(i);
        end
    end
end
put = sum(continue_value)/NumSim;
put_value = put*exp(-r*(lapso-1)*dt)
option = max(media-k,0);
continue_value = option(:,DivTemp);
for j = (DivTemp-1):(-1):lapso
    temp = regress_mod_131109(exp(-r*dt)*continue_value,
        prices(:,j),media(:,j),option_pu
        t(:,j),grau);
    for i = 1:NumSim
        if (option(i,j) > temp(i))
            continue_value(i) = option(i,j);
        else
            continue_value(i) = exp(-r*dt)*continue_value(i);
        end
    end
end
call = sum(continue_value)/NumSim;
call_value = call*exp(-r*(lapso-1)*dt)

```

### E.3 Amerasian Floating Strike Aritmética – Base de Potência

```

function [put_value,call_value] =
Amerasian_Arit_Float_Strike(s,T,r,v,y,lapso,NumSim,DivTemp,grau)
% painel de controle
s = 100;
T = 1;
r = 0.05;
v = 0.2;
y = 0;
lapso = 3;
NumSim = 100;
DivTemp = 12;
grau = 2;
%
dt = T/DivTemp;
prices = zeros(NumSim,DivTemp);
for i = 1:NumSim;
prices(i,1) = s;
end
for j = 2:DivTemp
for i = 1:(NumSim/2)
eps = normrnd(0,1);
prices(i,j) = prices(i,j-1)*exp((r-y-((v^2)/2))*dt +
v*eps*sqrt(dt));
prices(i+(NumSim/2), j) = prices(i+(NumSim/2), j-1)*exp((r-y-
((v^2)/2))*dt - v*eps*sqrt(dt));
end
end
media = zeros(NumSim,DivTemp);
for j = DivTemp:(-1):1
for i = 1:NumSim
media(i,j) = (sum(prices(i,1:j)))/j;
end
end
option_put = zeros(NumSim,DivTemp);
for j=DivTemp:(-1):1
for i = 1:NumSim
option_put(i,j) = max(((media(i,j))-(prices(i,j))),0);
end
end
continue_value = option_put(:,DivTemp);
for j = (DivTemp-1):(-1):lapso
temp = regress_mod_l3l109(exp(-r*dt)*continue_value,
prices(:,j),media(:,j),option_pu
t(:,j),grau);
for i = 1:NumSim
if (option_put(i,j)>temp(i))
continue_value(i) = option_put(i,j);
else
continue_value(i) = exp(-r*dt)*continue_value(i);
end
end
end
put = sum(continue_value)/NumSim;
put_value = put*exp(-r*(lapso-1)*dt)
option = zeros(NumSim,DivTemp);
for j=DivTemp:(-1):1
for i = 1:NumSim
option(i,j) = max(((prices(i,j))-(media(i,j))),0);
end

```

```

end
continue_value = option(:,DivTemp);
for j = (DivTemp-1):(-1):lapso
temp = regress_mod_131109(exp(-r*dt)*continue_value,
prices(:,j),media(:,j),option_pu
t(:,j),grau);
for i = 1:NumSim
if (option(i,j) > temp(i))
continue_value(i) = option(i,j);
else
continue_value(i) = exp(-r*dt)*continue_value(i);
end
end
end
call = sum(continue_value)/NumSim;
call_value = call*exp(-r*(lapso-1)*dt)

```

#### E.4 Amerasian Fixed Strike Geométrica – Base de Potência

```

function [put_value,call_value] =
geom_american_fixedstrike_MQMC(s,k,T,r,v,y,lapso,NumSim,DivTemp,g
rau)
% painel de controle
s = 100;
k = 100;
T = 1;
r = 0.05;
v = 0.2;
y = 0;
lapso = 3;
NumSim = 100;
DivTemp = 12;
grau = 2;
%
dt = T/DivTemp;
prices = zeros(NumSim,DivTemp);
for i = 1:NumSim;
prices(i,1) = s;
end
for j = 2:DivTemp
for i = 1:(NumSim/2)
eps = normrnd(0,1);
prices(i,j) = prices(i,j-1)*exp((r-y-((v^2)/2))*dt +
v*eps*sqrt(dt));
prices(i+(NumSim/2), j) = prices(i+(NumSim/2), j-1)*exp((r-y-
((v^2)/2))*dt - v*eps*sqrt(dt));
end
end
media = ones(NumSim,DivTemp);
for j = DivTemp:(-1):1
for i = 1:NumSim
media(i,j) = prod((prices(i,1:j)).^(1/j));
end
end
option_put = max(k-media,0);
continue_value = option_put(:,DivTemp);
for j = (DivTemp-1):(-1):lapso
temp = regress_mod_131109(exp(-r*dt)*continue_value,
prices(:,j),media(:,j),option_pu
t(:,j),grau);
for i = 1:NumSim

```

```

if (option_put(i,j)>temp(i))
continue_value(i) = option_put(i,j);
else
continue_value(i) = exp(-r*dt)*continue_value(i);
end
end
end
put = sum(continue_value)/NumSim;
put_value = put*exp(-r*(lapso-1)*dt)
option = max(media-k,0);
continue_value = option(:,DivTemp);
for j = (DivTemp-1):(-1):lapso
temp = regress_mod_131109(exp(-r*dt)*continue_value,
prices(:,j),media(:,j),option_pu
t(:,j),grau);
for i = 1:NumSim
if (option(i,j) > temp(i))
continue_value(i) = option(i,j);
else
continue_value(i) = exp(-r*dt)*continue_value(i);
end
end
end
call = sum(continue_value)/NumSim;
call_value = call*exp(-r*(lapso-1)*dt)

```

## E.5 Amerasian Floating Strike Geométrica – Base de Potência

```

function [put_value,call_value] =
geom_amerasian_floatstrike_MQMC(s,T,r,v,y,lapso,NumSim,DivTemp,grau)
% painel de controle
s = 100;
T = 1;
r = 0.05;
v = 0.2;
y = 0;
lapso = 3;
NumSim = 10000;
DivTemp = 12;
grau = 2;
%
dt = T/DivTemp;
prices = zeros(NumSim,DivTemp);
for i = 1:NumSim;
prices(i,1) = s;
end
for j = 2:DivTemp
for i = 1:(NumSim/2)
eps = normrnd(0,1);
prices(i,j) = prices(i,j-1)*exp((r-y-((v^2)/2))*dt +
v*eps*sqrt(dt));
prices(i+(NumSim/2), j) = prices(i+(NumSim/2), j-1)*exp((r-y-
((v^2)/2))*dt - v*eps*sqrt(dt));
end
end
media = zeros(NumSim,DivTemp);
for j = DivTemp:(-1):1
for i = 1:NumSim
media(i,j) = prod((prices(i,1:j)).^(1/j));
end

```

```

end
option_put = zeros(NumSim,DivTemp);
for j=DivTemp:(-1):1
for i = 1:NumSim
option_put(i,j) = max(((media(i,j))-(prices(i,j)))),0);
end
end
continue_value = option_put(:,DivTemp);
for j = (DivTemp-1):(-1):lapso
temp = regress_mod_131109(exp(-r*dt)*continue_value,
prices(:,j),media(:,j),option_pu
t(:,j),grau);
for i = 1:NumSim
if (option_put(i,j)>temp(i))
continue_value(i) = option_put(i,j);
else
continue_value(i) = exp(-r*dt)*continue_value(i);
end
end
end
put = sum(continue_value)/NumSim;
put_value = put*exp(-r*(lapso-1)*dt)
option = zeros(NumSim,DivTemp);
for j=DivTemp:(-1):1
for i = 1:NumSim
option(i,j) = max(((prices(i,j))-(media(i,j)))),0);
end
end
continue_value = option(:,DivTemp);
for j = (DivTemp-1):(-1):lapso
temp = regress_mod_131109(exp(-r*dt)*continue_value,
prices(:,j),media(:,j),option_pu
t(:,j),grau);
for i = 1:NumSim
if (option(i,j) > temp(i))
continue_value(i) = option(i,j);
else
continue_value(i) = exp(-r*dt)*continue_value(i);
end
end
end
call = sum(continue_value)/NumSim;
call_value = call*exp(-r*(lapso-1)*dt)

```

## E.6 Regressão Base de Hermite A

```

function [vector] = regress_Hermite(A,B,D,C,grau)
%
Y = [];
Y = A;
%
X = [];
X = B;
%
M = [];
M = D;
%
Z = [];
Z = C;
%
W = [];

```

```

n = length(Y);
YY = [];
XX = [];
MM = [];
i = 1;
for j = 1:n
    if Z(j)==0;
        W(j,1)=0;
    else
        YY(i,1)=Y(j);
        XX(i,1)=X(j);
        MM(i,1)=M(j);
        W(j,1)=1;
        i = i+1;
    end
end
%% definindo matrizes auxiliares
H = zeros(size(X));
HH = zeros(size(XX));
J = zeros(size(M));
JJ = zeros(size(MM));
F = zeros(size(X));
FF = zeros(size(XX));
G = zeros(size(M));
GG = zeros(size(MM));
E = zeros(size(X));
EE = zeros(size(XX));
U = zeros(size(M));
UU = zeros(size(MM));
%
H = 1;
HH = 1;
J = 1;
JJ = 1;
%%
F = 2.*X;
FF = 2.*XX;
G = 2.*M;
GG = 2.*MM;
%%
E = ((4.*X.*X)-2);
EE = ((4.*XX.*XX)-2);
U = ((4.*M.*M)-2);
UU = ((4.*MM.*MM)-2);
if (grau==1)
    matrixx = [ones(size(FF)) FF GG FF.*GG];
    matrix = [ones(size(F)) F G F.*G];
elseif (grau==2)
    matrixx = [ones(size(FF)) FF GG EE UU FF.*EE FF.*UU EE.*GG];
    matrix = [ones(size(F)) F G E U F.*E F.*U E.*G];
end
coef = matrixx\YY;
vector = (matrix*coef).*W;

```

## E.7 Amerasian Fixed Strike Aritmética – Base de Hermite A

```

function [put_value,call_value] =
Amerasian_Arit_Fixed_Strike_Hermite_A(s,k,T,r,v,y,lapso,NumSim,Div
Temp,grau)
% painel de controle
s = 100;

```



```

k = 100;
T = 1;
r = 0.05;
v = 0.2;
y = 0;
lapso = 3;
NumSim = 100;
DivTemp = 12;
grau = 2;
%
dt = T/DivTemp;
prices = zeros(NumSim,DivTemp);
for i = 1:NumSim;
prices(i,1) = s;
end
for j = 2:DivTemp
for i = 1:(NumSim/2)
eps = normrnd(0,1);
prices(i,j) = prices(i,j-1)*exp((r-y-((v^2)/2))*dt +
v*eps*sqrt(dt));
prices(i+(NumSim/2), j) = prices(i+(NumSim/2), j-1)*exp((r-y-
((v^2)/2))*dt - v*eps*sqrt(dt));
end
end
media = zeros(NumSim,DivTemp);
for j = DivTemp:(-1):1
for i = 1:NumSim
media(i,j) = (sum(prices(i,1:j)))/j;
end
end
option_put = max(k-media,0);
continue_value = option_put(:,DivTemp);
for j = (DivTemp-1):(-1):lapso
temp = regress_Hermite(exp(-r*dt)*continue_value,
prices(:,j),media(:,j),option_put(:,j),grau);
for i = 1:NumSim
if (option_put(i,j)>temp(i))
continue_value(i) = option_put(i,j);
else
continue_value(i) = exp(-r*dt)*continue_value(i);
end
end
end
put = sum(continue_value)/NumSim;
put_value = put*exp(-r*(lapso-1)*dt)
option = max(media-k,0);
continue_value = option(:,DivTemp);
for j = (DivTemp-1):(-1):lapso
temp = regress_Hermite(exp(-r*dt)*continue_value,
prices(:,j),media(:,j),option_put(:,j),grau);
for i = 1:NumSim
if (option(i,j) > temp(i))
continue_value(i) = option(i,j);
else
continue_value(i) = exp(-r*dt)*continue_value(i);
end
end
end
call = sum(continue_value)/NumSim;
call_value = call*exp(-r*(lapso-1)*dt)

```

## E.8 Amerasian Floating Strike Aritmética – Base de Hermite A

```
function [put_value,call_value] =
Amerasian_Arit_Float_Strike_Hermite_A(s,T,r,v,y,lapso,NumSim,DivTemp, grau)
% painel de controle
s = 100;
T = 1;
r = 0.05;
v = 0.2;
y = 0;
lapso = 3;
NumSim = 100;
DivTemp = 12;
grau = 2;
%
dt = T/DivTemp;
prices = zeros(NumSim,DivTemp);
for i = 1:NumSim;
prices(i,1) = s;
end
for j = 2:DivTemp
for i = 1:(NumSim/2)
eps = normrnd(0,1);
prices(i,j) = prices(i,j-1)*exp((r-y-((v^2)/2))*dt +
v*eps*sqrt(dt));
prices(i+(NumSim/2), j) = prices(i+(NumSim/2), j-1)*exp((r-y-
((v^2)/2))*dt - v*eps*sqrt(dt));
end
end
media = zeros(NumSim,DivTemp);
for j = DivTemp:(-1):1
for i = 1:NumSim
media(i,j) = (sum(prices(i,1:j)))/j;
end
end
option_put = zeros(NumSim,DivTemp);
for j=DivTemp:(-1):1
for i = 1:NumSim
option_put(i,j) = max(((media(i,j))-(prices(i,j)))),0);
end
end
continue_value = option_put(:,DivTemp);
for j = (DivTemp-1):(-1):lapso
temp = regress_Hermite(exp(-r*dt)*continue_value,
prices(:,j),media(:,j),option_put(:,j),grau);
for i = 1:NumSim
if (option_put(i,j)>temp(i))
continue_value(i) = option_put(i,j);
else
continue_value(i) = exp(-r*dt)*continue_value(i);
end
end
end
put = sum(continue_value)/NumSim;
put_value = put*exp(-r*(lapso-1)*dt)
option = zeros(NumSim,DivTemp);
for j=DivTemp:(-1):1
for i = 1:NumSim
option(i,j) = max(((prices(i,j))-(media(i,j)))),0);
```

```

end
end
continue_value = option(:,DivTemp);
for j = (DivTemp-1):(-1):lapso
temp = regress_Hermite(exp(-r*dt)*continue_value,
prices(:,j),media(:,j),option_put(:,j),grau);
for i = 1:NumSim
if (option(i,j) > temp(i))
continue_value(i) = option(i,j);
else
continue_value(i) = exp(-r*dt)*continue_value(i);
end
end
end
call = sum(continue_value)/NumSim;
call_value = call*exp(-r*(lapso-1)*dt)

```

### E.9 Amerasian Fixed Strike Geométrica – Base de Hermite A

```

function [put_value,call_value] =
Amerasian_Geom_Fixed_Strike_Hermite_A(s,k,T,r,v,y,lapso,NumSim,Div
Temp,grau)
% painel de controle
s = 100;
k = 100;
T = 1;
r = 0.05;
v = 0.2;
y = 0;
lapso = 3;
NumSim = 100;
DivTemp = 12;
grau = 2;
%
dt = T/DivTemp;
prices = zeros(NumSim,DivTemp);
for i = 1:NumSim;
prices(i,1) = s;
end
for j = 2:DivTemp
for i = 1:(NumSim/2)
eps = normrnd(0,1);
prices(i,j) = prices(i,j-1)*exp((r-y-((v^2)/2))*dt +
v*eps*sqrt(dt));
prices(i+(NumSim/2), j) = prices(i+(NumSim/2), j-1)*exp((r-y-
((v^2)/2))*dt - v*eps*sqrt(dt));
end
end
media = ones(NumSim,DivTemp);
for j = DivTemp:(-1):1
for i = 1:NumSim
media(i,j) = prod((prices(i,1:j)).^(1/j));
end
end
option_put = max(k-media,0);
continue_value = option_put(:,DivTemp);
for j = (DivTemp-1):(-1):lapso
temp = regress_Hermite(exp(-r*dt)*continue_value,
prices(:,j),media(:,j),option_put(:,j),grau);

```

```

for i = 1:NumSim
if (option_put(i,j)>temp(i))
continue_value(i) = option_put(i,j);
else
continue_value(i) = exp(-r*dt)*continue_value(i);
end
end
end
put = sum(continue_value)/NumSim;
put_value = put*exp(-r*(lapso-1)*dt)
option = max(media-k,0);
continue_value = option(:,DivTemp);
for j = (DivTemp-1):(-1):lapso
temp = regress_Hermite(exp(-r*dt)*continue_value,
prices(:,j),media(:,j),option_put(:,j),grau);
for i = 1:NumSim
if (option(i,j) > temp(i))
continue_value(i) = option(i,j);
else
continue_value(i) = exp(-r*dt)*continue_value(i);
end
end
end
call = sum(continue_value)/NumSim;
call_value = call*exp(-r*(lapso-1)*dt)

```

## E.10 Amerasian Floating Strike Geométrica – Base de Hermite A

```

function [put_value,call_value] =
Amerasian_Geom_Float_Strike_Hermite_A(s,T,r,v,y,lapso,NumSim,DivTemp, grau)
% painel de controle
s = 100;
T = 1;
r = 0.05;
v = 0.2;
y = 0;
lapso = 3;
NumSim = 100;
DivTemp = 12;
grau = 2;
%
dt = T/DivTemp;
prices = zeros(NumSim,DivTemp);
for i = 1:NumSim;
prices(i,1) = s;
end
for j = 2:DivTemp
for i = 1:(NumSim/2)
eps = normrnd(0,1);
prices(i,j) = prices(i,j-1)*exp((r-y-((v^2)/2))*dt +
v*eps*sqrt(dt));
prices(i+(NumSim/2), j) = prices(i+(NumSim/2), j-1)*exp((r-y-
((v^2)/2))*dt - v*eps*sqrt(dt));
end
end
media = zeros(NumSim,DivTemp);
for j = DivTemp:(-1):1
for i = 1:NumSim
media(i,j) = prod((prices(i,1:j)).^(1/j));
end

```

```

end
option_put = zeros(NumSim,DivTemp);
for j=DivTemp:(-1):1
for i = 1:NumSim
option_put(i,j) = max(((media(i,j))-(prices(i,j)))),0);
end
end
continue_value = option_put(:,DivTemp);
for j = (DivTemp-1):(-1):lapso
temp = regress_Hermite(exp(-r*dt)*continue_value,
prices(:,j),media(:,j),option_put(:,j),grau);
for i = 1:NumSim
if (option_put(i,j)>temp(i))
continue_value(i) = option_put(i,j);
else
continue_value(i) = exp(-r*dt)*continue_value(i);
end
end
end
put = sum(continue_value)/NumSim;
put_value = put*exp(-r*(lapso-1)*dt)
option = zeros(NumSim,DivTemp);
for j=DivTemp:(-1):1
for i = 1:NumSim
option(i,j) = max(((prices(i,j))-(media(i,j)))),0);
end
end
continue_value = option(:,DivTemp);
for j = (DivTemp-1):(-1):lapso
temp = regress_Hermite(exp(-r*dt)*continue_value,
prices(:,j),media(:,j),option_put(:,j),grau);
for i = 1:NumSim
if (option(i,j) > temp(i))
continue_value(i) = option(i,j);
else
continue_value(i) = exp(-r*dt)*continue_value(i);
end
end
end
call = sum(continue_value)/NumSim;
call_value = call*exp(-r*(lapso-1)*dt)

```

## E.11 Regressão Base de Legendre

```

function [vector] = regress_Legendre(A,B,D,C,grau)
%
Y = [];
Y = A;
%
X = [];
X = B;
%
M = [];
M = D;
%
Z = [];
Z = C;
%
W = [];
n = length(Y);

```

```

YY = [];
XX = [];
MM = [];
i = 1;
for j = 1:n
    if Z(j)==0;
        W(j,1)=0;
    else
        YY(i,1)=Y(j);
        XX(i,1)=X(j);
        MM(i,1)=M(j);
        W(j,1)=1;
        i = i+1;
    end
end
%% definindo matrizes auxiliares
H = zeros(size(X));
HH = zeros(size(XX));
J = zeros(size(M));
JJ = zeros(size(MM));
F = zeros(size(X));
FF = zeros(size(XX));
G = zeros(size(M));
GG = zeros(size(MM));
E = zeros(size(X));
EE = zeros(size(XX));
U = zeros(size(M));
UU = zeros(size(MM));
%
H = 1;
HH = 1;
J = 1;
JJ = 1;
%%
F = X;
FF = XX;
G = M;
GG = MM;
%%%
E = ((3.*(X.^2))-1)./2;
EE = ((3.*(XX.^2))-1)./2;
U = ((3.*(M.^2))-1)./2;
UU = ((3.*(MM.^2))-1)./2;
if (grau==1)
    matrixx = [ones(size(FF)) FF GG FF.*GG];
    matrix = [ones(size(F)) F G F.*G];
elseif (grau==2)
    matrixx = [ones(size(FF)) FF GG EE UU FF.*EE FF.*UU EE.*GG];
    matrix = [ones(size(F)) F G E U F.*E F.*U E.*G];
end
coef = matrixx\YY;
vector = (matrix*coef).*W;

```

## E.12 Amerasian Fixed Strike Aritmética – Base de Legendre

```

function [put_value,call_value] =
Amerasian_Arit_Fixed_Strike_Legendre(s,k,T,r,v,y,lapso,NumSim,DivT
emp,grau)
% painel de controle
s = 100;
k = 100;

```

```

T = 1;
r = 0.05;
v = 0.2;
y = 0;
lapso = 3;
NumSim = 100;
DivTemp = 12;
grau = 2;
%
dt = T/DivTemp;
prices = zeros(NumSim,DivTemp);
for i = 1:NumSim;
prices(i,1) = s;
end
for j = 2:DivTemp
for i = 1:(NumSim/2)
eps = normrnd(0,1);
prices(i,j) = prices(i,j-1)*exp((r-y-((v^2)/2))*dt +
v*eps*sqrt(dt));
prices(i+(NumSim/2), j) = prices(i+(NumSim/2), j-1)*exp((r-y-
((v^2)/2))*dt - v*eps*sqrt(dt));
end
end
media = zeros(NumSim,DivTemp);
for j = DivTemp:(-1):1
for i = 1:NumSim
media(i,j) = (sum(prices(i,1:j)))/j;
end
end
option_put = max(k-media,0);
continue_value = option_put(:,DivTemp);
for j = (DivTemp-1):(-1):lapso
temp = regress_Legendre(exp(-r*dt)*continue_value,
prices(:,j),media(:,j),option_put(:,j),grau);
for i = 1:NumSim
if (option_put(i,j)>temp(i))
continue_value(i) = option_put(i,j);
else
continue_value(i) = exp(-r*dt)*continue_value(i);
end
end
end
put = sum(continue_value)/NumSim;
put_value = put*exp(-r*(lapso-1)*dt)
option = max(media-k,0);
continue_value = option(:,DivTemp);
for j = (DivTemp-1):(-1):lapso
temp = regress_Legendre(exp(-r*dt)*continue_value,
prices(:,j),media(:,j),option_put(:,j),grau);
for i = 1:NumSim
if (option(i,j) > temp(i))
continue_value(i) = option(i,j);
else
continue_value(i) = exp(-r*dt)*continue_value(i);
end
end
end
call = sum(continue_value)/NumSim;
call_value = call*exp(-r*(lapso-1)*dt)

```

### E.13 Amerasian Floating Strike Aritmética – Base de Legendre

```
function [put_value,call_value] =
Amerasian_Arit_Float_Strike_Legendre(s,T,r,v,y,lapso,NumSim,DivTemp,
p, grau)
% painel de controle
s = 100;
T = 1;
r = 0.05;
v = 0.2;
y = 0;
lapso = 3;
NumSim = 100;
DivTemp = 12;
grau = 2;
%
dt = T/DivTemp;
prices = zeros(NumSim,DivTemp);
for i = 1:NumSim;
prices(i,1) = s;
end
for j = 2:DivTemp
for i = 1:(NumSim/2)
eps = normrnd(0,1);
prices(i,j) = prices(i,j-1)*exp((r-y-((v^2)/2))*dt +
v*eps*sqrt(dt));
prices(i+(NumSim/2), j) = prices(i+(NumSim/2), j-1)*exp((r-y-
((v^2)/2))*dt - v*eps*sqrt(dt));
end
end
media = zeros(NumSim,DivTemp);
for j = DivTemp:(-1):1
for i = 1:NumSim
media(i,j) = (sum(prices(i,1:j)))/j;
end
end
option_put = zeros(NumSim,DivTemp);
for j=DivTemp:(-1):1
for i = 1:NumSim
option_put(i,j) = max(((media(i,j))-(prices(i,j)))),0);
end
end
continue_value = option_put(:,DivTemp);
for j = (DivTemp-1):(-1):lapso
temp = regress_Legendre(exp(-r*dt)*continue_value,
prices(:,j),media(:,j),option_put(
(:,j),grau);
for i = 1:NumSim
if (option_put(i,j)>temp(i))
continue_value(i) = option_put(i,j);
else
continue_value(i) = exp(-r*dt)*continue_value(i);
end
end
end
put = sum(continue_value)/NumSim;
put_value = put*exp(-r*(lapso-1)*dt)
option = zeros(NumSim,DivTemp);
for j=DivTemp:(-1):1
for i = 1:NumSim
option(i,j) = max(((prices(i,j))-(media(i,j)))),0);
```



```

end
end
continue_value = option(:,DivTemp);
for j = (DivTemp-1):(-1):lapso
temp = regress_Legendre(exp(-r*dt)*continue_value,
prices(:,j),media(:,j),option_put(:,j),grau);
for i = 1:NumSim
if (option(i,j) > temp(i))
continue_value(i) = option(i,j);
else
continue_value(i) = exp(-r*dt)*continue_value(i);
end
end
end
call = sum(continue_value)/NumSim;
call_value = call*exp(-r*(lapso-1)*dt)

```

### E.14 Amerasian Fixed Strike Geométrica – Base de Legendre

```

function [put_value,call_value] =
Amerasian_Geom_Fixed_Strike_Legendre(s,k,T,r,v,y,lapso,NumSim,DivT
emp,grau)
% painel de controle
s = 100;
k = 100;
T = 1;
r = 0.05;
v = 0.2;
y = 0;
lapso = 3;
NumSim = 100;
DivTemp = 12;
grau = 2;
%
dt = T/DivTemp;
prices = zeros(NumSim,DivTemp);
for i = 1:NumSim;
prices(i,1) = s;
end
for j = 2:DivTemp
for i = 1:(NumSim/2)
eps = normrnd(0,1);
prices(i,j) = prices(i,j-1)*exp((r-y-((v^2)/2))*dt +
v*eps*sqrt(dt));
prices(i+(NumSim/2), j) = prices(i+(NumSim/2), j-1)*exp((r-y-
((v^2)/2))*dt - v*eps*sqrt(dt));
end
end
media = ones(NumSim,DivTemp);
for j = DivTemp:(-1):1
for i = 1:NumSim
media(i,j) = prod((prices(i,1:j)).^(1/j));
end
end
option_put = max(k-media,0);
continue_value = option_put(:,DivTemp);
for j = (DivTemp-1):(-1):lapso
temp = regress_Legendre(exp(-r*dt)*continue_value,
prices(:,j),media(:,j),option_put(
(:,j),grau);
for i = 1:NumSim

```

```

if (option_put(i,j)>temp(i))
continue_value(i) = option_put(i,j);
else
continue_value(i) = exp(-r*dt)*continue_value(i);
end
end
end
put = sum(continue_value)/NumSim;
put_value = put*exp(-r*(lapso-1)*dt)
option = max(media-k,0);
continue_value = option(:,DivTemp);
for j = (DivTemp-1):(-1):lapso
temp = regress_Legendre(exp(-r*dt)*continue_value,
prices(:,j),media(:,j),option_put(:,j),grau);
for i = 1:NumSim
if (option(i,j) > temp(i))
continue_value(i) = option(i,j);
else
continue_value(i) = exp(-r*dt)*continue_value(i);
end
end
end
call = sum(continue_value)/NumSim;
call_value = call*exp(-r*(lapso-1)*dt)

```

### E.15 Amerasian Floating Strike Geométrica – Base de Legendre

```

function [put_value,call_value] =
Amerasian_Geom_Float_Strike_Legendre(s,T,r,v,y,lapso,NumSim,DivTem
p, grau)
% painel de controle
s = 100;
T = 1;
r = 0.05;
v = 0.2;
y = 0;
lapso = 3;
NumSim = 100;
DivTemp = 12;
grau = 2;
%
dt = T/DivTemp;
prices = zeros(NumSim,DivTemp);
for i = 1:NumSim;
prices(i,1) = s;
end
for j = 2:DivTemp
for i = 1:(NumSim/2)
eps = normrnd(0,1);
prices(i,j) = prices(i,j-1)*exp((r-y-((v^2)/2))*dt +
v*eps*sqrt(dt));
prices(i+(NumSim/2), j) = prices(i+(NumSim/2), j-1)*exp((r-y-
((v^2)/2))*dt - v*eps*sqrt(dt));
end
end
media = zeros(NumSim,DivTemp);
for j = DivTemp:(-1):1
for i = 1:NumSim
media(i,j) = prod((prices(i,1:j)).^(1/j));
end
end

```

```

option_put = zeros(NumSim,DivTemp);
for j=DivTemp:(-1):1
for i = 1:NumSim
option_put(i,j) = max(((media(i,j))-(prices(i,j)))),0);
end
end
continue_value = option_put(:,DivTemp);
for j = (DivTemp-1):(-1):lapso
temp = regress_Legendre(exp(-r*dt)*continue_value,
prices(:,j),media(:,j),option_put(
:,j),grau);
for i = 1:NumSim
if (option_put(i,j)>temp(i))
continue_value(i) = option_put(i,j);
else
continue_value(i) = exp(-r*dt)*continue_value(i);
end
end
end
put = sum(continue_value)/NumSim;
put_value = put*exp(-r*(lapso-1)*dt)
option = zeros(NumSim,DivTemp);
for j=DivTemp:(-1):1
for i = 1:NumSim
option(i,j) = max(((prices(i,j))-(media(i,j)))),0);
end
end
continue_value = option(:,DivTemp);
for j = (DivTemp-1):(-1):lapso
temp = regress_Legendre(exp(-r*dt)*continue_value,
prices(:,j),media(:,j),option_put(:,j),grau);
for i = 1:NumSim
if (option(i,j) > temp(i))
continue_value(i) = option(i,j);
else
continue_value(i) = exp(-r*dt)*continue_value(i);
end
end
end
call = sum(continue_value)/NumSim;
call_value = call*exp(-r*(lapso-1)*dt)

```

## E.16 Regressão Base de Laguerre

```

function [vector] = regress_Laguerre(A,B,D,C,grau)
%
Y = [];
Y = A;
%
X = [];
X = B;
%
M = [];
M = D;
%
Z = [];
Z = C;
%
W = [];
n = length(Y);
YY = [];

```

```

XX = [];
MM = [];
i = 1;
for j = 1:n
    if Z(j)==0;
        W(j,1)=0;
    else
        YY(i,1)=Y(j);
        XX(i,1)=X(j);
        MM(i,1)=M(j);
        W(j,1)=1;
        i = i+1;
    end
end
%% definindo matrizes auxiliares
H = zeros(size(X));
HH = zeros(size(XX));
J = zeros(size(M));
JJ = zeros(size(MM));
%
F = zeros(size(X));
FF = zeros(size(XX));
G = zeros(size(M));
GG = zeros(size(MM));
%
E = zeros(size(X));
EE = zeros(size(XX));
U = zeros(size(M));
UU = zeros(size(MM));
%
H = 1;
HH = 1;
J = 1;
JJ = 1;
%%
F = (1-X);
FF = (1-XX);
G = (1-M);
GG = (1-MM);
%%%
E = (1 - 2.*X + ((1/2).*(X.^2)));
EE = (1 - 2.*XX + ((1/2).*(XX.^2)));
U = (1 - 2.*M + ((1/2).*(M.^2)));
UU = (1 - 2.*MM + ((1/2).*(MM.^2)));
%
if (grau==1)
    matrixx = [ones(size(FF)) FF GG FF.*GG];
    matrix = [ones(size(F)) F G F.*G];
elseif (grau==2)
    matrixx = [ones(size(FF)) FF GG EE UU FF.*EE FF.*UU EE.*GG];
    matrix = [ones(size(F)) F G E U F.*E F.*U E.*G];
end
%
coef = matrixx\YY;
vector = (matrix*coef).*W;

```

### E.17 Amerasian Fixed Strike Aritmética – Base de Laguerre

```

function [put_value,call_value] =
Amerasian_Arit_Fixed_Strike_Laguerre(s,k,T,r,v,y,lapso,NumSim,DivT
emp,grau)

```

```

% painel de controle
s = 100;
k = 100;
T = 1;
r = 0.05;
v = 0.2;
y = 0;
lapso = 3;
NumSim = 100;
DivTemp = 12;
grau = 2;
%
dt = T/DivTemp;
prices = zeros(NumSim,DivTemp);
for i = 1:NumSim;
prices(i,1) = s;
end
for j = 2:DivTemp
for i = 1:(NumSim/2)
eps = normrnd(0,1);
prices(i,j) = prices(i,j-1)*exp((r-y-((v^2)/2))*dt +
v*eps*sqrt(dt));
prices(i+(NumSim/2), j) = prices(i+(NumSim/2), j-1)*exp((r-y-
((v^2)/2))*dt - v*eps*sqrt(dt));
end
end
media = zeros(NumSim,DivTemp);
for j = DivTemp:(-1):1
for i = 1:NumSim
media(i,j) = (sum(prices(i,1:j)))/j;
end
end
option_put = max(k-media,0);
continue_value = option_put(:,DivTemp);
for j = (DivTemp-1):(-1):lapso
temp = regress_Laguerre(exp(-r*dt)*continue_value,
prices(:,j),media(:,j),option_put(
:,j),grau);
for i = 1:NumSim
if (option_put(i,j)>temp(i))
continue_value(i) = option_put(i,j);
else
continue_value(i) = exp(-r*dt)*continue_value(i);
end
end
end
put = sum(continue_value)/NumSim;
put_value = put*exp(-r*(lapso-1)*dt)
%
option = max(media-k,0);
continue_value = option(:,DivTemp);
for j = (DivTemp-1):(-1):lapso
temp = regress_Laguerre(exp(-r*dt)*continue_value,
prices(:,j),media(:,j),option_put(:,j),grau);
for i = 1:NumSim
if (option(i,j) > temp(i))
continue_value(i) = option(i,j);
else
continue_value(i) = exp(-r*dt)*continue_value(i);
end
end
end

```

```

end
call = sum(continue_value)/NumSim;
call_value = call*exp(-r*(lapso-1)*dt)

```

## E.18 Amerasian Floating Strike Aritmética – Base de Laguerre

```

function [put_value,call_value] =
Amerasian_Arit_Float_Strike_Laguerre(s,T,r,v,y,lapso,NumSim,DivTem
p, grau)
% painel de controle
s = 100;
T = 1;
r = 0.05;
v = 0.2;
y = 0;
lapso = 3;
NumSim = 100;
DivTemp = 12;
grau = 2;
%
dt = T/DivTemp;
prices = zeros(NumSim,DivTemp);
for i = 1:NumSim;
prices(i,1) = s;
end
for j = 2:DivTemp
for i = 1:(NumSim/2)
eps = normrnd(0,1);
prices(i,j) = prices(i,j-1)*exp((r-y-((v^2)/2))*dt +
v*eps*sqrt(dt));
prices(i+(NumSim/2), j) = prices(i+(NumSim/2), j-1)*exp((r-y-
((v^2)/2))*dt - v*eps*sqrt(dt));
end
end
media = zeros(NumSim,DivTemp);
for j = DivTemp:(-1):1
for i = 1:NumSim
media(i,j) = (sum(prices(i,1:j)))/j;
end
end
option_put = zeros(NumSim,DivTemp);
for j=DivTemp:(-1):1
for i = 1:NumSim
option_put(i,j) = max(((media(i,j))-(prices(i,j)))),0);
end
end
continue_value = option_put(:,DivTemp);
for j = (DivTemp-1):(-1):lapso
temp = regress_Laguerre(exp(-r*dt)*continue_value,
prices(:,j),media(:,j),option_put(:,j),grau);
for i = 1:NumSim
if (option_put(i,j)>temp(i))
continue_value(i) = option_put(i,j);
else
continue_value(i) = exp(-r*dt)*continue_value(i);
end
end
end
put = sum(continue_value)/NumSim;
put_value = put*exp(-r*(lapso-1)*dt)
option = zeros(NumSim,DivTemp);

```

```

for j=DivTemp:(-1):1
for i = 1:NumSim
option(i,j) = max(((prices(i,j))-(media(i,j))),0);
end
end
continue_value = option(:,DivTemp);
for j = (DivTemp-1):(-1):lapso
temp = regress_Laguerre(exp(-r*dt)*continue_value,
prices(:,j),media(:,j),option_put(:,j),grau);
for i = 1:NumSim
if (option(i,j) > temp(i))
continue_value(i) = option(i,j);
else
continue_value(i) = exp(-r*dt)*continue_value(i);
end
end
end
call = sum(continue_value)/NumSim;
call_value = call*exp(-r*(lapso-1)*dt)

```

### E.19 Amerasian Fixed Strike Geométrica – Base de Laguerre

```

function [put_value,call_value] =
Amerasian_Geom_Fixed_Strike_Laguerre(s,k,T,r,v,y,lapso,NumSim,DivT
emp,grau)
% painel de controle
s = 100;
k = 100;
T = 1;
r = 0.05;
v = 0.2;
y = 0;
lapso = 3;
NumSim = 100;
DivTemp = 12;
grau = 2;
%
dt = T/DivTemp;
prices = zeros(NumSim,DivTemp);
for i = 1:NumSim;
prices(i,1) = s;
end
for j = 2:DivTemp
for i = 1:(NumSim/2)
eps = normrnd(0,1);
prices(i,j) = prices(i,j-1)*exp((r-y-((v^2)/2))*dt +
v*eps*sqrt(dt));
prices(i+(NumSim/2), j) = prices(i+(NumSim/2), j-1)*exp((r-y-
((v^2)/2))*dt - v*eps*sqrt(dt));
end
end
media = ones(NumSim,DivTemp);
for j = DivTemp:(-1):1
for i = 1:NumSim
media(i,j) = prod((prices(i,1:j)).^(1/j));
end
end
option_put = max(k-media,0);
continue_value = option_put(:,DivTemp);
for j = (DivTemp-1):(-1):lapso

```

```

temp = regress_Laguerre(exp(-r*dt)*continue_value,
prices(:,j),media(:,j),option_put(:,j),grau);
for i = 1:NumSim
if (option_put(i,j)>temp(i))
continue_value(i) = option_put(i,j);
else
continue_value(i) = exp(-r*dt)*continue_value(i);
end
end
end
put = sum(continue_value)/NumSim;
put_value = put*exp(-r*(lapso-1)*dt)
option = max(media-k,0);
continue_value = option(:,DivTemp);
for j = (DivTemp-1):(-1):lapso
temp = regress_Laguerre(exp(-r*dt)*continue_value,
prices(:,j),media(:,j),option_put(:,j),grau);
for i = 1:NumSim
if (option(i,j) > temp(i))
continue_value(i) = option(i,j);
else
continue_value(i) = exp(-r*dt)*continue_value(i);
end
end
end
call = sum(continue_value)/NumSim;
call_value = call*exp(-r*(lapso-1)*dt)

```

## E.20 Amerasian Floating Strike Geométrica – Base de Laguerre

```

function [put_value,call_value] =
Amerasian_Geom_Float_Strike_Laguerre(s,T,r,v,y,lapso,NumSim,DivTem
p, grau)
% painel de controle
s = 100;
T = 1;
r = 0.05;
v = 0.2;
y = 0;
lapso = 3;
NumSim = 100;
DivTemp = 12;
grau = 2;
%
dt = T/DivTemp;
prices = zeros(NumSim,DivTemp);
for i = 1:NumSim;
prices(i,1) = s;
end
for j = 2:DivTemp
for i = 1:(NumSim/2)
eps = normrnd(0,1);
prices(i,j) = prices(i,j-1)*exp((r-y-((v^2)/2))*dt +
v*eps*sqrt(dt));
prices(i+(NumSim/2), j) = prices(i+(NumSim/2), j-1)*exp((r-y-
((v^2)/2))*dt - v*eps*sqrt(dt));
end
end
media = zeros(NumSim,DivTemp);
for j = DivTemp:(-1):1
for i = 1:NumSim

```



```

media(i,j) = prod((prices(i,1:j)).^(1/j));
end
end
option_put = zeros(NumSim,DivTemp);
for j=DivTemp:(-1):1
for i = 1:NumSim
option_put(i,j) = max(((media(i,j))-(prices(i,j))),0);
end
end
continue_value = option_put(:,DivTemp);
for j = (DivTemp-1):(-1):lapso
temp = regress_Laguerre(exp(-r*dt)*continue_value,
prices(:,j),media(:,j),option_put(
:,j),grau);
for i = 1:NumSim
if (option_put(i,j)>temp(i))
continue_value(i) = option_put(i,j);
else
continue_value(i) = exp(-r*dt)*continue_value(i);
end
end
end
put = sum(continue_value)/NumSim;
put_value = put*exp(-r*(lapso-1)*dt)
option = zeros(NumSim,DivTemp);
for j=DivTemp:(-1):1
for i = 1:NumSim
option(i,j) = max(((prices(i,j))-(media(i,j))),0);
end
end
continue_value = option(:,DivTemp);
for j = (DivTemp-1):(-1):lapso
temp = regress_Laguerre(exp(-r*dt)*continue_value,
prices(:,j),media(:,j),option_put(:,j),grau);
for i = 1:NumSim
if (option(i,j) > temp(i))
continue_value(i) = option(i,j);
else
continue_value(i) = exp(-r*dt)*continue_value(i);
end
end
end
call = sum(continue_value)/NumSim;
call_value = call*exp(-r*(lapso-1)*dt)

```

## Anexo I

Esquema Básico de classificação de uma opção asiática

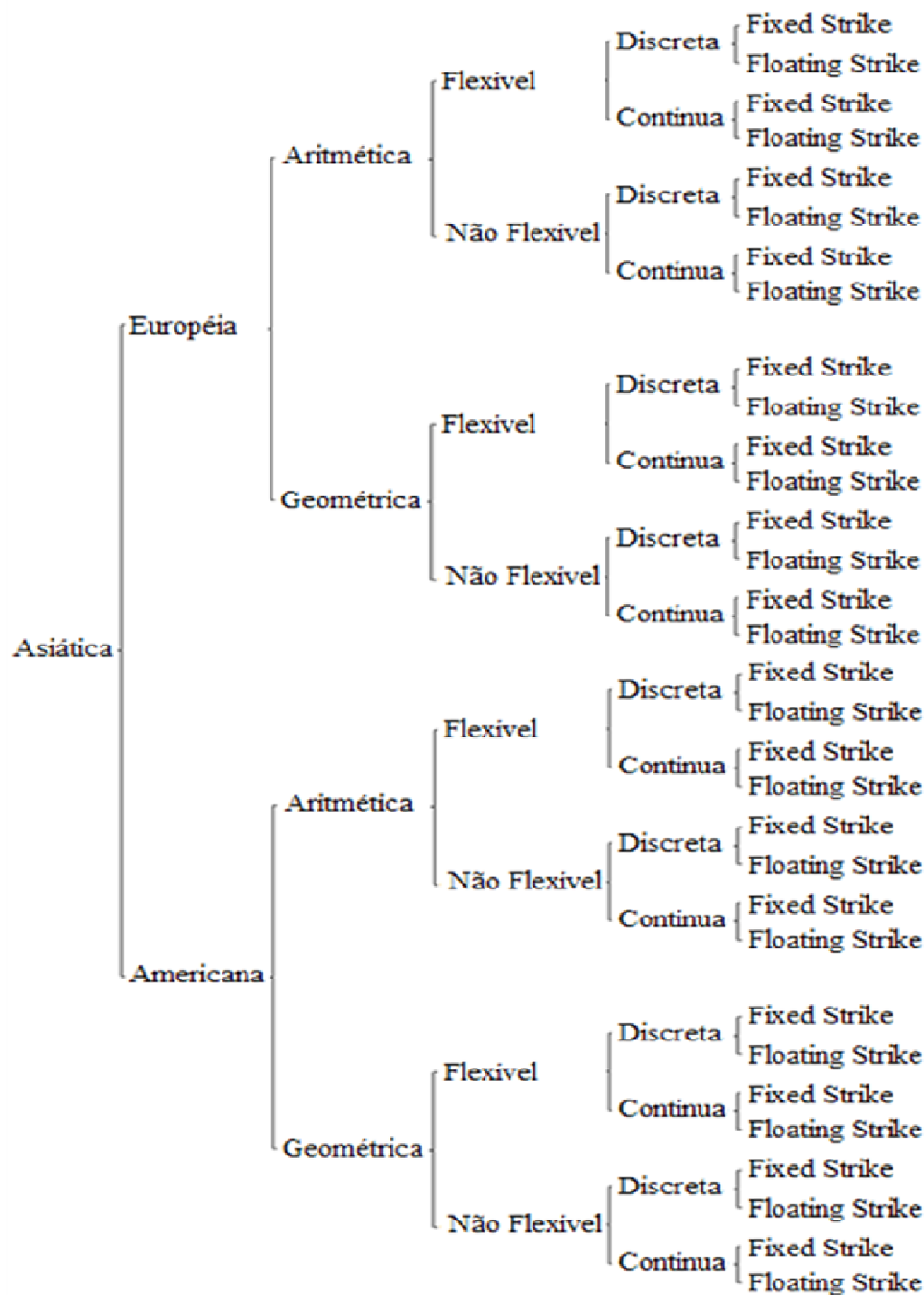


Figura 5 – Estrutura Esquemática de Classificação Básica