

Referências Bibliográficas

- ANDERSEN BROTHERTON-RATCLIFF, “**Exact exotics**”, 1996.
- BLACK F. & SHOLES M. "The Pricing of Options & Corporate Liabilities", The Journal of Political Economy. 1973
- BOYLE BROADIE GLASSERMAN, “**Monte carlo methods for security pricing**”. Journal of Economic Dynamics and Control, 1997.
- COX ROSS RUBINSTEIN, “**Option Pricing: A Simplified Approach.**” Journal of Financial Economics, 1979.
- DOUADY, “**Monte-Carlo Path Weighing**”. Working paper (CIBC and Ecole Normale Superieure of Cachan), 1999.
- DUPIRE B, “**Monte-Carlo Methodologies and Applications for Pricing and Risk Management**”,1998.
- FISHMAN, “**Monte Carlo: Concepts, Algorithms, and Applications**”, 1996.
- GLASSERMAN, “**Monte Carlo Methods in Financial Engineering**”, 2004.
- GRANT D, G. VORA, D. WEEKS. “**Simulation and the early-exercise option problem. Journal of Financial Engineering**”, 1996.
- HEYNEN R., & KAT H. "Lookback options with Discrete and Partial Monitoring of the Underlying Price", Applied Mathematical Finance, Vol 2, p273-283. 1995
- HIGHAM DESMOND J., “**An Introduction to Financial Option Valuation**”, 2004.
- HULL JOHN C., “**Options, Futures and Other Derivatives**”, 2000.
- KEMNA A. G. Z. & VORST A. C. F., "A Pricing Method for Options Based on Average Asset Values", Journal of Banking and Finance, 14, p113-129, 1990
- LONGSTAFF SCHWARTZ: “**Valuing American Options by Simulation: A Simple Least-Squares Approach**”, Review of Financial Studies vol. 14, 2001
- REINER E. & RUBINSTEIN M., “**Unscrambling the Binary Code**”. 1991
- RUBINSTEIN M., "Options for the Undecided", 1991
- ZHOU AGLEHOLE DYBVIG , “**Going to extremes: Correcting simulation in exotic option valuation**”. Financial Analysts Journal, 1997.

Apêndice 1

Calculo Estocástico

Para um título S que segue um movimento geométrico browniano:

$$dS = \mu.S.dt + \sigma.S.dZ \quad (1)$$

μ : taxa ajustada ao risco do título

σ : volatilidade do título

dt : incremento de tempo

dZ : incremento de Wiener (dZ segue $N(0,dt)$)

e trocando μ por r a taxa livre de risco na equação (1),

$$dS = r.S.dt + \sigma.S.dZ \quad (2)$$

assim podemos escrever:

$$d(\ln(S)) = \frac{dS}{S} = r.dt + \sigma.dZ \quad (3)$$

Para um derivativo $F(S,t) = \ln(S)$ seguindo um processo d'Itô temos:

$$dF(S,t) = \frac{\partial F}{\partial S} dS + \frac{\partial F}{\partial t} dt + \frac{1}{2} \frac{\partial^2 F}{\partial S^2} dS^2 \quad (4)$$

Análise do incremento de Wiener dZ :

Os termos dt^i com $i > 1$ são considerados nulos, considerando as expressões truncadas na ordem 1 somente.

$$dt^i = 0 \text{ para } i > 1 \quad (5)$$

Calcular $(dS)^2$:

$$(dS)^2 = (r.S.dt + \sigma.S.dZ)^2 = r^2.S^2.dt^2 + 2.r.\sigma.S^2.dt.dZ + \sigma^2.S^2.dZ^2 \quad (6)$$

Distribuição de dZ :

dZ segue por definição uma distribuição normal, media 0 e variância dt : seja $N(0, dt)$.

Distribuição de dZ^2 :

Sabendo a distribuição acima, podemos escrever:

$$Var(dZ) = E[dZ^2] - (E[dZ])^2 = E[dZ^2] - 0 = dt \quad (7)$$

Então

$$E[dZ^2] = dt \quad (8)$$

$$Var(dZ^2) = E[(dZ^2)^2] - (E[dZ^2])^2 = E[dZ^4] - (E[dZ^2])^2 = 3.dt^2 - dt^2 = 2.dt^2$$

$$Var(dZ^2) = 0 \quad (9)$$

Então com (8) e (9), dZ^2 segue a distribuição $N(dt, 0)$, seja é determinístico.

$$dZ^2 = dt \quad (10)$$

Finalmente usando (5) e (10), (6) da:

$$(dS)^2 = \sigma^2.S^2.dt \quad (11)$$

Calculo das derivadas de $F(S,t) = \ln(S)$:

$$\frac{\partial F(S,t)}{\partial S} = \frac{\partial(\ln S)}{\partial S} = \frac{dS}{S}$$

$$\frac{\partial^2 F(S,t)}{\partial S^2} = \frac{\partial^2(\ln S)}{\partial S^2} = \frac{\partial}{\partial S} \left(\frac{dS}{S} \right) = -\frac{1}{2} \frac{dS^2}{S^2}$$

$$\frac{\partial F(S,t)}{\partial t} = \frac{\partial(\ln S)}{\partial t} = 0$$

Substituindo as expressões, temos finalmente a formula (4) que define o movimento do derivativo F, e sendo o movimento geométrico browniano do ativo:

$$dF(S,t) = \left(r - \frac{\sigma^2}{2} \right) dt + \sigma \cdot dZ \quad (12)$$

Terminando o calculo onde "eps" esta uma variável aleatória tirada da distribuição $N(0,1)$:

$$\Delta F(S,t) = \Delta \ln S = \left(r - \frac{\sigma^2}{2} \right) \Delta t + \sigma \cdot eps \cdot \sqrt{\Delta t}$$

$$\Delta \ln S = \ln S_t - \ln S_0 = \ln \frac{S_t}{S_0} = \left(r - \frac{\sigma^2}{2} \right) \Delta t + \sigma \cdot eps \cdot \sqrt{\Delta t}$$

$$\frac{S_t}{S_0} = \exp \left(\left(r - \frac{\sigma^2}{2} \right) \Delta t + \sigma \cdot eps \cdot \sqrt{\Delta t} \right)$$

$$\boxed{S_t = S_0 e^{\left(r - \frac{\sigma^2}{2} \right) \Delta t + \sigma \cdot eps \cdot \sqrt{\Delta t}} \quad (13)}$$

Assim podemos simular o preço S_t para um ativo qualquer, de preço inicio S_0 , volatilidade σ e para um tempo futuro distante de Δt .

Dividindo o tempo total da simulação T em vários intervalos de tempo " dt " todos iguais, cada $S_{n.dt}$ pode ser calculado usando o termo precedente da seguinte maneira:

$$S_{n.dt} = S_{(n-1).dt} e^{\left(r - \frac{\sigma^2}{2}\right)dt + \sigma \cdot eps \cdot \sqrt{dt}} \quad (14)$$

iniciando com S_0 , o preço da ação no instante inicial.

A fórmula com "*dividend yield*" esta quase igual, trocando o r acima para " $r - y$ " onde y representa a taxa de dividendos continua da ação:

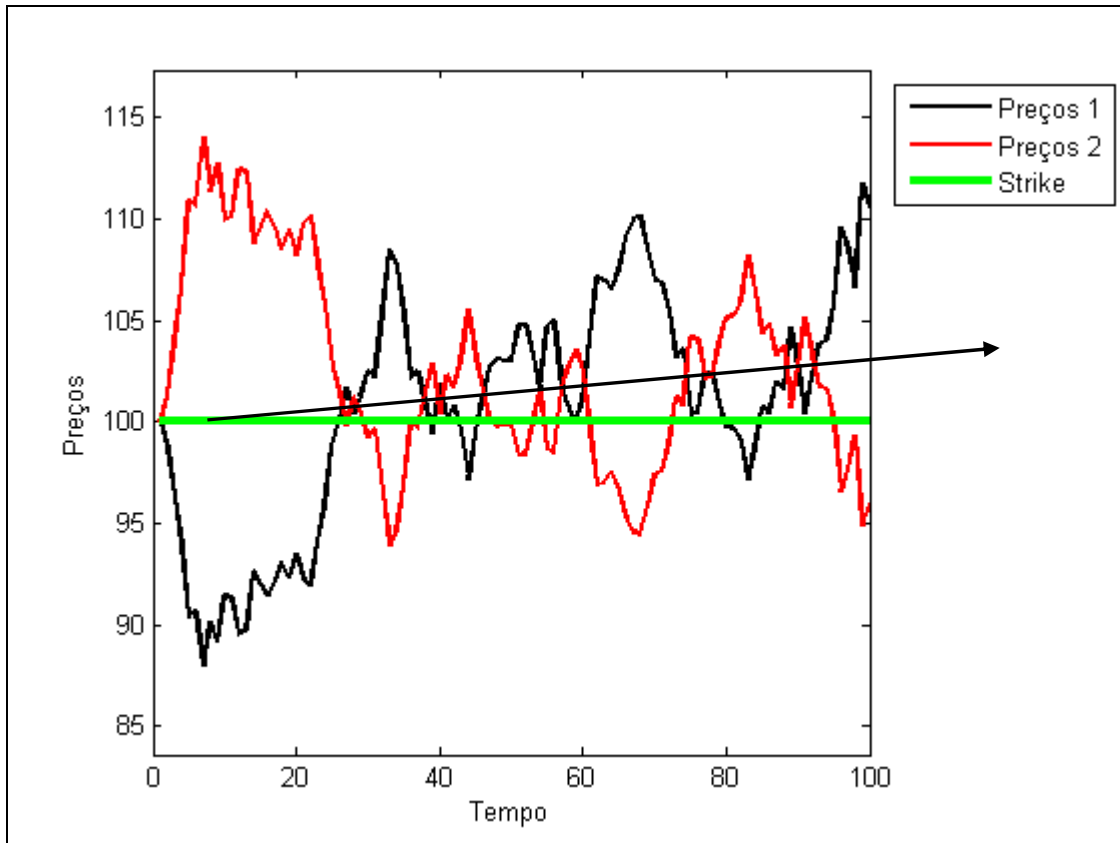
$$S_{n.dt} = S_{(n-1).dt} e^{\left(r - y - \frac{\sigma^2}{2}\right)dt + \sigma \cdot eps \cdot \sqrt{dt}} \quad (15)$$

Exemplo 1:

Preços do ativo no inicio	S=100
Strike da opção	k=100
Tempo a expiração	T=1
Taxa livre de risco	r=5%
Volatilidade do ativo	v=20%
Dividend Yield	y=0

A simulação dos preços foi feita sobre 100 divisões iguais do tempo de um ano da opção.

O "preços 1" representa o caminho normal, e "preços 2" o caminho "antitético" correspondente.



Podemos ver que ao longo do tempo, a tendência normal do preço está de aumentar. A simulação dos preços com esses parâmetros tem um "drift" como visto no modelo que explica a variação da variável $dF(S,t)$ em função de dt e de dZ . O "drift" é a tendência que ocorre no tempo, seja o parâmetro fator de dt :

$$dF(S,t) = \left(r - y - \frac{\sigma^2}{2} \right) dt + \sigma dZ \quad \text{Drift} = \left(r - y - \frac{\sigma^2}{2} \right)$$

Nesse exemplo:

$$\text{Drift} = \left(r - y - \frac{\sigma^2}{2} \right) = \left(0.05 - \frac{0.2^2}{2} \right) = 0.03 = 3\%$$

Exemplo 2: Sem drift

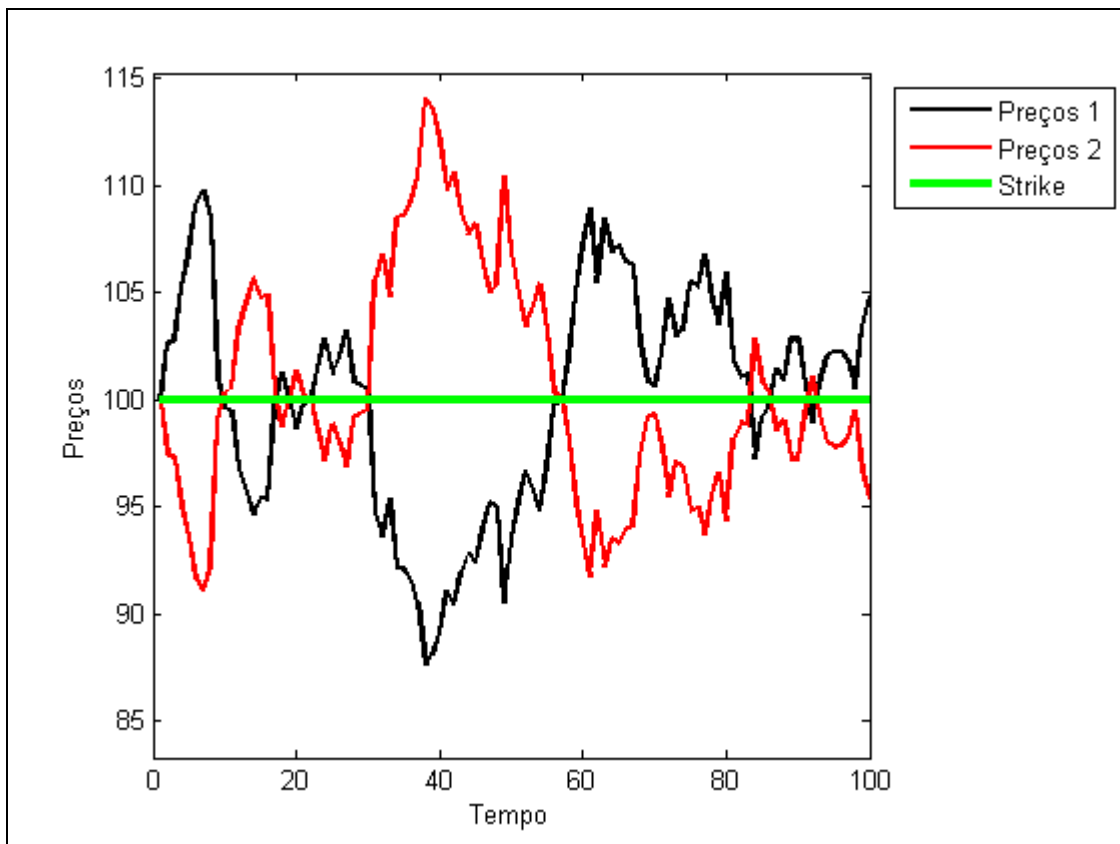
Usando parâmetros ajustado tais que $Drift = \left(r - y + \frac{\sigma^2}{2} \right) = 0$, seja $r = \frac{\sigma^2}{2}$ com

$y = 0$, a média dos dois caminhos antitéticos será exatamente igual a zero.

Preços do ativo no início	S=100
Strike da opção	k=100
Tempo a expiração	T=1
Taxa livre de risco	r=2%
Volatilidade do ativo	v=20%
Dividend Yield	y=0

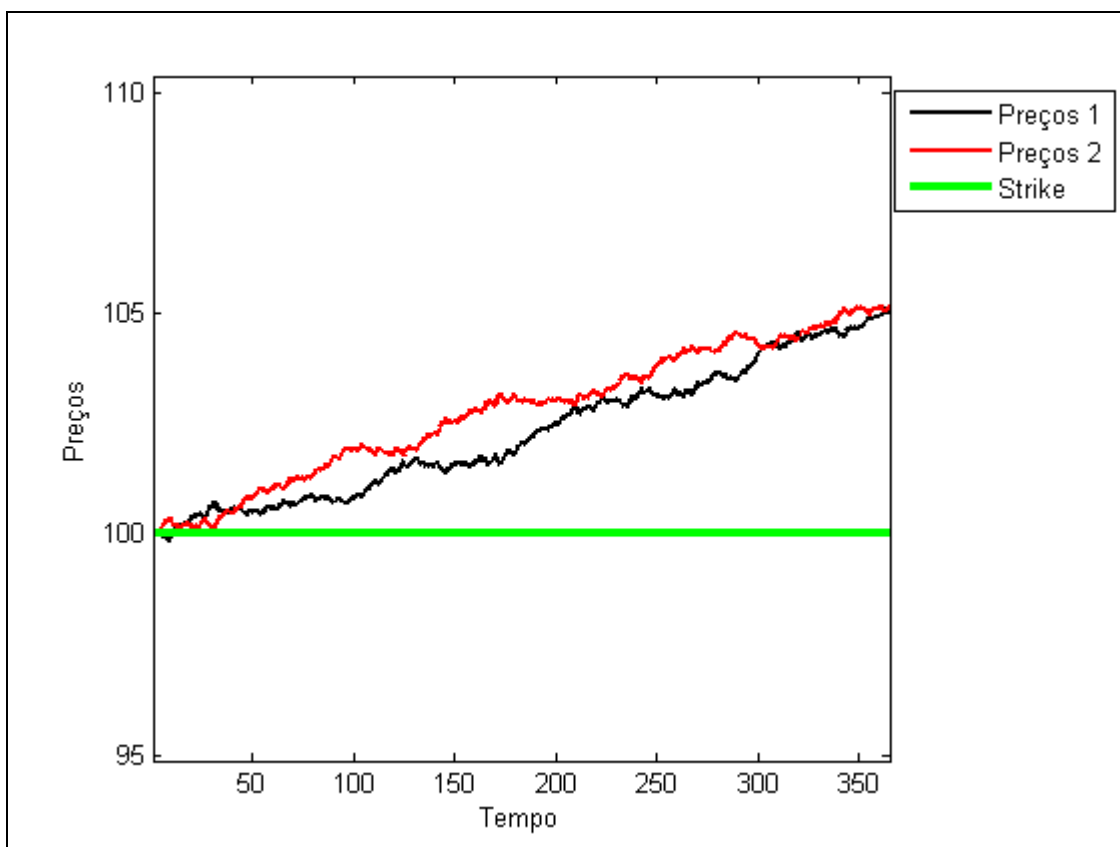
Assim, $r = 2\%$ e $\frac{\sigma^2}{2} = \frac{0.2^2}{2} = 2\%$. Segue o gráfico abaixo onde as duas curvas

são perfeitamente simétricas para o valor inicial de 100:



Exemplo 3: Volatilidade baixa

Preços do ativo no início	S=100
Strike da opção	k=100
Tempo a expiração	T=1
Taxa livre de risco	r=5%
Volatilidade do ativo	v=0.1%
Dividend Yield	y=0



Com volatilidade fraca, os valores simulados ficam mais perto do valor esperado, seguindo o aumento clássica de 5% da taxa livre de risco, para atingir 105 em 1 ano.

$$Drift = \left(r - y - \frac{\sigma^2}{2} \right) = 0.05 - \frac{0.01^2}{2} \approx 5\%$$

O valor esperado da simulação, com valor 100 no instante inicial, e um drift de 5% é:

$$E[Preços] = 100 \cdot (1 + 0.05) = 105$$

Apêndice 2

Demonstração Black-Scholes

O objetivo é de provar a formula de Black-Scholes para precificação de opções, ou seja:

$$w = x \cdot \underline{N}(d_1) - C \cdot e^{-r(T-t)} \cdot \underline{N}(d_2)$$

Com

$$d_1 = \frac{\ln(X/C) + (r + \frac{v^2}{2})(T-t)}{v \cdot \sqrt{T-t}} \quad \text{e} \quad d_2 = \frac{\ln(X/C) + (r - \frac{v^2}{2})(T-t)}{v \cdot \sqrt{T-t}}$$

Onde

w	valor da opção
x	valor do ativo objeto
C	preço de exercício da opção
t	data atual
T	data de exercício da opção
r	taxa livre de risco
v	volatilidade do ativo

Queremos achar uma formula para $w(x,t)$ em qualquer instante de tempo. Mas sabemos que no vencimento ($T = t$):

$$w(x,T) = \max(x_T - C, 0) \quad (1)$$

Para encontramos $w(x,t)$, podemos trazer a expressão $w(x,T)$ a valor presente, de forma que sendo neutro ao risco:

$$w(x,t) = e^{-r(T-t)} \cdot E[\max(x_T - C, 0)] \quad (2)$$

Considerando que a variação do preço das opções siga um movimento geométrico browniano MGB:

$$dx = r \cdot x \cdot dt + \sigma \cdot x \cdot dZ \quad (3)$$

Onde o incremento de Wiener $dZ \sim N(0, dt)$

Seja $f = f(x)$ uma função apenas da variável estocástica x . Do lema de Itô, teremos:

$$df(x) = \frac{\partial f}{\partial x} dx + \frac{1}{2} \frac{\partial^2 f}{\partial^2 x} dx^2$$

Usando $f(x) = \ln(x)$, e aplicando o lema de Itô:

$$d \ln(x) = \frac{\partial \ln(x)}{\partial x} dx + \frac{1}{2} \frac{\partial^2 \ln(x)}{\partial^2 x} dx^2$$

$$d \ln(x) = \frac{1}{x} dx + \frac{1}{2} \left(-\frac{1}{x^2} \right) dx^2$$

$$d \ln(x) = \frac{1}{x} dx - \frac{dx^2}{2x^2} \quad (6)$$

Do MGB, na equação (4) temos:

$$(dx)^2 = (r \cdot x \cdot dt + \sigma \cdot x \cdot dZ)^2 = r^2 \cdot x^2 \cdot dt^2 + 2 \cdot r \cdot \sigma \cdot x^2 \cdot dt \cdot dZ + \sigma^2 \cdot x^2 \cdot dZ^2$$

e finalmente:

$$(dx)^2 = \sigma^2 \cdot x^2 \cdot dt \quad (7) \quad (\text{ver anexo 1 p2 para detalho})$$

Substituindo (4) e (7) em (6):

$$d \ln(x) = \frac{1}{x}(r.x.dt + v.x.dz) - \frac{1}{2x^2}v^2.x^2.dt$$

$$d \ln(x) = \left(r - \frac{v^2}{2}\right).dt + v.dz \quad (9)$$

Integrando a expressão acima (9) de t a T :

$$\int_t^T d \ln(x) = \int_t^T \left(r - \frac{v^2}{2}\right).dt + \int_t^T v.dz_x \quad (\text{a segunda parte é uma integral estocástica que sabemos resolver})$$

$$\ln(x_T) - \ln(x_t) = \left(r - \frac{v^2}{2}\right)(T-t) + v.(z_T - z_t) \quad (10) \quad (\text{onde } (z_T - z_t) \sim N(0, T-t))$$

$$\ln(x_T) = \ln(x_t) + \left(r - \frac{v^2}{2}\right)(T-t) + v.(z_T - z_t) \quad (11)$$

Seja y_T a função definida para:
$$y_T = \left(r - \frac{v^2}{2}\right)(T-t) + v.(z_T - z_t) \quad (12)$$

Como $(z_T - z_t)$ é a única parcela estocástica com distribuição $N(0, T-t)$, y_T também terá distribuição normal com parâmetros:

Media:

$$E[y_T] = \left(r - \frac{v^2}{2}\right)(T-t) + E[v.(z_T - z_t)]$$

$$E[y_T] = \left(r - \frac{v^2}{2}\right)(T-t) \quad \text{devido á distribuição da variável estocástica } z.$$

Variância:

$$\text{Var}[y_T] = \text{Var}[v.(z_T - z_t)]$$

$$\text{Var}[y_T] = v^2.\text{Var}(z_T - z_t)$$

$$\text{Var}[y_T] = v^2.(T-t)$$

$$y_T \text{ tem a distribuição } N\left(\left(r - \frac{v^2}{2}\right)(T-t), v^2.(T-t)\right) \quad (13)$$

Ainda sobre y_T , da expressão (11):

$$\ln\left(\frac{x_T}{x_t}\right) = y_T \quad x_T = x_t \cdot e^{y_T} \quad (14)$$

Pela teoria das probabilidades temos para quaisquer funções g e h :

$$g(y_T) = h(x_T) \cdot \left| \frac{dx_T}{dy_T} \right|$$

Mas temos $\frac{dx_T}{dy_T} = x_T \cdot e^{y_T}$ que é sempre positivo, então

$$g(y_T) \cdot dy_T = h(x_T) \cdot dx_T \quad (15)$$

Voltando na definição de $w(x, t)$ (2) que queremos calcular:

$$w(x, t) = e^{-r(T-t)} \cdot E[\max(x_T - C, 0)]$$

Da definição de valor esperado temos:

$$w(x, t) = e^{-r(T-t)} \cdot \int_{-\infty}^{+\infty} \max(x_T - C, 0) \cdot f(x_T) \cdot d(x_T)$$

Considerando que a integral é nula para $x_T < C$ pela definição da função

$$\max(x_T - C, 0): \quad w(x, t) = e^{-r(T-t)} \cdot \int_C^{+\infty} (x_T - C) \cdot f(x_T) \cdot d(x_T)$$

E usando (15):

$$w(x, t) = e^{-r(T-t)} \cdot \int_u^{+\infty} (x_t \cdot e^{y_T} - C) \cdot g(y_T) \cdot d(y_T)$$

Onde u é o novo limite inferior para a integral em $d(y_T)$ de forma que:

$$u = \ln\left(\frac{C}{x_t}\right).$$

Finalmente

$$w(x, t) = e^{-r(T-t)} \cdot \int_{\ln(C/x_t)}^{+\infty} (x_t \cdot e^{y_T} - C) \cdot g(y_T) \cdot d(y_T)$$

$$w(x, t) = x_t \cdot I_1 - C \cdot e^{-r(T-t)} \cdot I_2 = x_t \cdot e^{-r(T-t)} \cdot \int_{\ln(C/x_t)}^{+\infty} e^{y_T} \cdot g(y_T) \cdot d(y_T) - C \cdot e^{-r(T-t)} \cdot \int_{\ln(C/x_t)}^{+\infty} g(y_T) \cdot d(y_T) \quad (16)$$

Calcular I_1 :

Como y_T tem distribuição normal com parâmetros dados na expressão (13):

$$I_1 = e^{-r(T-t)} \cdot \int_{\ln(C/x_t)}^{+\infty} e^{y_T} \cdot \frac{1}{\sqrt{2\pi \cdot v^2 \cdot (T-t)}} \cdot \exp \left[-\frac{1}{2} \left(\frac{y_T - (r - \frac{v^2}{2}) \cdot (T-t)}{v \cdot \sqrt{(T-t)}} \right)^2 \right] \cdot dy_T$$

$$I_1 = \int_{\ln(C/x_t)}^{+\infty} e^{y_T} \cdot \frac{1}{\sqrt{2\pi \cdot v^2 \cdot (T-t)}} \cdot \exp \left[-r \cdot (T-t) + y_T - \frac{1}{2} \left(\frac{y_T - (r - \frac{v^2}{2}) \cdot (T-t)}{v \cdot \sqrt{(T-t)}} \right)^2 \right] \cdot dy_T$$

(17)

Abrindo a expressão do expoente de e na integração, teremos:

$$A = -r \cdot (T-t) + y_T - \frac{1}{2} \left(\frac{y_T - (r - \frac{v^2}{2}) \cdot (T-t)}{v \cdot \sqrt{(T-t)}} \right)^2$$

$$A = -\frac{1}{2 \cdot v^2 \cdot (T-t)} \left[\left(y_T - (r - \frac{v^2}{2}) \cdot (T-t) \right)^2 + (-2) \cdot (y_T - r \cdot (T-t)) \cdot v^2 \cdot (T-t) \right]$$

$$A = -\frac{1}{2} \left[\frac{y_T - (r + \frac{v^2}{2}) \cdot (T-t)}{v \cdot \sqrt{(T-t)}} \right]^2$$

(18)

Observação:

usando uma distribuição normal com média μ_x e variância σ_x^2 , temos:

$$I = \frac{1}{\sqrt{2\pi \cdot \sigma_x^2}} \int_b^{+\infty} \exp \left[-\frac{1}{2} \left(\frac{x - \mu_x}{\sigma_x} \right)^2 \right] \cdot dx$$

fazendo $Z = \frac{x - \mu_x}{\sigma_x}$, $dx = \sigma_x \cdot dZ$. Para $x = b$, temos: $Z = \frac{b - \mu_x}{\sigma_x}$

$$I = \frac{1}{\sqrt{2\pi}} \int_{\frac{b-\mu_x}{\sigma_x}}^{+\infty} \exp\left[-\frac{1}{2}(Z)^2\right].dx \quad \text{e} \quad I = \int_{\frac{b-\mu_x}{\sigma_x}}^{+\infty} N(Z).dZ = \int_{-\infty}^{\frac{\mu_x-b}{\sigma_x}} N(Z).dZ$$

Onde $N(Z)$ é a função de densidade de probabilidade da $N(0,1)$, função simétrica.

De forma análoga em I_1 temos com esses parâmetros:

$$l = \ln(c/x) \quad \mu_x = \left(r + \frac{v^2}{2}\right).(T-t) \quad \sigma_x = \sqrt{v^2.(T-t)}$$

$$I_1 = \underline{N} \left(\frac{\ln(x/c) + \left(r + \frac{v^2}{2}\right).(T-t)}{\sqrt{v^2.(T-t)}} \right) \quad (19)$$

onde \underline{N} é a função cumulativa da densidade de probabilidade normal N .

$$I_1 = \underline{N}(d_1) \quad \text{onde} \quad d_1 = \frac{\ln(x/c) + \left(r + \frac{v^2}{2}\right).(T-t)}{\sqrt{v^2.(T-t)}}$$

Calcular I_2 :

$$I_2 = \int_{\ln(c/x_t)}^{+\infty} e^{y_T} \cdot \frac{1}{\sqrt{2\pi.v^2.(T-t)}} \cdot \exp\left[-\frac{1}{2}\left(\frac{y_T - \left(r - \frac{v^2}{2}\right).(T-t)}{v.\sqrt{(T-t)}}\right)^2\right].dy_T$$

Da mesma forma e usando os parâmetros seguintes:

$$l = \ln(c/x) \quad \mu_x = \left(r - \frac{v^2}{2}\right).(T-t) \quad \sigma_x = \sqrt{v^2.(T-t)}$$

$$I_1 = \underline{N} \left(\frac{\ln(x/c) + \left(r - \frac{v^2}{2}\right).(T-t)}{\sqrt{v^2.(T-t)}} \right) \quad (20)$$

$$I_2 = \underline{N}(d_2) \quad \text{onde} \quad d_2 = \frac{\ln(x/c) + (r - \frac{v^2}{2}) \cdot (T - t)}{\sqrt{v^2 \cdot (T - t)}}$$

Usando (16), (19) e (20):

A fórmula de Black-Scholes

$$w(x, t) = x_t \cdot \underline{N}(d_1) - C \cdot e^{-r(T-t)} \cdot \underline{N}(d_2)$$

Onde

$$d_1 = \frac{\ln(x/c) + (r + \frac{v^2}{2}) \cdot (T - t)}{\sqrt{v^2 \cdot (T - t)}} \quad d_2 = \frac{\ln(x/c) + (r - \frac{v^2}{2}) \cdot (T - t)}{\sqrt{v^2 \cdot (T - t)}}$$

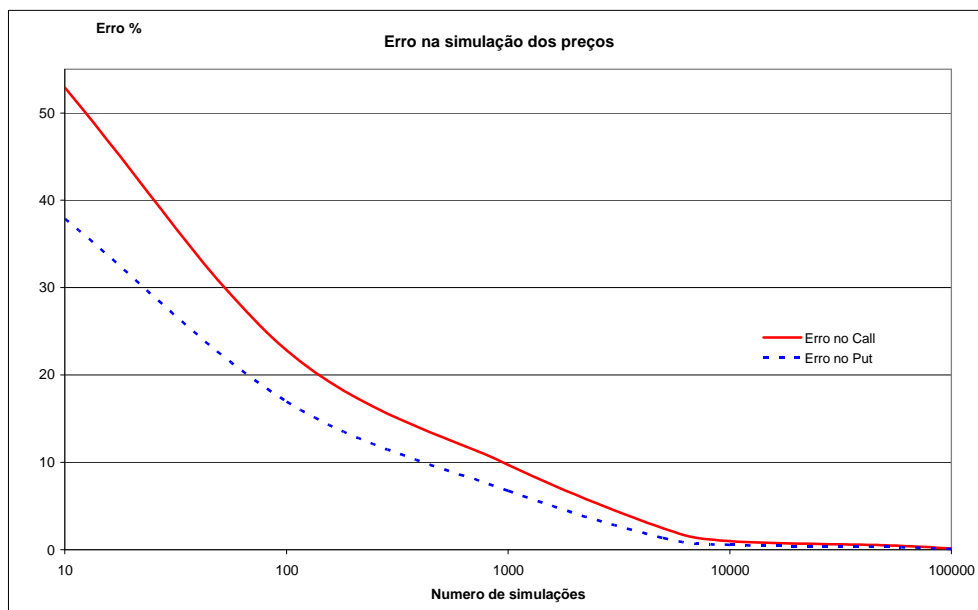
Apêndice 3

Gráficos de sensibilidade

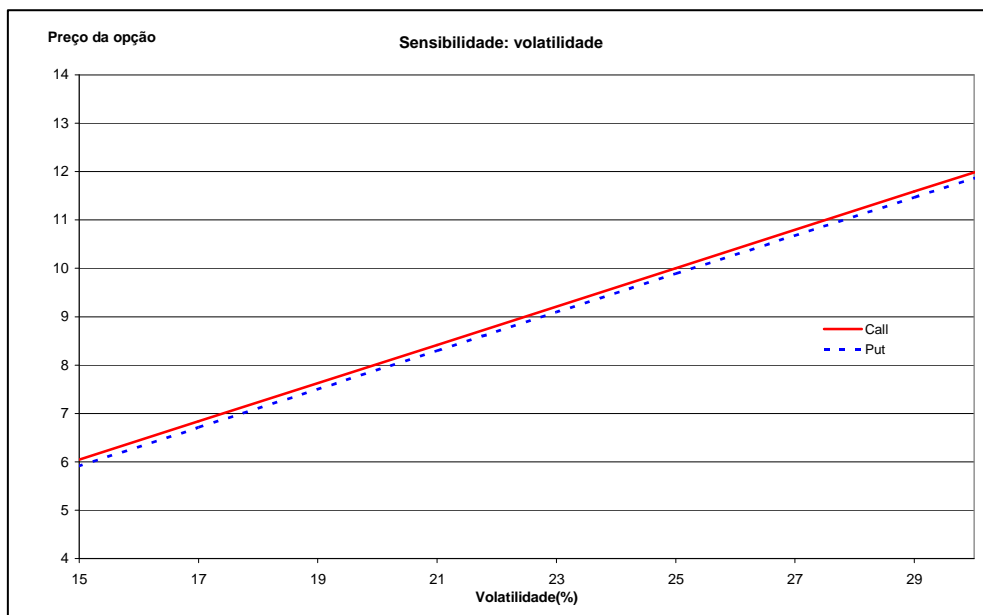
Opção europeia

Sensibilidade em relação ao número de simulações

Evolução do erro com o aumento do número de simulações, em comparação no valor teórico da fórmula de Black e Scholes.



Sensibilidade em relação a mudanças na volatilidade

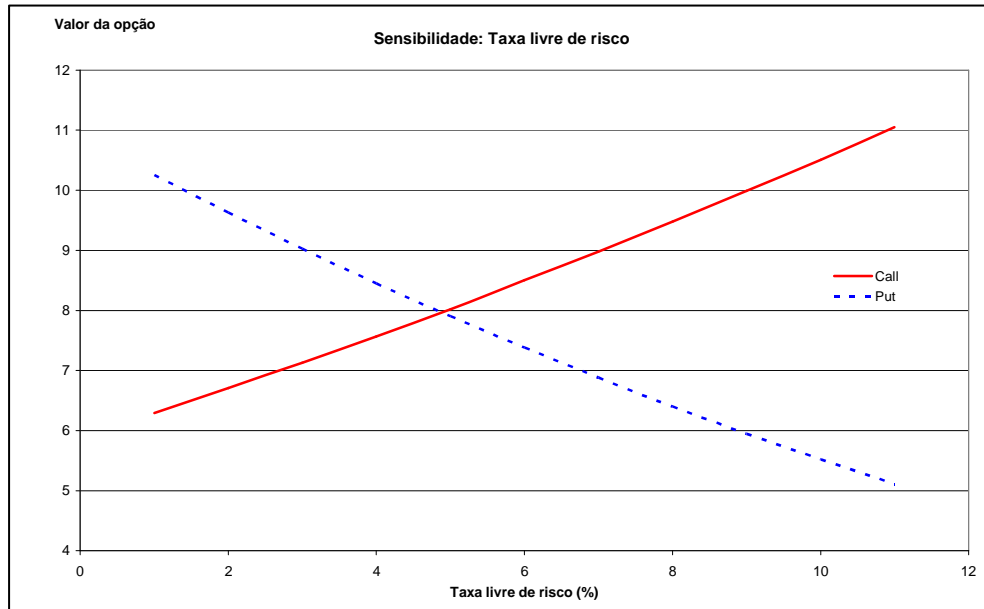


Ilustrando a linearidade com a volatilidade v , uma regressão linear mostra que:

$$Pr eçoCall = 0,3951 \cdot v + 0.107$$

$$Pr eçoPut = 0,3953 \cdot v - 0,0167$$

Sensibilidade em relação a mudanças na taxa livre de risco



Usando uma regressão quadrática para aproximar as curvas, obtemos:

$$\text{PreçoCall} = 0,007.r^2 + 0,3921.r + 5,8922$$

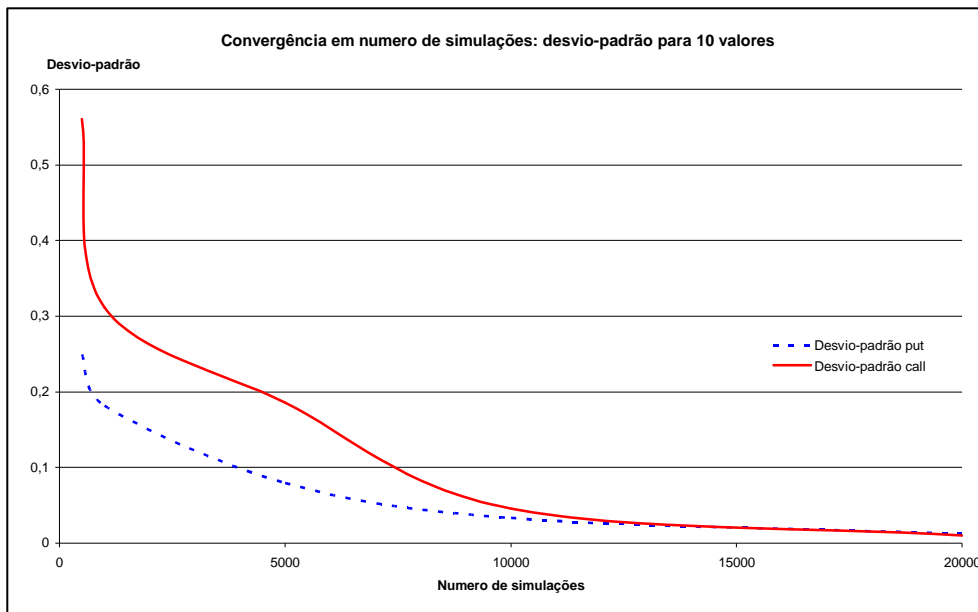
$$R^2 = 1$$

$$\text{PreçoPut} = 0,0121.r^2 - 0,6591.r + 10,897$$

$$R^2 = 1$$

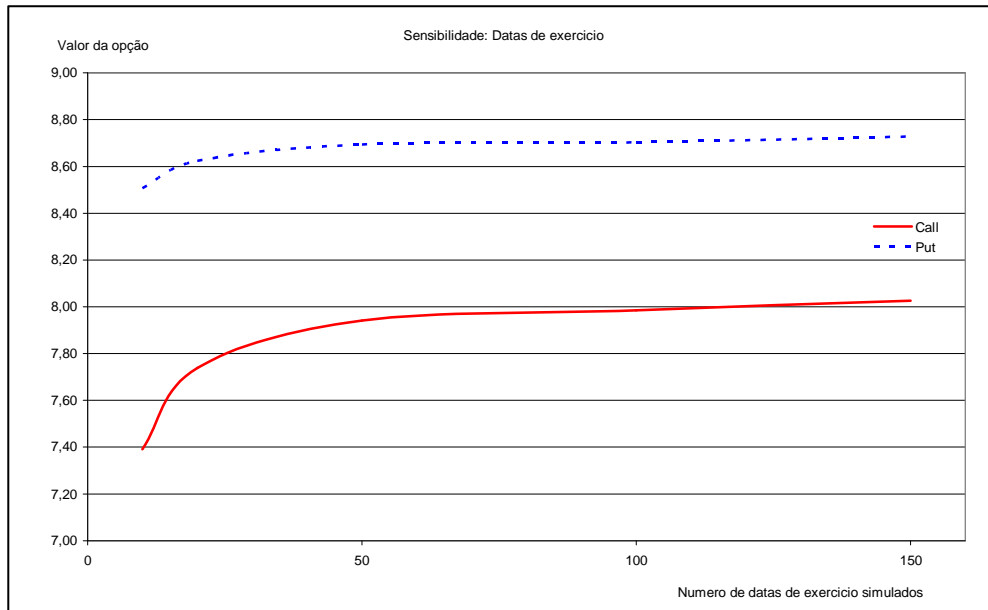
Opção americana

Sensibilidade em mudança no numero de simulações



A precisão no método de Monte-Carlo é obtida através o aumento dos eventos possíveis simulados.

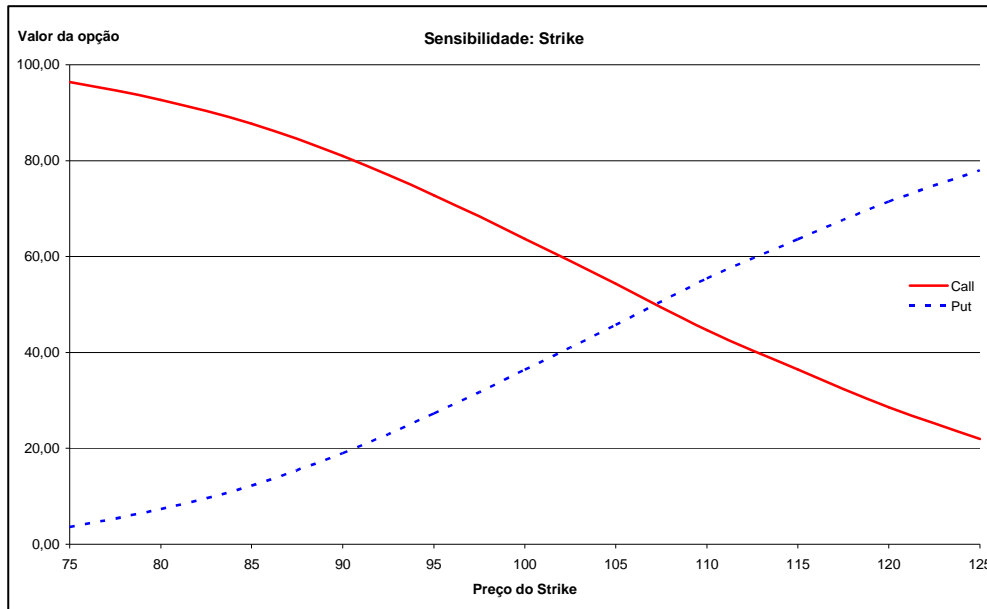
Sensibilidade em mudança no numero de datas de exercício simuladas



Com o aumento de datas de exercício simuladas, os preços aumentam junto com a flexibilidade de poder exercer em qualquer momento.

Opção *asset-or-nothing*

Sensibilidade em relação à mudança no preço do strike



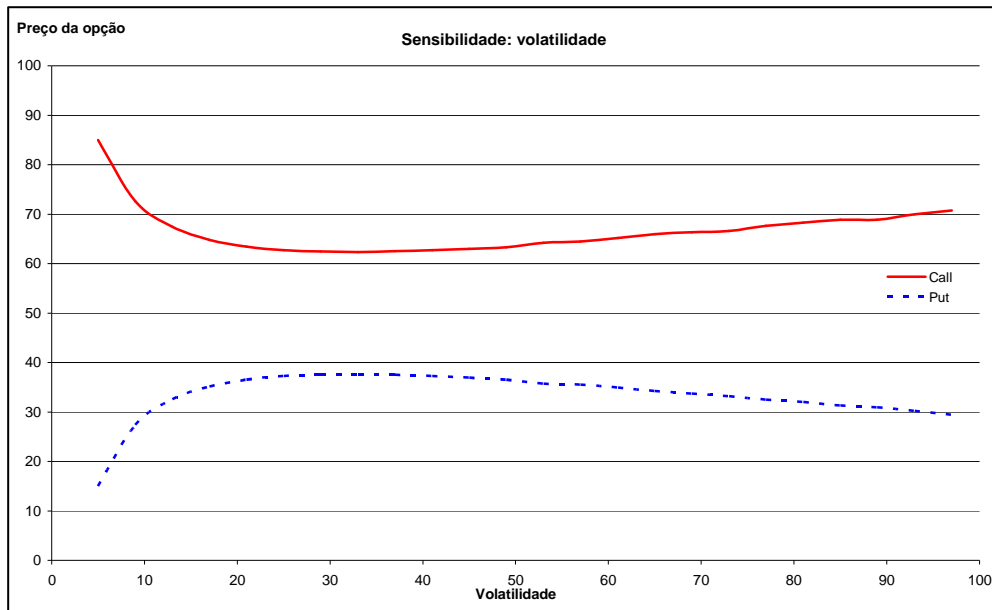
As curvas são simétricas, conseqüentemente a: $Call + Put = C$, com C uma constante. (aqui $C = 100$ nesse exemplo).

Os limites são:

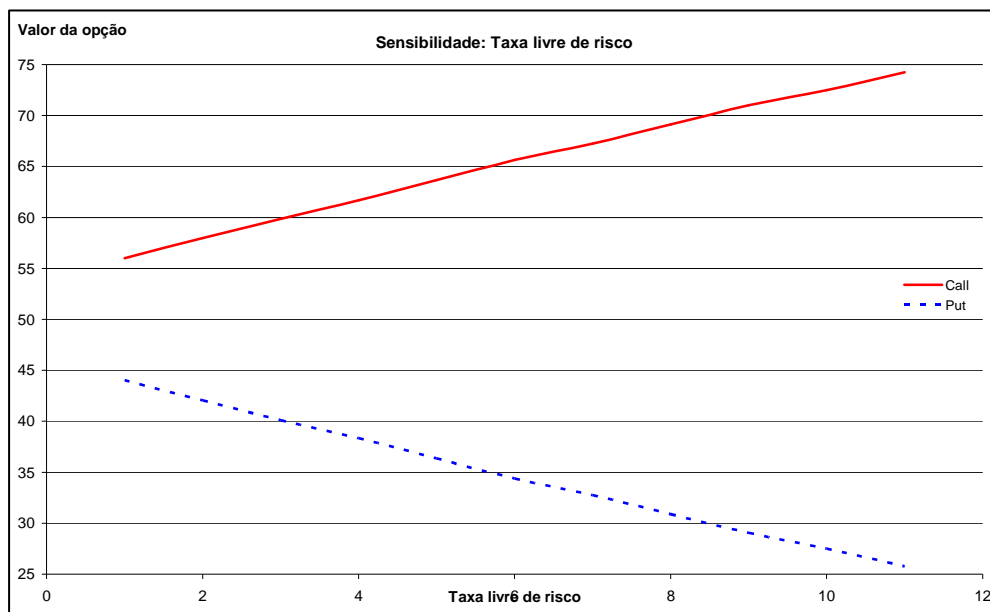
$Strike \rightarrow 0$: $Call \rightarrow 100$ porque sempre vai ficar "*in-the-money*", sendo o preço do ativo maior que o strike
 $Put \rightarrow 0$ porque sempre "*out-of-the-money*", pela mesma razão.

$Strike \rightarrow +\infty$: $Call \rightarrow 0$ O preço do ativo tem maior probabilidade de ficar menor que o strike.
 $Put \rightarrow 100$ pela mesma razão.

Sensibilidade em relação à mudança na volatilidade



Sensibilidade em relação à mudança na taxa livre de risco



Usando uma regressão linear, obtemos as equações seguintes:

$$\text{PreçoCall} = 1,8287.r + 54,372$$

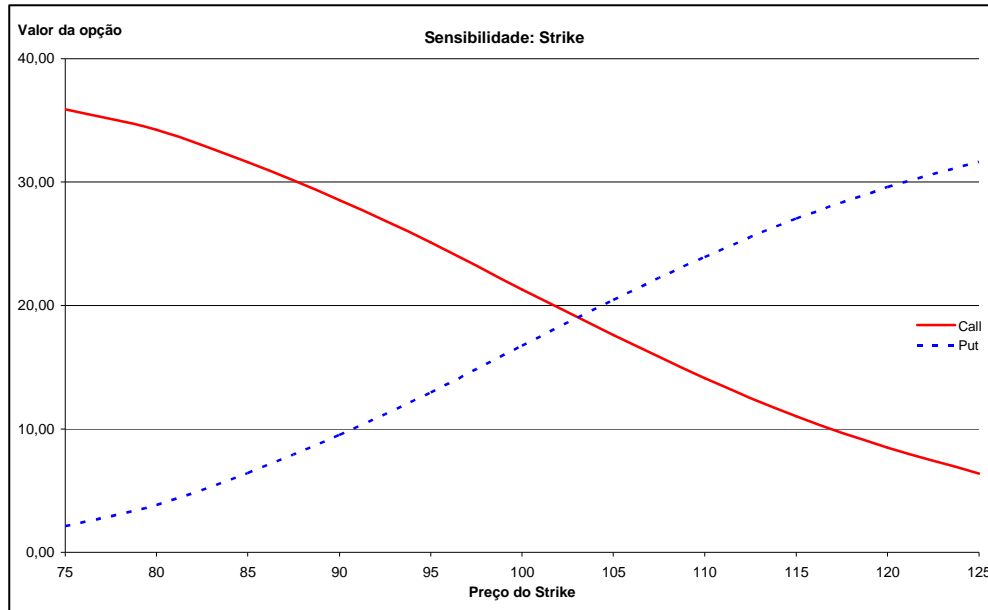
$$R^2 = 0,9993$$

$$\text{PreçoPut} = -1,8286.r + 45,624$$

$$R^2 = 0,9992$$

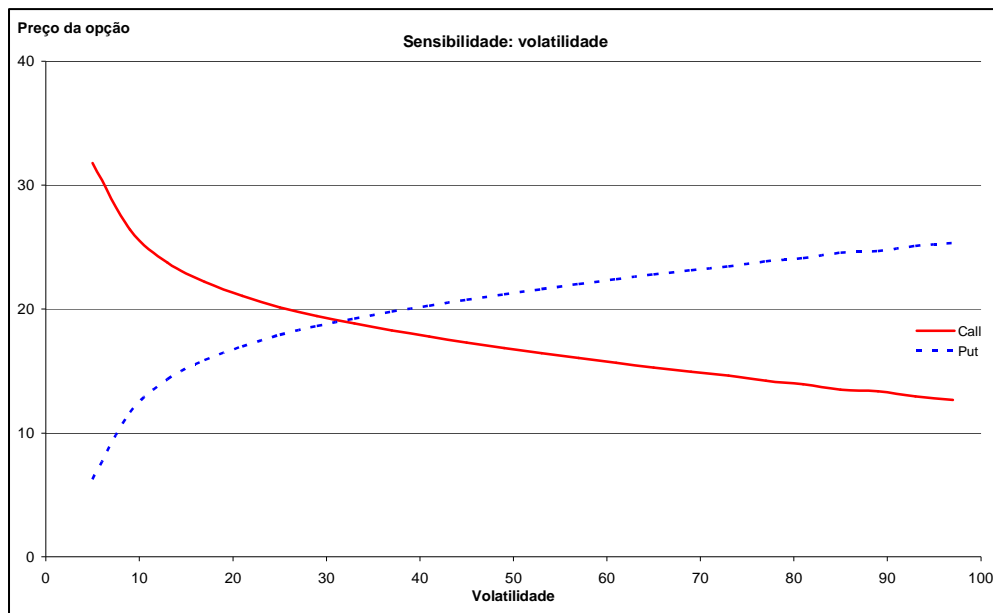
Opção *cash-or-nothing*

Sensibilidade em relação à mudança no preço do strike

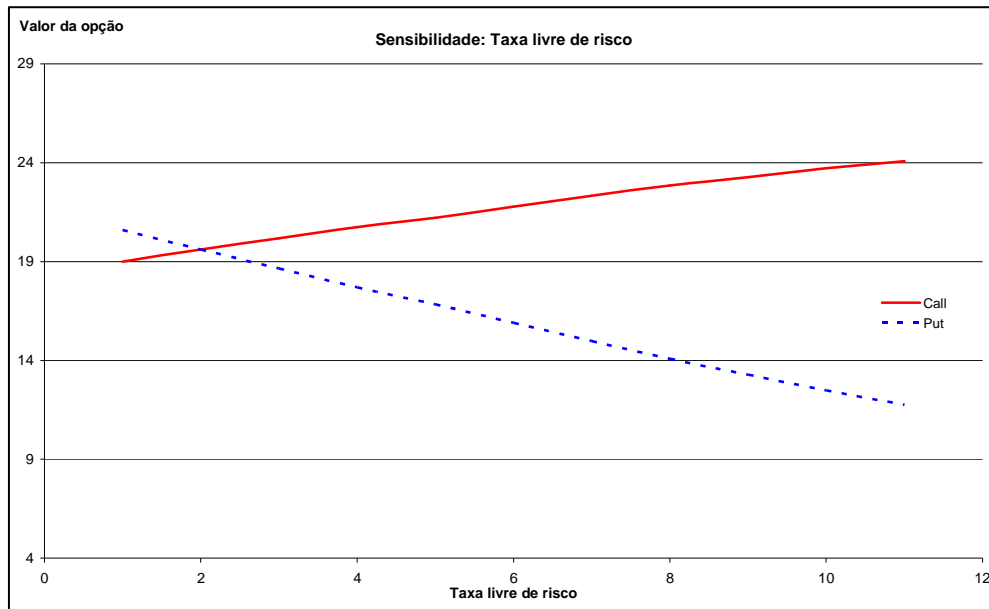


Da mesma forma que para a opção *asset-or-nothing*, as curvas são simétricas e as mesmas conclusões podem ser tiradas quando as limites das curvas em 0 e em $+\infty$

Sensibilidade em relação à mudança na volatilidade



Sensibilidade em relação à mudança na taxa livre de risco



Usando uma regressão linear, obtemos as equações seguintes:

$$\text{PreçoCall} = 0,5137.r + 18,621$$

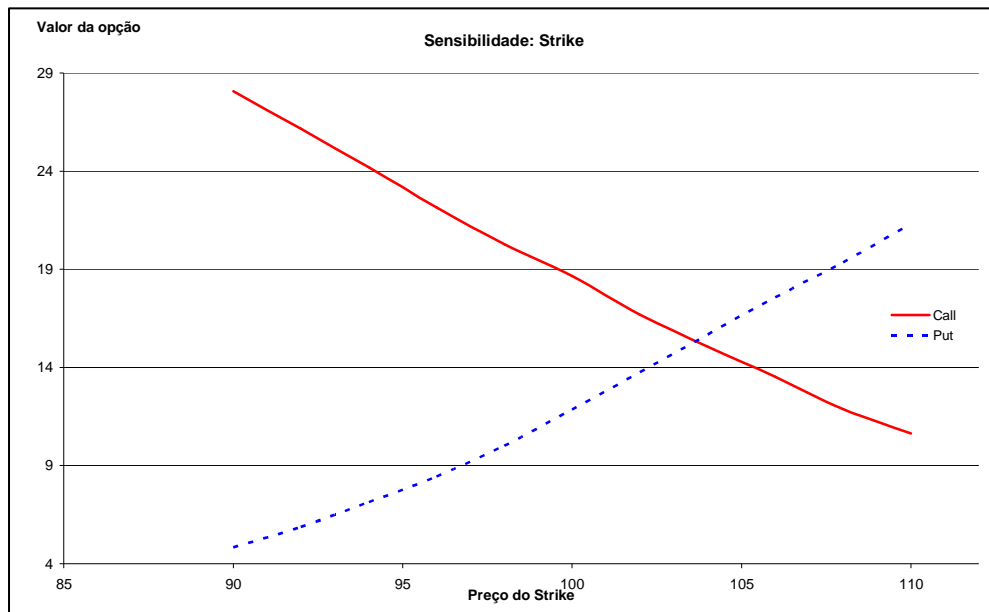
$$R^2 = 0,9969$$

$$\text{PreçoPut} = -0,8905.r + 21,329$$

$$R^2 = 0,9986$$

Opção *Lookback fixed strike*

Sensibilidade em relação à mudança no preço do strike



Usando uma regressão quadrática, obtemos as equações seguintes:

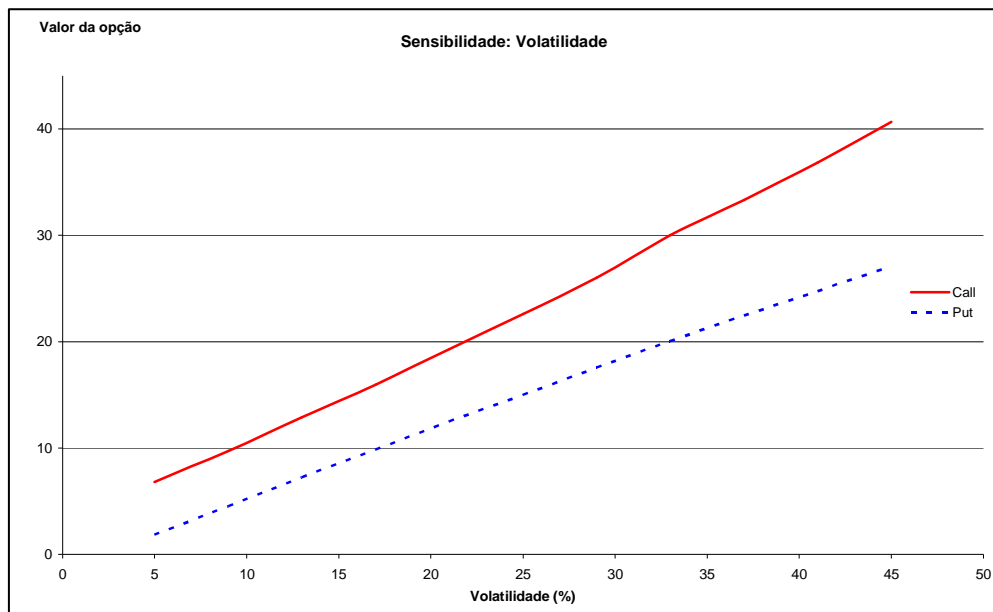
$$\text{PreçoCall} = 0,0083.k^2 - 2,5413.k + 189,66$$

$$R^2 = 0,9998$$

$$\text{PreçoPut} = 0,0122.k^2 - 1,5959.k + 49,449$$

$$R^2 = 0,9996$$

Sensibilidade em relação à mudança na volatilidade



Usando uma regressão linear, obtemos as equações seguintes:

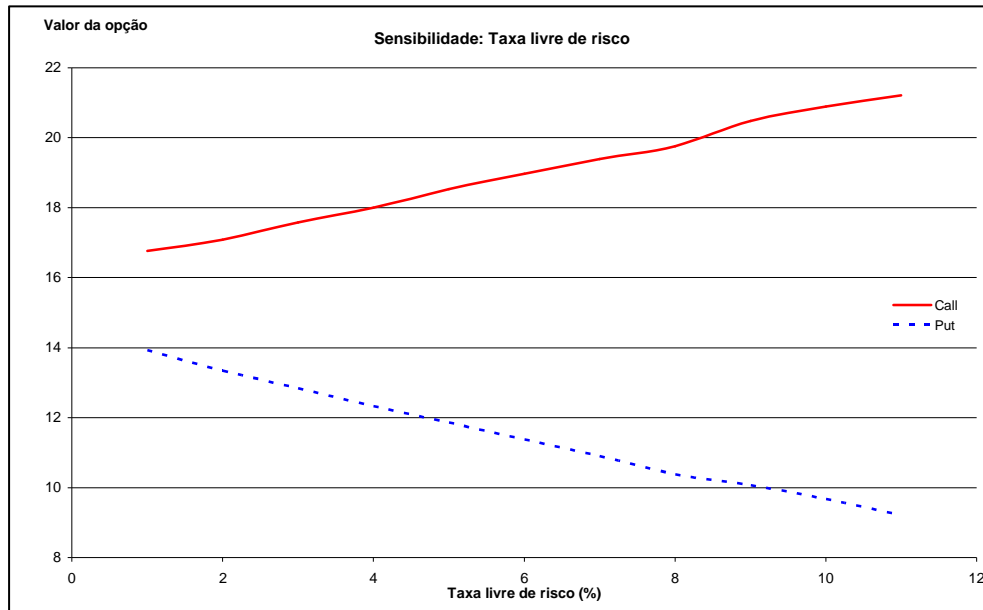
$$\text{PreçoCall} = 0,8493.v + 1,8699$$

$$R^2 = 0,9987$$

$$\text{PreçoPut} = 0,6325.v - 1,0124$$

$$R^2 = 0,9993$$

Sensibilidade em relação à mudança na taxa livre de risco



Usando uma regressão linear, obtemos as equações seguintes:

$$\text{PreçoCall} = 0,4591.r + 16,215$$

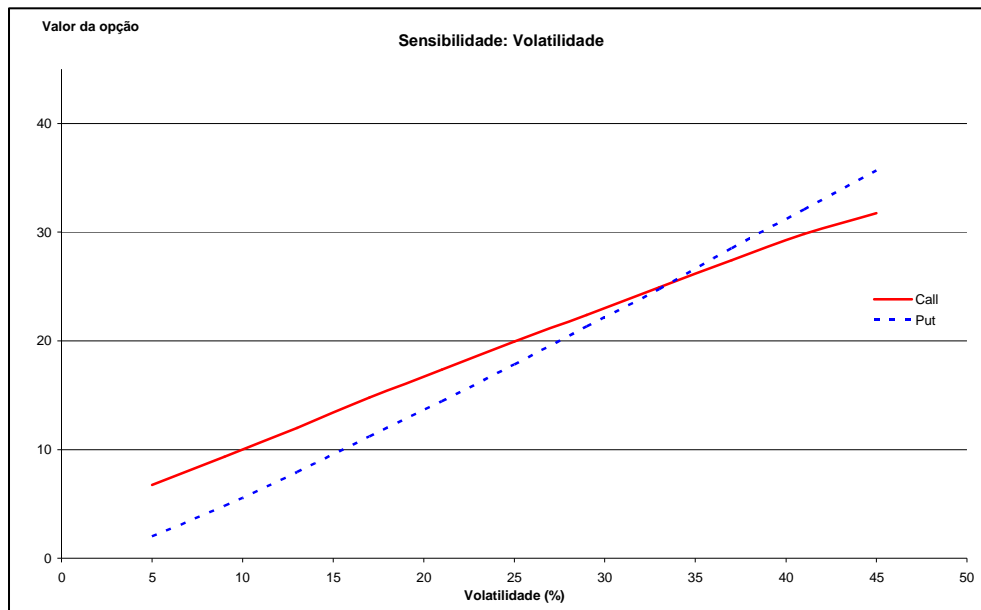
$$R^2 = 0,9975$$

$$\text{PreçoPut} = -0,4675.r + 14,253$$

$$R^2 = 0,9969$$

Opção *Lookback floating strike*

Sensibilidade em relação à mudança na volatilidade



Usando uma regressão linear, obtemos as equações seguintes:

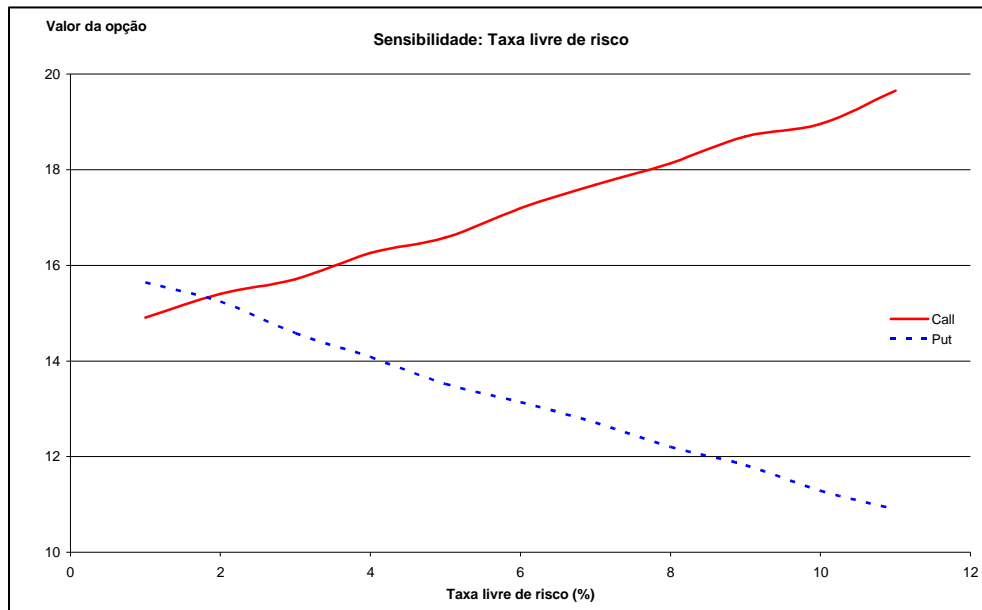
$$\text{PreçoCall} = 0,633.v + 3,8557$$

$$R^2 = 0,999$$

$$\text{PreçoPut} = -0,8482.v - 2,9737$$

$$R^2 = 0,9989$$

Sensibilidade em relação à mudança na taxa livre de risco



Usando uma regressão linear, obtemos as equações seguintes:

$$\text{PreçoCall} = 0,4704.r + 14,374$$

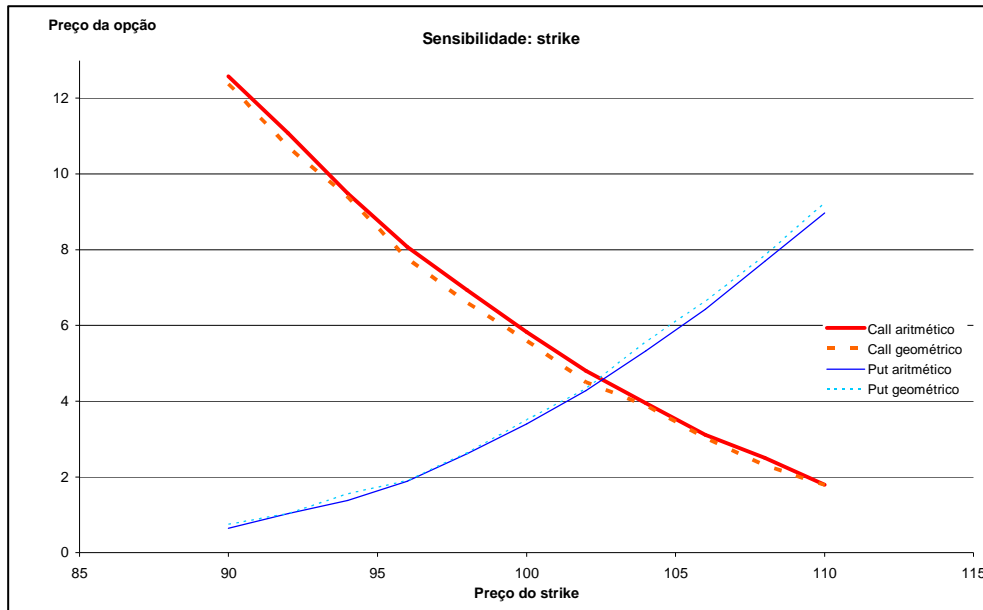
$$R^2 = 0,9971$$

$$\text{PreçoPut} = -0,4761.r + 16,049$$

$$R^2 = 0,9974$$

Opção Asiática

Sensibilidade em relação à mudança no preço do strike



Usando uma regressão quadrática, obtemos as equações seguintes:

$$\text{PreçoCall} = 0,0141.k^2 - 1,5959.k + 49,449$$

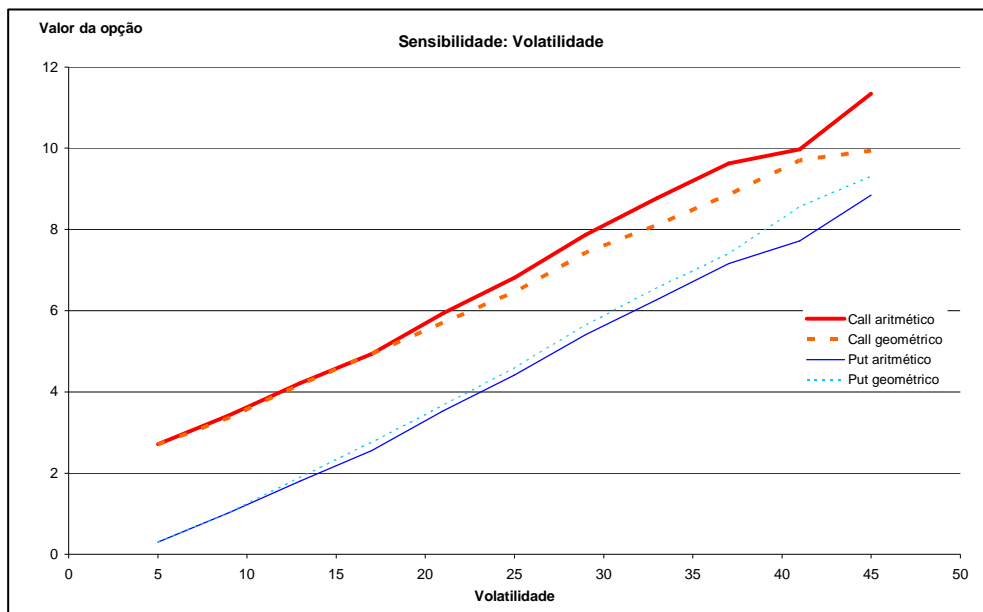
$$R^2 = 0,9998$$

$$\text{PreçoPut} = 0,0144.k^2 - 2,4689.k + 105,93$$

$$R^2 = 0,9998$$

Sensibilidade em relação à mudança na volatilidade

Usando uma regressão linear, obtemos as equações seguintes:



Aritmético

$$\text{PreçoCall} = 0,2163.v + 1,4683$$

$$R^2 = 0,9963$$

$$\text{PreçoPut} = 0,2156.v - 0,9331$$

$$R^2 = 0,9983$$

Geométrico

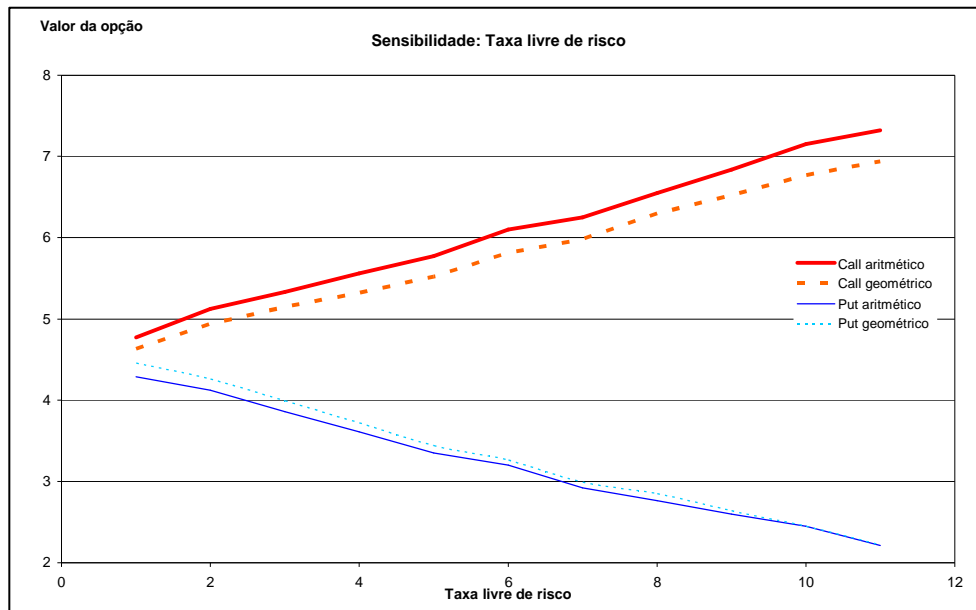
$$\text{PreçoCall} = 0,1901.v + 1,7335$$

$$R^2 = 0,9967$$

$$\text{PreçoPut} = -0,23.v - 1,0466$$

$$R^2 = 0,9987$$

Sensibilidade em relação à mudança na taxa livre de risco



Usando uma regressão linear, obtemos as equações seguintes:

Aritmético

$$\text{PreçoCall} = 0,2533.r + 4,5495$$

$$R^2 = 0,9972$$

$$\text{PreçoPut} = -0,209.r + 4,4695$$

$$R^2 = 0,9943$$

Geométrico

$$\text{PreçoCall} = 0,2309.r + 4,4218$$

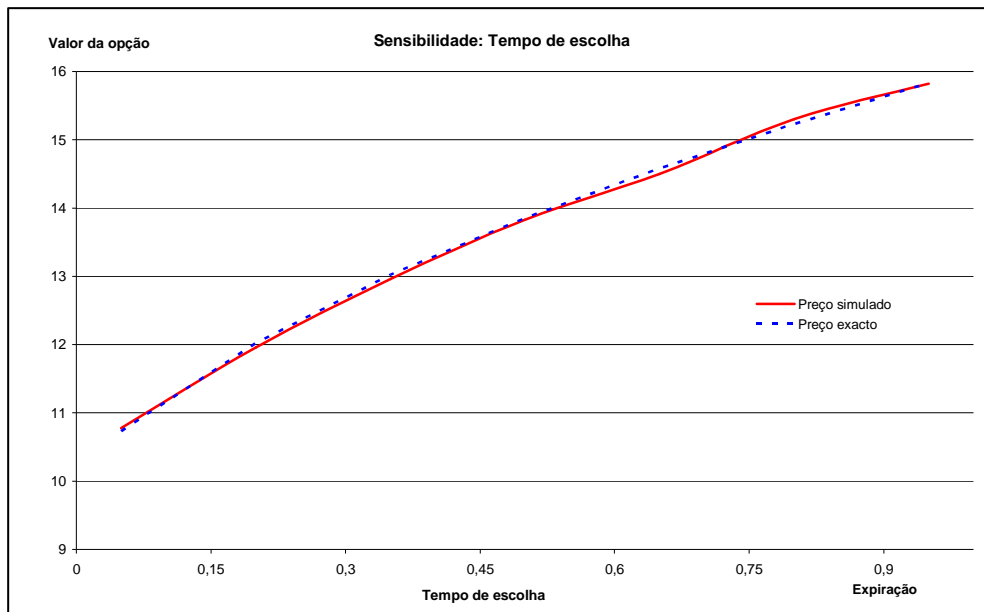
$$R^2 = 0,9974$$

$$\text{PreçoPut} = -0,2244.r + 4,6444$$

$$R^2 = 0,9952$$

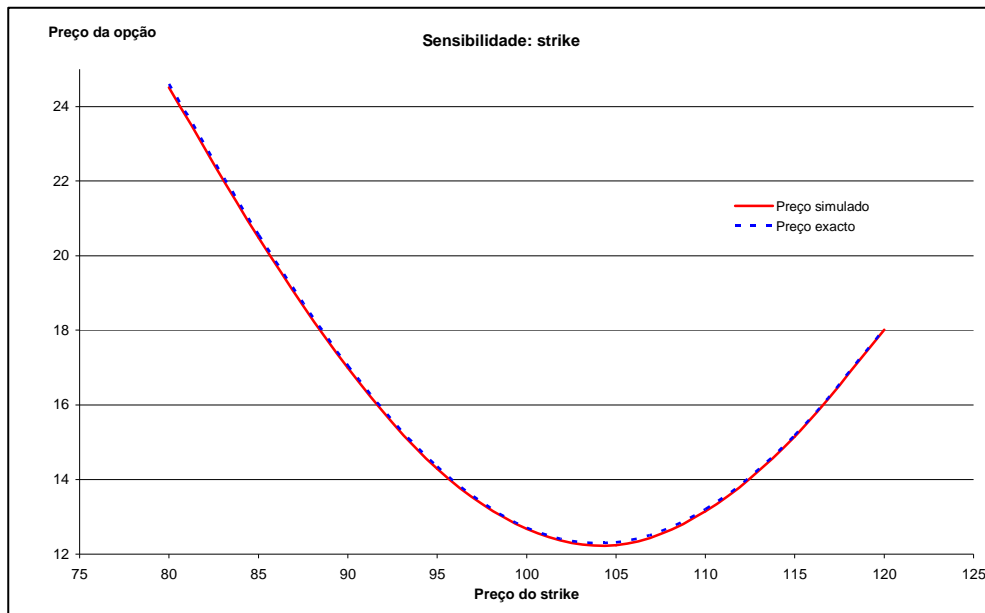
Opção Chooser

Sensibilidade em relação à mudança no tempo de escolha



O preço da opção aumenta com o tempo de escolha, a incerteza sobre o valor final do ativo sendo reduzida.

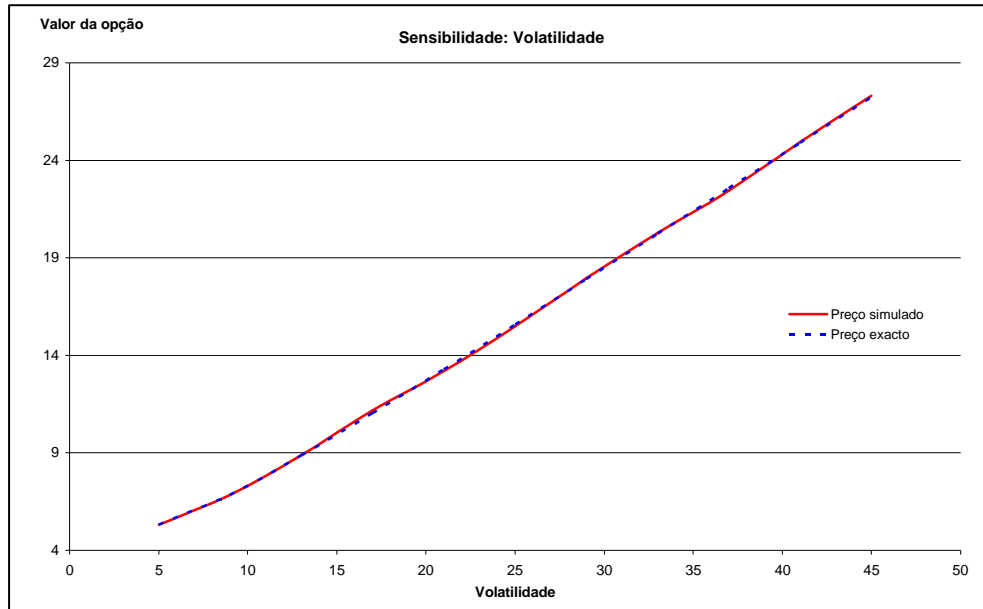
Sensibilidade em relação à mudança no preço do strike



O mínimo do valor do preço da opção depende do *drift* da simulação, então dos parâmetros do ativo.

O preço da opção aumenta com strike baixo, e com strike alto, a opção *chooser* sendo um call e um put no mesmo tempo até o tempo de escolher.

Sensibilidade em relação à mudança na volatilidade

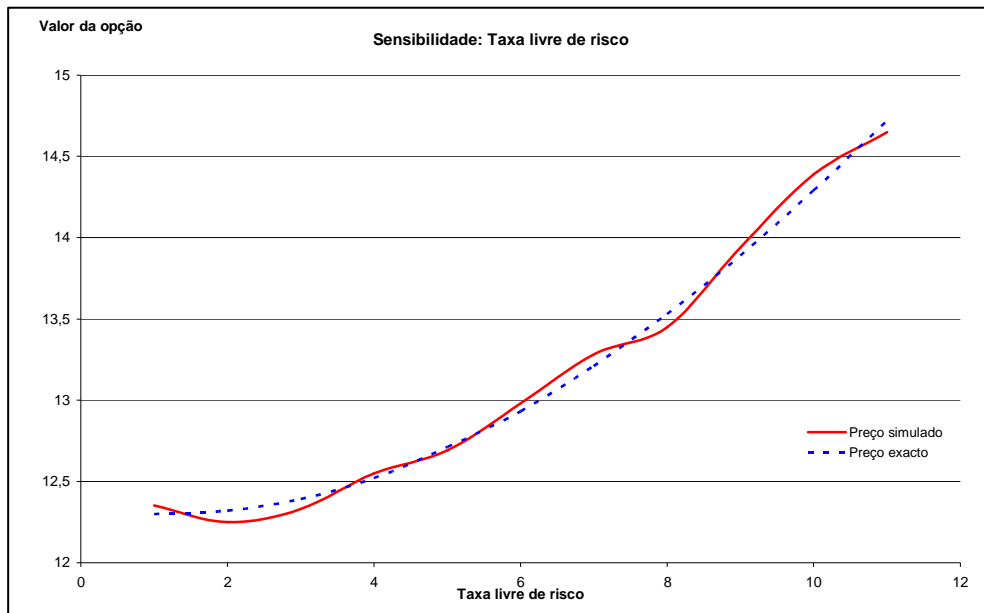


Uma relação linear existe para valores maiores que 10%:

$$\text{Preço Chooser} = 0,5588.v + 11,8189$$

$$R^2 = 0,9984$$

Sensibilidade em relação à mudança na taxa livre de risco



Para o preço teórico existe uma relação quadrática:

$$\text{Preço Chooser} = 0,0231.r^2 - 0,0315.r + 12,292$$

$$R^2 = 0,9998$$

O preço simulado converge para todos os valores da taxa livre de risco.

Apêndice 4

Apresentação dos programas

Para iniciar um cálculo de opção, abrir "*Matlab*" e iniciar o programa:

"*maingui.m*" (RUN).

A janela seguinte aparece:



Selecionar o tipo de opção desejado, e apertar "*Go to calculate*" para entrar no programa particular da opção.

Para cada tipo de opção, a janela que aparece é a seguinte:

European option

The screenshot shows a software window titled "mceuropean" with the subtitle "Monte-Carlo Simulation: European Option". The window is divided into two main sections: "Parameters" and "Value".

Parameters:

Stock price	100
Strike price	105
Time to maturity	1
Risk-free rate (%)	5
Volatility (%)	20
Dividend Yield (%)	0
Number of simulations	100000

Value:

Calculate

Call price	8.01182
Put price	7.89555
Black-Sholes prices	
Call price	8.02135
Put price	7.90044
Errors	
Call price (%)	0.118847
Put price (%)	0.0619514

American option Least-square method

The screenshot shows a software window titled "minquad" with the subtitle "Monte-Carlo Simulation: Least-Square Method American Option". The interface is divided into two main sections: "Parameters" and "Value".

Parameters:

Stock price	100
Strike price	105
Time to maturity	1
Risk-free rate (%)	5
Volatility (%)	20
Dividend Yield (%)	0
Number of simulations	10000
Number of exercising dates	100
Regression degree: 1,2,3,4 Default: use 2	2

Value:

Call price	8.22592
Put price	8.7176

A "Calculate" button is located at the top of the Value section.

Asian option

Monte-Carlo Simulation: Asian Option

Parameters	Value	
Stock price	100	
Strike price	100	
Time to maturity	1	
Risk-free rate (%)	5	
Volatility (%)	20	
Dividend Yield (%)	0	
Number of simulations	2000	
Number of exercising dates	365	
<input type="button" value="Calculate"/>		
<input checked="" type="checkbox"/> Geometric Method		
Call price	5.52367	
Put price	3.44472	
<input checked="" type="checkbox"/> Arithmetic Method		
Call price	5.86822	
Put price	3.42399	
Black-Scholes exacts prices		
Call price	5.94218	
Put price	3.20848	
Errors %		
	Geometric	Arithmetic
Call price	7.04293	1.24468
Put price	7.36299	6.71691

Asset-or-nothing option

The screenshot shows a software window titled "assetornothing" with a subtitle "Monte-Carlo Simulation: European Asset-or-nothing Option". The interface is divided into two main sections: "Parameters" and "Value".

Parameters:

Stock price	100
Strike price	100
Time to maturity	1
Risk-free rate (%)	5
Volatility (%)	20
Dividend rate (%)	0
Number of simulations	1000000

Value:

Calculate

Call price	63.6739
Put price	36.3197

Cash-or-nothing option

The screenshot shows a software window titled "cashornothing" with a subtitle "Monte-Carlo Simulation: European Cash-or-nothing Option". The interface is divided into two main sections: "Parameters" and "Value".

Parameters:

Stock price	100
Strike price	100
Time to maturity	1
Risk-free rate (%)	5
Volatility (%)	20
Dividend rate (%)	0
Premium	40
Number of simulations	1000000

Value:

Calculate	
Call price	21.311
Put price	16.7382
Exact call price	21.293
Exact put price	16.7562
Error call (%)	0.0843983
Error put (%)	0.107249

Lookback option: fixed strike

The screenshot shows a software window titled "fixedstrikelookback" with a subtitle "Monte-Carlo Simulation: Fixed Strike Lookback Option". The interface is divided into two main sections: "Parameters" and "Value".

Parameters:

Stock price	100
Strike price	100
Time to maturity	1
Risk-free rate (%)	5
Volatility (%)	20
Dividend Yield (%)	0
Number of simulations	10000
Number of exercising dates	365

Value:

Calculate

Call price	18.5412
Put price	11.8536

Lookback option: floating strike

The screenshot shows a software window titled "floatingstrikelookback" with a subtitle "Monte-Carlo Simulation: Floating Strike Lookback Option". The interface is divided into two main sections: "Parameters" and "Value".

Parameters:

Stock price	100
Time to maturity	1
Risk-free rate (%)	5
Volatility (%)	20
Dividend Yield (%)	0
Number of simulations	10000
Number of exercising dates	365

Value:

Calculate

Call price	16.7375
Put price	13.6329

Chooser option

The screenshot shows a software window titled "chooser" with a subtitle "Monte-Carlo Simulation: Chooser Option". The interface is divided into two main sections: "Parameters" and "Value".

Parameters:

Stock price	100
Strike price Put/Call	100
Time to maturity Put/call	1
Risk-free rate (%)	5
Volatility (%)	20
Dividend Yield (%)	0
Chooser Time	0.3
Number of simulations	100000

Value:

Calculate

Chooser price	12.5903
Exact chooser price	12.7094
Error (%)	0.936409

Barrier option

The screenshot shows a software window titled "simplebarrier" with the subtitle "Monte-Carlo Simulation: European Barrier Option". The interface is divided into two main sections: "Parameters" and "Value".

Parameters:

Stock price	100
Strike price	105
Barrier	95
Time to maturity	1
Risk-free rate (%)	5
Volatility (%)	20
Dividend Yield (%)	0
Number of simulations	10000
Number of exercising dates	365

Value:

Calculate

Option type

1: Up and out
2: Up and in
3: Down and out
4: Down and in

Call price: 4.75462
Put price: 0.0932278

Apêndice 5

Método de Monte-Carlo

Código "Matlab" dos programas

Variáveis padrões dos programas:

s: Preço inicial
k: Strike
T: Tempo
r: Taxa livre de risco
v: Volatilidade
y: Dividend yield

NumSim: Numero de simulações
DivTemp: Numero de datas de exercício

grau: Grau da regressão

premium: Prêmio

b: Barrier
type: Tipo de opção *barrier*

Opção europeia

```

function [call,put] =
mceuropeanoption(price,strike,time,riskfreerate,volatility,dividendyield,NumSim)

x=price;
k=strike;
r=riskfreerate;
v=volatility;
n=NumSim;
t=time;
y=dividendyield;

%Inicializacao e calculo das variaveis intermediarios
F=(r-y-v^2/2)*t;
FF=v*sqrt(t);
call_vector =0;
call_vector_inv=0;
put_vector =0;
put_vector_inv=0;

for i=1:n/2
%Optimizar por "antithetic"
eps =random('Normal',0,1);
eps_inv=-eps;

%Calculo do valor da opcao para cada simulacao
call_vector =max( x*exp(F+FF*eps)-k,0) +call_vector;
call_vector_inv =max( x*exp(F+FF*eps_inv)-k,0) +call_vector_inv;
put_vector =max(-x*exp(F+FF*eps)+k,0) +put_vector;
put_vector_inv =max(-x*exp(F+FF*eps_inv)+k,0) +put_vector_inv;
end

%Calculo da media do valor do payoff
c=(1/n)*(call_vector + call_vector_inv);
p=(1/n)*( put_vector + put_vector_inv);

% Valor da opcao
call=exp(-r*t)*c;
put=exp(-r*t)*p;

```

Opção americana: Método dos mínimos quadrados de Monte-Carlo

```

function [ put_value,call_value ] = minquadooption(s,k,T,r,v,y,NumSim,DivTemp,grau)

% Divisão de tempo usada pela simulação, (tempo total dividido pelo numero de datas de
exercicios)
dt = T/DivTemp;
% Simulação dos preços
prices=zeros(NumSim,DivTemp);
% Inicialização da primeira coluna da simulação dos preços
for i=1:NumSim
    prices(i,1)=s;
end
% Simulação dos preços do ativo básico
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

% Calculo do Put
% Criação da matriz de calculo do preço da opção no exercicio imediato em cada tempo.
option = max( k-prices , 0);
% Inicialização do valor de continuação ao tempo final, seja o preço da opção.
continue_value=option(:,DivTemp);
% Metodo do MQ para calculo da opção
for j=DivTemp-1:-1:2
    temp=regression( exp(-r*dt)*continue_value , prices(:,j) , option(:,j) , grau);
    % Condicao de exercicio da opcao (exercicio immediatio > valor esperado).
    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
put_value=sum(continue_value)/NumSim;

% Calculo do Call
% Criação da matriz de calculo do preço da opção no exercicio imediato em cada tempo.
option = max( prices-k , 0);
% Inicialização do valor de continuação ao tempo final, seja o preço da opção.
continue_value=option(:,DivTemp);
% Metodo do MQ para calculo da opção
for j=DivTemp-1:-1:2
    temp=regression( exp(-r*dt)*continue_value , prices(:,j) , option(:,j) , grau);
    % Condicao de exercicio da opcao (exercicio immediatio > valor esperado).

```

```

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_value=sum(continue_value)/NumSim;

```

Função de regressão

```

function [vector]=regression(A,B,C,grau)
% Função de calculo dos parametros da regreção e do valor de continuacao da opcao.

% Vetor de valor de continuacao
Y=[];
Y=A;
% Vetor de valor dos precos
X=[];
X=B;
% Vetor de valor da opcao corespondante
Z=[];
Z=C;
% Vetor marcando os caminhos "ITM"
W=[];
n=length(Y);
YY=[];
XX=[];

% Selecao dos preços "in-the-money" e do vetor dos caminhos "in-the-money"
% A linha tem o valor "1" se "ITM", "0" se "out-of-the-money"
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);
        W(j,1)=1;
        i=i+1;
    end
end
end

% Calculo da matriz dos valores de (XX^0,XX,XX^2,XX^3...) para o calculo dos
% coeficientes da regressao, so com os caminhos "in-the-money".
% Dependendo do grau desejado:
if (grau==1) % regressao linear
    matrixx=[ones(size(XX)) XX];
    matrix=[ones(size(X)) X];
elseif (grau==2) % regressao quadratica
    matrixx=[ones(size(XX)) XX XX.*XX];
    matrix=[ones(size(X)) X X.*X];

```

```

elseif (grau==3) %regressao grau 3
    matrixx=[ones(size(XX)) XX XX.*XX XX.*XX.*XX];
    matrix=[ones(size(X)) X X.*X X.*X.*X];
elseif (grau==4) %regressao grau 4
    matrixx=[ones(size(XX)) XX XX.*XX XX.*XX.*XX XX.*XX.*XX.*XX];
    matrix=[ones(size(X)) X X.*X X.*X.*X X.*X.*X.*X];
elseif (grau==5) %regressao grau 5
    matrixx=[ones(size(XX)) XX XX.*XX XX.*XX.*XX XX.*XX.*XX.*XX XX.*XX.*XX.*XX.*XX];
    matrix=[ones(size(X)) X X.*X X.*X.*X X.*X.*X.*X X.*X.*X.*X.*X];
end
%Calculo dos coeficientes da regressão
coef = matrixx\YY;
%Calculo dos precos esperados
vector = (matrix*coef).*W;

```

Opção asset-or-nothing

```
function [call,put] = assetnothingoption(s,k,T,r,v,y,NumSim)
```

```

%Inicialização das variaveis usadas
eps =0;
price =0;
price_inv=0;
valuecall=0;
valueput =0;
call =0;
put =0;
%Simulação dos preços do ativo basico e calculo do valor do payoff em cada
tempo
for i=1:NumSim/2
    eps =normrnd(0,1);
    price =s*exp( (r-y-v^2/2)*T + v*eps*sqrt(T) );
    price_inv=s*exp( (r-y-v^2/2)*T - v*eps*sqrt(T) );
    %Calculo do payoff do call
    if(price>k)
        valuecall =valuecall + price;
    end
    if(price_inv>k)
        valuecall =valuecall + price_inv;
    end
    %Calculo do payoff do put
    if(k>price)
        valueput =valueput + price;
    end
    if(k>price_inv)
        valueput =valueput + price_inv;
    end
end
call=exp(-r*T)*(valuecall/NumSim);
put=exp(-r*T)* (valueput /NumSim);

```

Opção cash-or-nothing

```
function [call,put] = cashornothingoption(s,k,T,r,v,y,premium,NumSim)
```

```
% Inicialização das variáveis usadas
```

```
eps = 0;
price = 0;
price_inv = 0;
valuecall = 0;
valueput = 0;
call = 0;
put = 0;
```

```
% Simulação dos preços do ativo básico e cálculo do valor do payoff em cada
% tempo
```

```
for i=1:NumSim/2
    eps = normrnd(0,1);
    price = s*exp((r-y-v^2/2)*T + v*eps*sqrt(T));
    price_inv = s*exp((r-y-v^2/2)*T - v*eps*sqrt(T));
```

```
% Cálculo do payoff do call
```

```
if(price > k)
    valuecall = valuecall + premium;
end
if(price_inv > k)
    valuecall = valuecall + premium;
end
```

```
% Cálculo do payoff do put
```

```
if(k > price)
    valueput = valueput + premium;
end
if(k > price_inv)
    valueput = valueput + premium;
end
```

```
end
```

```
call = exp(-r*T)*(valuecall/NumSim);
put = exp(-r*T)*(valueput/NumSim);
```

```
% Função de cálculo exacto:
```

```
function [call,put] =
exactcashornothingoption(s,k,T,r,v,y,premium)
m)
```

```
% Reiner e Rubinstein 1991 solução
```

```
d = (log(s/k) + (r-y-v^2/2)*T) / (v*sqrt(T));
call = exp(-r*T)*normcdf(d)*premium;
put = exp(-r*T)*normcdf(-d)*premium;
```

Opção lookback: *fixed strike*

```

function [call,put] = fixedstrikelookbackoption(s,k,T,r,v,y,NumSim,DivTemp)

% Divisão de tempo usada pela simulação, (tempo total dividido pelo numero de
% datas de exercicios)
dt = T/DivTemp;
% Simulação dos preços
prices=zeros(NumSim,DivTemp);
% Inicialização da primeira coluna da simulação dos preços
for i=1:NumSim
    prices(i,1)=s;
end
% Simulação dos preços do ativo basico
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

% Pesquisando o minimo e o maximo para cada caminho
for i=1:NumSim
    callpayoff(i,1)=max( max(prices(i,:))-k ,0);
    putpayoff (i,1)=max( k-min(prices(i,:)) ,0);
end

call=exp(-r*T)*(sum(callpayoff)/NumSim);
put =exp(-r*T)*(sum(putpayoff) /NumSim);

```

Opção lookback: *floating strike*

```
function [call,put] = floatingstrikelookbackoption(s,T,r,v,y,NumSim,DivTemp)
```

```
% Divisão de tempo usada pela simulação, (tempo total dividido pelo numero de  
datas de exercicios)
```

```
dt = T/DivTemp;
```

```
% Simulação dos preços
```

```
prices=zeros(NumSim,DivTemp);
```

```
% Inicialização da primeira coluna da simulação dos preços
```

```
for i=1:NumSim
```

```
    prices(i,1)=s;
```

```
end
```

```
% Simulação dos preços do ativo basico
```

```
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
```

```
% Pesquisando o minimo e o maximo para cada caminho
```

```
for i=1:NumSim
```

```
    callpayoff(i,1)=max( prices(i,DivTemp)-min(prices(i,:)) ,0);
```

```
    putpayoff (i,1)=max( max(prices(i,:))-prices(i,DivTemp) ,0);
```

```
end
```

```
call=exp(-r*T)*(sum(callpayoff)/NumSim);
```

```
put =exp(-r*T)*(sum(putpayoff) /NumSim);
```


Opção asiática: método geométrico

```
function [call,put] =
geometricasianoptionmontecarlo(s,k,T,r,v,y,NumSim,DivTemp)

% Divisão de tempo usada pela simulação, (tempo total dividido pelo numero de
datas de exercicios)
dt = T/DivTemp;
% Simulação dos preços
prices=zeros(NumSim,DivTemp);
% Inicialização da primeira coluna da simulação dos preços
for i=1:NumSim
    prices(i,1)=s;
end
% Simulação dos preços do ativo basico
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
% Inicialização do vetor coluna recebendo o produto dos preços
for i=1:NumSim
    payoff(i,1)=1;
end
% Calculo do producto dos preços e da raiz "^(1/DivTemp)"
for j=1:DivTemp
    payoff(:,1) = payoff(:,1).*(prices(:,j).^(1/DivTemp));
end
% Calculo do payoff final de cada simulação
callpayoff(:,1)= max(payoff(:,1)-k,0);
putpayoff(:,1) = max(k-payoff(:,1),0);

call=sum(callpayoff(:,1))/NumSim;
call=call*exp(-r*T);

put=sum(putpayoff(:,1))/NumSim;
put=put*exp(-r*T);
```

```
% Função de calculo exacto:
```

```
function [ call,put ] =
geometricasianoption(s,k,T,r,v,y)
% Modificação dos parametros: volatilidade e
dividend yield
va = v/sqrt(3);
ya = (r-y-v^2/6)/2;
[call,put]=blsprice(s,k,r,T,va,ya);
```

Opção asiática: método aritmético

```

function [call,put] =
arithmeticsianoptionmontecarlo(s,k,T,r,v,y,NumSim,DivTemp)

% Divisão de tempo usada pela simulação, (tempo total dividido pelo numero de
datas de exercicios)
dt = T/DivTemp;
% Simulação dos preços
prices=zeros(NumSim,DivTemp);
% Inicialização da primeira coluna da simulação dos preços
for i=1:NumSim
    prices(i,1)=s;
end
% Simulação dos preços do ativo basico
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

% Inicialização do vetor coluna recebendo a soma dos preços
for i=1:NumSim
    payoff(i,1)=0;
end

% Calculo da soma dos preços
for j=1:DivTemp
    payoff(:,1) = payoff(:,1)+prices(:,j);
end

% Calculo da media do payoff em cada simulação
payoff(:,1)=payoff(:,1)/DivTemp;

% Calculo do payoff final de cada simulação
callpayoff(:,1)= max(payoff(:,1)-k,0);
putpayoff(:,1) = max(k-payoff(:,1),0);

call=sum(callpayoff(:,1))/NumSim;
call=call*exp(-r*T);

put=sum(putpayoff(:,1))/NumSim;
put=put*exp(-r*T);

```

Opção chooser

```

function [chooservalue ] = chooseroption(s,k,T,t,r,v,y,NumSim)

price=0;
price_inv=0;
valuecall=0;
valueput =0;
numbercall=0;
numberput=0;
unpriceablepaths=0;

% Simulação dos preços do ativo básico e cálculo do valor do payoff em cada
tempo
for i=1:NumSim/2
    eps =normrnd(0,1);
    epss=normrnd(0,1);

    % Simulação do preço dependendo do call ou do put escolhido no tempo t
    price      =s      *exp( (r-y-v^2/2)*t  + v*eps*sqrt(t) );
    pricee     =price*exp( (r-y-v^2/2)*(T-t) + v*epss*sqrt(T-t) );
    [blscall,blsput] =blsprice(price,k,r,T-t,v,y);

    if (blscall>blsput)
        valuecall      =valuecall  + max(pricee-k,0);
        numbercall     =numbercall+1;
    elseif (blsput>blscall)
        valueput       =valueput   + max(k-pricee,0);
        numberput      =numberput+1;
    else
        unpriceablepaths =unpriceablepaths+1;
    end

    price_inv      =s      *exp( (r-y-v^2/2)*t  - v*eps*sqrt(t) );
    price_invv     =price_inv*exp( (r-y-v^2/2)*(T-t) - v*epss*sqrt(T-t) );
    [blscalltemp,blsputtemp] =blsprice(price_inv,k,r,T-t,v,y);

    if (blscalltemp>blsputtemp)
        valuecall      =valuecall  + max(price_invv-k,0);
        numbercall     =numbercall+1;
    elseif (blsputtemp>blscalltemp)
        valueput       =valueput   + max(k-price_invv,0);
        numberput      =numberput+1;
    else
        unpriceablepaths=unpriceablepaths+1;
    end
end

average=(valuecall+valueput)/(numbercall+numberput);
chooservalue=average*exp(-r*T);

```

Opção *barrier*

```
function [call,put] = simplebarrieroption(s,k,b,T,r,v,y,NumSim,DivTemp,type)
```

```
% valor do "type":
% 1: up and out
% 2: up and in
% 3: down and out
% 4: down and in
```

```
% Inicialização das variáveis
valuecall =0;
valueput =0;
optionactive=0;
```

```
% Divisão de tempo usada pela simulação, (tempo total dividido pelo numero de
datas de exercicios)
```

```
dt = T/DivTemp;
```

```
% Simulação dos preços
```

```
prices=zeros(NumSim,DivTemp);
```

```
% Inicialização da primeira coluna da simulação dos preços
```

```
for i=1:NumSim
```

```
    prices(i,1)=s;
```

```
end
```

```
% Simulação dos preços do ativo basico
```

```
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
```

```
% type=1: up and out
```

```
if (type==1)
```

```
    for i=1:NumSim
```

```
        optionactive=b-max(prices(i,:));
```

```
        % Opção esta ativa se optionactive positivo
```

```
        if (optionactive>0)
```

```
            valuecall=max(prices(i,DivTemp)-k,0)+valuecall;
```

```
            valueput =max(k-prices(i,DivTemp),0)+valueput ;
```

```
        end
```

```
    end
```

```
    call=exp(-r*T)*valuecall/NumSim;
```

```

    put =exp(-r*T)*valueput /NumSim;
end

```

```

%type=2: up and in

```

```

if (type==2)
    for i=1:NumSim
        optionactive=max(prices(i,:))-b;
        % Opção esta ativa se optionactive positivo
        if (optionactive>0)
            valuecall=max(prices(i,DivTemp)-k,0)+valuecall;
            valueput =max(k-prices(i,DivTemp),0)+valueput ;
        end
    end
    call=exp(-r*T)*valuecall/NumSim;
    put =exp(-r*T)*valueput /NumSim;
end

```

```

%type=3: down and out

```

```

if (type==3)
    for i=1:NumSim
        optionactive=min(prices(i,:))-b;
        % Opção esta ativa se optionactive positivo
        if (optionactive>0)
            valuecall=max(prices(i,DivTemp)-k,0)+valuecall;
            valueput =max(k-prices(i,DivTemp),0)+valueput ;
        end
    end
    call=exp(-r*T)*valuecall/NumSim;
    put =exp(-r*T)*valueput /NumSim;
end

```

```

%type=4: down and in

```

```

if (type==4)
    for i=1:NumSim
        optionactive=b-min(prices(i,:));
        % Opção esta ativa se optionactive positivo
        if (optionactive>0)
            valuecall=max(prices(i,DivTemp)-k,0)+valuecall;
            valueput =max(k-prices(i,DivTemp),0)+valueput ;
        end
    end
    call=exp(-r*T)*valuecall/NumSim;
    put =exp(-r*T)*valueput /NumSim;
end

```

Apêndice 6

Opção americana sem dividendos Modelo de Coxx-Ross-Rubinstein

Nesse modelo, r a taxa livre de risco é considerada constante.

Seja S_n o valor do ativo no instante n :

Assim existe duas constantes a e b tais que: $-1 < a < b$ e que $\frac{S_{n+1}}{S_n}$ só pode pegar os valores “ $(1+a)$ ” ou “ $(1+b)$ ” no instante $(n+1)$, com probabilidades “ p ” e “ $(p-1)$ ” respectivamente:

Instante n	Instante $(n+1)$
S_n	$S_{n+1} = S_n \cdot (1+a)$ Valor S_U com probabilidade p
	$S_{n+1} = S_n \cdot (1+b)$ Valor S_D com probabilidade $(p-1)$

Usando a definição da opção de venda seguinte:

Preços do ativo no inicio	$S=100$
Strike da opção	$k=100$
Tempo a expiração (ano)	$T=1$
Taxa livre de risco	$R=5\%$
Volatilidade do ativo	$v=20\%$
Dividend Yield	$y=0$

Para avaliar as opções americanas nesse contexto e usando o método binomial devemos escolher os parâmetros assim:

- N Numero de simulações
- $r = \frac{R.T}{N}$ A taxa livre de risco no período.
- $(1+a) = (1+r).e^{-\sigma.\sqrt{\frac{T}{N}}}$ Definição do parâmetro a
- $(1+b) = (1+r).e^{\sigma.\sqrt{\frac{T}{N}}}$ Definição do parâmetro b
- $p = \frac{b-r}{b-a}$ Definição da probabilidade neutra ao risco

Foi provado que com esses parâmetros o modelo binomial converge, por N grande, no valor da opção americana sem dividendos.

Obtemos um algoritmo rápido e determinístico para avaliar uma opção americana sem dividendos e onde o único parâmetro importante é “ N ” que deve ser alto para ter precisão no calculo. O tempo computacional para $N = 5000$ está de 1 minuto aproximadamente.

Os resultados da convergência são:

Numero de simulações	Valor da opção de venda
<i>10</i>	<i>6.1842</i>
<i>50</i>	<i>6.0917</i>
<i>100</i>	<i>6.0945</i>
<i>500</i>	<i>6.0915</i>
<i>1000</i>	<i>6.0898</i>
<i>5000</i>	<i>6.0904</i>

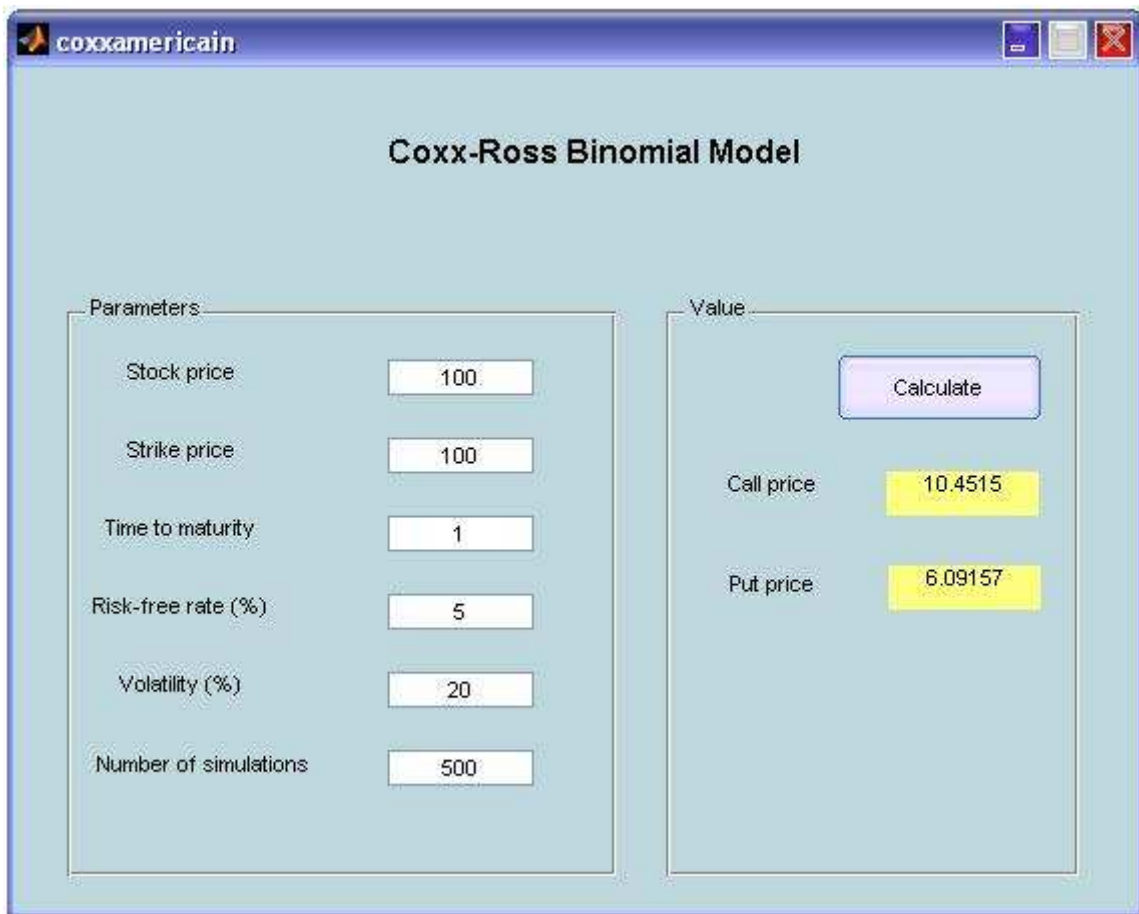
A precisão do calculo esta rapidamente boa, porque com $N = 1000$ o erro é só de 0,01% em comparação ao valor teórico, obtido com $N=5000$.

O preço da opção europeia de venda com os mesmos parâmetros é 5.5735, valor menor que a opção americana.

O preço da opção de compra americana sem dividendos é o mesmo que a opção europeia, porque foi provado que o exercício antecipado nunca é ótimo. Esse preço teórico é dado pela formula de Black-Scholes. (ver anexo 2).

Os resultados da convergência são:

Numero de simulações	Valor do Call	Valor Black-Scholes	Erro (%)
<i>10</i>	<i>10.6085</i>		<i>1.51</i>
<i>50</i>	<i>10.4423</i>		<i>0.08</i>
<i>100</i>	<i>10.4586</i>		<i>0.07</i>
<i>500</i>	<i>10.4515</i>	<i>10.4506</i>	<i>0.009</i>
<i>1000</i>	<i>10.4493</i>		<i>0.008</i>
<i>5000</i>	<i>10.4506</i>		<i>0</i>



Código Matlab do programa:

```
function [call,put] =
coxxamericainoption(price,strike,time,riskfreerate,volatility,NumSim)

%Definição das constantes no calculo
s=price; k=strike; t=time; R=riskfreerate; v=volatility; n=NumSim;
%Taxa ajustada no período
r=R*t/n;

A=(1+r)*exp(-v*sqrt(t/n));
B=(1+r)*exp(v*sqrt(t/n));

p=((B-1)-r) / ((B-1)-(A-1)); %Probabilidade neutra ao risco de queda
q=1-p; %Probabilidade neutra ao risco de subida
m=zeros(n+1,n+1);

%Calculo da opção call
for j=1:n+1
for i=1:j
m(i,j)= max ( s*(B)^(i-1)*(A)^(j-i)-k , 0 );
end
```

```

end

for j=n+1:-1:2
  for i=j:-1:2
    m(i-1,j-1)= max ( (q*m(i,j)+p*m(i-1,j))/(1+r) , m(i-1,j-1) );
  end
end
call = m(1,1);

```

%Calculo da opção put

```

for j=1:n+1
  for i=1:j
    m(i,j)= max ( k-s*(B)^(i-1)*(A)^(j-i) , 0 );
  end
end

for j=n+1:-1:2
  for i=j:-1:2
    m(i-1,j-1)= max ( (q*m(i,j)+p*m(i-1,j))/(1+r) , m(i-1,j-1) );
  end
end
put = m(1,1);

```