

Fernando Prieto

**Estudo do apreçamento de opções com dois ativos,
volatilidade estocástica e custos de transação,
por um processo de diferenças finitas**

Área de concentração: Engenharia Financeira

Monografia apresentada à Escola Politécnica da Universidade de São Paulo como requisito para obtenção do título MBA em Engenharia Financeira.

Orientador: Prof. Dr. André Cury Maialy

São Paulo

2014

MBA-EF
2014
P933-e

591



Escola Politécnica - EPEL



31500023591

Ficha Catalográfica

M2014AP *

Prieto, Fernando

Estudo do apreçamento de opções com dois ativos, volatilidade estocástica e custos de transação, por um processo de diferenças finitas / F. Prieto - São Paulo, 2014.

162p.

Monografia (Especialização em Engenharia Financeira) - Escola Politécnica da Universidade de São Paulo. Programa de Educação Continuada em Engenharia.

1. Apreçamento de opções 2. Custos de transação 3. Diferenças finitas 4. Volatilidade estocástica. Universidade de São Paulo. Escola Politécnica. Programa de Educação Continuada em Engenharia. II.t

[2716337]

Agradecimento

*Ao Prof. Dr. André Cury Maialy pelos esclarecimentos durante
a execução deste trabalho e pela confiança.*

*À memória de meus avós Armando e Vicente,
pelo que pude aprender com eles e ao meu filho Gabriel,
para que sirva como um pequeno exemplo na sua caminhada.*

Abstract

In this work we make a careful and detailed analysis of some aspects present in option pricing, taking into account transaction costs and stochastic volatility. We make use of a finite difference method for the solution of the nonlinear partial differential equation models.

Initially we do a brief introduction on the finite difference method and some basic transaction costs models on option pricing.

Then we show the derivation of a mathematical model for option pricing with transaction costs and stochastic volatility, based on a modified Hull-White process containing general drift in terms of asset and volatility. After that we show that, with minor changes to this model, we can obtain another model for the case of an option with n correlated assets and deterministic volatilities.

Finite difference models were developed for pricing “plain vanilla” options, exotic “up & out call” barrier options and “basket” options with two assets. The goal was to show not only some cases in which the models can be applied but also that they are able to deal with many other “payoffs” with European characteristics and also for n assets. Various graphical outputs were implemented in Matlab® pricing scripts, so that we could clearly see the power of the models in the evaluation of options Greek’s, fundamental for financial market professionals in their risk analysis and portfolios hedge.

To make possible a comparison and evaluation of the quality, efficiency and accuracy of the models, Monte Carlo simulations were developed to price the same types of options considered by the finite difference models.

Various validation and robustness tests were developed, analyzing particular cases of “payoffs” and comparing the solution provided by the models with closed formula analytical solutions and the results from other work found in the finance literature.

Finally, several simulations were performed and the results were presented in tables, with error estimates relative to other pricing methods. All Matlab® scripts used in this work are presented in the C Annex.

Aplicação da teoria e verificação dos modelos pela simulação de alguns casos específicos	49
4.1 Introdução.....	49
4.2 Simulação de uma opção “ <i>basket call</i> ” com volatilidades determinísticas composta por dois ativos.....	51
4.3 Simulação de uma opção “ <i>basket call</i> ” com barreira e volatilidades determinísticas composta por dois ativos.....	56
4.4 Análise de uma opção de compra europeia “ <i>vanilla</i> ” com volatilidade estocástica.....	60
4.5 Análise de uma opção de compra europeia com barreira “ <i>up&out call</i> ” e volatilidade estocástica	67
Discussão dos resultados.....	73
Conclusões	77
Pesquisas futuras.....	79
7.1 Discretização implícita	79
7.2 Implementação de um método ADI.....	79
7.3 Utilização de malha de discretização não uniforme.....	80
7.4 Implementação do modelo de Heston em diferenças finitas	80
7.5 Treinamento de uma rede neural artificial a partir dos modelos desenvolvidos	80
7.6 Utilização de variáveis de controle nas simulações de Monte Carlo	80
7.7 Custos de transação nas simulações de Monte Carlo	81
7.8 Brownian Bridge para as simulações de Monte Carlo das opções com barreiras	81
Anexo A.....	83
A1 Análise do modelo de apreçamento de uma opção de compra europeia “ <i>vanilla</i> ” para dois ativos com volatilidades determinísticas	83
A2 Análise do modelo de apreçamento de uma opção de compra europeia com barreira “ <i>up&out call</i> ” para dois ativos com volatilidades determinísticas	87
A3 Análise de simetria do modelo desenvolvido para uma opção de compra europeia “ <i>vanilla</i> ” com dois ativos e volatilidades determinísticas	91
A4 Análise de simetria do modelo desenvolvido para uma opção de compra europeia com barreira “ <i>up&out call</i> ”, dois ativos e volatilidades determinísticas.....	95
Anexo B.....	99
B1 Análise do modelo para uma opção de compra europeia “ <i>vanilla</i> ” e volatilidade estocástica - Comparação com outro trabalho	99
B2 Análise do modelo de volatilidade estocástica em relação a modelos com volatilidade determinística para uma opção de compra europeia “ <i>vanilla</i> ”	103
B3 Análise do modelo de volatilidade estocástica em relação a modelos com volatilidade determinística para uma opção de compra europeia com barreira do tipo “ <i>up&out call</i> ”	108
Anexo C - Códigos Matlab®	113
C1 Opção europeia “ <i>call</i> ” com modelo de Hoggard, Whalley e Wilmott (1992) para os custos de transação	113
C2 Opção europeia com barreira “ <i>up&out call</i> ” com modelo de Hoggard, Whalley e Wilmott (1992) para os custos de transação.....	116
C3 Opção europeia “ <i>call</i> ” com volatilidade estocástica e custos de transação.....	119
C4 Opção europeia com barreira “ <i>up&out call</i> ” com volatilidade estocástica e custos de transação	127
C5 Simulação de Monte Carlo para opção europeia “ <i>call</i> ” com volatilidade estocástica - Heston - Milstein	135

Capítulo 1

Introdução

1.1 Motivação para este trabalho

No mercado financeiro, cada vez mais são lançados produtos derivativos cujo apreçamento correto levando-se em conta os custos de transação depende de soluções de equações complexas ou simulações numéricas que podem ser muito demoradas. Além disso tais produtos podem ser exóticos, compostos por vários ativos, ou ainda apresentar volatilidade estocástica. Este último caso é um problema de longa data em finanças e sua solução torna-se importante quando dados de alta frequência são considerados. Nestes casos a volatilidade é estocástica e portanto é necessário trabalhar com modelos que levam em conta volatilidade como uma variável estocástica [1].

Além da solução matemática do problema, para a implementação prática no mercado financeiro, necessita-se obter o preço considerando os custos de transação. Surge então a necessidade da obtenção de modelos matemáticos capazes de fazer o apreçamento de derivativos cujo ativo negociado requer o pagamento de taxas de transação que são proporcionais à quantidade e ao valor dos ativos base.

Neste trabalho pretendemos investigar com cuidado, ferramentas de apreçamento de alta relevância para profissionais do mercado financeiro que trabalham com opções ou análises de risco de investimentos.

Vamos estudar alguns casos de apreçamento de opções, pela da solução numérica da equação diferencial parcial não linear, levando-se em conta custos de transação, volatilidade estocástica e ainda a presença de dois ativos base no produto.

Os modelos parecem bastante próximos da realidade do mercado, pois podem levar em conta os custos de transação em função do valor dos ativos, assim como uma volatilidade estocástica, como é o caso de um mercado completo, onde ela é um ativo negociável. Os modelos também podem auxiliar a análise do VaR de uma carteira, ou ainda possibilitar de forma bastante simples e direta o cálculo de Δ e Γ hedge.

1.2 Objetivos do trabalho

Este trabalho apresenta um método explícito de diferenças finitas para a solução de uma equação diferencial parcial complexa aplicada em finanças, cuja solução determina o preço de opções incluindo os custos de transação e volatilidade estocástica. Uma pequena alteração na equação diferencial deste modelo permite o cálculo do preço de opções exóticas compostas por dois ativos e volatilidades determinísticas. Neste trabalho consideramos um modelo de mercado no qual a transação de ativos requer o pagamento de taxas que são proporcionais ao valor e a quantidade do ativo negociado, e esse ativo pode ser aproximado usando-se um modelo de volatilidade estocástica.

Casos de opções de compra europeias tipo “*plain vanilla*” e exóticas com barreiras tipo “*up&out call*” serão simuladas numericamente e os resultados comparados com os resultados de trabalhos

Capítulo 2

Revisão Bibliográfica

2.1 Equação diferencial parcial (EDP) para volatilidade estocástica

Uma das maneiras de se fazer o apreçamento de opções com volatilidade estocástica pelo processo de diferenças finitas é apresentado em [2] resolvendo a equação diferencial parcial (EDP) de Heston ao longo de uma malha bidimensional que representa o preço do ativo e a volatilidade deste ativo. O texto mostra como construir malhas uniformes e não uniformes para a discretização do valor dos ativos e da volatilidade e apresenta fórmulas de aproximação por diferenças finitas para as derivadas contidas na EDP de Heston.

São apresentados os métodos mais comuns que incluem esquemas explícito, implícito e o de Crank-Nicolson como um caso especial. O autor mostra ainda as condições de contorno da EDP para uma opção de compra europeia “*vanilla*”. O texto apresenta também um método ADI (Alternate Direction Implicit) que produz resultados precisos para um pequeno número de intervalos de tempo de integração. O autor não comenta nada sobre custos de transação.

A equação a seguir é a EDP de Heston para volatilidade estocástica

$$\frac{\partial V}{\partial t} + \frac{1}{2} \nu S^2 \frac{\partial^2 V}{\partial S^2} + \rho \sigma \nu S \frac{\partial^2 V}{\partial \nu \partial S} + \frac{1}{2} \nu \sigma^2 \frac{\partial^2 V}{\partial \nu^2} + (r - D) S \frac{\partial V}{\partial S} + [\kappa(\theta - \nu) - \lambda(S, \nu, t)] \frac{\partial V}{\partial \nu} - rV = 0$$

onde ν é a variância, S é o valor do ativo base, ρ é a correlação entre o valor do ativo e a variância, σ é a volatilidade da variância, r é a taxa de juros livre de risco, D é a taxa de dividendos, κ é a velocidade de reversão à média, θ é a variância média e $\lambda(S, \nu, t)$ representa um prêmio de risco da volatilidade.

Podemos usar uma notação compacta da EDP, onde L é um operador linear.

$$\frac{\partial V}{\partial t} = LV(t)$$

$$L = \frac{1}{2} \nu S^2 \frac{\partial^2}{\partial S^2} + \rho \sigma \nu S \frac{\partial^2}{\partial \nu \partial S} + \frac{1}{2} \nu \sigma^2 \frac{\partial^2}{\partial \nu^2} + (r - D) S \frac{\partial}{\partial S} + \kappa(\theta - \nu) \frac{\partial}{\partial \nu} - r$$

2.2 Malhas de discretização para diferenças finitas

Malhas uniformes são aquelas nas quais as distâncias entre os pontos discretizados são constantes, podendo diferir para cada uma das variáveis. Estas malhas possuem duas vantagens: são mais fáceis de se construir e como os incrementos são iguais, as aproximações das derivadas do operador L por diferenças finitas são mais simples de se obter.

Malhas não uniformes são mais difíceis de se obter e as aproximações das derivadas do operador L por diferenças finitas também serão mais complicadas. As malhas de discretização podem ser

2.4 Custos de transação

Custos de transação são os custos resultantes da compra e venda dos ativos base, relacionado ao “*spread*” entre os preços de compra e venda. A análise de Black-Scholes supõe o rebalanceamento contínuo do portfólio sem custos de transação. Na prática essa premissa é incorreta. Dependendo do mercado em questão os custos de transação podem ou não ser importantes. Num mercado com altos custos de transação, ações nos mercados emergentes por exemplo, seria muito custoso fazer o “*hedge*” frequentemente. Em mercados mais líquidos, papéis do governo em países do primeiro mundo por exemplo, os custos são menores e os portfólios poderão ser rebalanceados com maior frequência.

Os modelos de custos de transação foram iniciados por Hayne Leland [6]. Existem outros modelos de custos de transação, e um modelo mais elaborado mais ainda bastante simples, é modelo de Hoggard-Whalley-Wilmott, que também é baseado na estratégia de “*hedge*” e no modelo de Leland.

A análise de Black-Scholes assume que as operações de “*hedge*” são realizadas continuamente, ou seja em intervalos de tempo que tendem a zero. Devido à diferença entre os preços de compra e venda do ativo base, as operações de “*hedge*” terão um custo infinito.

Os custos de transação são sempre um gasto de dinheiro para os operadores que estejam fazendo “*hedge*”, eles sempre vão perder na diferença entre compra e venda. Desta forma espera-se que uma posição comprada seja avaliada abaixo do valor de Black-Scholes e que uma posição vendida seja avaliada acima do valor de Black-Scholes. Independentemente da posição ser comprada ou vendida, os custos das operações de “*hedge*” devem ser deduzidos do valor da opção. Desta forma o sinal do “*payoff*” é importante. É natural pensar que uma posição comprada tenha um “*payoff*” positivo e uma posição vendida tenha um “*payoff*” negativo.

2.4.1 O modelo de Leland 1985

O trabalho base de modelamento dos custos de transação foi feito por Leland (1985). Ele adotou uma estratégia de fazer o “*hedge*” a cada intervalo δt independentemente se essa estratégia é ótima ou não.

Os custos de transação são proporcionais ao valor dos ativos base negociados, portanto se v ações forem compradas ($v > 0$) ou vendidas ($v < 0$) pelo preço S , então o custo será $\kappa |v| S$, onde κ é uma constante que representa o custo.

Leland reconheceu que os custos de transação, igualando o valor do portfólio ao retorno esperado do mesmo portfólio corrigido por uma taxa de retorno livre de risco, devia ser avaliado através de uma volatilidade ajustada.

$$\bar{\sigma} = \sigma \left(1 \pm \sqrt{\frac{8}{\pi \delta t}} \frac{\kappa}{\sigma} \right)^{1/2}$$

2.4.2 O modelo de Hoggard, Whalley e Wilmott 1992

A estratégia de Leland pode ser aplicada a “*payoffs*” arbitrários e a portfólios de opções mas o

O parâmetro H_0 é uma medida do máximo risco esperado para o portfólio. Quando o “*hedge perfeito*” $\partial V / \partial S$ e o “*hedge atual*” Q se movem sobre uma linha em que a equação acima é violada, a posição do portfólio deve ser rebalanceada. A equação define então uma Largura de Banda (*Bandwidth*) da posição de “*hedge*”.

O modelo de Whalley e Wilmott 1993 e Heron 1993 leva em conta essa estratégia de “*hedge*”. Portanto o investidor deve prescrever uma quantidade H_0 e rebalancear o “*hedge*” em $Q = \partial V / \partial S$.

O valor da opção deve então satisfazer a seguinte equação diferencial parcial não linear:

$$\frac{\partial V}{\partial t} + (r - D)S \frac{\partial V}{\partial S} + \frac{1}{2} \sigma^2 S^2 \frac{\partial^2 V}{\partial S^2} - rV = \frac{\sigma^2 S^4 \Gamma^2}{H_0} \left[k_1 + (k_2 + k_3 S) \frac{H_0^{1/2}}{S} \right]$$

onde

$$\Gamma = \frac{\partial^2 V}{\partial S^2}$$

Existem outros modelos de custos de transação que podem ser vistos em [6]. Neste trabalho modelaremos os custos de transação de uma forma parecida ao modelo de Hoggard, Whalley e Wilmott 1992.

2.5 Multi-asset options

Uma equação convectiva-difusiva de n dimensões contém termos convectivos e difusivos como o próprio nome já diz. Essas equações receberam muita atenção na literatura de engenharia nos últimos 50 anos pois elas modelam vários tipos de problemas físicos. Um dos principais em dinâmica dos fluidos é a equação de Navier-Stokes e seus casos particulares. Em engenharia financeira podemos ver a equação de Black-Scholes como um caso particular de uma equação convectiva-difusiva [7].

$$\frac{\partial V}{\partial t} + \frac{1}{2} \sum_{i,j=1}^n \rho_{ij} \sigma_i \sigma_j \frac{\partial^2 V}{\partial S_i \partial S_j} + \sum_{i=1}^n (r - D_i) S_i \frac{\partial V}{\partial S_i} - rV = 0$$

Neste caso temos n ativos base e U é o valor da opção. Podemos observar a presença de termos cruzados se os ativos apresentarem correlação entre eles e o esquema de diferenças finitas resultante deve produzir aproximações precisas para estes termos. Podemos desta forma modelar opções de mais de um ativo.

A seguir podemos ver um exemplo desta equação para dois ativos correlacionados.

$$\frac{\partial V}{\partial t} + \frac{1}{2} \sigma_1^2 S_1^2 \frac{\partial^2 V}{\partial S_1^2} + \rho \sigma_1 \sigma_2 S_1 S_2 \frac{\partial^2 V}{\partial S_1 \partial S_2} + \frac{1}{2} \sigma_2^2 S_2^2 \frac{\partial^2 V}{\partial S_2^2} + (r - D_1) S_1 \frac{\partial V}{\partial S_1} + (r - D_2) S_2 \frac{\partial V}{\partial S_2} - rV = 0$$

Opções europeias com barreira "up&out call"

$$V_{up\&out_call}(S_T, t) = S_t e^{-q(T-t)} \left(N(d_1) - N(d_3) - b(N(d_6) - N(d_8)) \right) \dots \\ - K e^{-r(T-t)} \left(N(d_2) - N(d_4) - a(N(d_5) - N(d_7)) \right)$$

onde

$$d_1 = \frac{\ln\left(\frac{S_0}{K}\right) + \left(r - q + \frac{1}{2}\sigma^2\right)(T-t)}{\sigma\sqrt{(T-t)}}; d_5 = \frac{\ln\left(\frac{S_0}{S_b}\right) + \left(r - q - \frac{1}{2}\sigma^2\right)(T-t)}{\sigma\sqrt{(T-t)}}$$

$$d_2 = \frac{\ln\left(\frac{S_0}{K}\right) + \left(r - q - \frac{1}{2}\sigma^2\right)(T-t)}{\sigma\sqrt{(T-t)}}; d_6 = \frac{\ln\left(\frac{S_0}{S_b}\right) + \left(r - q + \frac{1}{2}\sigma^2\right)(T-t)}{\sigma\sqrt{(T-t)}}$$

$$d_3 = \frac{\ln\left(\frac{S_0}{S_b}\right) + \left(r - q + \frac{1}{2}\sigma^2\right)(T-t)}{\sigma\sqrt{(T-t)}}; d_7 = \frac{\ln\left(\frac{SK}{S_b^2}\right) + \left(r - q - \frac{1}{2}\sigma^2\right)(T-t)}{\sigma\sqrt{(T-t)}}$$

$$d_4 = \frac{\ln\left(\frac{S_0}{S_b}\right) + \left(r - q - \frac{1}{2}\sigma^2\right)(T-t)}{\sigma\sqrt{(T-t)}}; d_8 = \frac{\ln\left(\frac{SK}{S_b^2}\right) + \left(r - q + \frac{1}{2}\sigma^2\right)(T-t)}{\sigma\sqrt{(T-t)}}$$

$$a = \left(\frac{S_b}{S_0}\right)^{-1+(2(r-q)/\sigma^2)}; b = \left(\frac{S_b}{S_0}\right)^{1+(2(r-q)/\sigma^2)}$$

$$N(x) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^x e^{-\frac{\xi^2}{2}} d\xi$$

Capítulo 3

Desenvolvimento dos Modelos Matemáticos

3.1 Modelagem matemática dos casos a serem estudados

Para analisar alguns casos de apreçamento de derivativos vamos estudar modelos matemáticos que resultem em equações diferenciais parciais e propor um método de solução numérica, que leve em conta volatilidades determinísticas e estocásticas. Os modelos matemáticos propostos baseiam-se em um processo estocástico bivariado, composto pelo preço do ativo S_t e pela variância v_t .

Os modelos de diferenças finitas estudados foram todos do tipo explícito para que se pudesse trabalhar mais facilmente com a não linearidade dos modelos, causada pelos termos de custos de transação. Alguns outros métodos do tipo implícito ou “predictor-corrector” poderiam ter sido estudados, mas não haveria tempo hábil para implementá-los e colher resultados para as conclusões deste trabalho e portanto foram deixados para estudo futuro, não fazendo parte do escopo deste trabalho.

Paralelamente aos modelos de diferenças finitas, alguns modelos baseados em simulação de Monte Carlo foram implementados para verificar a qualidade dos modelos estudados. Para comparações dos resultados dos modelos com volatilidade determinística, foi usada uma simulação baseada no esquema de Euler para o modelo de Black-Scholes. Para as comparações dos resultados dos modelos com volatilidade estocástica, foi usada uma simulação baseada no e esquema de Milstein para o modelo de Heston [2][9].

3.2 Limitações dos modelos propostos

Os modelos de diferenças finitas propostos tendem a ser mais rápidos na obtenção de uma solução do que modelos baseados em simulações de Monte Carlo, mas também podem apresentar alguma dificuldade na hora de gerar as condições iniciais e de contorno que sejam adequadas ao problema a ser estudado. Existem alguns casos específicos em que uma solução analítica pode ser obtida, mas via de regra não podemos contar com essa hipótese na solução de uma opção mais complexa. Normalmente, a precisão da solução a ser obtida com modelos de diferenças finitas está na escolha de uma malha de discretização adequada, que em alguns casos pode ser refinada a ponto da solução necessitar de uma máquina com muita memória, ou tornar-se muito demorada.

No caso dos modelos explícitos também existe a questão da discretização do tempo de integração para que não haja problema na convergência numérica, o que pode levar a intervalos de integração muito pequenos, aumentando ainda mais o tempo da solução. Um ponto positivo dos modelos explícitos é o fato de requererem menor quantidade de memória computacional e não ser necessária a solução de sistemas lineares a cada passo de integração da equação diferencial.

Alguns casos estudados levam em conta uma opção composta por dois ativos. Quando o número de ativos cresce a solução por diferenças finitas passa a ser mais complicada e demorada, sem mencionar a dificuldade em estabelecer as condições iniciais e de contorno. Nestes casos uma solução do tipo simulação de Monte Carlo parece muito mais simples de ser implementada.

$$\left(\frac{\partial V}{\partial S} - \Delta_s\right) = 0 \text{ e } \left(\frac{\partial V}{\partial \sigma} - \Delta_\sigma\right) = 0$$

Isso permite eliminar os termos de “drift” e a dinâmica do portfólio passa a ser

$$d\Pi = \left(\frac{\partial V}{\partial t} + \frac{1}{2} \sigma^2 S^2 \frac{\partial^2 V}{\partial S^2} + \frac{1}{2} \beta^2 \sigma^2 \frac{\partial^2 V}{\partial \sigma^2} + \rho \sigma^2 \beta S \frac{\partial^2 V}{\partial S \partial \sigma} \right) \delta t - \kappa_s S |v_s| - \kappa_\sigma \sigma |v_\sigma| \quad (3.4)$$

Vamos agora estimar os custos associados à transação do ativo.

Se a quantidade do ativo comprados no instante t for

$$\Delta_t = \frac{\partial V}{\partial S}(S, \sigma, t) \quad (3.5)$$

após um período δt e refazendo o hedge, a quantidade de ativos comprados será

$$\Delta_{t+\delta t} = \frac{\partial V}{\partial S}(S + \delta S, \sigma + \delta \sigma, t + \delta t)$$

Como o intervalo de tempo δt é pequeno, as variações no valor do ativo e do índice de volatilidade também são pequenas e portanto podemos aproximar o valor por uma expansão de Taylor.

$$\Delta_{t+\delta t} = \frac{\partial V}{\partial S}(S, \sigma, t) + \delta t \frac{\partial^2 V}{\partial t \partial S}(S, \sigma, t) + \delta S \frac{\partial^2 V}{\partial S^2}(S, \sigma, t) + \delta \sigma \frac{\partial^2 V}{\partial \sigma \partial S}(S, \sigma, t) + \dots$$

sendo que

$$\delta S = \sigma S \delta X_1 + O(\delta t) \text{ e } \delta \sigma = \beta \sigma \delta X_2 + O(\delta t)$$

e portanto

$$\Delta_{t+\delta t} \approx \frac{\partial V}{\partial S}(S, \sigma, t) + \sigma S \delta X_1 \frac{\partial^2 V}{\partial S^2}(S, \sigma, t) + \beta \sigma \delta X_2 \frac{\partial^2 V}{\partial \sigma \partial S}(S, \sigma, t) \quad (3.6)$$

Subtraindo 3.5 de 3.6 encontramos a quantidade de ativos negociados durante o intervalo de tempo δt .

$$\Delta_{t+\delta t} - \Delta_t = v_s = \sigma S \delta X_1 \frac{\partial^2 V}{\partial S^2}(S, \sigma, t) + \beta \sigma \delta X_2 \frac{\partial^2 V}{\partial \sigma \partial S}(S, \sigma, t) \quad (3.7)$$

Não sabemos de antemão a quantidade de ativos que serão negociadas, mas podemos calcular o valor esperado desta variável e portanto o custo esperado de transação.

Como X_1 e X_2 são movimentos Brownianos correlacionados, podemos considerar Z_1 e Z_2 duas variáveis normais independentes com média 0 e variância 1 e portanto as distribuições X_1 e X_2 serão:

$$\begin{aligned} \delta X_1 &= Z_1 \sqrt{\delta t} \\ \delta X_2 &= \rho Z_1 \sqrt{\delta t} + \sqrt{1 - \rho^2} Z_2 \sqrt{\delta t} \end{aligned}$$

após um período δt e refazendo o hedge, a quantidade de índice de volatilidade comprados será:

$$\Delta_{t+\delta t} = \frac{\partial V}{\partial \sigma}(S + \delta S, \sigma + \delta \sigma, t + \delta t)$$

Como o intervalo de tempo δt é pequeno, as variações no valor do ativo e do índice de volatilidade também são pequenas e portanto podemos aproximar o valor por uma expansão de Taylor.

$$\Delta_{t+\delta t} \approx \frac{\partial V}{\partial \sigma}(S, \sigma, t) + \delta t \frac{\partial^2 V}{\partial t \partial \sigma}(S, \sigma, t) + \delta S \frac{\partial^2 V}{\partial \sigma \partial S}(S, \sigma, t) + \delta \sigma \frac{\partial^2 V}{\partial \sigma^2}(S, \sigma, t) + \dots$$

sendo que

$$\delta S = \sigma S \delta X_1 + O(\delta t) \text{ e } \delta \sigma = \beta \sigma \delta X_2 + O(\delta t)$$

e portanto

$$\Delta_{t+\delta t} \approx \frac{\partial V}{\partial \sigma}(S, \sigma, t) + \sigma S \delta X_1 \frac{\partial^2 V}{\partial \sigma \partial S}(S, \sigma, t) + \beta \sigma \delta X_2 \frac{\partial^2 V}{\partial \sigma^2}(S, \sigma, t) \quad (3.12)$$

Subtraindo 3.11 de 3.12 encontramos a quantidade de índice de volatilidade negociados durante o intervalo de tempo δt .

$$\Delta_{t+\delta t} - \Delta_t = v_\sigma = \sigma S \delta X_1 \frac{\partial^2 V}{\partial \sigma \partial S}(S, \sigma, t) + \beta \sigma \delta X_2 \frac{\partial^2 V}{\partial \sigma^2}(S, \sigma, t) \quad (3.13)$$

Não sabemos de antemão a quantidade de índice de volatilidade que serão negociadas, mas podemos calcular o valor esperado desta variável e portanto o custo esperado de transação.

Como X_1 e X_2 são movimentos Brownianos correlacionados, podemos considerar Z_1 e Z_2 duas variáveis normais independentes com média 0 e variância 1 e portanto as distribuições X_1 e X_2 serão:

$$\begin{aligned} \delta X_1 &= Z_1 \sqrt{\delta t} \\ \delta X_2 &= \rho Z_1 \sqrt{\delta t} + \sqrt{1 - \rho^2} Z_2 \sqrt{\delta t} \end{aligned}$$

Substituindo estas expressões na quantidade de índice de volatilidade negociados v_σ teremos:

$$v_\sigma = \sigma S Z_1 \sqrt{\delta t} \frac{\partial^2 V}{\partial \sigma \partial S}(S, \sigma, t) + \beta \sigma \rho Z_1 \sqrt{\delta t} \frac{\partial^2 V}{\partial \sigma^2}(S, \sigma, t) + \beta \sigma \sqrt{1 - \rho^2} Z_2 \sqrt{\delta t} \frac{\partial^2 V}{\partial \sigma^2}(S, \sigma, t)$$

Fazendo as substituições

$$\begin{aligned} \alpha_\sigma &= \sigma S \sqrt{\delta t} \frac{\partial^2 V}{\partial \sigma \partial S}(S, \sigma, t) + \beta \sigma \rho \sqrt{\delta t} \frac{\partial^2 V}{\partial \sigma^2}(S, \sigma, t) \\ \beta_\sigma &= \beta \sigma \sqrt{1 - \rho^2} \sqrt{\delta t} \frac{\partial^2 V}{\partial \sigma^2}(S, \sigma, t) \end{aligned} \quad (3.14)$$

3.4 Obtenção do modelo com volatilidade determinística e custos de transação para dois ativos base

Vamos modelar uma opção com dois ativos usando um movimento Browniano Geométrico com volatilidade constante, representado pelas equações a seguir:

$$\begin{aligned} dS_{1t} &= \mu(S_{1t})dt + \sigma_1 S_{1t} dX_1(t) \\ dS_{2t} &= \mu(S_{2t})dt + \sigma_2 S_{2t} dX_2(t) \end{aligned} \quad (3.18)$$

onde os dois movimentos Brownianos Geométricos X_1 e X_2 são correlacionados entre si com o coeficiente ρ .

$$E[dX_1(t)dX_2(t)] = \rho\delta t \quad (3.19)$$

3.4.1 Dedução da equação diferencial parcial quando a opção é composta por dois ativos

Considere um portfólio Π que contém a opção, com valor $V(S_1, S_2, t)$ e quantidades Δ_{S_1} e Δ_{S_2} de S_1 e S_2 respectivamente. Ou seja:

$$\Pi = V - \Delta_{S_1}S_1 - \Delta_{S_2}S_2 \quad (3.20)$$

Aplicando o Lema de Itô para obter a dinâmica do valor da opção obtemos a variação do valor do portfólio Π como:

$$\begin{aligned} d\Pi &= \left(\frac{\partial V}{\partial t} + \frac{1}{2}\sigma_1^2 S_1^2 \frac{\partial^2 V}{\partial S_1^2} + \frac{1}{2}\sigma_2^2 S_2^2 \frac{\partial^2 V}{\partial S_2^2} + \rho\sigma_1\sigma_2 S_1 S_2 \frac{\partial^2 V}{\partial S_1 \partial S_2} \right) \delta t \\ &\quad + \left(\frac{\partial V}{\partial S_1} - \Delta_{S_1} \right) dS_1 + \left(\frac{\partial V}{\partial S_2} - \Delta_{S_2} \right) dS_2 - \kappa_{S_1} S_1 |v_{S_1}| - \kappa_{S_2} S_2 |v_{S_2}| \end{aligned}$$

onde $\kappa_{S_1} S_1 |v_{S_1}|$ e $\kappa_{S_2} S_2 |v_{S_2}|$ representam os custos de transação associados à uma quantidade v_{S_1} do ativo S_1 e v_{S_2} do ativo S_2 , durante o intervalo de tempo δt . Os custos de transação entre S_1 e S_2 são diferentes. Seleccionamos Δ_{S_1} e Δ_{S_2} como a solução das equações

$$\left(\frac{\partial V}{\partial S_1} - \Delta_{S_1} \right) = 0 \text{ e } \left(\frac{\partial V}{\partial S_2} - \Delta_{S_2} \right) = 0$$

Isso permite eliminar os termos de “drift” e a dinâmica do portfólio passa a ser

$$d\Pi = \left(\frac{\partial V}{\partial t} + \frac{1}{2}\sigma_1^2 S_1^2 \frac{\partial^2 V}{\partial S_1^2} + \frac{1}{2}\sigma_2^2 S_2^2 \frac{\partial^2 V}{\partial S_2^2} + \rho\sigma_1\sigma_2 S_1 S_2 \frac{\partial^2 V}{\partial S_1 \partial S_2} \right) \delta t - \kappa_{S_1} S_1 |v_{S_1}| - \kappa_{S_2} S_2 |v_{S_2}| \quad (3.21)$$

Vamos agora estimar os custos associados à transação do ativo S_1 .

Se a quantidade de ativos “comprados” no instante t for

$$\Delta_t = \frac{\partial V}{\partial S_1}(S_1, S_2, t) \quad (3.22)$$

e observando que

$$\begin{aligned}\alpha_{S_1}^2 &= \sigma_1^2 S_1^2 \delta t \left(\frac{\partial^2 V}{\partial S_1^2} \right)^2 + 2\rho\sigma_1\sigma_2 S_1 S_2 \delta t \left(\frac{\partial^2 V}{\partial S_1^2} \right) \left(\frac{\partial^2 V}{\partial S_1 \partial S_2} \right) + \rho^2 \sigma_2^2 S_2^2 \delta t \left(\frac{\partial^2 V}{\partial S_1 \partial S_2} \right)^2 \\ \beta_{S_1}^2 &= \sigma_2^2 S_2^2 \delta t \left(\frac{\partial^2 V}{\partial S_1 \partial S_2} \right)^2 - \rho^2 \sigma_2^2 S_2^2 \delta t \left(\frac{\partial^2 V}{\partial S_1 \partial S_2} \right)^2\end{aligned}$$

chegamos à seguinte expressão para descrever a variação da quantidade de ativos num intervalo de tempo δt .

$$v_{S_1} = \alpha_{S_1} Z_1 + \beta_{S_1} Z_2 \quad (3.26)$$

Lembrando que o valor esperado do módulo de uma variável aleatória normal, com média 0 e variância 1 é:

$$E[|Z|] = \sqrt{\frac{2}{\pi}}$$

é fácil observar que o valor esperado dos custos de transação associados à compra e venda do ativo S_1 será:

$$E[\kappa_{S_1} S_1 | v_{S_1}|] = \sqrt{\frac{2}{\pi}} \kappa_{S_1} S_1 \sqrt{\alpha_{S_1}^2 + \beta_{S_1}^2}$$

e portanto a expressão completa para os custos de transação de S_1 será:

$$E[\kappa_{S_1} S_1 | v_{S_1}|] = \sqrt{\frac{2}{\pi}} \kappa_{S_1} S_1 \sqrt{\sigma_1^2 S_1^2 \delta t \left(\frac{\partial^2 V}{\partial S_1^2} \right)^2 + 2\rho\sigma_1\sigma_2 S_1 S_2 \delta t \left(\frac{\partial^2 V}{\partial S_1^2} \right) \left(\frac{\partial^2 V}{\partial S_1 \partial S_2} \right) + \sigma_2^2 S_2^2 \delta t \left(\frac{\partial^2 V}{\partial S_1 \partial S_2} \right)^2} \quad (3.27)$$

Analisamos agora os custos de transação associados à compra e venda do ativo S_2 e procedendo exatamente da mesma maneira teremos:

Se a quantidade do ativo índice de volatilidade “comprados” no instante t for

$$\Delta_t = \frac{\partial V}{\partial S_2}(S_1, S_2, t) \quad (3.28)$$

após um período δt e refazendo o hedge, a quantidade de ativos comprados será

$$\Delta_{t+\delta t} = \frac{\partial V}{\partial S_2}(S_1 + \delta S_1, S_2 + \delta S_2, t + \delta t)$$

Como o intervalo de tempo δt é pequeno, as variações no valor dos ativos também são pequenas e portanto podemos aproximar o valor por uma expansão de Taylor.

$$\Delta_{t+\delta t} = \frac{\partial V}{\partial S_2}(S_1, S_2, t) + \delta t \frac{\partial^2 V}{\partial t \partial S_2}(S_1, S_2, t) + \delta S_1 \frac{\partial^2 V}{\partial S_1 \partial S_2}(S_1, S_2, t) + \delta S_2 \frac{\partial^2 V}{\partial S_2^2}(S_1, S_2, t) + \dots$$

Da mesma forma como foi para o ativo S_1 é fácil observar que o valor esperado dos custos de transação associados à compra e venda do ativo S_2 será:

$$E[\kappa_{S_2} S_2 | v_{S_2}] = \sqrt{\frac{2}{\pi}} \kappa_{S_2} S_2 \sqrt{\alpha_{S_2}^2 + \beta_{S_2}^2}$$

e portanto a expressão completa para os custos de transação de S_2 será:

$$E[\kappa_{S_2} S_2 | v_{S_2}] = \sqrt{\frac{2}{\pi}} \kappa_{S_2} S_2 \sqrt{\sigma_1^2 S_1^2 \delta t \left(\frac{\partial^2 V}{\partial S_1 \partial S_2} \right)^2 + 2\rho \sigma_1 \sigma_2 S_1 S_2 \delta t \left(\frac{\partial^2 V}{\partial S_1 \partial S_2} \right) \left(\frac{\partial^2 V}{\partial S_2^2} \right) + \sigma_2^2 S_2^2 \delta t \left(\frac{\partial^2 V}{\partial S_2^2} \right)^2} \quad (3.33)$$

Para que não haja arbitragem, deve haver uma taxa de juros livre de risco que satisfaça as 3 equações a seguir, simultaneamente:

$$\begin{aligned} \Pi &= V - \Delta_{S_1} S_1 - \Delta_{S_2} S_2 \\ d\Pi &= r\Pi \delta t \end{aligned} \quad (3.34)$$

$$d\Pi = \left(\frac{\partial V}{\partial t} + \frac{1}{2} \sigma_1^2 S_1^2 \frac{\partial^2 V}{\partial S_1^2} + \frac{1}{2} \sigma_2^2 S_2^2 \frac{\partial^2 V}{\partial S_2^2} + \rho \sigma_1 \sigma_2 S_1 S_2 \frac{\partial^2 V}{\partial S_1 \partial S_2} \right) \delta t - \kappa_{S_1} S_1 |v_{S_1}| - \kappa_{S_2} S_2 |v_{S_2}|$$

Portanto a expressão final da dinâmica da opção com custos de transação para dois ativos fica:

$$\begin{aligned} & \frac{\partial V}{\partial t} + \frac{1}{2} \sigma_1^2 S_1^2 \frac{\partial^2 V}{\partial S_1^2} + \frac{1}{2} \sigma_2^2 S_2^2 \frac{\partial^2 V}{\partial S_2^2} + \rho \sigma_1 \sigma_2 S_1 S_2 \frac{\partial^2 V}{\partial S_1 \partial S_2} + (r - D_1) S_1 \frac{\partial V}{\partial S_1} + (r - D_2) S_2 \frac{\partial V}{\partial S_2} - rV \\ & - \sqrt{\frac{2}{\pi \delta t}} \kappa_{S_1} S_1 \sqrt{\sigma_1^2 S_1^2 \left(\frac{\partial^2 V}{\partial S_1^2} \right)^2 + 2\rho \sigma_1 \sigma_2 S_1 S_2 \left(\frac{\partial^2 V}{\partial S_1^2} \right) \left(\frac{\partial^2 V}{\partial S_1 \partial S_2} \right) + \sigma_2^2 S_2^2 \left(\frac{\partial^2 V}{\partial S_1 \partial S_2} \right)^2} \\ & - \sqrt{\frac{2}{\pi \delta t}} \kappa_{S_2} S_2 \sqrt{\sigma_1^2 S_1^2 \left(\frac{\partial^2 V}{\partial S_1 \partial S_2} \right)^2 + 2\rho \sigma_1 \sigma_2 S_1 S_2 \left(\frac{\partial^2 V}{\partial S_1 \partial S_2} \right) \left(\frac{\partial^2 V}{\partial S_2^2} \right) + \sigma_2^2 S_2^2 \left(\frac{\partial^2 V}{\partial S_2^2} \right)^2} = 0 \end{aligned} \quad (3.35)$$

Os últimos 2 termos desta Equação Diferencial Parcial não Linear são resultantes dos custos de transação. A solução numérica desta equação vai permitir conhecer o preço da opção em função dos valores dos ativos S_1 e S_2 .

$$\frac{\partial^2 f}{\partial \sigma^2} = \frac{f_{i,j+1,k} - 2f_{i,j,k} + f_{i,j-1,k}}{(\Delta \sigma)^2} \Rightarrow \text{Gamma}\sigma$$

$$\frac{\partial^2 f}{\partial S \partial \sigma} = \frac{f_{i+1,j+1,k} - f_{i+1,j-1,k} - f_{i-1,j+1,k} + f_{i-1,j-1,k}}{4\Delta S \Delta \sigma} \Rightarrow \text{CrossVanna}\sigma$$

$$\frac{\partial f}{\partial t} = \frac{f_{i,j,k-1} - f_{i,j,k}}{\Delta t} \Rightarrow \text{Theta}$$

b) Discretização dos termos relativos aos custos de transação. Estes termos podem ser discretizados facilmente, usando-se a discretização das Gregas feitas anteriormente e usando os valores discretizados do ativo S_i e da volatilidade σ_j .

Temos portanto:

$$\text{Cost}_s = k_s S_i \sqrt{\frac{2}{\pi \delta t}} \sqrt{\sigma_j^2 S_i^2 \left(\frac{\partial^2 f}{\partial S^2} \right)^2 + 2\rho \beta \sigma_j^2 S_i \frac{\partial^2 f}{\partial S^2} \frac{\partial^2 f}{\partial S \partial \sigma} + \beta^2 \sigma_j^2 \left(\frac{\partial^2 f}{\partial S \partial \sigma} \right)^2}$$

$$\text{Cost}_\sigma = k_\sigma \sigma_j \sqrt{\frac{2}{\pi \delta t}} \sqrt{\sigma_j^2 S_i^2 \left(\frac{\partial^2 f}{\partial S \partial \sigma} \right)^2 + 2\rho \beta \sigma_j^2 S_i \frac{\partial^2 f}{\partial \sigma^2} \frac{\partial^2 f}{\partial S \partial \sigma} + \beta^2 \sigma_j^2 \left(\frac{\partial^2 f}{\partial \sigma^2} \right)^2}$$

$$\text{onde } S_i = \frac{i}{L+1} S_{\max} \text{ e } \sigma_j = \sigma_{\min} + \frac{j}{M+1} (\sigma_{\max} - \sigma_{\min})$$

c) Discretização para o cálculo das Gregas quando o ponto para cálculo está sobre uma das fronteiras do domínio, ou seja, deseja-se calcular uma derivada sobre uma das fronteiras do domínio.

O nosso domínio possui duas fronteiras para o valor do ativo base e duas fronteiras para o valor da volatilidade. No caso do ativo base esse problema não vai existir, pois os valores da opção quando $S=0$ ou $S=S_{\max}$ serão calculados pelas condições de contorno para o valor do ativo, logo teremos para o ativo sempre as expressões:

$$\frac{\partial f}{\partial S} = \frac{f_{i+1,j,k} - f_{i-1,j,k}}{2\Delta S} \Rightarrow \text{Delta}S$$

$$\frac{\partial^2 f}{\partial S^2} = \frac{f_{i+1,j,k} - 2f_{i,j,k} + f_{i-1,j,k}}{(\Delta S)^2} \Rightarrow \text{Gamma}S$$

$$\frac{\partial f}{\partial t} = \frac{f_{i,j,k-1} - f_{i,j,k}}{\Delta t} \Rightarrow \text{Theta}$$

No caso da volatilidade devemos modificar a maneira de calcular as derivadas para que um erro de mesma ordem de grandeza seja obtido quando o ponto de cálculo estiver sobre a fronteira do domínio e portanto vamos adotar as expressões a seguir para o caso da volatilidade:

$$\frac{\partial f}{\partial \sigma} = \frac{-3f_{i,1,k} + 4f_{i,2,k} - f_{i,3,k}}{2\Delta \sigma} \Rightarrow \frac{3f_{i,M,k} - 4f_{i,M-1,k} + f_{i,M-2,k}}{2\Delta \sigma} \Rightarrow \text{Delta}\sigma$$

3.6 Discretização da EDP do modelo com volatilidade determinística para implementação do método de diferenças finitas

Consideremos agora a EDP (3.35) para um modelo de opção de 2 ativos com volatilidades determinísticas e custos de transação.

$$\begin{aligned}
 & \frac{\partial f}{\partial t} + \frac{1}{2} \sigma_1^2 S_1^2 \frac{\partial^2 f}{\partial S_1^2} + \frac{1}{2} \sigma_2^2 S_2^2 \frac{\partial^2 f}{\partial S_2^2} + \rho \sigma_1 \sigma_2 S_1 S_2 \frac{\partial^2 f}{\partial S_1 \partial S_2} + (r - D_1) S_1 \frac{\partial f}{\partial S_1} + (r - D_2) S_2 \frac{\partial f}{\partial S_2} - rf + \\
 & -k_{S_1} S_1 \sqrt{\frac{2}{\pi \delta t}} \sqrt{\sigma_1^2 S_1^2 \left(\frac{\partial^2 f}{\partial S_1^2} \right)^2 + 2 \rho \sigma_1 \sigma_2 S_1 S_2 \frac{\partial^2 f}{\partial S_1^2} \frac{\partial^2 f}{\partial S_1 \partial S_2} + \sigma_2^2 S_2^2 \left(\frac{\partial^2 f}{\partial S_1 \partial S_2} \right)^2} + \\
 & -k_{S_2} S_2 \sqrt{\frac{2}{\pi \delta t}} \sqrt{\sigma_1^2 S_1^2 \left(\frac{\partial^2 f}{\partial S_1 \partial S_2} \right)^2 + 2 \rho \sigma_1 \sigma_2 S_1 S_2 \frac{\partial^2 f}{\partial S_2^2} \frac{\partial^2 f}{\partial S_1 \partial S_2} + \sigma_2^2 S_2^2 \left(\frac{\partial^2 f}{\partial S_2^2} \right)^2} = 0
 \end{aligned} \tag{3.39}$$

Analogamente ao modelo anterior, considerando f o valor da opção, a discretização será feita num domínio tridimensional, onde o valor do ativo 1 será discretizado na direção i , o valor do ativo 2 será discretizado na direção j e o tempo de integração será discretizado na direção k . Temos portanto os ativos S_1 e S_2 variando de 0 até um valor máximo $S_{1\max}$, $S_{2\max}$ e o tempo variando de trás para frente de T final até t inicial (ou zero).

Este esquema pode ser resumido pelas seguintes expressões:

$$\begin{aligned}
 & f(S_1, S_2, t) \rightarrow f(i\Delta S_1, j\Delta S_2, k\Delta t) \\
 & S_1 = 0, 1\Delta S_1, 2\Delta S_1, \dots, i\Delta S_1, \dots, L\Delta S_1 = S_{1\max} \\
 & S_2 = 0, 1\Delta S_2, 2\Delta S_2, \dots, j\Delta S_2, \dots, M\Delta S_2 = S_{2\max} \\
 & t = 0, 1\Delta t, 2\Delta t, \dots, k\Delta t, \dots, N\Delta t = T
 \end{aligned}$$

Métodos de discretização em função da posição do ponto em relação ao domínio

a) Discretização para o cálculo das Gregas quando o ponto de cálculo está dentro das fronteiras do domínio, ou seja, em nenhum momento deseja-se calcular uma derivada sobre a fronteira do domínio.

$$\begin{aligned}
 \frac{\partial f}{\partial S_1} &= \frac{f_{i+1,j,k} - f_{i-1,j,k}}{2\Delta S_1} \Rightarrow \text{Delta}_{S_1} \\
 \frac{\partial f}{\partial S_2} &= \frac{f_{i,j+1,k} - f_{i,j-1,k}}{2\Delta S_2} \Rightarrow \text{Delta}_{S_2} \\
 \frac{\partial^2 f}{\partial S_1^2} &= \frac{f_{i+1,j,k} - 2f_{i,j,k} + f_{i-1,j,k}}{(\Delta S_1)^2} \Rightarrow \text{Gamma}_{S_1}
 \end{aligned}$$

$$\frac{\partial^2 f}{\partial S_1 \partial S_2} = \frac{(-3f_{i+1,1,k} + 4f_{i+1,2,k} - f_{i+1,3,k}) - (-3f_{i-1,1,k} + 4f_{i-1,2,k} - f_{i-1,3,k})}{4\Delta S_1 \Delta S_2} \Rightarrow CrossGammaS_1S_2$$

$$\frac{\partial^2 f}{\partial S_1 \partial S_2} = \frac{(3f_{i+1,M,k} - 4f_{i+1,M-1,k} + f_{i+1,M-2,k}) - (3f_{i-1,M,k} - 4f_{i-1,M-1,k} + f_{i-1,M-2,k})}{4\Delta S_1 \Delta S_2} \Rightarrow CrossGammaS_1S_2$$

$$\frac{\partial^2 f}{\partial S_1 \partial S_2} = \frac{(-3f_{1,j+1,k} + 4f_{2,j+1,k} - f_{3,j+1,k}) - (-3f_{1,j-1,k} + 4f_{2,j-1,k} - f_{3,j-1,k})}{4\Delta S_1 \Delta S_2} \Rightarrow CrossGammaS_1S_2$$

$$\frac{\partial^2 f}{\partial S_1 \partial S_2} = \frac{(3f_{L,j+1,k} - 4f_{L-1,j+1,k} + f_{L-2,j+1,k}) - (3f_{L,j-1,k} - 4f_{L-1,j-1,k} + f_{L-2,j-1,k})}{4\Delta S_1 \Delta S_2} \Rightarrow CrossGammaS_1S_2$$

Cada caso de ΔS_i ou ΓS_i possui duas versões, que refletem o fato de estar na fronteira inicial ou final do domínio, desta forma o erro de discretização (ou do cálculo das derivadas) será de mesma ordem de grandeza dentro de todos os pontos do domínio. Para o casos do $CrossGammaS_iS_j$ existem 4 possibilidades, uma para cada aresta do retângulo do domínio.

A discretização dos termos relativos aos custos de transação quando o ponto de cálculo está sobre uma das fronteiras permanece da mesma forma, apenas que a maneira de calcular as Gregas deve ser substituída pelas expressões anteriores.

A equação final de diferenças finitas, para o caso de pontos calculados dentro das fronteiras é:

$$\begin{aligned} f_{i,j,k-1} = & f_{i,j,k} - (r - D_1)S_i \left(\frac{f_{i+1,j,k} - f_{i-1,j,k}}{2\Delta S_1} \right) \Delta t - \frac{\sigma_1^2 S_i^2}{2} \left(\frac{f_{i+1,j,k} - 2f_{i,j,k} + f_{i-1,j,k}}{(\Delta S_1)^2} \right) \Delta t \\ & - (r - D_2)S_j \left(\frac{f_{i,j+1,k} - f_{i,j-1,k}}{2\Delta S_2} \right) \Delta t - \frac{\sigma_2^2 S_j^2}{2} \left(\frac{f_{i,j+1,k} - 2f_{i,j,k} + f_{i,j-1,k}}{(\Delta S_2)^2} \right) \Delta t \\ & - \rho \sigma_1 \sigma_2 S_i S_j \left(\frac{f_{i+1,j+1,k} - f_{i+1,j-1,k} - f_{i-1,j+1,k} + f_{i-1,j-1,k}}{4\Delta S_1 \Delta S_2} \right) \Delta t + rf_{i,j,k} \Delta t + Cost_{S1} \Delta t + Cost_{S2} \Delta t \end{aligned} \quad (3.40)$$

onde

$$\begin{aligned} Cost_{S1} = & k_{S1} S_i \sqrt{\frac{2}{\pi \delta t}} \sqrt{\sigma_1^2 S_i^2 \left(\frac{\partial^2 f}{\partial S_1^2} \right)^2 + 2\rho \sigma_1 \sigma_2 S_i S_j \frac{\partial^2 f}{\partial S_1^2} \frac{\partial^2 f}{\partial S_1 \partial S_2} + \sigma_2^2 S_j^2 \left(\frac{\partial^2 f}{\partial S_1 \partial S_2} \right)^2} \\ Cost_{S2} = & k_{S2} S_j \sqrt{\frac{2}{\pi \delta t}} \sqrt{\sigma_1^2 S_i^2 \left(\frac{\partial^2 f}{\partial S_1 \partial S_2} \right)^2 + 2\rho \sigma_1 \sigma_2 S_i S_j \frac{\partial^2 f}{\partial S_2^2} \frac{\partial^2 f}{\partial S_1 \partial S_2} + \sigma_2^2 S_j^2 \left(\frac{\partial^2 f}{\partial S_2^2} \right)^2} \end{aligned} \quad (3.41)$$

$$S_i = \frac{i}{L+1} S_{1\max}; S_j = \frac{j}{M+1} S_{2\max}$$

Para os casos em que os pontos calculados estão sobre uma das fronteira, a equação de diferenças finitas deve ter as Gregas substituídas, de acordo com a posição dos pontos a serem calculados. Não vamos mostrar aqui neste trabalho como a equação ficaria, mas é trivial. Estas expressões foram implementadas em um código Matlab® para as verificações e simulações realizadas neste trabalho.

3.7.1.2 Condição de contorno para opção de compra “plain vanilla”

Será usada segunda derivada nula para o valor da opção, no início e no final do domínio, ou seja:

$$\frac{\partial^2 f}{\partial S^2} = \frac{2f_{1,j,k} - 5f_{2,j,k} + 4f_{3,j,k} - f_{4,j,k}}{(\Delta S)^2} = 0 \Rightarrow S = 0$$

$$\frac{\partial^2 f}{\partial S^2} = \frac{-2f_{L,j,k} + 5f_{L-1,j,k} - 4f_{L-2,j,k} + f_{L-3,j,k}}{(\Delta S)^2} = 0 \Rightarrow S = S_{\max}$$

Portanto as condições a seguir devem ser obtidas para quaisquer valores da volatilidade

$$f_{1,j,k} = \frac{5f_{2,j,k} - 4f_{3,j,k} + f_{4,j,k}}{2}$$

$$f_{M,j,k} = \frac{5f_{L-1,j,k} - 4f_{L-2,j,k} + f_{L-3,j,k}}{2}$$

3.7.1.3 Condição inicial para opção de compra com barreira superior “up&out call”

A condição inicial para este caso é $S_t < Sb, \forall t \Rightarrow V = \max(S_t - K, 0)$ ou $S_t \geq Sb, \forall t \Rightarrow V = 0$ para qualquer valor da volatilidade.

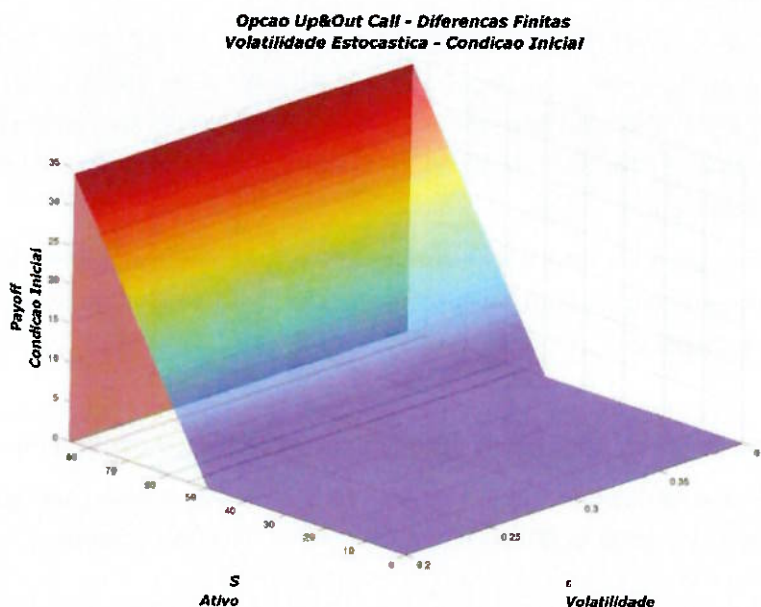


Figura 3.2- Condição Inicial para o modelo de opção de compra europeia com barreira superior e volatilidade estocástica

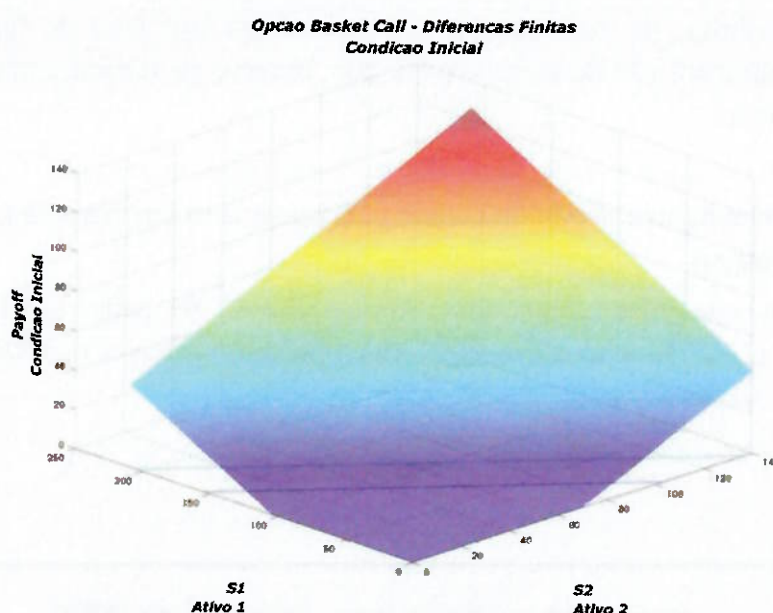


Figura 3.3- Condição inicial para o modelo de opção de compra europeia “basket call” com dois ativos
- $w=0,4$, $K=45$

3.7.2.2 Condição de contorno para opção de compra europeia “basket call” com dois ativos

Da mesma forma como foi feito para o caso de volatilidade estocástica, será empregado aqui uma condição de segunda derivada nula, mas agora esta condição deve ser imposta em todas as fronteiras do domínio, ou seja:

$$\frac{\partial^2 f}{\partial S_1^2} = \frac{2f_{1,j,k} - 5f_{2,j,k} + 4f_{3,j,k} - f_{4,j,k}}{(\Delta S_1)^2} = 0$$

$$\frac{\partial^2 f}{\partial S_1^2} = \frac{-2f_{L,j,k} + 5f_{L-1,j,k} - 4f_{L-2,j,k} + f_{L-3,j,k}}{(\Delta S_1)^2} = 0$$

$$\frac{\partial^2 f}{\partial S_2^2} = \frac{2f_{i,1,k} - 5f_{i,2,k} + 4f_{i,3,k} - f_{i,4,k}}{(\Delta S_2)^2} = 0$$

$$\frac{\partial^2 f}{\partial S_2^2} = \frac{-2f_{i,M,k} + 5f_{i,M-1,k} - 4f_{i,M-2,k} + f_{i,M-3,k}}{(\Delta S_2)^2} = 0$$

Portanto as condições a seguir devem ser obtidas nos pontos situados nas fronteiras do domínio.

$$S_1 = 0 \rightarrow f_{1,j,k} = \frac{5f_{2,j,k} - 4f_{3,j,k} + f_{4,j,k}}{2}$$

$$S_1 = S_{1\max} \rightarrow f_{L,j,k} = \frac{5f_{L-1,j,k} - 4f_{L-2,j,k} + f_{L-3,j,k}}{2}$$

$$S_2 = 0 \rightarrow f_{i,1,k} = \frac{5f_{i,2,k} - 4f_{i,3,k} + f_{i,4,k}}{2}$$

$$S_2 = S_{2\max} \rightarrow f_{i,M,k} = \frac{5f_{i,M-1,k} - 4f_{i,M-2,k} + f_{i,M-3,k}}{2}$$

Portanto as condições a seguir devem ser obtidas para os pontos situados nas fronteiras do domínio:

$$S_1 = 0 \rightarrow f_{1,j,k} = \frac{5f_{2,j,k} - 4f_{3,j,k} + f_{4,j,k}}{2}$$

$$S_2 = 0 \rightarrow f_{i,1,k} = \frac{5f_{i,2,k} - 4f_{i,3,k} + f_{i,4,k}}{2}$$

$$f = 0 \text{ se } wS_1 + (1-w)S_2 \geq Sb$$

3.8 Simulações de Monte Carlo

Simulações de Monte Carlo, no contexto de apreçamento de opções referem-se a um conjunto de técnicas para gerar valores típicos de ativos subjacentes, como os preços de ações ou de taxas de juros ao longo do tempo. Normalmente a dinâmica dos preços de ações e taxas de juros são resultantes de um processo estocástico em tempo contínuo. As simulações, entretanto, são feitas em intervalos de tempo discretos, portanto, o primeiro passo em qualquer esquema de simulação é encontrar uma maneira de discretizar um processo de tempo contínuo em um processo de tempo discreto. Apresentamos aqui dois esquemas de discretização, Euler e Milstein [3], que irão ilustrar os modelos de Black-Scholes e Heston usados na comparação dos resultados obtidos pelos modelos matemáticos de apreçamento de opções propostos neste trabalho.

3.8.1 Esquema de Euler para o Modelo de Black-Scholes

A dinâmica de Black-Scholes no preço de uma ação sob taxa de juros neutra ao risco é

$$dS_t = rS_t dt + \sigma S_t dW_t$$

Podemos fazer a discretização de Euler para o modelo de Black-Scholes da seguinte forma

$$S_{t+dt} = S_t + rS_t dt + \sigma S_t \sqrt{dt} Z$$

$$dt = t_i - t_{i-1}$$

Da mesma forma, podemos analisar o logaritmo dos preços e depois gerar um resultado exponencial. Pelo lema de Itô $\ln S_t$ segue o processo abaixo

$$d \ln S_{t+dt} = \left(r - \frac{1}{2} \sigma^2 \right) dt + \sigma S_t dW_t$$

A discretização de Euler produz a expressão

$$\ln S_{t+dt} = \ln S_t + \left(r - \frac{1}{2} \sigma^2 \right) dt + \sigma S_t \sqrt{dt} Z$$

e portanto temos

$$S_{t+dt} = S_t \exp \left[\left(r - \frac{1}{2} \sigma^2 \right) dt + \sigma S_t \sqrt{dt} Z \right]$$

3.8.4 Esquema de Milstein para modelo de Heston

Lembrando que o modelo de Heston é descrito pelo processo estocástico bivariado, composto pelo preço do ativo S_t e da variância v_t .

$$\begin{aligned} dS_t &= rS_t dt + \sqrt{v_t} S_t dW_{1,t} \\ dv_t &= \kappa(\theta - v_t) dt + \sigma \sqrt{v_t} dW_{2,t} \\ \text{onde } E[dW_{1,t} dW_{2,t}] &= \rho dt. \end{aligned}$$

Discretização da variância v_t

A equação diferencial estocástica da variância é discretizada conforme a expressão a seguir

$$v_{t+dt} = v_t + \kappa(\theta - v_t) dt + \beta \sqrt{v_t} dt Z_v + \frac{1}{4} \beta^2 dt (Z_v^2 - 1)$$

ou então

$$v_{t+dt} = \left(\sqrt{v_t} + \frac{1}{2} \beta \sqrt{v_t} \sqrt{dt} Z_v \right)^2 + \kappa(\theta - v_t) dt - \frac{1}{4} \beta^2 dt$$

O método de discretização de Milstein para o processo da variância produz menos valores negativos do que o método de Euler. Entretanto devemos aplicar o truncamento ou reflexão da variância, substituindo os valores de v_t por v_t^+ ou $|v_t|$, da mesma forma como foi feito para o modelo de Euler.

Discretização do valor do ativo S_t

O processo do valor do ativo é discretizado conforme a expressão a seguir:

$$S_{t+dt} = S_t + (r - D) S_t dt + \sqrt{v_t} S_t Z_s + \frac{1}{2} v_t S_t dt (Z_s^2 - 1)$$

Da mesma forma como feito para o modelo de Black-Scholes, para gerar Z_v e Z_s com correlação ρ , devemos gerar duas variáveis independentes Z_1 e Z_2 , e fazer $Z_v = Z_1$ e $Z_s = \rho Z_1 + \sqrt{1 - \rho^2} Z_2$.

Existem dois problemas que surgem nas simulações de processos bivariados. O primeiro é a baixa velocidade de convergência. O segundo e mais sério deles é que como a variância segue um processo de Cox-Ingersoll-Ross, muitos esquemas de simulação, incluindo os esquemas de Euler e Milstein gerarão valores negativos para a volatilidade, mesmo que a condição de Feller $2\kappa\theta > \sigma^2$ seja observada. Isto se deve ao fato da condição de Feller ser válida para processos contínuos, mas as simulações são feitas em tempo discreto como forma de simular um processo contínuo [2].

O modo mais simples de lidar com variâncias negativas é eliminá-las no momento em que elas aparecem, e existem duas formas simples de fazer isso:

- Truncamento - um valor negativo da variância v_t é automaticamente substituído por zero. Portanto v_t é substituído por v_t^+ em todos os locais da discretização.

3.9.2 Simulação de Monte Carlo para opções europeias de compra com barreira “up&out call” com 2 ativos e volatilidades determinísticas

$$j = 1, \dots, M$$

$$i = 0, \dots, N - 1$$

$$Z1_i \rightarrow N(0,1)$$

$$Z2_i \rightarrow N(0,1)$$

$$S1_{i+1} = S1_i e^{\left(r - D1 - \frac{\sigma_1^2}{2}\right)\Delta t + \sigma_1 \sqrt{\Delta t} Z1_i}$$

$$S2_{i+1} = S2_i e^{\left(r - D2 - \frac{\sigma_2^2}{2}\right)\Delta t + \sigma_2 \sqrt{\Delta t} Z2_i}$$

$$S_i = wS1_i + (1 - w)S2_i$$

$$P_{i+1} = e^{-2 \frac{(Sb - S_{i+1})(Sb - S_i)}{\sigma^2 S_i^2 \Delta t}}$$

$$u_{i+1} \rightarrow U(0,1)$$

$$se\{S_i > Sb \parallel P_i > u_i\} \rightarrow S_j = 0$$

$$V_j = e^{-rT} \max(S_T - K, 0)$$

$$V = \frac{1}{M} \sum_{j=1}^M V_j$$

3.9.4 Simulação de Monte Carlo para opção europeia de compra com barreira “up&out call” com volatilidade estocástica

$$j = 1, \dots, M$$

$$i = 0, \dots, N-1$$

$$Zv_i \rightarrow N(0,1)$$

$$ZS_i \rightarrow \rho Zv_i + \sqrt{1-\rho^2} N(0,1)$$

$$v_{i+1} = v_i + \kappa(\theta - v_i)dt + \beta\sqrt{v_i}dt Zv_i + \frac{\beta^2}{4}dt(Zv_i^2 - 1)$$

$$se\{v_i < 0\} \rightarrow v_i = 0$$

$$S_{i+1} = S_i \left[1 + (r - D)dt + \sqrt{v_i}dt ZS_i + \frac{1}{2}v_i dt(ZS_i^2 - 1) \right]$$

$$P_{i+1} = e^{-2 \frac{(Sb - S_{i+1})(Sb - S_i)}{\sigma^2 S_i^2 \Delta t}}$$

$$u_{i+1} \rightarrow U(0,1)$$

$$se\{S_i > Sb \parallel P_i > u_i\} \rightarrow S_j = 0$$

$$V_j = e^{-rT} \max(S_T - K, 0)$$

$$V = \frac{1}{M} \sum_{j=1}^M V_j$$

3.9.5 Brownian Bridge

Para acelerar o processo de convergência nas simulações das opções com barreiras, foi implementado o conceito de *Brownian Bridge*. Quando dois valores de uma trajetória simulada por Monte Carlo estiverem bem próximos do valor da barreira, existirá sempre uma probabilidade de que a trajetória entre estes dois pontos tenha atingido a barreira. Não fazer uso desta probabilidade nos obriga a aumentar drasticamente o número de intervalos em cada trajetória, aumentando muito o tempo de simulação [9].

Isto está representado pela seguinte parte dos algoritmos apresentados:

$$P_{i+1} = e^{-2 \frac{(Sb - S_{i+1})(Sb - S_i)}{\sigma^2 S_i^2 \Delta t}}$$

$$u_{i+1} \rightarrow U(0,1)$$

$$se\{S_i > Sb \parallel P_i > u_i\} \rightarrow S_j = 0$$

Um exemplo pode ser visto a seguir numa simulação de uma *Brownian Bridge* onde o objetivo foi comparar o valor teórico de “quebra da barreira” com o esperado pela simulação.

Capítulo 4

Aplicação da teoria e verificação dos modelos pela simulação de alguns casos específicos

4.1 Introdução

O objetivo deste capítulo é, a partir da teoria desenvolvida, simular alguns casos de interesse, para que os modelos possam ser avaliados como ferramenta de apreçamento de derivativos. Como foi visto no capítulo 3, alguns tipos de opções podem ser apreçadas facilmente bastando para tal modificar as condições iniciais e de contorno do problema e usar exatamente o mesmo algoritmo para solução numérica da equação diferencial parcial.

Todas as simulações apresentadas foram realizadas pelo método de diferenças finitas desenvolvidos a partir dos modelos desenvolvidos e os resultados foram comparados com simulações de Monte Carlo da mesma opção, com trabalhos encontrados na literatura de derivativos ou com a simulação de outros modelos. Exceção foi feita nos casos de volatilidade estocástica, onde as simulações de Monte Carlo foram feitas usando o modelo de Heston que não pressupõe a volatilidade como um ativo negociável. Mesmo assim foram feitas comparações entre os modelos para investigar o comportamento da solução.

Os modelos matemáticos desenvolvidos foram implementados no Matlab® em vários *scripts* diferentes com múltiplas saídas gráficas e estão disponíveis no anexo C. As saídas gráficas nos permitem avaliar o comportamento das Gregas em todos os casos, o que é bastante útil na execução do “*hedge*” ou avaliação de risco dos portfólios.

No início de cada simulação foi colocada uma tabela mostrando claramente todos os valores relevantes usados naquela simulação em questão, assim como dados auxiliares usados nas simulações adicionais para comparação dos resultados.

Abaixo de cada figura existe um texto descritivo dos resultados mostrados e um resumo dos valores usados como dados de entrada para a simulação que os gerou.

O resultado de cada simulação é mostrado pelas figuras que apresentam os seguintes gráficos: condição inicial, valor da opção, deltas, derivada cruzada e gammas para os casos sem e com custos de transação. Depois destas figuras aparece, dependendo do caso, uma “superfície de resposta” com os valores com e sem custos de transação em função do valor dos ativos. Nos casos com volatilidade estocástica também é mostrada uma “superfície de resposta” com e sem custos de transação em função do valor do ativo e da volatilidade. Também são mostradas duas outras figuras dos valores com e sem custos de transação em função da volatilidade e o valor de exercício.

Para as simulações de Monte Carlo, o resultado também é mostrado por uma “superfície de resposta” e a qualidade da simulação por ser avaliada por uma outra figura que mostra a distribuição dos valores do ativo, distribuição dos valores da opção e um pseudo histograma sobre uma curva normal dos valores obtidos em cada simulação. É fácil entender observando cada uma das figuras.

4.2 Simulação de uma opção “basket call” com volatilidades determinísticas composta por dois ativos

O modelo desenvolvido para o apreamento de uma opção com dois ativos é na realidade uma adaptação do modelo de apreamento de uma opção com volatilidade estocástica, onde a variância foi substituída pelo segundo ativo e a volatilidade da variância pela volatilidade do segundo ativo. Todos os termos da equação são determinísticos, como se pode concluir pelas equações 3.17 e 3.35.

O modelo desenvolvido supõe que a opção é uma média ponderada dos ativos, onde o fator de ponderação é a quantidade total de cada ativo na carteira. Teoricamente pode-se resolver este problema para maior número de ativos, mas a solução numérica do modelo passará a ser muito demorada e por que não dizer inviável. Neste caso deve-se optar por outras formas de solução, como por exemplo uma simulação de Monte Carlo com os ativos correlacionados.

Uma grande vantagem deste modelo de diferenças finitas é o fato de gerar como resposta uma “superfície de solução”, o que possibilita ao usuário determinar de maneira rápida e para quaisquer valores dos ativos, o valor das gregas, com uma interpolação numérica na região de interesse.

A seguir estão mostrados os resultados da simulação de uma opção de compra europeia tipo “basket call” com dois ativos, com custos de transação e volatilidades determinísticas, calculada pelo modelo de diferenças finitas. Os valores encontrados foram comparados com uma simulação de Monte Carlo para a mesma opção.

A análise foi feita com os seguintes dados de entrada:

Dados gerais							
w	0,4	r	0,1				
S ₁₀	30	σ_1	0,2	D1	0,02	dS1	1,68
S ₂₀	50	σ_2	0,3	D2	0,03	dS2	1,12
K	42	ρ	0,5				
Dados para Diferenças Finitas							
T	1	dt	1/15000	δ_{tc}	1/50	kS1	0,05
						kS2	0,05
Dados para Monte Carlo							
Intervalos	200	Trajetórias	5000	Repetições	200		

Tabela 4.1 - Dados de entrada para simulação de uma opção compra europeia tipo “basket” com dois ativos e volatilidades determinísticas

A seguir estão os resultados da simulação.

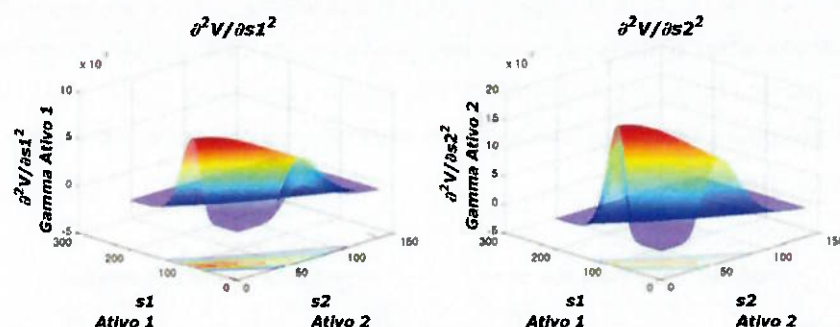


Figura 4.3 - Valores dos gammas sem custos de transação para a opção de compra europeia tipo “basket” com dois ativos, volatilidades determinísticas e custos de transação - $w=0,4$, $S10=30$, $S20=50$, $K=42$, $T=1$, $r=0,1$, $D1=0,02$, $D2=0,03$, $\rho=0,5$, $\sigma1=0,2$, $\sigma2=0,3$, $dS1=1,68$, $dS2=1,12$, $kS1=0,005$, $kS2=0,005$, $dt=1/12000$, $\delta tc=1/50$

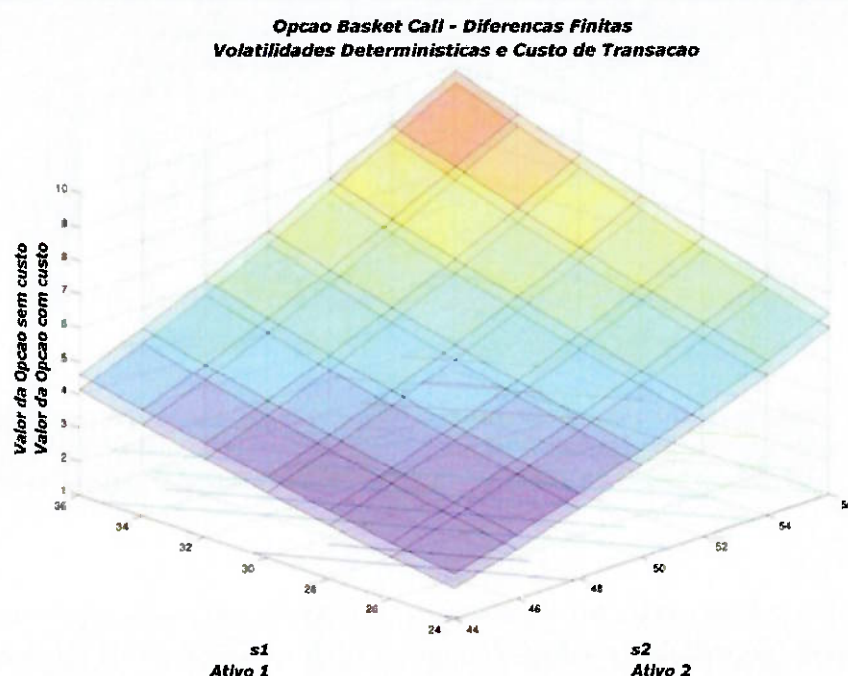


Figura 4.4 - Valores da opção com e sem custos de transação para a opção de compra europeia tipo “basket” com dois ativos, volatilidades determinísticas e custos de transação, em função dos valores dos ativos - $w=0,4$, $S10=30$, $S20=50$, $K=42$, $T=1$, $r=0,1$, $D1=0,02$, $D2=0,03$, $\rho=0,5$, $\sigma1=0,2$, $\sigma2=0,3$, $dS1=1,68$, $dS2=1,12$, $kS1=0,005$, $kS2=0,005$, $dt=1/12000$, $\delta tc=1/50$

A mesma análise foi feita por simulação de Monte Carlo para o caso sem custos de transação. Desta forma podemos comparar os valores obtidos pelos dois processos e avaliar a qualidade do modelo de apreçamento desenvolvido. Nos anexos A e B foram feitas algumas análises que avaliam a capacidade de apreçamento do modelo, quando introduzimos custos de transação.

As simulações de Monte Carlo foram implementadas de forma a fornecer algumas saídas gráficas que permitam uma avaliação do dimensionamento das simulações, ou seja, que permita determinar a quantidade de trajetórias, intervalos em cada trajetória e número de repetições para o cálculo de

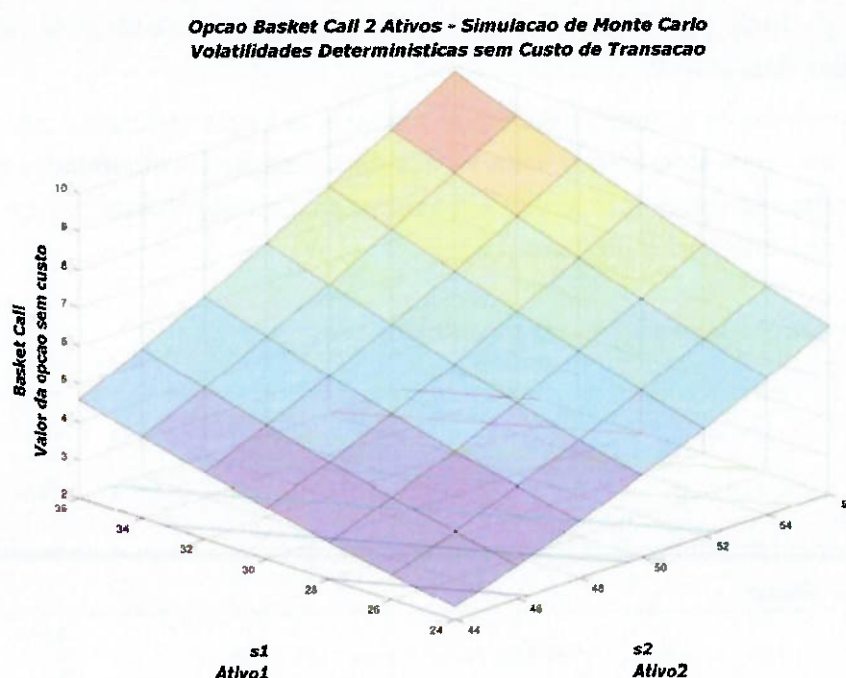


Figura 4.6 - Valores da opção sem custos de transação para a opção de compra europeia tipo “basket” com dois ativos e volatilidades determinísticas em função dos valores dos ativos, calculados por simulação de Monte Carlo - $w=0,4$, $K=42$, $T=1$, $r=0,1$, $D1=0,02$, $D2=0,03$, $\rho=0,5$, $\sigma1=0,2$, $\sigma2=0,3$, Trajetórias=5000, Intervalos=200, Repetições=200 para cada ponto da superfície

A tabela a seguir mostra um comparativo entre os valores da opção, obtidos pelo processo de diferenças finitas e simulação de Monte Carlo.

		Ativo 2						
		44	46	48	50	52	54	56
Ativo 1	24	2,3539	2,8903	3,4859	4,1383	4,8443	5,6004	6,4029
		2,349	2,891	3,4892	4,1342	4,8394	5,5988	6,4103
		0,21%	-0,02%	0,09%	0,10%	0,10%	0,03%	-0,12%
	26	2,658	3,2337	3,8679	4,5575	5,299	6,0887	6,9226
		2,6576	3,2342	3,8664	4,5546	5,2988	6,0864	6,9216
		0,02%	-0,02%	0,04%	0,06%	0,00%	0,04%	0,01%
	28	2,989	3,6041	4,2765	5,0027	5,7789	6,6009	7,4647
		2,9927	3,6035	4,2785	4,9968	5,7786	6,6087	7,4754
		0,12%	0,02%	0,05%	0,12%	0,01%	-0,12%	0,14%
	30	3,3475	4,0018	4,7118	5,4736	6,2831	7,1361	8,0284
		3,3417	4,0019	4,7154	5,4726	6,2818	7,1297	8,0303
		0,17%	0,00%	0,08%	0,02%	0,02%	0,09%	-0,02%
	32	3,7337	4,4266	5,1733	5,9695	6,811	7,6934	8,6126
		3,7436	4,4162	5,169	5,9715	6,8175	7,695	8,6012
		0,26%	0,24%	0,08%	-0,03%	-0,10%	-0,02%	0,13%
	34	4,1476	4,8783	5,6605	6,4898	7,3617	8,2719	9,2163
		4,1456	4,8846	5,6586	6,4896	7,3569	8,2629	9,2213
		0,05%	-0,13%	0,03%	0,00%	0,07%	0,11%	-0,05%
	36	4,589	5,3563	6,1727	7,0334	7,9342	8,8705	9,8383
		4,5806	5,353	6,1728	7,0346	7,9302	8,8705	9,8473
		0,18%	0,06%	0,00%	0,02%	0,05%	0,00%	-0,09%
Valores calculados com o modelo de diferenças finitas								
Valores calculados por simulação de Monte Carlo								
Erro entre os dois métodos								

Tabela 4.2 - Valores da opção sem custos de transação para a opção de compra europeia tipo “basket” com dois ativos e volatilidades determinísticas em função dos valores dos ativos - Comparação entre o método de diferenças finitas proposto e simulação de Monte Carlo - Erro médio 0,012%

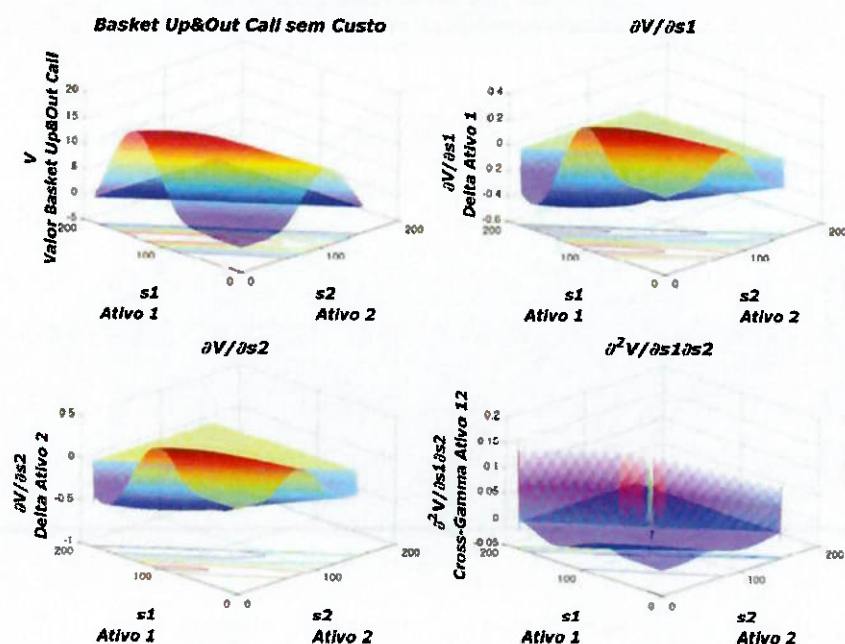


Figura 4.8 - Valores da opção e dos deltas sem custos de transação para a opção de compra europeia com barreira com dois ativos, volatilidades determinísticas e custos de transação - $w=0,45$, $S10=30$, $S20=50$, $K=42$, $Sb=85$, $T=1$, $r=0,1$, $D1=0,02$, $D2=0,03$, $\rho=0,5$, $\sigma_1=0,2$, $\sigma_2=0,3$, $dS1=1,25$, $dS2=1,0$, $kS1=0,005$, $kS2=0,005$, $dt=1/15000$, $\delta t_c=1/50$

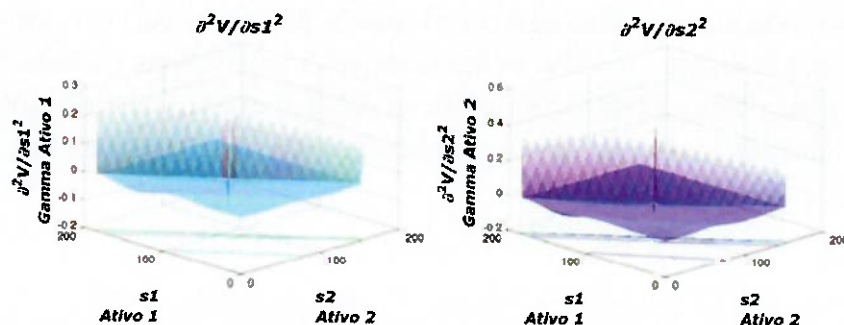


Figura 4.9 - Valores dos gammas sem custos de transação para a opção de compra europeia com barreira e dois ativos, volatilidades determinísticas e custos de transação - $w=0,45$, $S10=30$, $S20=50$, $K=42$, $Sb=85$, $T=1$, $r=0,1$, $D1=0,02$, $D2=0,03$, $\rho=0,5$, $\sigma_1=0,2$, $\sigma_2=0,3$, $dS1=1,25$, $dS2=1,0$, $kS1=0,005$, $kS2=0,005$, $dt=1/15000$, $\delta t_c=1/50$

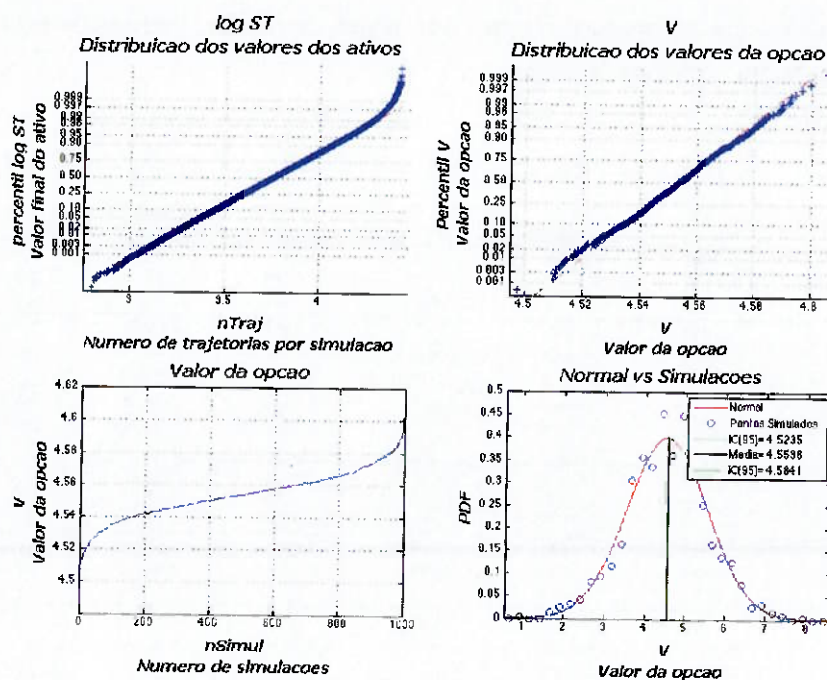


Figura 4.11 - Valor da opção sem custos de transação para a opção de compra europeia com barreira com dois ativos, volatilidades determinísticas - $w=0,45$, $S10=30$, $S20=50$, $K=42$, $Sb=85$, $T=1$, $r=0,1$, $D1=0,02$, $D2=0,03$, $\rho=0,5$, $\sigma1=0,2$, $\sigma2=0,3$, Trajetórias=50000, Intervalos=300, Repetições=1000

A simulação de Monte Carlo pode ser repetida diversas vezes em função dos valores dos ativos para determinar uma “superfície de solução” como foi feito no modelo de diferenças finitas. Calculando os valores no mesmo intervalo adotado para o cálculo anterior, teremos.

Opcao Basket Up&Out Call 2 Ativos - Simulacao de Monte Carlo
Volatilidades Determinísticas sem Custo de Transacao

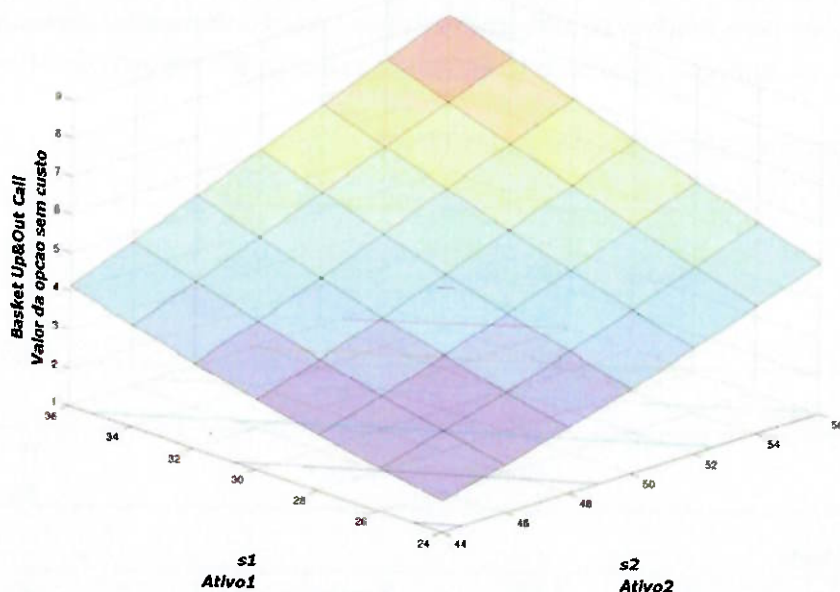


Figura 4.12 - Valores da opção sem custos de transação para a opção de compra europeia com barreira e dois ativos e volatilidades determinísticas em função dos valores dos ativos, calculados por simulação de Monte Carlo - $w=0,45$, $S10=30$, $S20=50$, $K=42$, $Sb=85$, $T=1$, $r=0,1$, $D1=0,02$, $D2=0,03$, $\rho=0,5$, $\sigma1=0,2$, $\sigma2=0,3$, Trajetórias=10000, Intervalos=300, Repetições=200

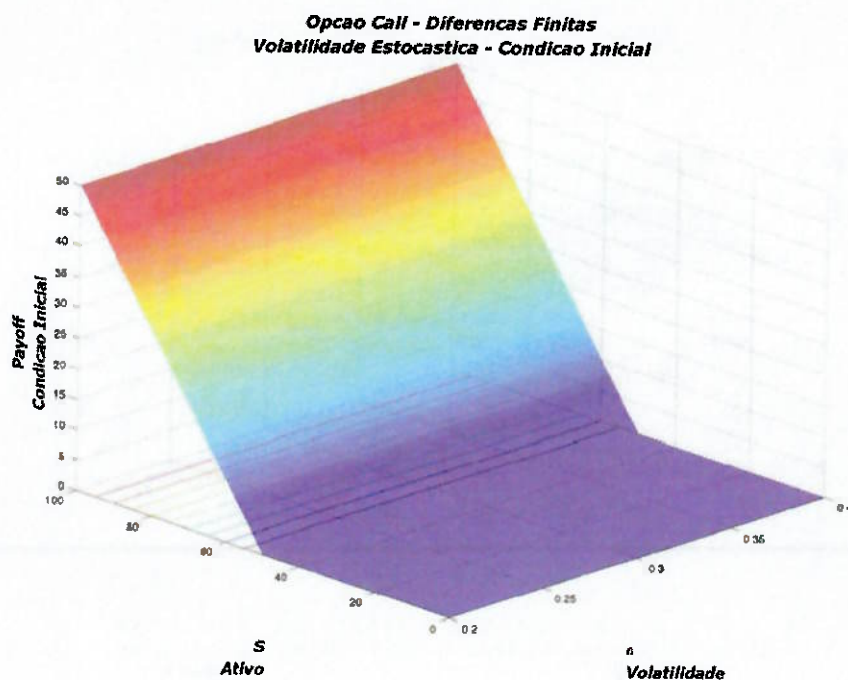


Figura 4.13 - Condição Inicial para o modelo de opção de compra europeia "vanilla" calculada com o modelo de volatilidade estocástica - $S_0=50$, $K=50$, $T=1$, $r=0,1$, $D=0,05$, $\rho=0,5$, $\beta=0,3$, $\sigma_1=0,2$, $\sigma_2=0,4$, $dS=0,8$, $d\sigma=0,0016$, $kS=0,005$, $k\sigma=0,005$, $dt=1/15000$, $\delta t_c=1/50$

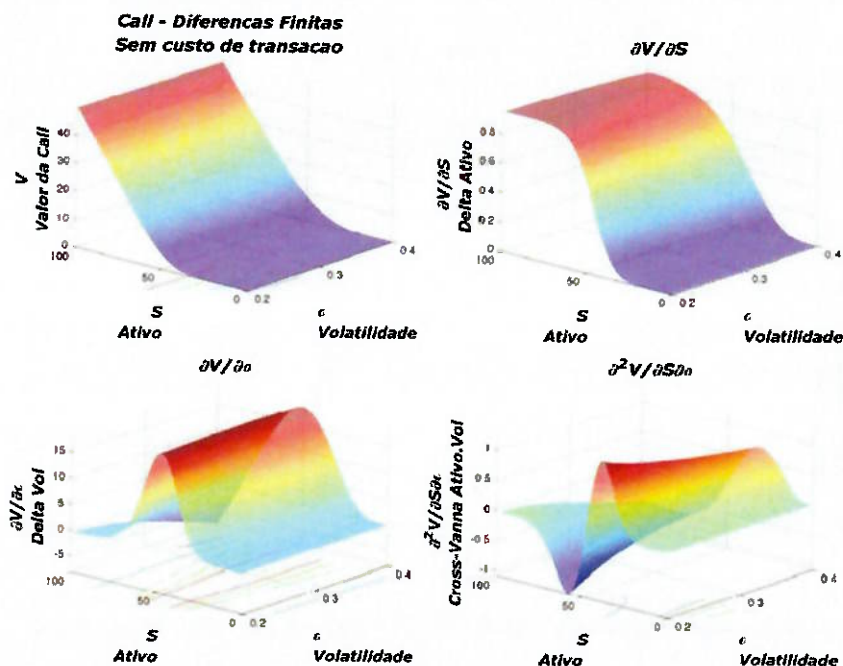


Figura 4.14 - Valores da opção e dos deltas sem custo para o opção de compra europeia "vanilla" calculada com o modelo de volatilidade estocástica - $S_0=50$, $K=50$, $T=1$, $r=0,1$, $D=0,05$, $\rho=0,5$, $\beta=0,3$, $\sigma_1=0,2$, $\sigma_2=0,4$, $dS=0,8$, $d\sigma=0,0016$, $kS=0,005$, $k\sigma=0,005$, $dt=1/15000$, $\delta t_c=1/50$

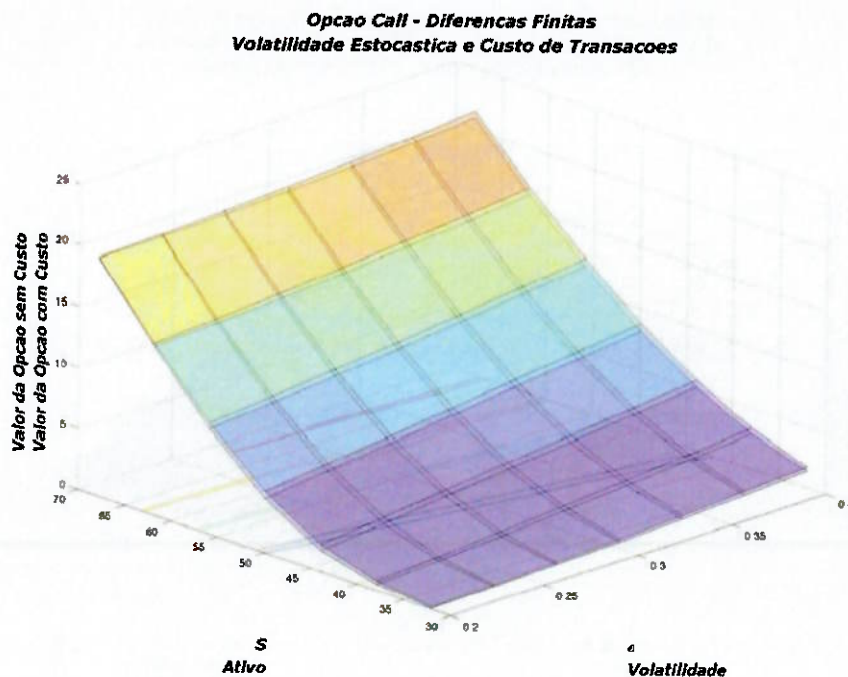


Figura 4.17 - Valor da opção com e sem custos de transação em função da volatilidade e valores do ativo, para o modelo de opção de compra europeia “vanilla” calculada com o modelo de volatilidade estocástica - $S_0=50$, $K=50$, $T=1$, $r=0,1$, $D=0,05$, $\rho=0,5$, $\beta=0,3$, $\sigma_{inicial}=0,2$, $\sigma_{final}=0,4$, $dS=0,8$, $d\sigma=0,0016$, $kS=0,005$, $k\sigma=0,005$, $dt=1/15000$, $\delta t_c=1/50$

A mesma análise foi feita por uma simulação de Monte Carlo usando o modelo de Heston, implementado pelo método de Milstein, para o caso sem custos de transação. Desta forma podemos comparar os valores obtidos pelos dois processos e avaliar a qualidade do modelo de apreçamento desenvolvido. Nos anexo B foram feitas algumas análises que avaliam a capacidade de apreçamento do modelo, quando introduzimos custos de transação.

Para gerar a superfície de pontos pela simulação de Monte Carlo, foram feitas 49 simulações diferentes, cada uma delas com um par de valores diferentes para o ativo e a variância $V(S, \sigma^2)$. O valor da opção para cada uma das simulações foi tomado como a média de 50 repetições de uma simulação gerada a partir de 10000 trajetórias de 200 intervalos dt . Vários valores da velocidade de reversão a média foram testados e o valor $\kappa=2$ foi o adotado para esta análise.

Os resultados estão apresentados a seguir.

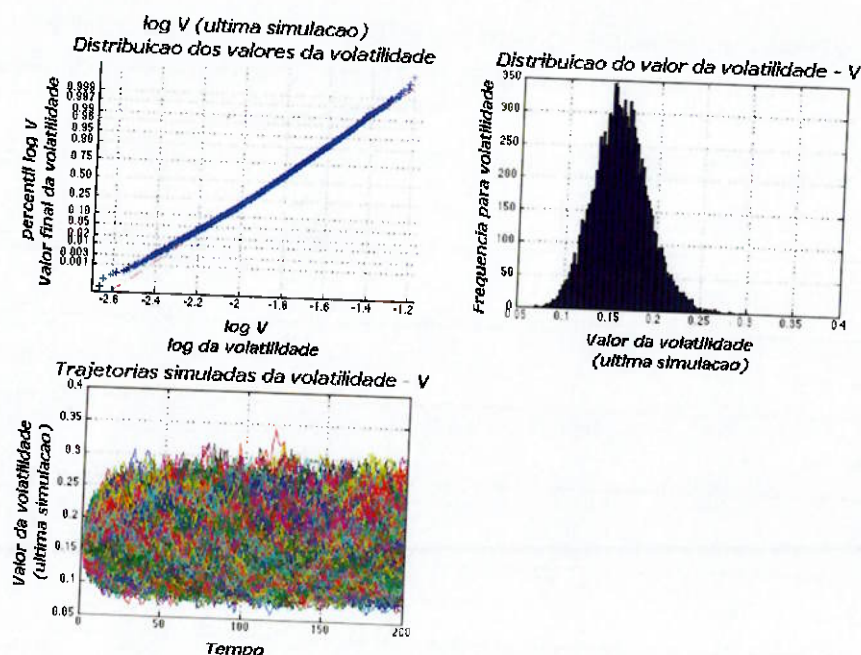


Figura 4.20 - Análise das trajetórias e distribuição do valor da volatilidade para o modelo de Heston nas simulações de Monte Carlo para o modelo de opção de compra europeia "vanilla" com volatilidade estocástica (valores da última simulação $S=68$, $\sigma=0,4$) - $K=50$, $T=1$, $r=0,1$, $D=0,05$, $\rho=0,5$, $\beta=0,3$, $\kappa=2$, Trajetórias=10000, Intervalos=200

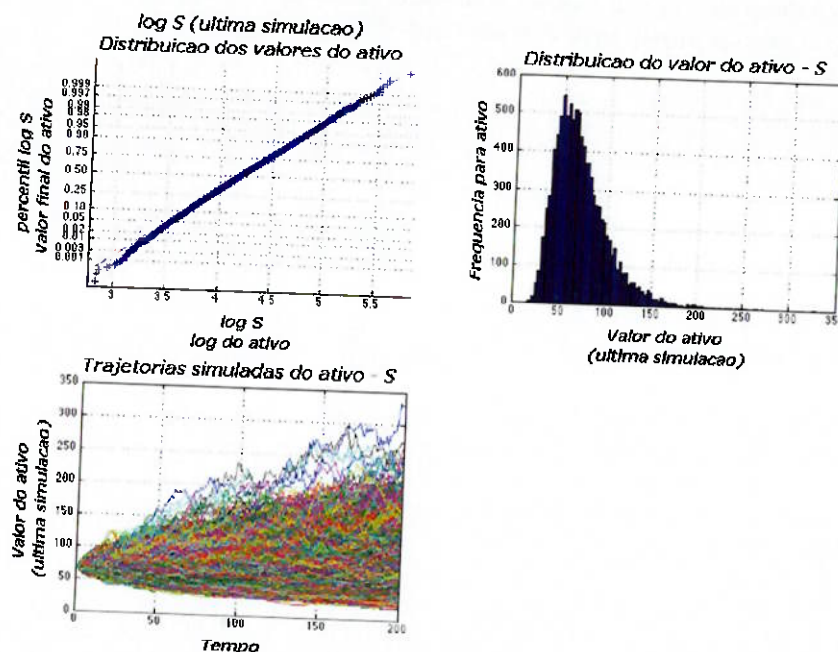


Figura 4.21 - Análise das trajetórias e distribuição do valor do ativo para o modelo de Heston nas simulações de Monte Carlo para o modelo de opção de compra europeia "vanilla" com volatilidade estocástica (valores da última simulação $S=68$, $\sigma=0,4$) - $K=50$, $T=1$, $r=0,1$, $D=0,05$, $\rho=0,5$, $\beta=0,3$, $\kappa=2$, Trajetórias=10000, Intervalos=200

4.5 Análise de uma opção de compra europeia com barreira “up&out call” e volatilidade estocástica

A seguir apresentamos os resultados do apreçamento de uma opção de compra europeia com barreira utilizando o modelo de diferenças finitas com volatilidade estocástica. Os resultados obtidos pelo modelo de diferenças finitas sem custos de transação foram comparados com uma simulação de Monte Carlo da mesma opção, usando o modelo de Heston com o método de Milstein.

A análise foi feita com os seguintes dados de entrada:

Dados gerais

S_0	50	β	0,3	D	0,05
S_b	85	$\sigma_{inicial}$	0,2	dS	0,68
K	50	σ_{final}	0,4	$d\sigma$	0,0016
r	0,1	ρ	0,5		

Dados para Diferenças Finitas

T	1	dt	1/15000	δ_{tc}	1/50	kS	0,05
						$k\sigma$	0,05

Dados para Monte Carlo

Intervalos	300	Trajeto�rias	10000	Repeti��es	50	κ	1
------------	-----	--------------	-------	------------	----	----------	---

Tabela 4.7 - Dados de entrada para simula  o de uma op  o compra europeia com barreira “up&out call” calculada com o modelo de volatilidade estoc  tica

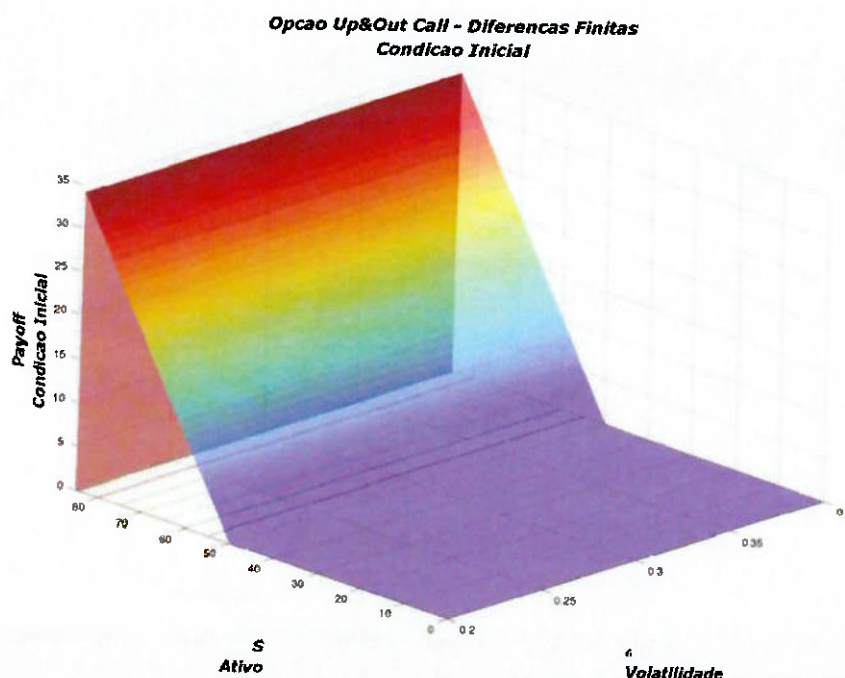


Figura 4.22 - Condi  o inicial para o modelo de op  o de compra europeia com barreira “up&out call” calculada com o modelo de volatilidade estoc  tica - $S_0=50$, $K=50$, $T=1$, $r=0,1$, $D=0,05$, $\rho=0,5$, $\beta=0,3$, $\sigma_{inicial}=0,2$, $\sigma_{final}=0,4$, $dS=0,68$, $d\sigma=0,0016$, $kS=0,005$, $k\sigma=0,005$, $dt=1/15000$, $\delta_{tc}=1/50$

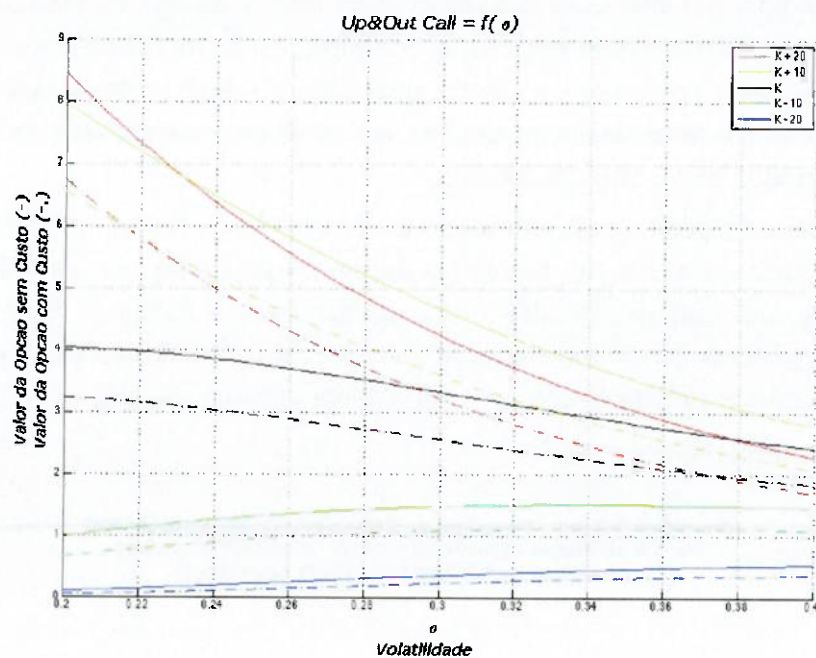


Figura 4.25 - Valor da opção com e sem custos de transação em função da volatilidade e valores de exercício, para o modelo de opção de compra europeia com barreira "up&out call" e volatilidade estocástica - $S_0=50$, $K=50$, $T=1$, $r=0,1$, $D=0,05$, $\rho=0,5$, $\beta=0,3$, $\sigma_1=0,2$, $\sigma_2=0,4$, $dS=0,68$, $d\sigma=0,0016$, $kS=0,005$, $k\sigma=0,005$, $dt=1/15000$, $\delta t_c=1/50$

Opcao Up&Out Call - Diferencas Finitas
Volatilidade Estocastica e Custo de Transacoes

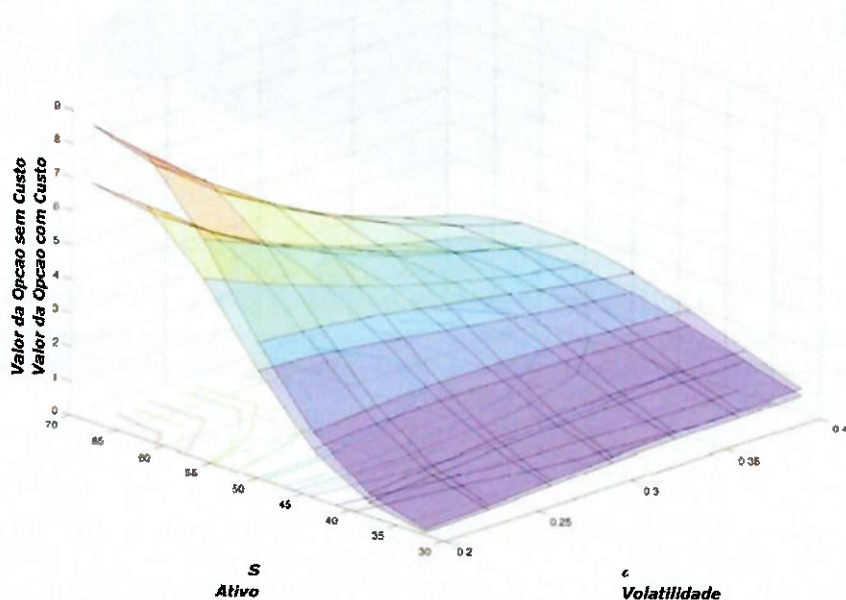


Figura 4.26 - Valor da opção com e sem custos de transação em função da volatilidade e valores do ativo, para o modelo de opção de compra europeia com barreira "up&out call" e volatilidade estocástica - $S_0=50$, $K=50$, $S_b=85$, $T=1$, $r=0,1$, $D=0,05$, $\rho=0,5$, $\beta=0,3$, $\sigma_1=0,2$, $\sigma_2=0,4$, $dS=0,85$, $d\sigma=0,0016$, $kS=0,005$, $k\sigma=0,005$, $dt=1/10000$, $\delta t_c=1/50$

Opcao Up&Out Call - Simulacao de Monte Carlo vs Diferencas Finitas
Modelo de Heston - Metodo de Milstein - Variaveis Antiteticas
Diferencas Finitas com Volatilidade Estocastica

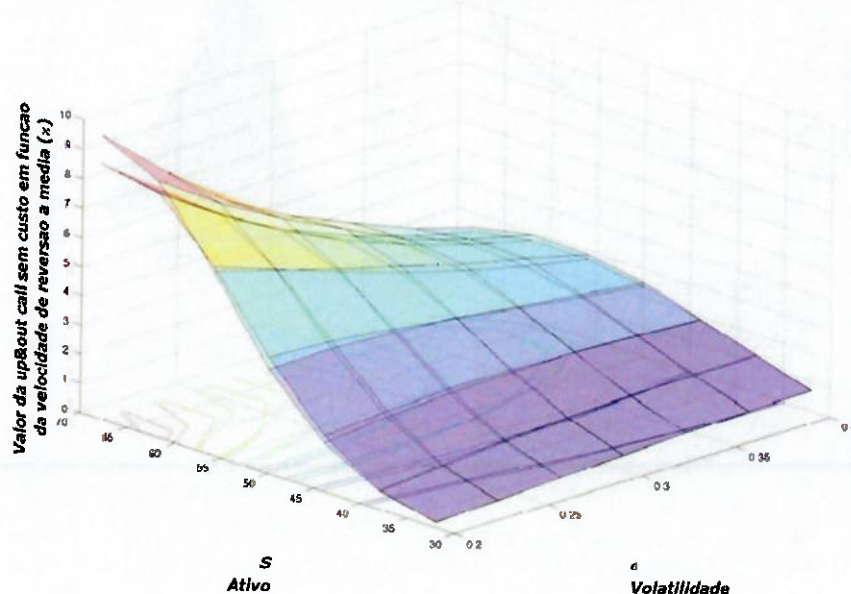


Figura 4.28 - Valor da opção sem custos de transação em função do valor do ativo e volatilidade para o modelo de opção de compra europeia com barreira “up&out call” e volatilidade estocástica - Comparativo com a simulação de Monte Carlo usando o modelo de Heston - $K=50$, $S_b=85$, $T=1$, $r=0,1$, $D=0,05$, $\sigma=0,4$, $\rho=0,5$, $\beta=0,3$, $\kappa=1$

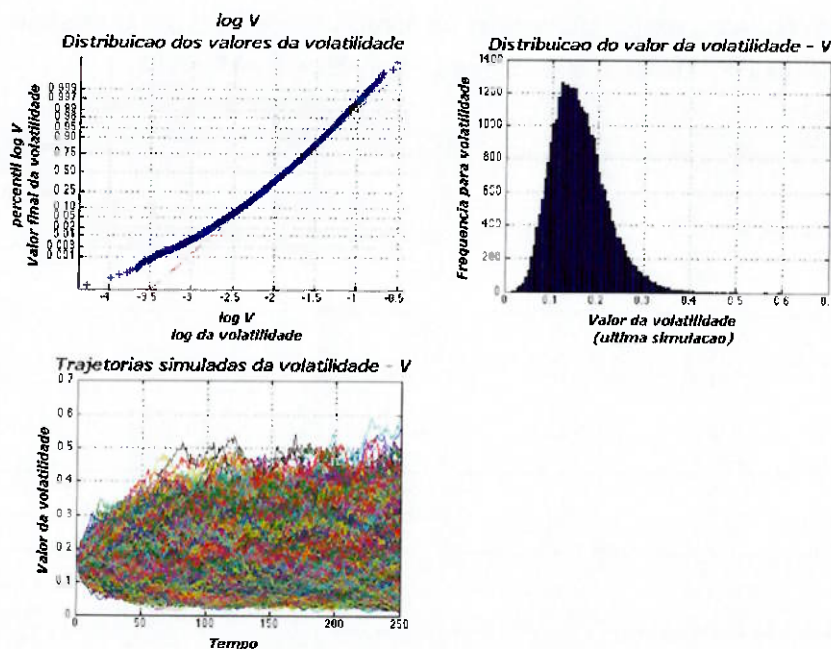


Figura 4.29 - Análise das trajetórias e distribuição do valor da volatilidade para o modelo de opção de compra europeia com barreira “up&out call” e volatilidade estocástica, calculadas por simulação de Monte Carlo usando o modelo de Heston - $K=50$, $S_b=85$, $T=1$, $r=0,1$, $D=0,05$, $\sigma=0,4$, $\rho=0,5$, $\beta=0,3$, $\kappa=2$, Trajetórias=5000, Intervalos=200 (valores da última simulação $S=68$, $\sigma=0,4$)

Capítulo 5

Discussão dos resultados

Como já vimos anteriormente, foram feitos vários testes de verificação e aceitação dos modelos desenvolvidos, comparando os resultados obtidos com as soluções de outros modelos, com trabalhos encontrados na literatura e simulações de Monte Carlo especialmente desenvolvidas para esse propósito. É necessário, antes de começar a colher resultados obtidos através das ferramentas desenvolvidas, conhecer suas capacidades e limitações.

A primeira parte do anexo A mostra uma comparação entre os resultados dos modelos de apreçamento para dois ativos com volatilidades determinísticas e outros modelos para um ativo usando o modelo de Hoggard, Whalley e Wilmott (1992) para os custos de transação. Usando como dado de entrada o valor de um dos ativos como zero, devemos esperar que os resultados do apreçamento sejam iguais para os dois modelos, tanto com ou sem custos de transação. De fato, observando as tabelas comparativas dos anexos A1 e A2, podemos observar que a implementação do modelo em duas dimensões não restringe sua aplicação para apenas um ativo. Os erros obtidos entre as duas soluções foram bastante baixos, comprovando a afirmação acima.

A segunda parte do anexo A mostra testes de simetria da solução quando os dados de entrada também são simétricos. Esse teste foi necessário para avaliar se havia algum erro de discretização das derivadas no modelo de diferenças finitas. Analogamente, observando as tabelas comparativas dos anexos A3 e A4, podemos ver que não há diferença de valores entre pontos simétricos na “superfície de resposta”. Podemos observar também que para o apreçamento sem custo, o erro em relação a simulações de Monte Carlo foi bastante pequeno, principalmente no caso da opção “*vanilla*”. No caso da opção com barreira “*up&out call*”, o tratamento da “quebra da barreira” pela Brownian Bridge parece introduzir um pequeno erro. De qualquer forma os resultados foram muito bons.

O anexo B mostra a comparação dos resultados entre o modelo de diferenças finitas desenvolvido para o caso de volatilidade estocástica e um outro trabalho [1]. Neste trabalho os autores usaram um modelo de elementos finitos para integração da EDP característica do problema, o que nos induz a concluir que a solução apresenta uma boa precisão. Nesta análise usamos uma malha de discretização bastante refinada 300x300 associada a um passo de integração bastante pequeno 1/40000s para que a solução não apresentasse instabilidade. Quatro casos foram comparados, com e sem custos de transação e quando as volatilidades são $\sigma=0,05$ e $\sigma=0,2$. Observando as tabelas comparativas do anexo B1 podemos observar que o modelo desenvolvido apresenta uma ótima precisão quando comparado com a solução por elementos finitos, tanto no apreçamento com, como no sem custos de transação. Apenas dois valores apresentaram erro elevado (7,5% e 4,3%) mas imagino que no primeiro caso isso se deve a um problema de arredondamento na apresentação dos resultados e no segundo caso a um erro de digitação em [1].

Mais dois testes foram realizados, comparando o modelo de apreçamento com volatilidade estocástica e os outros modelos para volatilidade determinística e custos de transação, usando o modelo de Hoggard, Whalley e Wilmott (1992). Nestas análises usamos o valor zero para a volatilidade da vo-

implementamos o modelo de Heston, usando o método de Milstein com variáveis antitéticas. Essas simulações são bem pesadas, sobretudo quando queremos obter a “superfície de resposta” para estes casos de volatilidade estocástica.

Aqui encontramos um problema maior, o ajuste do parâmetro κ (velocidade de reversão à média da variância para o modelo de Heston). Adotamos uma estratégia para o ajuste deste parâmetro, para que pudéssemos comparar os resultados obtidos entre os modelos desenvolvidos e as simulações de Monte Carlo com volatilidade estocástica. Foram analisados 5 casos, adotando para κ os valores de 0,5, 1, 2, 4, 8 e em cada simulação. Para cada κ foi gerada uma “superfície de resposta” e o erro quadrático médio foi calculado entre cada superfície e a superfície gerada pelo método de diferenças finitas. Desta forma obtemos uma curva de erro quadrático médio em função de κ e podemos estimar qual κ produzirá o erro mínimo. Após a determinação do κ ótimo refazemos a simulação adotando valores maiores para os parâmetros número de trajetórias, intervalos e repetições, plotando as superfícies geradas pelos dois métodos no mesmo gráfico.

Observando as tabelas comparativas 4.6 e 4.8 verificamos que os valores obtidos por diferenças finitas e Monte Carlo não são tão próximos quanto eram nos modelos de volatilidade determinística, tanto para a opção “*vanilla*” quanto para a opção com barreira “*up&out call*”, entretanto através do gráfico das superfícies podemos observar que apresentam exatamente o mesmo comportamento.

Existe uma explicação para esta diferença entre as duas simulações. Na realidade não estamos simulando o mesmo modelo. O modelo de Heston, usado nas simulações de Monte Carlo, não pressupõe a volatilidade com um ativo negociável e há uma pequena diferença entre termos convectivos da EDP de cada modelo. Na EDP do modelo de Heston existe o termo $\kappa(\theta - \nu)\partial V / \partial \nu$, ao passo que no modelo de diferenças finitas desenvolvido temos um termo ligeiramente diferente $r\sigma \partial V / \partial \sigma$.

Capítulo 6

Conclusões

O objeto deste trabalho foi estudar o apreamento de alguns tipos de derivativos, considerando volatilidades determinísticas e estocásticas, com custos de transação. Foram desenvolvidos dois modelos matemáticos para o apreamento de opções, e solução da equação diferencial parcial não linear foi obtida numericamente por um processo de diferenças finitas.

O primeiro modelo visa fazer o apreamento de opções europeias, considerando a volatilidade como um ativo negociável e considerando os custos de transação. Estudamos a solução de uma opção de compra “*plain vanilla*” e uma outra opção com barreira do tipo “*up&out call*”. O objetivo foi mostrar como podemos adaptar as condições iniciais e de contorno para resolver vários tipos de problemas.

O segundo modelo visa fazer o apreamento de opções europeias do tipo “*basket*”, considerando 2 ativos no portfólio, volatilidades determinísticas e custos de transação. Da mesma forma como foi feito para o caso anterior, estudamos a solução de uma opção de compra “*plain vanilla*” e uma outra opção com barreira do tipo “*up&out call*”.

Os modelos matemáticos propostos foram desenvolvidos no capítulo 3 e as EDPs resultantes (3.17 e 3.35) foram discretizadas por um processo de diferenças finitas explícito. Neste desenvolvimento procuramos mostrar como tratar as condições iniciais para cada tipo de opção de uma forma bidimensional, já que as EDPs resultantes dos modelos estudados apresentam duas dimensões no espaço. Procuramos também mostrar como tratar as derivadas parciais e as condições de contorno para os pontos situados sobre as fronteiras do domínio, de tal forma que a ordem de grandeza do erro de diferenciação seja constante em todo o domínio [12].

As equações discretizadas foram implementadas em *scripts Matlab®* e inicialmente vários casos foram simulados para se fazer uma validação dos modelos em termos de robustez, confiabilidade e precisão dos resultados. Essas validações foram realizadas comparando-se os casos simulados com soluções analíticas de problemas unidimensionais, com resultados apresentados na literatura de derivativos e também com simulações de Monte Carlo especialmente desenvolvidas.

Os resultados obtidos foram muito bons, como pode ser visto nas tabelas A1.2, A2.2, A3.2, A4.2. Os erros máximos foram da ordem de 3% para a opção com barreira. Podemos concluir que os modelos desenvolvidos podem ser usados no apreamento de derivativos com vantagens sobre métodos como por exemplo as simulações de Monte Carlo.

Observando as tabelas B1.2 e B1.3 podemos concluir que o modelo desenvolvido para o apreamento de opções com volatilidade estocástica reproduz com precisão os valores obtidos por elementos finitos em [1], tanto para a análise com como sem custos de transação.

Após os testes de validação foram simulados outros quatro casos de interesse, cujas variáveis de entrada representavam um problema geral e novamente os resultados foram comparados com simulações de Monte Carlo para o mesmo problema. Os resultados para as opções do tipo “*basket*”

Capítulo 7

Pesquisas futuras

7.1 Discretização implícita

A discretização da equação diferencial parcial para o processo de diferenças finitas foi feita explicitamente, devido à não linearidade do modelo quanto aos custos de transação. Desta forma estamos sempre usando as informações do passo anterior para calcular o novo passo e a não linearidade do modelo passa a não representar nenhum problema.

Entretanto a discretização explícita apresenta algumas limitações. Dependendo do grau de refinamento da malha a ser usada, do passo de integração e, no casos destes modelos, dos custos de transação, a solução pode tornar-se bastante instável. Uma forma de contornar esse problema é adotar um passo de integração bem pequeno, evitando assim a instabilidade, mas tornando a solução muito demorada.

A simulação do modelo da opção de compra europeia “*call*” com volatilidade estocástica, para uma malha de 300x300 demorou mais de 3 horas para ser realizada. Foi necessário usar um passo de 1/40000s para que a solução não apresentasse instabilidade.

Uma forma de contornar este problema seria fazer uma discretização implícita do método de diferenças finitas. Mas como acomodar a não linearidade do modelo? Imagino fazer algo do tipo “*predictor-corrector*”, ou seja, podemos escrever um método de Crank-Nicolson calculando as parcelas de custo com a informação do passo anterior e determinar uma solução intermediária para o passo seguinte. Com esta solução intermediária avaliamos novamente as parcelas dos custos de transação e calculamos novamente o passo seguinte. O custo para contornar a não linearidade será resolver o problema todo duas vezes, solução intermediária e solução final.

É preciso testar para verificar se não haverá muito erro numérico acumulado forçando a redução do passo de integração novamente. Vale observar que neste caso haverá a necessidade da solução de sistemas lineares, e como os problemas resolvidos neste trabalho são bidimensionais no espaço, os sistemas lineares serão grandes. Só haveria um ganho se o passo de integração pudesse ser aumentado consideravelmente.

7.2 Implementação de um método ADI

Se o método de discretização fosse implícito, os sistemas lineares a serem resolvidos seriam muito grandes. No caso da solução apresentada no anexo B1, deveríamos resolver a cada passo um sistema linear 90000x90000. Mesmo que sejam usados métodos de solução para sistemas esparsos, a solução seria muito demorada, inviabilizando talvez a solução destes problemas de apreçamento por diferenças finitas, elementos finitos ou outro método numérico semelhante.

Uma alternativa a este problema seria implementar um método ADI (Alternating Direction Implicit). Este método tem por objetivo reduzir a dimensionalidade do problema e, no exemplo citado acima, passaríamos a resolver 2x300 vezes um sistema linear 300x300. Pode não parecer mas é

Se desejarmos uma precisão razoável, o número de trajetórias, intervalos e repetições vai sem dúvida desencadear uma simulação muito demorada. Ao longo deste trabalho foram feitas algumas simulações que levaram cerca de 28 horas de processamento.

Uma das maneiras de tentar reduzir o tempo de simulação sem prejuízo da precisão seria a adoção de variáveis de controle nas simulações de Monte Carlo, que na realidade não era o foco principal deste trabalho. Neste trabalho em todas as simulações de Monte Carlo foram empregadas variáveis antitéticas.

7.7 Custos de transação nas simulações de Monte Carlo

Foi necessário recorrer a um modelo de custos de transação (Hoggard, Whalley e Wilmott (1992)), também implementado em diferenças finitas, para avaliar os modelos desenvolvidos neste trabalho. Teria sido bem melhor se o método de comparação fosse de outra natureza.

Não pensei muito, nem investiguei esse assunto, mas me parece possível fazê-lo. Imagino que a cada intervalo preestabelecido δt , devemos deduzir do valor na trajetória simulada, a parcela dos custos de transação. Precisamos avaliar as Gregas a cada intervalo δt , o que não é muito trivial, mas me parece um desafio interessante.

7.8 Brownian Bridge para as simulações de Monte Carlo das opções com barreiras

O problema da convergência das simulações de Monte Carlo para as opções com barreiras me preocupava há algum tempo. Tentei fazer algumas simulações, mas sem sucesso prático. Precisava aumentar em muito o número de intervalos na discretização para aproximar os resultados do valor exato calculado pelas fórmulas analíticas [6][10].

Não fazia parte do escopo deste trabalho mas uma probabilidade teórica de Brownian Bridge foi usada nos algoritmos das simulações de Monte Carlo com bastante sucesso.

Algumas simulações de Brownian Bridge puras foram feitas, para comparar a probabilidade simulada com a probabilidade teórica de uma Trajetória Browniana cruzar uma barreira.

Anexo A

A1 Análise do modelo de apreçamento de uma opção de compra europeia “vanilla” para dois ativos com volatilidades determinísticas

Antes de iniciar qualquer trabalho de simulação e análise de resultados com os modelos implementados, alguns testes foram realizados para verificar sua aplicabilidade e confiabilidade e estão descritos a seguir.

Análise do modelo desenvolvido para uma opção de compra europeia “vanilla” com dois ativos, e volatilidades determinísticas, no apreçamento de uma opção semelhante de um ativo

Inicialmente verificaremos a aplicabilidade do modelo desenvolvido para dois ativos no caso onde $w=1$, ou seja, o apreçamento de uma opção composta por um ativo apenas. Os resultados obtidos desta forma poderão ser comparados com soluções analíticas fechadas encontradas na literatura [1], ou então a outras simulações numéricas para o caso de um ativo apenas. A utilização $w=1$, deve fazer com que o modelo desenvolvido para dois ativos se comporte como um modelo de um ativo apenas. Na realidade isso deve valer tanto para $w=1$ como para $w=0$.

Vamos fazer uma simulação considerando $w=1$, ou seja 100% do ativo 1 e 0% do ativo 2 e ainda considerar a correlação entre os ativos nula, $\rho=0$. Fazendo-se isto espera-se que o modelo implementado se comporte como se fosse um modelo unidimensional para um ativo apenas. Os resultados encontrados devem ser rigorosamente os mesmos. Desta forma poderemos verificar a robustez do modelo desenvolvido.

Vamos considerar uma análise com os seguintes dados de entrada:

Dados gerais							
w	1	r	0,1				
S ₁₀	50	σ_1	0,4	D1	0,05		
S ₂₀	50	σ_2	0,4	D2	0,05		
K	50	ρ	0,0				
Dados para Diferenças Finitas							
T	1	dt	1/12000	dS1	0,8	kS1	0,05
		δt_c	1/50	dS2	0,8	kS2	0,05

Tabela A1.1 - Dados de entrada para simulação de uma opção compra europeia tipo “basket” com dois ativos e volatilidades determinísticas - Comparação com um modelo de um ativo

A seguir estão mostrados os resultados da análise descrita.

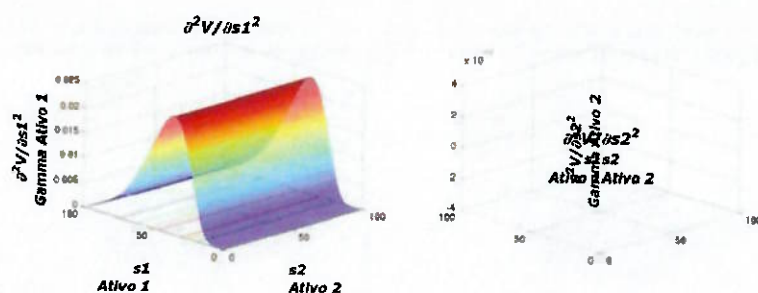


Figura A1.3 - Valor de gamma sem custo, em relação ao ativo 1, para o modelo de apreçamento de opção de compra europeia com dois ativos - $w=1$, $S_0=50$, $K=50$, $T=1$, $r=0,1$, $D1=0,05$, $\rho=0$, $\sigma_1=0,4$, $dS1=0,8$, $kS1=0,005$, $dt=1/12000$, $dte=1/50$

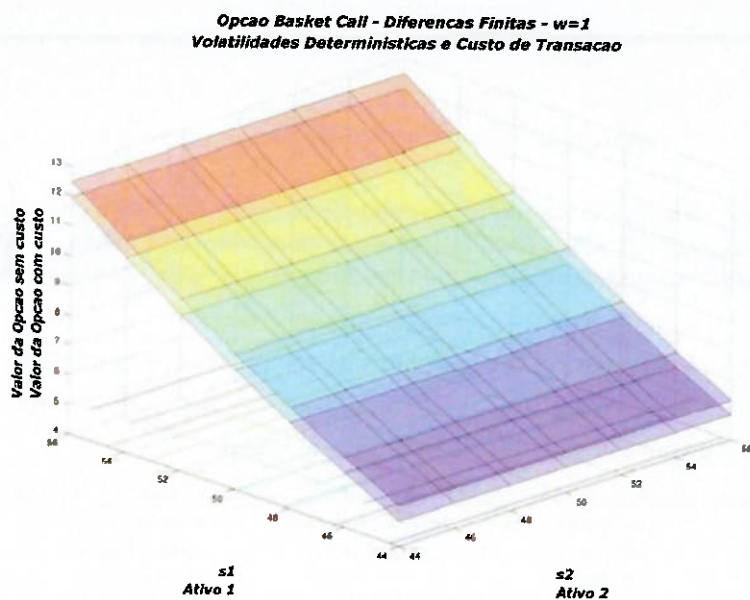


Figura A1.4 - Valor da opção com e sem custos de transação em função dos valores dos ativos (ativo 2 nulo) para o modelo de apreçamento de opção de compra europeia com dois ativos - $w=1$, $S_0=50$, $K=50$, $T=1$, $r=0,1$, $D1=0,05$, $\rho=0$, $\sigma_1=0,4$, $dS1=0,8$, $kS1=0,005$, $dt=1/12000$, $dte=1/50$

Para comparação dos valores da opção com custos de transação, vamos usar um modelo de diferenças finitas desenvolvido para um ativo apenas, considerando o modelo de Hoggard, Whalley e Wilmott (1992) para os custos de transação. Nesta análise será considerado apenas o custo proporcional ao valor do ativo, conforme implementado nos modelos para dois ativos.

A seguir encontram-se os resultados para o apreçamento de uma opção de compra europeia “vanilla” com custos de transação.

A2 Análise do modelo de apreçamento de uma opção de compra europeia com barreira "up&out call" para dois ativos com volatilidades determinísticas

Antes de iniciar qualquer trabalho de simulação e análise de resultados com os modelos implementados, alguns testes foram realizados para verificar sua aplicabilidade e confiabilidade e estão descritos a seguir.

Análise do modelo desenvolvido para uma opção de compra europeia com barreira, dois ativos e volatilidades determinísticas, no apreçamento de uma opção semelhante de um ativo

Da mesma forma utilizada para a opção "vanilla", para que possamos fazer uma comparação entre os resultados obtidos pelo modelo de apreçamento desenvolvido para dois ativos com a solução analítica fechada, vamos fazer uma simulação considerando uma opção de compra com barreira superior, do tipo "up&out call", com $w=1$, ou seja 100% do ativo 1 e 0% do ativo 2 e ainda considerar a correlação entre os ativos nula, $\rho=0$. Os resultados obtidos desta forma poderão ser comparados com soluções analíticas fechadas encontradas na literatura [1], ou então a outras simulações numéricas para o caso de um ativo apenas. Fazendo-se isto espera-se que o modelo implementado se comporte como se fosse um modelo unidimensional para um ativo apenas. Os resultados encontrados devem ser rigorosamente os mesmos. Desta forma poderemos verificar a robustez do modelo desenvolvido.

Vamos considerar uma análise com os seguintes dados de entrada:

Dados gerais							
w	1	r	0,1	Sb	85		
S10	50	σ_1	0,4	D1	0,05		
S20	50	σ_2	0,4	D2	0,05		
K	50	ρ	0,0				
Dados para Diferenças Finitas							
T	1	dt	1/15000	dS1	0,68	kS1	0,05
		δt_c	1/50	dS2	0,68	kS2	0,05

Tabela A2.1 - Dados de entrada para simulação de uma opção compra europeia com barreira tipo "basket" com dois ativos e volatilidades determinísticas - Comparação com um modelo de um ativo

A seguir estão mostrados os resultados da análise descrita.

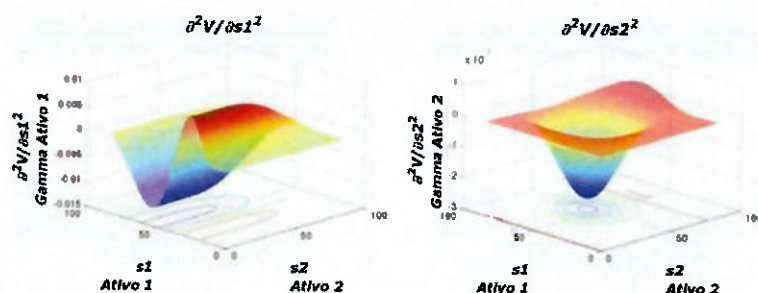


Figura A2.3 - Valor dos gammas sem custo para o modelo de apreçamento de opção de compra europeia com barreira do tipo “up&out call” e dois ativos - $w=1$, $S_0=50$, $K=50$, $S_b=85$, $T=1$, $r=0,1$, $D1=0,05$, $\rho=0$, $\sigma_1=0,4$, $dS1=0,68$, $kS1=0,005$, $dt=1/15000$, $\delta tc=1/50$

Opcao Basket Up&Out Cali - Diferencas Finitas - $w=1$
Volatilidades Deterministicas e Custo de Transacao

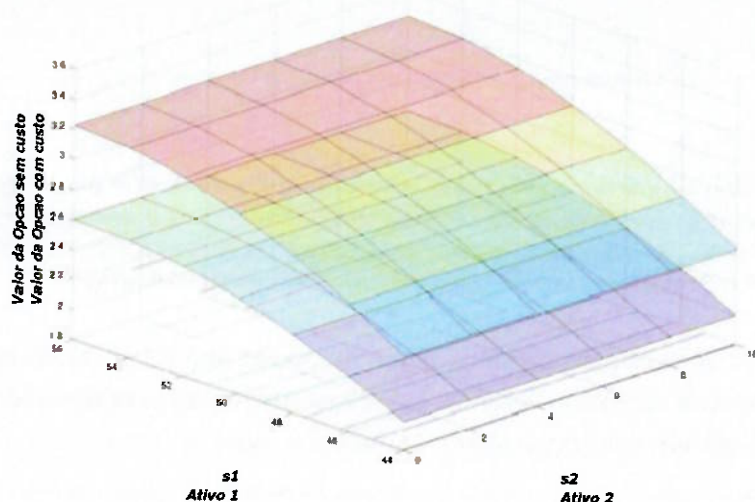


Figura A2.4 - Valor da opção com e sem custos de transação em função dos valores dos ativos para o modelo de apreçamento de opção de compra europeia com barreira do tipo “up&out call” e dois ativos - $w=1$, $S_0=50$, $K=50$, $S_b=85$, $T=1$, $r=0,1$, $D1=0,05$, $\rho=0$, $\sigma_1=0,4$, $dS1=0,68$, $kS1=0,005$, $dt=1/15000$, $\delta tc=1/50$

Para comparação dos valores da opção com custos de transação, vamos usar um modelo de diferenças finitas desenvolvido para um ativo apenas, considerando o modelo de Hoggard, Whalley e Wilmott (1992) para os custos de transação. Nesta análise será considerado apenas o custo proporcional ao valor do ativo, conforme implementado nos modelos para dois ativos.

A seguir encontram-se os resultados para o apreçamento de uma opção de compra europeia com barreira do tipo “up&out call” com custos de transação.

A3 Análise de simetria do modelo desenvolvido para uma opção de compra europeia “vanilla” com dois ativos e volatilidades determinísticas

Fazemos agora uma análise de simetria do modelo de apreçamento desenvolvido para uma opção de compra europeia “vanilla”, cujo objetivo é comprovar se para quaisquer dados de entrada simétricos, obtemos também uma saída simétrica, comprovando não haver nenhum problema na discretização da equação diferencial (3-35). Foi adotada uma carteira com $w=0,5$, 50% de cada um dos ativos e todas as características de cada ativo foram adotadas iguais.

Os resultados obtidos desta forma poderão ser comparados com simulações de Monte Carlo para os mesmos valores de entrada. Fazendo-se isto espera-se que o modelo implementado se comporte de maneira simétrica em relação aos valores dos ativos.

Vamos considerar uma análise com os seguintes dados de entrada:

Dados gerais							
w	0,5	r	0,1				
S10	50	σ_1	0,4	D1	0,05		
S20	50	σ_2	0,4	D2	0,05		
K	50	ρ	0,5				
Dados para Diferenças Finitas							
T	1	dt	1/12000	dS1	1,33	kS1	0,05
		δ_{tc}	1/50	dS2	1,33	kS2	0,05
Dados para Monte Carlo							
Intervalos	200	Trajetórias	10000	Repetições	100		

Tabela A3.1 - Dados de entrada para simulação de uma opção compra europeia tipo “basket” com dois ativos e volatilidades determinísticas - Verificação de simetria dos resultados

A seguir estão mostrados os resultados da análise descrita.

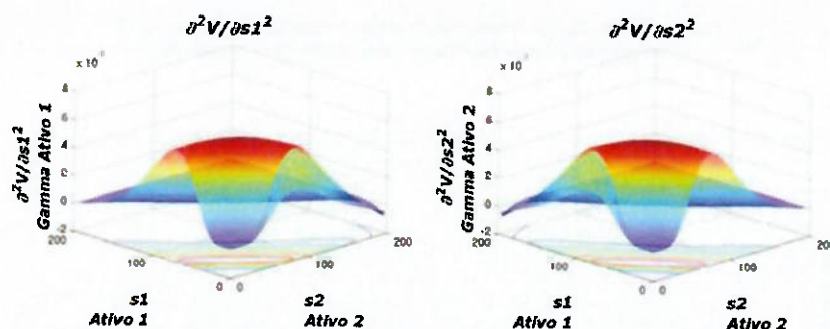


Figura A3.3 - Valores dos gammas sem custo para o modelo de opção de compra europeia “vanilla” com dois ativos, volatilidades determinísticas e dados de entrada simétricos - $w=0,5$, $S10=S20=50$, $K=50$, $T=1$, $r=0,1$, $D1=D2=0,05$, $\rho=0,5$, $\sigma1=\sigma2=0,4$, $dS1=dS2=1,33$, $kS1=kS2=0,005$, $dt=1/12000$, $\delta tc=1/50$

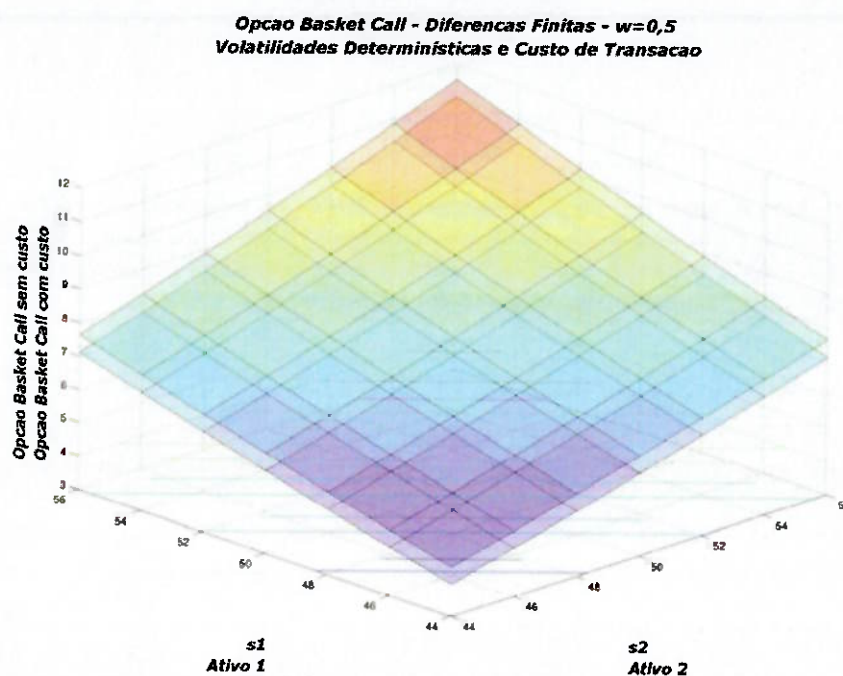


Figura A3.4 - Valor da Opção com e sem custos de transação em função dos valores dos ativos, para o modelo de opção de compra europeia “vanilla” com dois ativos, volatilidades determinísticas e dados de entrada simétricos - $w=0,5$, $S10=S20=50$, $K=50$, $T=1$, $r=0,1$, $D1=D2=0,05$, $\rho=0,5$, $\sigma1=\sigma2=0,4$, $dS1=dS2=1,33$, $kS1=kS2=0,005$, $dt=1/12000$, $\delta tc=1/50$

Comparamos os valores da opção sem custos de transação, fazendo uma simulação de Monte Carlo com os mesmos valores utilizados.

A4 Análise de simetria do modelo desenvolvido para uma opção de compra europeia com barreira “up&out call”, dois ativos e volatilidades determinísticas

Fazemos agora uma análise de simetria do modelo de apreçamento desenvolvido para uma opção de compra com barreira do tipo “up&out call”, cujo objetivo é comprovar se para quaisquer dados de entrada simétricos, obtemos também uma saída simétrica, comprovando não haver nenhum problema na discretização da equação diferencial (3-35). Foi adotada uma carteira com $w=0,5$, 50% de cada um dos ativos e todas as características de cada ativo foram adotadas iguais. Os resultados obtidos desta forma poderão ser comparados com simulações de Monte Carlo para os mesmos valores de entrada. Fazendo-se isto espera-se que o modelo implementado se comporte de maneira simétrica em relação aos valores dos ativos.

Vamos considerar uma análise com os seguintes dados de entrada:

Dados gerais

w	0,5	r	0,1	Sb	85
S ₁₀	50	σ_1	0,4	D1	0,05
S ₂₀	50	σ_2	0,4	D2	0,05
K	50	ρ	0,5		

Dados para Diferenças Finitas

T	1	dt	1/12000	dS1	1,36	kS1	0,05
		δt_c	1/50	dS2	1,36	kS2	0,05

Dados para Monte Carlo

Intervalos	300	Trajetórias	10000	Repetições	100
------------	-----	-------------	-------	------------	-----

Tabela A4.1 - Dados de entrada para simulação de uma opção compra europeia tipo “basket” com barreira, dois ativos e volatilidades determinísticas - Verificação de simetria dos resultados

A seguir estão mostrados os resultados da análise descrita.

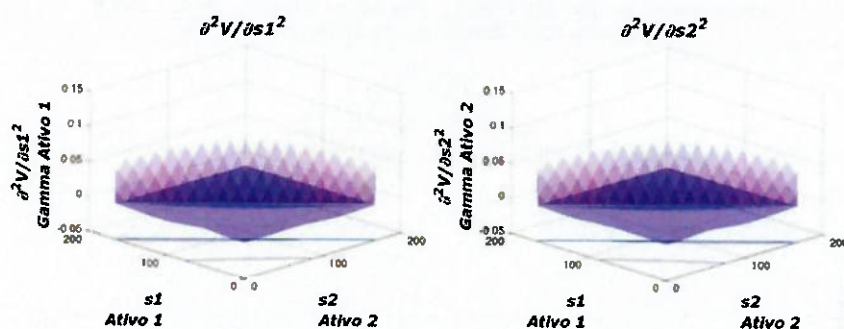


Figura A4.3 - Valor dos gammas sem custo para o modelo de opção de compra europeia com barreira “up&out call”, com dois ativos, volatilidades determinísticas e dados de entrada simétricos - $w=0,5$, $S10=S20=50$, $K=50$, $Sb=85$, $T=1$, $r=0,1$, $D1=D2=0,05$, $\rho=0,5$, $\sigma1=\sigma2=0,4$, $dS1=dS2=1,36$, $kS1=kS2=0,005$, $dt=1/15000$, $\delta tc=1/50$

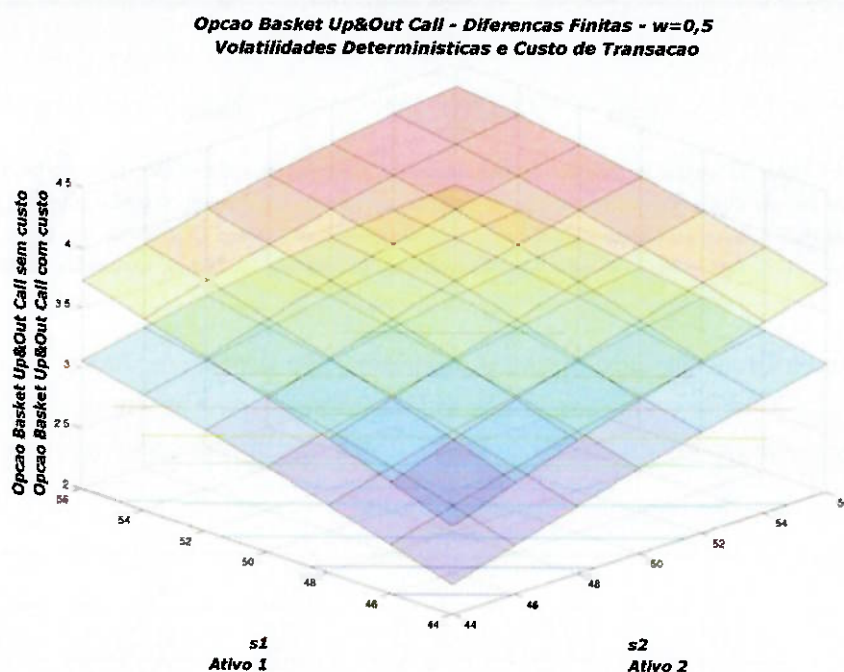


Figura A4.4 - Valor da opção com e sem custos de transação em função dos valores dos ativos, para o modelo de opção de compra europeia com barreira “up&out call” com dois ativos, volatilidades determinísticas e dados de entrada simétricos - $w=0,5$, $S10=S20=50$, $K=50$, $Sb=85$, $T=1$, $r=0,1$, $D1=D2=0,05$, $\rho=0,5$, $\sigma1=\sigma2=0,4$, $dS1=dS2=1,36$, $kS1=kS2=0,005$, $dt=1/15000$, $\delta tc=1/50$

Comparamos os valores da opção sem custos de transação, fazendo uma simulação de Monte Carlo com os mesmos valores utilizados.

Anexo B

B1 Análise do modelo para uma opção de compra europeia “vanilla” e volatilidade estocástica - Comparação com outro trabalho

Para que possamos verificar a qualidade do modelo desenvolvido para o apreçamento de opções de compra europeias com volatilidade estocástica, vamos comparar os resultados obtidos por este modelo com simulações encontrados na literatura, onde a solução da equação diferencial parcial foi obtida por um processo de elementos finitos [1], com e sem custos de transação. Com esta análise desejamos verificar a capacidade do modelo desenvolvido no apreçamento de opções com volatilidade estocástica e custos de transação. Curiosamente, no trabalho referido, o intervalo entre “hedges” foi considerado igual ao período de vencimento e portanto fizemos o mesmo.

Vamos considerar uma análise com os mesmos dados de entrada usados em [1]:

Dados gerais

S_0	100	r	0,05	σ_{inicial}	0,05	ρ	0,8
K	100	D	0	σ_{final}	0,2	β	0,4
S_{max}	200						

Dados para Diferenças Finitas

T	1	dt	1/40000	dS	0,666	kS	0,02
		δt_c	1/1	$d\sigma$	0,0005	$k\sigma$	0,02

Tabela B1.1 - Dados de entrada para simulação de uma opção compra europeia “vanilla” com volatilidade estocástica - Comparação com outro trabalho

A seguir estão mostrados os resultados da análise descrita.

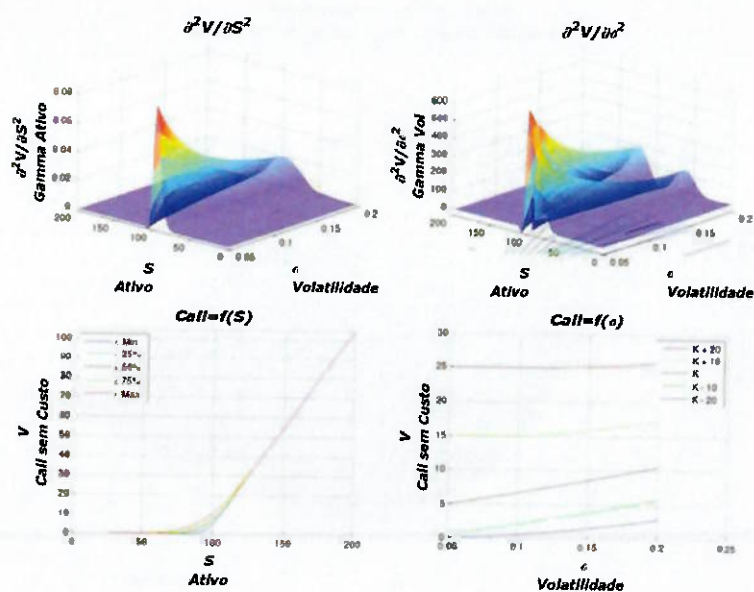


Figura B1.3 - Valores dos gamma sem custo para o modelo de apreçamento de opção de compra europeia “vanilla” com volatilidade estocástica - $S_0=100$, $K=100$, $T=1$, $r=0,05$, $D=0$, $\rho=0,8$, $\beta=0,4$, $\sigma_{\text{inic}}=0,05$, $\sigma_{\text{final}}=0,2$, $dS=0,66$, $d\sigma=0,0005$, $kS=0,2$, $k\sigma=0,2$, $dt=1/40000$, $\delta t_c=1$

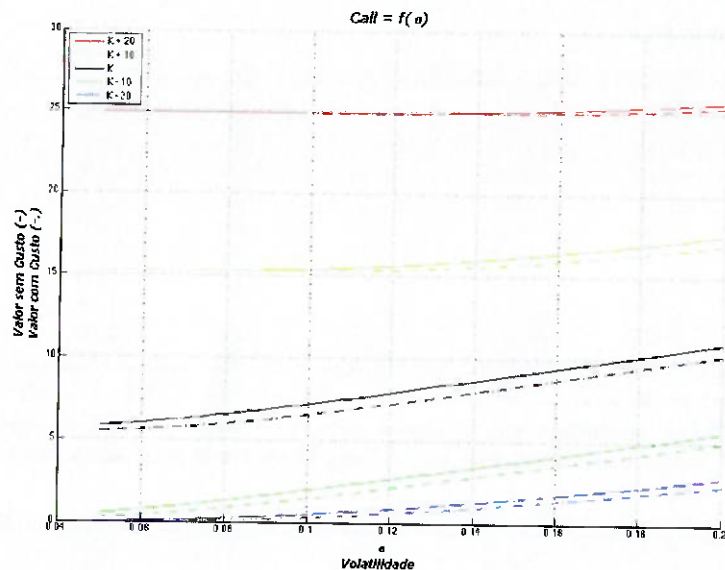


Figura B1.4 - Valor da opção com e sem custos de transação em função dos valores de volatilidade e exercício, para o modelo de apreçamento de opção de compra europeia “vanilla” com volatilidade estocástica - $S_0=100$, $K=100$, $T=1$, $r=0,05$, $D=0$, $\rho=0,8$, $\beta=0,4$, $\sigma_{\text{inic}}=0,05$, $\sigma_{\text{final}}=0,2$, $dS=0,66$, $d\sigma=0,0005$, $kS=0,2$, $k\sigma=0,2$, $dt=1/40000$, $\delta t_c=1$

B2 Análise do modelo de volatilidade estocástica em relação a modelos com volatilidade determinística para uma opção de compra europeia “vanilla”

Vamos comparar os resultados obtidos com o modelo de volatilidade estocástica e o modelo de volatilidade determinística, considerando o modelo de Hoggard, Whalley e Wilmott (1992) para os custos de transação. Para esta análise vamos adotar $\beta=0$ e $k\sigma=0$. Fazendo $\beta=0$ estamos admitindo volatilidade nula para a variância e fazendo $k\sigma=0$ estamos calculando apenas o custo devido ao valor do ativo. Desta forma o modelo de volatilidade estocástica deve apresentar um comportamento semelhante ao modelo de volatilidade determinística, permitindo assim comparar os resultados obtidos por eles. Para os custos de transação no modelo de Hoggard, Whalley e Wilmott (1992) vamos considerar apenas o custo proporcional ao valor do ativo, ou seja, $k1=0$, $k2=0$ e $k3=0,005$.

Vamos considerar uma análise com os seguintes dados de entrada:

Dados gerais							
S_0	50	r	0,1	σ_{inicial}	0,2	ρ	0
K	50	D	0,05	σ_{final}	0,4	β	0
S_{max}	100						
Dados para Diferenças Finitas - Modelo com Volatilidade Estocástica							
T	1	dt	1/10000	dS	0,8	kS	0,005
		δ_{tc}	1/50	$d\sigma$	0,0016	$k\sigma$	0
Dados para Diferenças Finitas - Modelo com Volatilidade Determinística							
T	1	dt	1/10000	$k1$	0	$\sigma1$	0,2
		δ_{tc}	1/50	$k2$	0	$\sigma2$	0,4
		dS	0,5	$k3$	0,005		

Tabela B2.1 - Dados de entrada para simulação de uma opção compra europeia “vanilla” com volatilidade estocástica - Comparação com outro modelo com volatilidade determinística

A seguir estão os resultados da simulação proposta para o caso de uma opção de compra europeia “vanilla”.

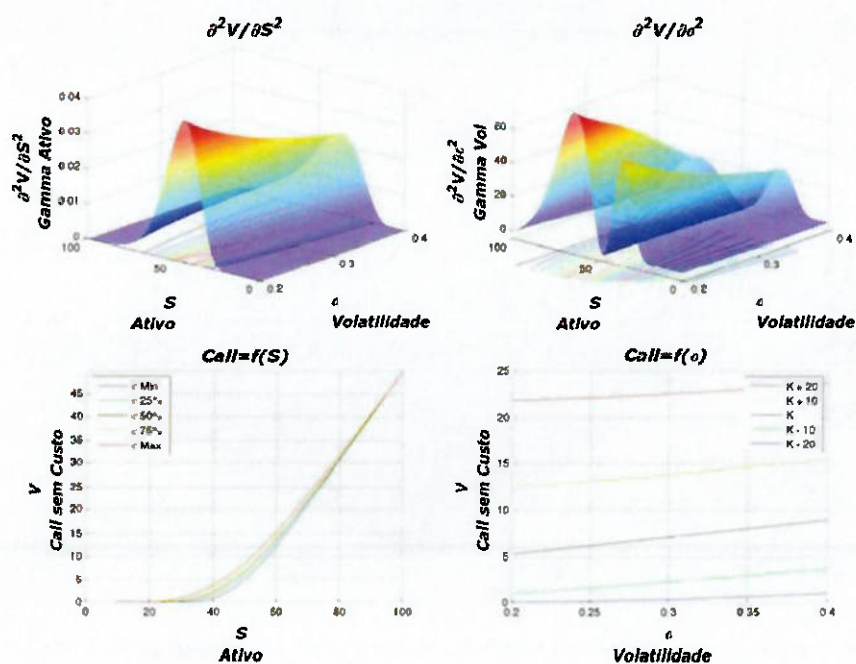


Figura B2.3 - Valores dos gammas sem custo para o opção de compra europeia “vanilla” de volatilidade estocástica, analisando um caso de volatilidade determinística - $S_0=50$, $K=50$, $T=1$, $r=0,1$, $D=0,05$, $\rho=0$, $\beta=0$, $\sigma_1=0,2$, $\sigma_2=0,4$, $dS=0,8$, $d\sigma=0,0016$, $kS=0,005$, $k\sigma=0$, $dt=1/10000$, $\delta tc=1/50$

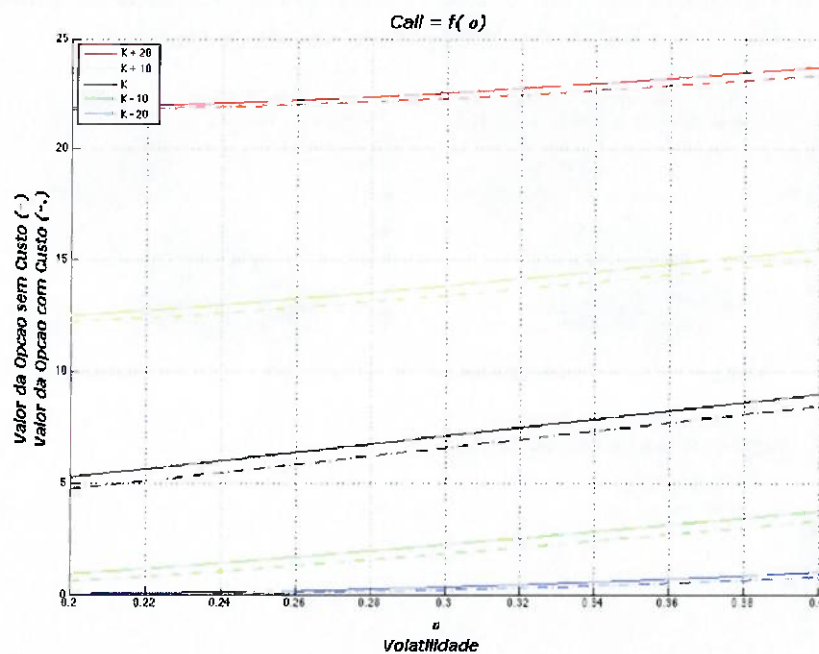


Figura B2.4 - Valor da opção com e sem custos de transação em função da volatilidade e valores de exercício, para o modelo de opção de compra europeia “vanilla” de volatilidade estocástica, analisando um caso com volatilidade determinística - $S_0=50$, $K=50$, $T=1$, $r=0,1$, $D=0,05$, $\rho=0$, $\beta=0$, $\sigma_1=0,2$, $\sigma_2=0,4$, $dS=0,8$, $d\sigma=0,0016$, $kS=0,005$, $k\sigma=0$, $dt=1/10000$, $\delta tc=1/50$

A tabela a seguir mostra um comparativo entre as soluções obtidas com o modelo de volatilidade estocástica e o modelo de volatilidade determinística usando o modelo de Hoggard, Whalley e Wilmott (1992) para os custos de transação.

Valor do ativo	$\sigma=0,2$				$\sigma=0,4$			
	Black-Scholes	Modelo de volatilidade estocástica sem custo	Modelo de volatilidade determinística com custo	Modelo Volatilidade estocástica com custo	Black-Scholes	Modelo de volatilidade estocástica sem custo	Modelo de volatilidade determinística com custo	Modelo Volatilidade estocástica com custo
44	2.0333	2.1146	1.5510	1.6311	5.3378	5.5071	4.8489	5.0190
46	2.8510	2.9383	2.3202	2.4081	6.3356	6.5109	5.8273	6.0041
48	3.8328	3.9247	3.2819	3.3753	7.4152	7.5955	6.8942	7.0764
50	4.9705	5.0545	4.4279	4.5110	8.5720	8.7506	8.0444	8.2250
52	6.2506	6.3368	5.7402	5.8242	9.8007	9.9822	9.2724	9.4562
54	7.6565	7.7340	7.1953	7.2676	11.0962	11.2744	10.5721	10.7528
56	9.1696	9.2392	8.7677	8.8295	12.4532	12.6273	11.9377	12.1147

Tabela B2.2 - Valores da opção com e sem custos de transação, para o modelo de opção de compra europeia "vanilla", com volatilidade estocástica, em função dos valores do ativo e volatilidades médias de $\sigma_1=0,2$ e $\sigma_2=0,4$, $S_0=50$, $K=50$, $T=1$, $r=0,1$, $D=0,05$, $\rho=0$, $\beta=0$, $dS=0,8$, $d\sigma=0,0016$, $kS=0,005$, $k\sigma=0$, $dt=1/10000$, $\delta tc=1/50$ - Comparativo com modelo de volatilidade determinística usando o modelo de Hoggard, Whalley e Wilmott (1992) para os custos de transação - $S_0=50$, $K=50$, $T=1$, $r=0,1$, $D=0,05$, $\sigma=0,2$ e $\sigma=0,4$, $dS=0,5$, $k1=0$, $k2=0$, $k3=0,005$, $dt=1/10000$, $\delta tc=1/50$

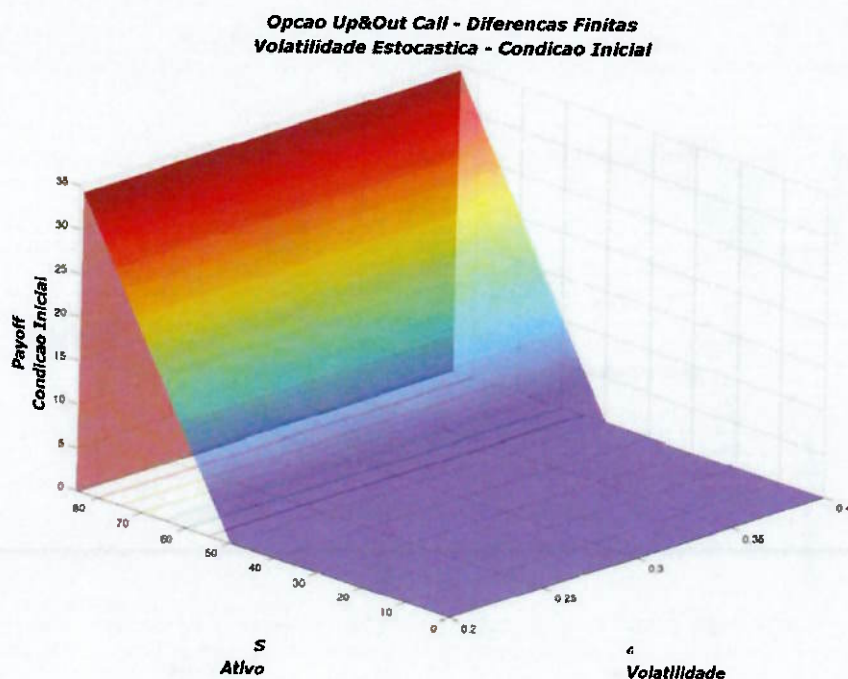


Figura B3.1 - Condição inicial para o modelo de opção de compra europeia com barreira “up&out call” de volatilidade estocástica, analisando um caso de volatilidade determinística - $S_0=50$, $K=50$, $T=1$, $r=0,1$, $D=0,05$, $\rho=0$, $\beta=0$, $\sigma_1=0,2$, $\sigma_2=0,4$, $dS=0,68$, $d\sigma=0,0016$, $kS=0,005$, $k\sigma=0$, $dt=1/10000$, $\delta t_c=1/50$

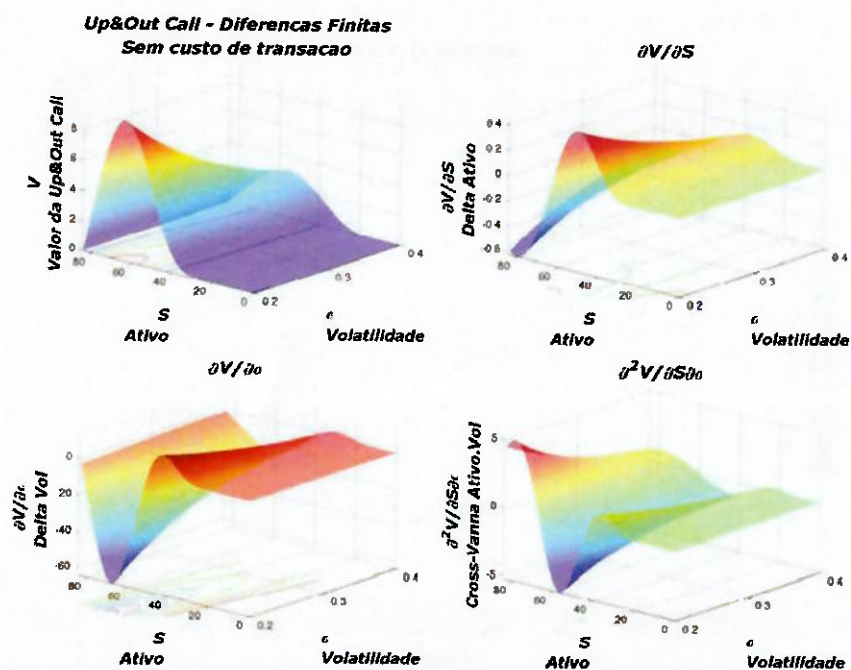


Figura B3.2 - Valores da opção e dos deltas sem custo para o modelo de opção de compra europeia com barreira “up&out call” de volatilidade estocástica, analisando um caso de volatilidade determinística - $S_0=50$, $K=50$, $T=1$, $r=0,1$, $D=0,05$, $\rho=0$, $\beta=0$, $\sigma_1=0,2$, $\sigma_2=0,4$, $dS=0,68$, $d\sigma=0,0016$, $kS=0,005$, $k\sigma=0$, $dt=1/10000$, $\delta t_c=1/50$

**Opcao Up&Out Call - Diferencas Finitas
Volatilidade Estocastica e Custo de Transacoes**

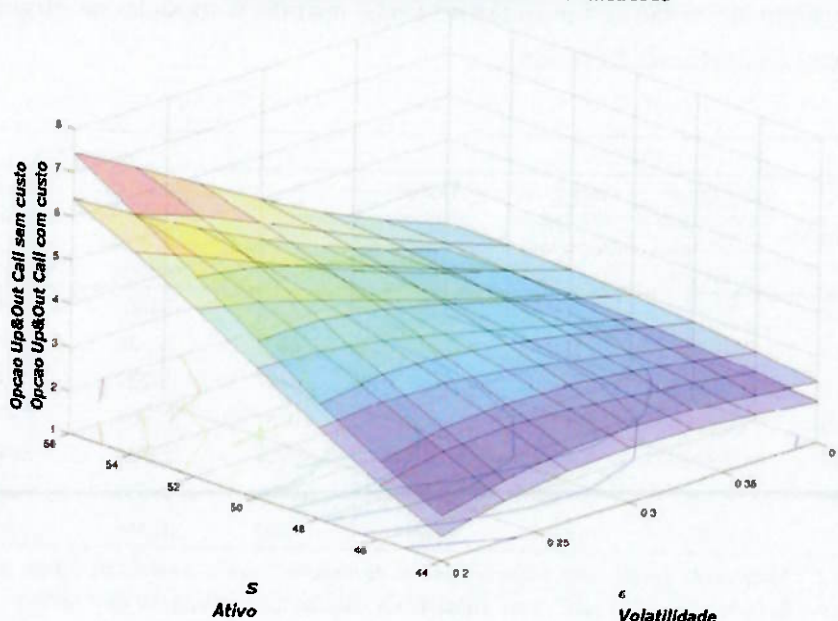


Figura B3.5 - Valor da opção com e sem custos de transação em função da volatilidade e valores do ativo, para o modelo de opção de compra europeia com barreira “up&out call” de volatilidade estocástica, analisando um caso de volatilidade determinística - $S_0=50, K=50, T=1, r=0,1, D=0,05, \rho=0, \beta=0, \sigma_1=0,2, \sigma_2=0,4, dS=0,68, d\sigma=0,0016, kS=0,005, k\sigma=0, dt=1/10000, \delta t_c=1/50$

Agora vamos analisar o mesmo problema com o modelo de volatilidade determinística, considerando o modelo de Hoggard, Whalley e Wilmott (1992) para os custos de transação.

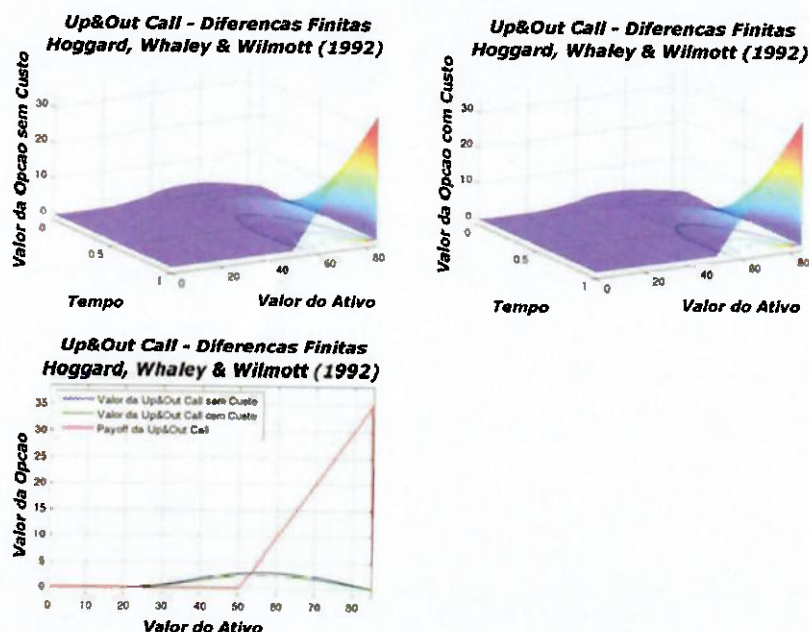


Figura B3.6 - Valor da opção com e sem custos de transação em função do valor do ativo, para o modelo de opção de compra europeia com barreira “up&out call” de volatilidade determinística, usando o modelo de Hoggard, Whalley e Wilmott (1992) para os custos de transação - $S_0=50, K=50, T=1, r=0,1, D=0,05, \sigma=0,4, dS=0,5, k_1=0, k_2=0, k_3=0,005, dt=1/10000, \delta t_c=1/50$

Anexo C - Códigos Matlab®

C1 Opção europeia “call” com modelo de Hoggard, Whalley e Wilmott (1992) para os custos de transação

```
%Aprecamento de Opcao Europeia Call pelo metodo de diferencas finitas
%usando um algoritmo explicito e o modelo de Hoggard, Whalley e Wilmott
%para o custo de transacoes.
%Adotados os seguintes valores de entrada
%(St,K,r,D,T,Sigma,Smax,dS,dt,dtc,k1,k2,k3)
%Referencia - Paul Wilmott, Quantitative Finance
clear all;close all;clc;
format short;

%Dados de entrada para os calculos
S0=50;%Valor do ativo
K=50;%Strike
r=.1;%Taxa de juros livre de risco
D=.05;%Taxa de dividendos
t=0;T=1;%Tempo para o vencimento
Sigma=.2;%Volatilidade
Smax=100;%Valor maximo do ativo
dS=.5;%Discretizacao do valor do ativo
dt=T/10000;%Discretizacao do tempo
k1=.0;%Custo fixo por transacao
k2=.0;%Custo proporcional ao volume
k3=.005;%Custo proporcional ao valor
dtc=T/50;%Intervalo entre operacoes de hedge

%Dados para obtencao de resultados interpolados
vectorInterpS=[S0-20:2:S0+20];%Vetor dos valores do ativo para interpolacao

ticID = tic;%Inicia o timer

%Ajuste do grid e dos incrementos
M = round(Smax/dS);
dS = Smax/M;
N = round(T/dt);
dt = T/N;

%Inicializacao de variaveis
Valor = zeros(M+1,N+1);
Delta = zeros(M+1,N+1);
Gamma = zeros(M+1,N+1);
Theta = zeros(M+1,N+1);

%Definicao da Condicoes de Contorno
vetor_S = linspace(0,Smax,M+1)';
Valor(:,N+1) = max(vetor_S-K,0);
ValorC=Valor;

%Calculo do valor da opcao sem custo
for j=N:-1:1
    for i=2:M
        Delta(i,j)=(Valor(i+1,j+1)-Valor(i-1,j+1));
        Gamma(i,j)=(Valor(i+1,j+1)-2*Valor(i,j+1)+Valor(i-1,j+1));
        Theta(i,j)=r*Valor(i,j+1)-Delta(i,j)/2*(r-D)*(i-1)-
        Gamma(i,j)/2*Sigma^2*(i-1)^2;
        Valor(i,j)=Valor(i,j+1)-dt*Theta(i,j);
    end
    %Condiacao de contorno nas extremidades
    %Segunda derivada nula
    Valor(1,j)=(5*Valor(2,j)-4*Valor(3,j)+Valor(4,j))/2;
    Valor(M+1,j)=(5*Valor(M,j)-4*Valor(M-1,j)+Valor(M-2,j))/2;
end

%Calculo do valor da opcao com custo
for j=N:-1:1
    for i=2:M
        Delta(i,j)=(ValorC(i+1,j+1)-ValorC(i-1,j+1));
```

```

weight','b','color','k');
%legend('\fontname{Verdana}\fontsize{11}\bf\itValor Up&Out Call sem custo de
transacao','\fontname{Verdana}\fontsize{11}\bf\itValor Up&Out Call com custo de
transacao','\fontname{Verdana}\fontsize{11}\bf\itPayoff da Up&Out Call',2)
legend('Valor da Call sem Custo','Valor da Call com Custo','Payoff da Call',2)

elapsedTime = toc(ticID);%Para o timer

%Mostra os valores calculados com este algoritmo vs formula fechada para
%a solucao deste problema - ref. Wilmott
fprintf('\n -----')
fprintf('\n Apreçamento de Opcao Call pelo metodo de Diferencas Finitas')
fprintf('\n Algoritmo explicito e modelo de Hoggard, Whalley e Wilmott (1992)')
fprintf('\n para o custo de transacoes')
fprintf('\n -----')
fprintf('\n Valores usados para este calculo')
fprintf('\n dS [intervalo do valor do ativo] : %-.2f',dS)
fprintf('\n dt [intervalo de tempo de integracao] : %-.6f',dt)
fprintf('\n St [valor do ativo] : %-.2f',S0)
fprintf('\n K [strike] : %-.2f',K)
fprintf('\n Smax [valor maximo do ativo] : %-.2f',Smax)
fprintf('\n T [tempo de vencimento] : %-.2f',T)
fprintf('\n r [taxa de juros livre de risco] : %-.2f',r)
fprintf('\n D [taxa de dividendo] : %-.2f',D)
fprintf('\n Sigma [volatilidade] : %-.2f',Sigma)
fprintf('\n -----')
fprintf('\n Valores de custo para o modelo de Hoggard, Whalley e Wilmott')
fprintf('\n dtc [intervalo de tempo de hedge] : %-.4f',dtc)
fprintf('\n k1 [custo fixo por transacao] : %-.4f',k1)
fprintf('\n k2 [custo proporcional ao volume] : %-.4f',k2)
fprintf('\n k3 [custo proporcional ao valor] : %-.4f',k3)
fprintf('\n -----')
fprintf('\n Valor calculado pelo formula fechada : %-.4f',blsprice)
fprintf('\n Valor calculado pelo algoritmo explicito sem custo : %-.4f',price)
fprintf('\n Valor calculado pelo algoritmo explicito com custo : %-.4f',priceC)
erro=(price-blsprice)/blsprice*100;
fprintf('\n Erro percentual deste calculo : %-.3f',erro)
fprintf('\n -----')
fprintf('\n Tempo do calculo (com os graficos) : %-.2f [seg]',elapsedTime)
fprintf('\n -----')
%Valores para os dados de saida
fprintf('\n Resultados - Valores Interpolados sem custo, formula analitica e com
custo')
interpValor=[vectorInterpS' interpValue' interpCall' interpValueC']
fprintf('\n -----')
xlswrite('Call_1Asset_FDExpCost.xlsx',interpValor);

```

```

        Theta(i,j)=r*ValorC(i,j+1)-Delta(i,j)/2*(r-D)*(i-1)-Gamma(i,j)/2*Sigma^2*(i-
1)^2+Cost(i,j);
        ValorC(i,j)=ValorC(i,j+1)-dt*Theta(i,j);
    end
    %Condicao de contorno nas extremidades
    %Segunda derivada nula
    ValorC(1,j)=(5*ValorC(2,j)-4*ValorC(3,j)+ValorC(4,j))/2;
end

%Retorna o valor interpolando os dados de saida
price=interp1(vetor_S, Valor(:,1),S0,'spline');
priceC=interp1(vetor_S, ValorC(:,1),S0,'spline');
priceUpOutCall=UpOutCall_Analytic(S0,Sb,K,r,D,Sigma,T,t);

%Interpolacao para tabela de resultados
interpValue=interp1(vetor_S, Valor(:,1),vectorInterpS,'spline');
interpValueC=interp1(vetor_S, ValorC(:,1),vectorInterpS,'spline');
interpUpOutCall=UpOutCall_Analytic(vectorInterpS,Sb,K,r,D,Sigma,T,t);

%Definicao dos eixos dos graficos
vetor_j = 0:N;
xt=vetor_j.*dt;
vetor_i = 0:M;
xS=vetor_i.*dS;

%Definicao dos graficos
set(0,'DefaultFigurePosition',[100 100 900 640])
tf=16;%Tamanho do font dos titulos
af=14;%Tamanho do font dos eixos

%Geracao dos graficos
figure(1)
set(gcf,'Color',[1,1,1])%Cor do fundo da figura
subplot(2,2,1)
surf(xt,xS,Valor,'EdgeColor','None');
title({'Up&Out Call - Diferencas Finitas';'Hoggard, Whaley & Wilmott (1992) '},'fontsize',tf,'fontname','Verdana','fontangle','Italic','fontweight','b','color','k');
xlabel('Tempo','fontsize',af,'fontname','Verdana','fontangle','Italic','fontweight','b','color','k');
ylabel('Valor do Ativo ','fontsize',af,'fontname','Verdana','fontangle','Italic','fontweight','b','color','k');
zlabel('Valor da Opcao sem Custo ','fontsize',af,'fontname','Verdana','fontangle','Italic','fontweight','b','color','k');
axis([0 T 0 Sb -Sb/50 max(max(Valor))])
alpha(.5);
view(60,25);
subplot(2,2,2)
surf(xt,xS,ValorC,'EdgeColor','None');
title({'Up&Out Call - Diferencas Finitas';'Hoggard, Whaley & Wilmott (1992) '},'fontsize',tf,'fontname','Verdana','fontangle','Italic','fontweight','b','color','k');
xlabel('Tempo','fontsize',af,'fontname','Verdana','fontangle','Italic','fontweight','b','color','k');
ylabel('Valor do Ativo ','fontsize',af,'fontname','Verdana','fontangle','Italic','fontweight','b','color','k');
zlabel('Valor da Opcao com Custo ','fontsize',af,'fontname','Verdana','fontangle','Italic','fontweight','b','color','k');
axis([0 T 0 Sb -Sb/50 max(max(Valor))])
alpha(.5);
view(60,25);
subplot(2,2,3)
plot(xS,Valor(:,1),xS,ValorC(:,1),xS,Valor(:,N+1),'LineWidth',1);
axis([0 Sb -Sb/50 1.1*(Sb-K)]); grid on
title({'Up&Out Call - Diferencas Finitas';'Hoggard, Whaley & Wilmott (1992) '},'fontsize',tf,'fontname','Verdana','fontangle','Italic','fontweight','b','color','k');
xlabel('Valor do Ativo ','fontsize',af,'fontname','Verdana','fontangle','Italic','fontweight','b','color','k');
ylabel('Valor da Opcao ','fontsize',af,'fontname','Verdana','fontangle','Italic','fontweight','b','color','k');
%legend({'fontname{Verdana}\fontsize{11}\bf\itValor Up&Out Call sem custo de transacao','fontname{Verdana}\fontsize{11}\bf\itValor Up&Out Call com custo de transacao','fontname{Verdana}\fontsize{11}\bf\itPayoff da Up&Out Call',2)
legend('Valor da Up&Out Call sem Custo','Valor da Up&Out Call com Custo','Payoff da

```

C3 Opção europeia “call” com volatilidade estocástica e custos de transação

```
%Aprecamento de Opcao Europeia Call pelo metodo de Diferencas Finitas
%usando um algoritmo explicito e o modelo de Volatilidade Estocastica
%com Custo de Transacoes
%Ref. Numerical Methods applied to Option Pricing Models with Transactional
%Costs and Stochastic Volatility - Mariani, SenGupta e Sewell
%Adotados os seguintes valores de entrada
%(S0,K,r,D,T,Sigma_i,Sigma_f,Smax,dS,dt,dtc,kS,kSigma)
clear all;close all;clc;
format short;

%Dados de entrada para os calculos
S0=50;%Valor do ativo
K=50;%Strike
r=.1;%Taxa de juros livre de risco
D=.05;%Taxa de dividendos
T=1;%Tempo para o vencimento
t=0;%Inicio
Beta=.3;%Volatilidade da volatilidade
Rho=.5;%Coeficiente de correlacao entre ativo e volatilidade
Sigma_i=.2;%Volatilidade inicial
Sigma_f=.4;%Volatilidade final
Smax=100;%Valor maximo do ativo
dS=.8;%Discretizacao do valor do ativo
dSigma=.0016;%Discretizacao da volatilidade
dt=T/15000;%Discretizacao do tempo
kS=.005;%Custo proporcional ao valor do ativo
kSigma=.005;%Custo proporcional ao valor da volatilidade
dtc=T/50;%Intervalo entre operacoes de hedge

%Dados para obtencao de resultados interpolados
vectorS=[S0-18:6:S0+18];%Vetor dos valores do ativo para interpolacao
vectorSigma=[Sigma_i:(Sigma_f-Sigma_i)/6:Sigma_f];%Vetor dos valores da volatilidade
para interpolacao

ticID = tic;%Inicia o timer

%Ajuste do grid e dos incrementos
L = round(Smax/dS);
dS = Smax/L;
M = round((Sigma_f-Sigma_i)/dSigma);
dSigma = (Sigma_f-Sigma_i)/M;
N = round(T/dt);
dt = T/N;

%Inicializacao das variaveis
ValueOld = zeros(L+1,M+1);ValueOldC= zeros(L+1,M+1);
ValueNew = zeros(L+1,M+1);ValueNewC = zeros(L+1,M+1);
DeltaS = zeros(L+1,M+1);DeltaSC = zeros(L+1,M+1);
DeltaSigma = zeros(L+1,M+1);DeltaSigmaC = zeros(L+1,M+1);
CrossVannaSSigma = zeros(L+1,M+1);CrossVannaSSigmaC = zeros(L+1,M+1);
GammaS = zeros(L+1,M+1);GammaSC = zeros(L+1,M+1);
GammaSigma = zeros(L+1,M+1);GammaSigmaC = zeros(L+1,M+1);
Theta = zeros(L+1,M+1);ThetaC = zeros(L+1,M+1);

%Definicao da Condicao Inicial
S = linspace(0,Smax,L+1)';
Sigma = linspace(Sigma_i,Sigma_f,M+1)';
for j=1:M+1
    ValueOld(:,j) = max(S-K,0);
    ValueOldC(:,j) = max(S-K,0);
end

%Definicao dos graficos
set(0,'DefaultFigurePosition',[100 100 900 640])
tf=16;%Tamanho do font dos titulos
af=14;%Tamanho do font dos eixos

figure(1)
```

```

SSigmaC(i,j)^2+2*Rho*Beta*Sigma(j)^2*S(i)*GammaSigmaC(i,j)*CrossVannaSSigmaC(i,j)+Beta^
2*Sigma(j)^2*GammaSigmaC(i,j)^2);
ThetaC(i,j)=-(r-D)*S(i)*DeltaSC(i,j)-(r-D)*Sigma(j)*DeltaSigmaC(i,j)-1/2*Sigma(
j)^2*S(i)^2*GammaSC(i,j)-1/2*Beta^2*Sigma(j)^2*GammaSigmaC(i,j)-Rho*Beta*Sigma(j)^2*S(i
)*CrossVannaSSigmaC(i,j)+r*ValueOldC(i,j)+CostS(i,j)+CostSigma(i,j);
ValueNewC(i,j)=ValueOldC(i,j)-dt*ThetaC(i,j);
end
end
ValueOld=ValueNew;
ValueOldC=ValueNewC;
for j=1:M+1
%Condicao de contorno nas extremidades
%Segunda derivada nula
ValueOld(1,j)=(5*ValueOld(2,j)-4*ValueOld(3,j)+ValueOld(4,j))/2;
ValueOld(L+1,j)=(5*ValueOld(L,j)-4*ValueOld(L-1,j)+ValueOld(L-2,j))/2;
ValueOldC(1,j)=(5*ValueOldC(2,j)-4*ValueOldC(3,j)+ValueOldC(4,j))/2;
ValueOldC(L+1,j)=(5*ValueOldC(L,j)-4*ValueOldC(L-1,j)+ValueOldC(L-2,j))/2;
%Primeira derivada unitaria
%ValueOld(1,j)=(1-4*ValueOld(2,j)+ValueOld(3,j))/3;
%ValueOld(L+1,j)=(1+4*ValueOld(L,j)-ValueOld(L-1,j))/3;
%ValueOldC(1,j)=(1-4*ValueOldC(2,j)+ValueOldC(3,j))/3;
%ValueOldC(L+1,j)=(1+4*ValueOldC(L,j)-ValueOldC(L-1,j))/3;
end
end
elapsedTime = toc(ticID);%Para o timer

figure(2)
set(gcf,'Color',[1,1,1])%Cor do fundo da figura
subplot(2,2,1)
surf(Sigma,S,ValueOld,'EdgeColor','None');
title({'Call - Diferencas Finitas ','Sem custo de transacao ','fontsize',tf,'fontnam
e','Verdana','fontangle','Italic','fontweight','b','color','k'});
xlabel({'\sigma ','Volatilidade'},'fontsize',af,'fontname','Verdana','fontangle','Ital
ic','fontweight','b','color','k');
ylabel({'S ','Ativo ','fontsize',af,'fontname','Verdana','fontangle','Italic','fontwei
ght','b','color','k'});
zlabel({'V ','Valor da Call ','fontsize',af,'fontname','Verdana','fontangle','Italic',
'fontweight','b','color','k'});
axis([Sigma_i Sigma_f 0 Smax min(min(ValueOld)) max(max(ValueOld))])
alpha(.5);
view(-50,25);
subplot(2,2,2)
surf(Sigma,S(2:L),DeltaS(2:L,:), 'EdgeColor','None');
title({'\partial V/\partial S ','fontsize',tf,'fontname','Verdana','fontangle','Italic','f
ontweight','b','color','k'});
xlabel({'\sigma ','Volatilidade'},'fontsize',af,'fontname','Verdana','fontangle','Ital
ic','fontweight','b','color','k');
ylabel({'S ','Ativo ','fontsize',af,'fontname','Verdana','fontangle','Italic','fontwei
ght','b','color','k'});
zlabel({'\partial V/\partial S ','Delta Ativo ','fontsize',af,'fontname','Verdana','font
angle','Italic','fontweight','b','color','k'});
axis([Sigma_i Sigma_f 0 Smax min(min(DeltaS)) max(max(DeltaS))])
alpha(.5);
view(-50,25);
subplot(2,2,3)
surf(Sigma,S(2:L),DeltaSigma(2:L,:), 'EdgeColor','None');
title({'\partial V/\partial \sigma ','fontsize',tf,'fontname','Verdana','fontangle','Ital
ic','fontweight','b','color','k'});
xlabel({'\sigma ','Volatilidade'},'fontsize',af,'fontname','Verdana','fontangle','Ita
lic','fontweight','b','color','k');
ylabel({'S ','Ativo ','fontsize',af,'fontname','Verdana','fontangle','Italic','fontwei
ght','b','color','k'});
zlabel({'\partial V/\partial \sigma ','Delta Vol ','fontsize',af,'fontname','Verdana','
fontangle','Italic','fontweight','b','color','k'});
axis([Sigma_i Sigma_f 0 Smax min(min(DeltaSigma)) max(max(DeltaSigma))])
alpha(.5);
view(-50,25);
subplot(2,2,4)
surf(Sigma,S(2:L),CrossVannaSSigma(2:L,:), 'EdgeColor','None');
title({'\partial^2 V/\partial S \partial \sigma ','fontsize',tf,'fontname','Verdana','fonta
ngle','Italic','fontweight','b','color','k'});

```

```

figure(4)
set(gcf,'Color',[1,1,1])%Cor do fundo da figura
subplot(2,2,1)
surf(Sigma,S,ValueOldC,'EdgeColor','None');
title({'Call - Diferencas Finitas ','Com custo de transacao ','fontsize',tf,'fontname','Verdana','fontangle','Italic','fontweight','b','color','k');
xlabel({'\sigma ','Volatilidade ','fontsize',af,'fontname','Verdana','fontangle','Italic','fontweight','b','color','k');
ylabel({'S ','Ativo ','fontsize',af,'fontname','Verdana','fontangle','Italic','fontweight','b','color','k');
zlabel({'V ','Valor da Call ','fontsize',af,'fontname','Verdana','fontangle','Italic','fontweight','b','color','k');
axis([Sigma_i Sigma_f 0 Smax min(min(ValueOldC)) max(max(ValueOldC))])
alpha(.5);
view(-50,25);
subplot(2,2,2)
surf(Sigma,S(2:L),DeltaSC(2:L,:), 'EdgeColor','None');
title({'\partial V/\partial S ','fontsize',tf,'fontname','Verdana','fontangle','Italic','fontweight','b','color','k');
xlabel({'\sigma ','Volatilidade ','fontsize',af,'fontname','Verdana','fontangle','Italic','fontweight','b','color','k');
ylabel({'S ','Ativo ','fontsize',af,'fontname','Verdana','fontangle','Italic','fontweight','b','color','k');
zlabel({'\partial V/\partial S ','Delta Ativo ','fontsize',af,'fontname','Verdana','fontangle','Italic','fontweight','b','color','k');
axis([Sigma_i Sigma_f 0 Smax min(min(DeltaSC)) max(max(DeltaSC))])
alpha(.5);
view(-50,25);
subplot(2,2,3)
surf(Sigma,S(2:L),DeltaSigmaC(2:L,:), 'EdgeColor','None');
title({'\partial V/\partial \sigma ','fontsize',tf,'fontname','Verdana','fontangle','Italic','fontweight','b','color','k');
xlabel({'\sigma ','Volatilidade ','fontsize',af,'fontname','Verdana','fontangle','Italic','fontweight','b','color','k');
ylabel({'S ','Ativo ','fontsize',af,'fontname','Verdana','fontangle','Italic','fontweight','b','color','k');
zlabel({'\partial V/\partial \sigma ','Delta Vol ','fontsize',af,'fontname','Verdana','fontangle','Italic','fontweight','b','color','k');
axis([Sigma_i Sigma_f 0 Smax min(min(DeltaSigmaC)) max(max(DeltaSigmaC))])
alpha(.5);
view(-50,25);
subplot(2,2,4)
surf(Sigma,S(2:L),CrossVannaSSigmaC(2:L,:), 'EdgeColor','None');
title({'\partial^2 V/\partial S \partial \sigma ','fontsize',tf,'fontname','Verdana','fontangle','Italic','fontweight','b','color','k');
xlabel({'\sigma ','Volatilidade ','fontsize',af,'fontname','Verdana','fontangle','Italic','fontweight','b','color','k');
ylabel({'S ','Ativo ','fontsize',af,'fontname','Verdana','fontangle','Italic','fontweight','b','color','k');
zlabel({'\partial^2 V/\partial S \partial \sigma ','Cross-Vanna Ativo.Vol ','fontsize',af,'fontname','Verdana','fontangle','Italic','fontweight','b','color','k');
axis([Sigma_i Sigma_f 0 Smax min(min(CrossVannaSSigmaC)) max(max(CrossVannaSSigmaC))])
alpha(.5);
view(-50,25);

figure(5)
set(gcf,'Color',[1,1,1])%Cor do fundo da figura
subplot(2,2,1)
surf(Sigma,S(2:L),GammaSC(2:L,:), 'EdgeColor','None');
title({'\partial^2 V/\partial S^2 ','fontsize',tf,'fontname','Verdana','fontangle','Italic','fontweight','b','color','k');
xlabel({'\sigma ','Volatilidade ','fontsize',af,'fontname','Verdana','fontangle','Italic','fontweight','b','color','k');
ylabel({'S ','Ativo ','fontsize',af,'fontname','Verdana','fontangle','Italic','fontweight','b','color','k');
zlabel({'\partial^2 V/\partial S^2 ','Gamma Ativo ','fontsize',af,'fontname','Verdana','fontangle','Italic','fontweight','b','color','k');
axis([Sigma_i Sigma_f 0 Smax min(min(GammaSC)) max(max(GammaSC))])
alpha(.5);
view(-50,25);
subplot(2,2,2)

```

```

[originalS,originalSigma]=meshgrid(Sigma_i:dSigma:Sigma_f,0:dS:Smax);
%Mesh para valores a serem interpolados
[interpS,interpSigma]=meshgrid(vectorSigma,vectorS);
%Interpolacao bidimensional dentro do mesh especificado
interpValue=interp2(originalS,originalSigma,ValueOld,interpS,interpSigma);
interpValueC=interp2(originalS,originalSigma,ValueOldC,interpS,interpSigma);

%Grafico dos valores interpolados para saida
figure(7)
set(gcf,'Color',[1,1,1])%Cor do fundo da figura
surfc(interpS,interpSigma,interpValue)
alpha(.2);hold on
surfc(interpS,interpSigma,interpValueC)
alpha(.2);
title({'Opcao Call - Diferencas Finitas ','Volatilidade Estocastica e Custo de
Transacoes ','fontsize',tf,'fontname','Verdana','fontangle','Italic','fontweight','b',
'color','k'});
xlabel({'\sigma ','Volatilidade ','fontsize',af,'fontname','Verdana','fontangle','Ital
ic','fontweight','b','color','k'});
ylabel({'S ','Ativo ','fontsize',af,'fontname','Verdana','fontangle','Italic','fontwei
ght','b','color','k'});
zlabel({'Valor da Opcao sem Custo ','Valor da Opcao com Custo ','fontsize',af,'fontn
ame','Verdana','fontangle','Italic','fontweight','b','color','k'});
view(-45,30);

%Valores para os dados de saida
price_i=interp1(S,ValueOld(:,1),S0,'spline');
priceC_i=interp1(S,ValueOldC(:,1),S0,'spline');
price_f=interp1(S,ValueOld(:,M+1),S0,'spline');
priceC_f=interp1(S,ValueOldC(:,M+1),S0,'spline');
Call_i=Call_Analytic(S0,K,r,D,Sigma_i,T,t);
Call_f=Call_Analytic(S0,K,r,D,Sigma_f,T,t);

kSmin=Sigma_i/2*sqrt(pi()/2*dtc*(1-Rho^2));
kSigmamin=Beta/2*sqrt(pi()/2*dtc*(1-Rho^2));

%Mostra os valores calculados com este algoritmo vs formula fechada para
%a solucao deste problema - ref. Sewell
fprintf('\n -----')
fprintf('\n European Call - Metodo de Diferencas Finitas')
fprintf('\n Volatilidade Estocastica com Custo')
fprintf('\n -----')
fprintf('\n Valores usados para este calculo')
fprintf('\n dS      [intervalo do valor do ativo]      : %-.2f',dS)
fprintf('\n L      [numero de intervalos do ativo]      : %-.0f',L)
fprintf('\n dSigma [intervalo do valor da volatilidade]      : %-.4f',dSigma)
fprintf('\n M      [numero de intervalos da volatilidade]      : %-.0f',M)
fprintf('\n dt      [intervalo de tempo de integracao]      : %-.6f',dt)
fprintf('\n N      [numero de intervalos de tempo]      : %-.0f',N)
fprintf('\n S0      [valor do ativo]      : %-.2f',S0)
fprintf('\n K      [strike]      : %-.2f',K)
fprintf('\n Smax    [valor maximo do ativo]      : %-.2f',Smax)
fprintf('\n T      [tempo para o vencimento]      : %-.2f',T)
fprintf('\n r      [taxa de juros livre de risco]      : %-.2f',r)
fprintf('\n D      [taxa de dividendo]      : %-.2f',D)
fprintf('\n Beta    [volatilidade da volatilidade]      : %-.2f',Beta)
fprintf('\n Rho     [correlacao ativo-volatilidade]      : %-.2f',Rho)
fprintf('\n Sigma_i [volatilidade inicial]      : %-.2f',Sigma_i)
fprintf('\n Sigma_f [volatilidade final]      : %-.2f',Sigma_f)
fprintf('\n -----')
fprintf('\n Valores de custo para o modelo estocastico')
fprintf('\n dtc     [intervalo de tempo de hedge]      : %-.4f',dtc)
fprintf('\n kS      [custo proporcional ao ativo]      : %-.4f',kS)
fprintf('\n kSigma  [custo proporcional a volatilidade]      : %-.4f',kSigma)
fprintf('\n -----')
fprintf('\n Valor Black-Scholes vol deterministica - Sigma Min : %-.4f',Call_i)
fprintf('\n Valor Black-Scholes vol deterministica - Sigma Max : %-.4f',Call_f)
fprintf('\n Valor calculado vol estocastica sem custo - Sigma Min : %-.4f',price_i)
fprintf('\n Valor calculado vol estocastica sem custo - Sigma Max : %-.4f',price_f)
fprintf('\n Valor calculado vol estocastica com custo - Sigma Min : %-.4f',priceC_i)
fprintf('\n Valor calculado vol estocastica com custo - Sigma Max : %-.4f',priceC_f)
fprintf('\n -----')

```

C4 Opção europeia com barreira “up&out call” com volatilidade estocástica e custos de transação

```
%Aprecamento de Opcao Europeia Up&Out Call pelo metodo de Diferencas Finitas
%usando um algoritmo explicito e o modelo de Volatilidade Estocastica
%com Custo de Transacoes
%Ref. Numerical Methods applied to Option Pricing Models with Transactional
%Costs and Stochastic Volatility - Mariani, SenGupta e Sewell
%Adotados os seguintes valores de entrada
%(S0,K,r,D,T,Sigma_i,Sigma_f,Sb,dS,dt,dtc,kS,kSigma)
clear all;close all;clc;
format short;

%Dados de entrada para os calculos
S0=50;%Valor do ativo
K=50;%Strike
r=.1;%Taxa de juros livre de risco
D=.05;%Taxa de dividendos
T=1;%Tempo para o vencimento
t=0;%Inicio
Beta=.3;%Volatilidade da volatilidade
Rho=.5;%Coeficiente de correlacao entre ativo e volatilidade
Sigma_i=.2;%Volatilidade inicial
Sigma_f=.4;%Volatilidade final
Sb=85;%Valor da barreira do ativo
dS=.68;%Discretizacao do valor do ativo
dSigma=.0016;%Discretizacao da volatilidade
dt=T/15000;%Discretizacao do tempo
kS=.005;%Custo proporcional ao valor do ativo
kSigma=.005;%Custo proporcional ao valor da volatilidade
dtc=T/50;%Intervalo entre operacoes de hedge

%Dados para obtencao de resultados interpolados
vectorS=[S0-18:6:S0+18];%Vetor dos valores do ativo para interpolacao
vectorSigma=[Sigma_i:(Sigma_f-Sigma_i)/6:Sigma_f];%Vetor dos valores da volatilidade
para interpolacao

ticID = tic;%Inicia o timer

%Ajuste do grid e dos incrementos
L = round(Sb/dS);
dS = Sb/L;
M = round((Sigma_f-Sigma_i)/dSigma);
dSigma = (Sigma_f-Sigma_i)/M;
N = round(T/dt);
dt = T/N;

%Inicializacao das variaveis
ValueOld = zeros(L+1,M+1);ValueOldC= zeros(L+1,M+1);
ValueNew = zeros(L+1,M+1);ValueNewC = zeros(L+1,M+1);
DeltaS = zeros(L+1,M+1);DeltaSC = zeros(L+1,M+1);
DeltaSigma = zeros(L+1,M+1);DeltaSigmaC = zeros(L+1,M+1);
CrossVannaSSigma = zeros(L+1,M+1);CrossVannaSSigmaC = zeros(L+1,M+1);
GammaS = zeros(L+1,M+1);GammaSC = zeros(L+1,M+1);
GammaSigma = zeros(L+1,M+1);GammaSigmaC = zeros(L+1,M+1);
Theta = zeros(L+1,M+1);ThetaC = zeros(L+1,M+1);

%Definicao da Condicao Inicial
S = linspace(0,Sb,L+1)';S(L+1)=0;
Sigma = linspace(Sigma_i,Sigma_f,M+1)';
for j=1:M+1
    ValueOld(:,j) = max(S-K,0);
    ValueOldC(:,j) = max(S-K,0);
end

%Definicao dos graficos
set(0,'DefaultFigurePosition',[100 100 900 640])
tf=16;%Tamanho do font dos titulos
af=14;%Tamanho do font dos eixos
S(L+1)=Sb;%Para acertar a interpolacao dos resultados fazendo S monotonic
```

```

SSigmaC(i,j)^2+2*Rho*Beta*Sigma(j)^2*S(i)*GammaSigmaC(i,j)*CrossVannaSSigmaC(i,j)+Beta^
2*Sigma(j)^2*GammaSigmaC(i,j)^2;
ThetaC(i,j)=-(r-D)*S(i)*DeltaSC(i,j)-(r-D)*Sigma(j)*DeltaSigmaC(i,j)-1/2*Sigma(
j)^2*S(i)^2*GammaSC(i,j)-1/2*Beta^2*Sigma(j)^2*GammaSigmaC(i,j)-Rho*Beta*Sigma(j)^2*S(i
)*CrossVannaSSigmaC(i,j)+r*ValueOldC(i,j)+CostS(i,j)+CostSigma(i,j);
ValueNewC(i,j)=ValueOldC(i,j)-dt*ThetaC(i,j);
end
end
ValueOld=ValueNew;
ValueOldC=ValueNewC;
for j=1:M+1
%Condicao de contorno nas extremidades
%Segunda derivada nula
ValueOld(1,j)=(5*ValueOld(2,j)-4*ValueOld(3,j)+ValueOld(4,j))/2;
%Up&OutCall NAO - ValueOld(L+1,j)=(5*ValueOld(L,j)-4*ValueOld(L-1,j)+ValueOld(L-
2,j))/2;
ValueOldC(1,j)=(5*ValueOldC(2,j)-4*ValueOldC(3,j)+ValueOldC(4,j))/2;
%Up&OutCall NAO - ValueOldC(L+1,j)=(5*ValueOldC(L,j)-4*ValueOldC(L-
1,j)+ValueOldC(L-2,j))/2;
%Primeira derivada unitaria
%ValueOld(1,j)=(1-4*ValueOld(2,j)+ValueOld(3,j))/-3;
%ValueOld(L+1,j)=(1+4*ValueOld(L,j)-ValueOld(L-1,j))/3;
%ValueOldC(1,j)=(1-4*ValueOldC(2,j)+ValueOldC(3,j))/-3;
%ValueOldC(L+1,j)=(1+4*ValueOldC(L,j)-ValueOldC(L-1,j))/3;
end
end

elapsedTime = toc(ticID);%Para o timer

figure(2)
set(gcf,'Color',[1,1,1])%Cor do fundo da figura
subplot(2,2,1)
surf(Sigma,S,ValueOld,'EdgeColor','None');
title({'Up&Out Call - Diferencas Finitas ','Sem custo de transacao ','fontsize',tf,'
fontname','Verdana','fontangle','Italic','fontweight','b','color','k'});
xlabel({'\sigma ','Volatilidade ','fontsize',af,'fontname','Verdana','fontangle','Ital
ic','fontweight','b','color','k'});
ylabel({'S ','Ativo ','fontsize',af,'fontname','Verdana','fontangle','Italic','fontwei
ght','b','color','k'});
zlabel({'V ','Valor da Up&Out Call ','fontsize',af,'fontname','Verdana','fontangle','I
talic','fontweight','b','color','k'});
axis([Sigma_i Sigma_f 0 Sb min(min(ValueOld)) max(max(ValueOld))])
alpha(.5);
view(-50,25);
subplot(2,2,2)
surf(Sigma,S(2:L,:),DeltaS(2:L,:), 'EdgeColor','None');
title({'\partial V/\partial \sigma ','fontsize',tf,'fontname','Verdana','fontangle','Italic','f
ontweight','b','color','k'});
xlabel({'\sigma ','Volatilidade ','fontsize',af,'fontname','Verdana','fontangle','Ital
ic','fontweight','b','color','k'});
ylabel({'S ','Ativo ','fontsize',af,'fontname','Verdana','fontangle','Italic','fontwei
ght','b','color','k'});
zlabel({'\partial V/\partial \sigma ','Delta Ativo ','fontsize',af,'fontname','Verdana','font
angle','Italic','fontweight','b','color','k'});
axis([Sigma_i Sigma_f 0 Sb min(min(DeltaS)) max(max(DeltaS))])
alpha(.5);
view(-50,25);
subplot(2,2,3)
surf(Sigma,S(2:L,:),DeltaSigma(2:L,:), 'EdgeColor','None');
title({'\partial V/\partial \sigma ','fontsize',tf,'fontname','Verdana','fontangle','Itali
c','fontweight','b','color','k'});
xlabel({'\sigma ','Volatilidade ','fontsize',af,'fontname','Verdana','fontangle','Ital
ic','fontweight','b','color','k'});
ylabel({'S ','Ativo ','fontsize',af,'fontname','Verdana','fontangle','Italic','fontwei
ght','b','color','k'});
zlabel({'\partial V/\partial \sigma ','Delta Vol ','fontsize',af,'fontname','Verdana','f
ontangle','Italic','fontweight','b','color','k'});
axis([Sigma_i Sigma_f 0 Sb min(min(DeltaSigma)) max(max(DeltaSigma))])
alpha(.5);
view(-50,25);
subplot(2,2,4)
surf(Sigma,S(2:L,:),CrossVannaSSigma(2:L,:), 'EdgeColor','None');

```

```

%axis([Sigma_i Sigma_f min(min(ValueOldC)) max(max(ValueOldC))])
hold off

figure(4)
set(gcf,'Color',[1,1,1])%Cor do fundo da figura
subplot(2,2,1)
surfc(Sigma,S,ValueOldC,'EdgeColor','None');
title({'Up&Out Call - Diferencas Finitas ','Com custo de transacao ','fontsize',tf,'fontname','Verdana','fontangle','Italic','fontweight','b','color','k');
xlabel({'\sigma ','Volatilidade ','fontsize',af,'fontname','Verdana','fontangle','Italic','fontweight','b','color','k');
ylabel({'S ','Ativo ','fontsize',af,'fontname','Verdana','fontangle','Italic','fontweight','b','color','k');
zlabel({'V ','Valor Up&Out Call ','fontsize',af,'fontname','Verdana','fontangle','Italic','fontweight','b','color','k');
axis([Sigma_i Sigma_f 0 Sb min(min(ValueOldC)) max(max(ValueOldC))])
alpha(.5);
view(-50,25);
subplot(2,2,2)
surfc(Sigma,S(2:L),DeltaSC(2:L,:), 'EdgeColor','None');
title({'\partial V/\partial S ','fontsize',tf,'fontname','Verdana','fontangle','Italic','fontweight','b','color','k');
xlabel({'\sigma ','Volatilidade ','fontsize',af,'fontname','Verdana','fontangle','Italic','fontweight','b','color','k');
ylabel({'S ','Ativo ','fontsize',af,'fontname','Verdana','fontangle','Italic','fontweight','b','color','k');
zlabel({'\partial V/\partial S ','Delta Ativo ','fontsize',af,'fontname','Verdana','fontangle','Italic','fontweight','b','color','k');
axis([Sigma_i Sigma_f 0 Sb min(min(DeltaSC)) max(max(DeltaSC))])
alpha(.5);
view(-50,25);
subplot(2,2,3)
surfc(Sigma,S(2:L),DeltaSigmaC(2:L,:), 'EdgeColor','None');
title({'\partial V/\partial \sigma ','fontsize',tf,'fontname','Verdana','fontangle','Italic','fontweight','b','color','k');
xlabel({'\sigma ','Volatilidade ','fontsize',af,'fontname','Verdana','fontangle','Italic','fontweight','b','color','k');
ylabel({'S ','Ativo ','fontsize',af,'fontname','Verdana','fontangle','Italic','fontweight','b','color','k');
zlabel({'\partial V/\partial \sigma ','Delta Vol ','fontsize',af,'fontname','Verdana','fontangle','Italic','fontweight','b','color','k');
axis([Sigma_i Sigma_f 0 Sb min(min(DeltaSigmaC)) max(max(DeltaSigmaC))])
alpha(.5);
view(-50,25);
subplot(2,2,4)
surfc(Sigma,S(2:L),CrossVannaSSigmaC(2:L,:), 'EdgeColor','None');
title({'\partial^2 V/\partial S \partial \sigma ','fontsize',tf,'fontname','Verdana','fontangle','Italic','fontweight','b','color','k');
xlabel({'\sigma ','Volatilidade ','fontsize',af,'fontname','Verdana','fontangle','Italic','fontweight','b','color','k');
ylabel({'S ','Ativo ','fontsize',af,'fontname','Verdana','fontangle','Italic','fontweight','b','color','k');
zlabel({'\partial^2 V/\partial S \partial \sigma ','Cross-Vanna Ativo.Vol ','fontsize',af,'fontname','Verdana','fontangle','Italic','fontweight','b','color','k');
axis([Sigma_i Sigma_f 0 Sb min(min(CrossVannaSSigmaC)) max(max(CrossVannaSSigmaC))])
alpha(.5);
view(-50,25);

figure(5)
set(gcf,'Color',[1,1,1])%Cor do fundo da figura
subplot(2,2,1)
surfc(Sigma,S(2:L),GammaSC(2:L,:), 'EdgeColor','None');
title({'\partial^2 V/\partial S^2 ','fontsize',tf,'fontname','Verdana','fontangle','Italic','fontweight','b','color','k');
xlabel({'\sigma ','Volatilidade ','fontsize',af,'fontname','Verdana','fontangle','Italic','fontweight','b','color','k');
ylabel({'S ','Ativo ','fontsize',af,'fontname','Verdana','fontangle','Italic','fontweight','b','color','k');
zlabel({'\partial^2 V/\partial S^2 ','Gamma Ativo ','fontsize',af,'fontname','Verdana','fontangle','Italic','fontweight','b','color','k');
axis([Sigma_i Sigma_f 0 Sb min(min(GammaSC)) max(max(GammaSC))])
alpha(.5);

```

```

%Retorna o valor interpolando os dados de saida
%Mesh para valores originais
[originalS,originalSigma]=meshgrid(Sigma_i:dSigma:Sigma_f,0:dS:Sb);
%Mesh para valores interpolados
[interpS,interpSigma]=meshgrid(vectorSigma,vectorS);
%Interpolacao
interpValue=interp2(originalS,originalSigma,ValueOld,interpS,interpSigma);
interpValueC=interp2(originalS,originalSigma,ValueOldC,interpS,interpSigma);

%Grafico dos valores interpolados para saida
figure(7)
set(gcf,'Color',[1,1,1])%Cor do fundo da figura
surf(interpS,interpSigma,interpValue)
alpha(.2);hold on
surf(interpS,interpSigma,interpValueC)
alpha(.2);
title({'Opcao Up&Out Call - Diferencas Finitas ','Volatilidade Estocastica e Custo de Transacoes ','fontSize',tf,'fontName','Verdana','fontangle','Italic','fontweight','b','color','k');
xlabel({'\sigma ','Volatilidade ','fontSize',af,'fontName','Verdana','fontangle','Italic','fontweight','b','color','k');
ylabel({'S ','Ativo ','fontSize',af,'fontName','Verdana','fontangle','Italic','fontweight','b','color','k');
zlabel({'Valor da Opcao sem Custo ','Valor da Opcao com Custo ','fontSize',af,'fontName','Verdana','fontangle','Italic','fontweight','b','color','k');
view(-45,30);

%Retorna o valor interpolando os dados de saida
price_i=interp1(S,ValueOld(:,1),S0,'spline');
priceC_i=interp1(S,ValueOldC(:,1),S0,'spline');
price_f=interp1(S,ValueOld(:,M+1),S0,'spline');
priceC_f=interp1(S,ValueOldC(:,M+1),S0,'spline');
UpOutCall_i=UpOutCall_Analytic(S0,Sb,K,r,D,Sigma_i,T,t);
UpOutCall_f=UpOutCall_Analytic(S0,Sb,K,r,D,Sigma_f,T,t);

%Mostra os valores calculados com este algoritmo vs formula fechada para
%a solucao deste problema - ref. Sewell
fprintf('\n -----')
fprintf('\n Up&Out Call - Metodo de Diferencas Finitas')
fprintf('\n Volatilidade Estocastica com Custo de Transacao')
fprintf('\n -----')
fprintf('\n Valores usados para este calculo')
fprintf('\n dS      [intervalo do valor do ativo]      : %-.2f',dS)
fprintf('\n L       [numero de intervalos do ativo]       : %-.0f',L)
fprintf('\n dSigma  [intervalo do valor da volatilidade]  : %-.4f',dSigma)
fprintf('\n M       [numero de intervalos da volatilidade]: %-.0f',M)
fprintf('\n dt      [intervalo de tempo de integracao]    : %-.6f',dt)
fprintf('\n N       [numero de intervalos de tempo]       : %-.0f',N)
fprintf('\n S0      [valor do ativo]                      : %-.2f',S0)
fprintf('\n K       [strike]                             : %-.2f',K)
fprintf('\n Sb      [valor da barreira do ativo]          : %-.2f',Sb)
fprintf('\n T       [tempo para o vencimento]             : %-.2f',T)
fprintf('\n r       [taxa de juros livre de risco]        : %-.2f',r)
fprintf('\n D       [taxa de dividendo]                  : %-.2f',D)
fprintf('\n Beta    [volatilidade da volatilidade]        : %-.2f',Beta)
fprintf('\n Rho     [correlacao ativo-volatilidade]       : %-.2f',Rho)
fprintf('\n Sigma_i [volatilidade inicial]                : %-.2f',Sigma_i)
fprintf('\n Sigma_f [volatilidade final]                  : %-.2f',Sigma_f)
fprintf('\n -----')
fprintf('\n Valores de custo para o modelo estocastico')
fprintf('\n dtc     [intervalo de tempo de hedge]         : %-.4f',dtc)
fprintf('\n kS      [custo proporcional ao ativo]         : %-.4f',kS)
fprintf('\n kSigma  [custo proporcional a volatilidade]    : %-.4f',kSigma)
fprintf('\n -----')
fprintf('\n Valor formula fechada vol deterministica - Sigma Min : %-.4f',UpOutCall_i)
fprintf('\n Valor formula fechada vol deterministica - Sigma Max :

```

C5 Simulação de Monte Carlo para opção europeia "call" com volatilidade estocástica - Heston - Milstein

```
%Simulacao de Monte Carlo para uma Opcao de Compra Europeia - Call
%Modelo de Heston com Volatilidade Estocastica
%Metodo de Milstein com variaveis antiteticas
clear all;close all;clc;
format short;

S0=50;%Preco na data atual
K=50;%Preco de exercicio (strike)
r=.1;%Taxa de juros livre de risco
D=.05;%Taxa continua de dividendos pagos pelo ativo base
T=1;%Prazo de vencimento
beta=.3;%Volatilidade da volatilidade
rho=.5;%Correlacao ativo volatilidade
kappa=4;%Velocidade de reversao a media

vS=18;%Variacao do valor do ativo para as simulacoes
sigma_i=.2;%Volatilidade media de longo prazo inicial para as simulacoes
sigma_f=.4;%Volatilidade media de longo prazo final para as simulacoes
vectorS=linspace(S0-vS,S0+vS,7);%Vetor dos valores do ativo
vectorTheta=linspace(sigma_i,sigma_f,7).^2;%Variancia media de longo prazo

nInt=200;%Numero de intervalos de tempo
nTraj=1000;%Numero de caminhos
nSimul=5;%Numero de simulacoes
dt=T/nInt;%Intervalo de tempo dt para cada trajetoria
ticID = tic;%Inicia o timer

for dS=1:7
    for dSigma=1:7
        Va=zeros(nInt,nTraj);Vb=Va;
        Sa=zeros(nInt,nTraj);Sb=Sa;
        Va(1,:)=vectorTheta(dSigma);
        Vb(1,:)=vectorTheta(dSigma);
        Sa(1,:)=vectorS(dS);
        Sb(1,:)=vectorS(dS);
        for k=1:nSimul
            Xv=rand(nInt,nTraj);
            Xs=rand(nInt,nTraj);
            Zva=norminv(Xv,0,1);
            Zvb=norminv(1-Xv,0,1);
            Zsa=rho*Zva+sqrt(1-rho^2)*norminv(Xs,0,1);
            Zsb=rho*Zvb+sqrt(1-rho^2)*norminv(1-Xs,0,1);
            for j=1:nTraj
                for i=2:nInt
                    Va(i,j)=Va(i-1,j)+kappa*(vectorTheta(dSigma)-Va(i-1,j))*dt+beta*sqrt(Va(i-1,j)*dt)*Zva(i,j)+beta^2/4*dt*(Zva(i,j)^2-1);
                    if Va(i,j)<0
                        Va(i,j)=0;%Tratamento para volatilidade negativa
                    end
                    Vb(i,j)=Vb(i-1,j)+kappa*(vectorTheta(dSigma)-Vb(i-1,j))*dt+beta*sqrt(Vb(i-1,j)*dt)*Zvb(i,j)+beta^2/4*dt*(Zvb(i,j)^2-1);
                    if Vb(i,j)<0
                        Vb(i,j)=0;%Tratamento para volatilidade negativa
                    end
                    Sa(i,j)=Sa(i-1,j)*((1+(r-D)*dt)+sqrt(Va(i-1,j)*dt)*Zsa(i,j)+Va(i-1,j)/2*dt*(Zsa(i,j)^2-1));
                    Sb(i,j)=Sb(i-1,j)*((1+(r-D)*dt)+sqrt(Vb(i-1,j)*dt)*Zsb(i,j)+Vb(i-1,j)/2*dt*(Zsb(i,j)^2-1));
                end
            end
            Vab=[Va Vb];
            Sab=[Sa Sb];
            Vm(k)=mean(Vab(nInt,:));
            Sm(k)=mean(Sab(nInt,:));
            k
        end
        Vmean(dS,dSigma)=mean(Vm);
        Smean(dS,dSigma)=mean(Sm);
    end
end
```

```

set(gcf,'Color',[1,1,1])%Cor do fundo da figura
subplot(2,2,1)
normplot(log(Sms))
title({'log S (ultima simulacao) ','Distribuicao dos valores do ativo ','fontsize',tf,
'fontname','Verdana','fontangle','Italic','fontweight','b','color','k'});
xlabel({'log S ','log do ativo ','fontsize',af,'fontname','Verdana','fontangle','Itali
c','fontweight','b','color','k'});
ylabel({'percentil log S','Valor final do ativo ','fontsize',af,'fontname','Verdana','f
ontangle','Italic','fontweight','b','color','k'});
grid on
subplot(2,2,2)
hist(Sms,100)
xlabel({'Valor do ativo ','(ultima simulacao) ','fontsize',af,'fontname','Verdana','f
ontangle','Italic','fontweight','b','color','k'});
ylabel({'Frequencia para ativo ','fontsize',af,'fontname','Verdana','fontangle','Italic
','fontweight','b','color','k'});
title('Distribuicao do valor do ativo - S ','fontsize',tf,'fontname','Verdana','fontang
le','Italic','fontweight','b','color','k');
grid on
subplot(2,2,3)
plot(Sab)
xlabel('Tempo ','fontsize',af,'fontname','Verdana','fontangle','Italic','fontweight','b
','color','k');
ylabel({'Valor do ativo ','(ultima simulacao) ','fontsize',af,'fontname','Verdana','f
ontangle','Italic','fontweight','b','color','k'});
title('Trajetorias simuladas do ativo - S ','fontsize',tf,'fontname','Verdana','fontang
le','Italic','fontweight','b','color','k');
grid on

fprintf('\n -----')
fprintf('\n European Call')
fprintf('\n Volatilidade Estocastica sem Custo - Simulacao de Monte Carlo')
fprintf('\n Modelo de Heston - Metodo de Milstein com variaveis antiteticas')
fprintf('\n -----')
fprintf('\n Valores usados para este calculo')
fprintf('\n M      [numero de intervalos de cada trajetoria]      : %-.0f',nInt)
fprintf('\n N      [numero de trajetorias em cada simulacao]       : %-.0f',nTraj)
fprintf('\n O      [numero de simulacoes]                             : %-.0f',nSimul)
fprintf('\n dt     [intervalo de tempo de cada trajetoria]             : %-.2f',dt)
fprintf('\n S0     [valor inicial do ativo]                             : %-.2f',S0)
fprintf('\n K      [strike]                                              : %-.2f',K)
fprintf('\n K      [strike]                                              : %-.2f',K)
fprintf('\n T      [tempo para o vencimento]                             : %-.2f',T)
fprintf('\n r      [taxa de juros livre de risco]                       : %-.2f',r)
fprintf('\n D      [taxa de dividendo]                                   : %-.2f',D)
fprintf('\n Beta   [volatilidade da volatilidade]                      : %-.2f',beta)
fprintf('\n Rho    [correlacao ativo-volatilidade]                     : %-.2f',rho)
fprintf('\n Sigma_i [volatilidade media longo prazo inicial]           : %-.2f',sigma_i)
fprintf('\n Sigma_f [volatilidade media longo prazo final]             : %-.2f',sigma_f)
fprintf('\n Kappa  [velocidade de reversao a media]                     : %-.2f',kappa)
fprintf('\n -----')
fprintf('\n Tempo do calculo                %-.2f [seg]',elapsedTime)
fprintf('\n -----')

valorOpcao=[0 sqrt(vectorTheta);vectorS' STm]
stdv_valorOpcao=[0 sqrt(vectorTheta);vectorS' stdv_STm]
valorVar=[0 sqrt(vectorTheta);vectorS' Vmean]
valorAtivo=[0 sqrt(vectorTheta);vectorS' Smean]
xlswrite('Call_1Asset_MCMilstein_StochVolB.xlsx',[STm;Vmean;Smean]);

```

```

        end
    end
    un=rand(nInt,2*nTraj);
    for j=1:2*nTraj
        for i=2:nInt
            if (Sab(i,j)>Sbar | P(i,j)>un(i,j))
                Sab(:,j)=0;
            end
        end
    end
    Vm(k)=mean(Vab(nInt,:));
    Sm(k)=mean(Sab(nInt,:));
    k
    Bridge=sum(sum(P>un))/nInt/nTraj;%percentual de barreira pela Brownian Bridge
end
Vmean(dS,dSigma)=mean(Vm);
Smean(dS,dSigma)=mean(Sm);
ST=Sab(nInt,1:2*nTraj)-K;
for i=1:2*nTraj
    if ST(i)<0
        ST(i)=0;
    end
end
STm(dS,dSigma)=exp(-r*T)*mean(ST);
stdv_STm(dS,dSigma)=std(ST);
end

elapsedTime = toc(ticID)%Para o timer

%Graficos do ultimo calculo apenas
Vms=sort(Vab(nInt,:));
Sms=sort(Sab(nInt,:));

nZero=1;
while Sms(nZero)==0
    nZero=nZero+1;
end

%Definicao dos graficos
set(0,'DefaultFigurePosition',[100 100 900 640])
tf=16;%Tamanho do font dos titulos
af=14;%Tamanho do font dos eixos

figure(1)
set(gcf,'Color',[1,1,1])%Cor do fundo da figura
surf(sqrt(vectorTheta),vectorS,STm);
title({'Opcao Up&Out Call - Simulacao de Monte Carlo ','Modelo de Heston - Metodo de
Milstein - Variaveis Antiteticas ','fontname','Verdana','fontangle','Ita
lic','fontweight','b','color','k');
xlabel({'\sigma ','Volatilidade ','fontname','Verdana','fontangle','Ital
ic','fontweight','b','color','k');
ylabel({'S ','Ativo ','fontname','Verdana','fontangle','Italic','fontwei
ght','b','color','k');
zlabel({'Opcao Up&Out Call ','Valor da opcao sem custo ','fontname','V
erdana','fontangle','Italic','fontweight','b','color','k');
%axis([min(sqrt(vectorTheta)) max(sqrt(vectorTheta)) min(vectorS) max(vectorS) 0
ceil(max(max(STm)))])
alpha(.25);
view(-45,30);

%Graficos relativos a ultima simulacao
figure(2)
set(gcf,'Color',[1,1,1])%Cor do fundo da figura
subplot(2,2,1)
normplot(log(Vms))%Normplot do logaritmo dos valores finais da volatilidade da ultima
simulacao
title({'log V (ultima simulacao) ','Distribuicao dos valores da volatilidade ','fontsi
ze',tf,'fontname','Verdana','fontangle','Italic','fontweight','b','color','k');
xlabel({'log V ','log da volatilidade ','fontname','Verdana','fontangle',
'','Italic','fontweight','b','color','k');
ylabel({'percentil log V ','Valor final da volatilidade ','fontname','Ver

```

```
fprintf('\n -----')
fprintf('\n Tempo do calculo          %-.2f [seg]',elapsedTime)
fprintf('\n -----')

valorOpcao=[0 sqrt(vectorTheta);vectorS' STm]
stdv_valorOpcao=[0 sqrt(vectorTheta);vectorS' stdv_STm]
valorVar=[0 sqrt(vectorTheta);vectorS' Vmean]
valorAtivo=[0 sqrt(vectorTheta);vectorS' Smean]
xlswrite('UpOutCall_1Asset_MCMilstein_StochVolB.xlsx',[STm;Vmean;Smean]);
```

```

figure(1)
set(gcf,'Color',[1,1,1])%Cor do fundo da figura
surf(S2,S1,ValueOld,'EdgeColor','None');
title({'Opcao Basket Call - Diferencas Finitas ','Condicao Inicial ','fontsize',tf,'fontname','Verdana','fontangle','Italic','fontweight','b','color','k');
xlabel({'S2 ','Ativo 2 ','fontsize',af,'fontname','Verdana','fontangle','Italic','fontweight','b','color','k');
ylabel({'S1 ','Ativo 1 ','fontsize',af,'fontname','Verdana','fontangle','Italic','fontweight','b','color','k');
zlabel({'Payoff ','Condicao Inicial ','fontsize',af,'fontname','Verdana','fontangle','Italic','fontweight','b','color','k');
alpha(.5);
view(-45,30);

%Calculo do valor da opcao com e sem custo de transacao
for k=N:-1:1%Tempo
    for i=2:L%Direcao do ativo 2
        for j=2:M%Direcao do tivo 1
            %Calculo da Gregas para integracao explicita
            DeltaS1(i,j)=(ValueOld(i+1,j)-ValueOld(i-1,j))/2/ds1;
            DeltaS2(i,j)=(ValueOld(i,j+1)-ValueOld(i,j-1))/2/ds2;
            CrossGammaS1S2(i,j)=(ValueOld(i+1,j+1)-ValueOld(i+1,j-1)-ValueOld(i-1,j+1)+ValueOld(i-1,j-1))/4/ds1/ds2;
            GammaS1(i,j)=(ValueOld(i+1,j)-2*ValueOld(i,j)+ValueOld(i-1,j))/ds1^2;
            GammaS2(i,j)=(ValueOld(i,j+1)-2*ValueOld(i,j)+ValueOld(i,j-1))/ds2^2;
            DeltaS1C(i,j)=(ValueOldC(i+1,j)-ValueOldC(i-1,j))/2/ds1;
            DeltaS2C(i,j)=(ValueOldC(i,j+1)-ValueOldC(i,j-1))/2/ds2;
            CrossGammaS1S2C(i,j)=(ValueOldC(i+1,j+1)-ValueOldC(i+1,j-1)-ValueOldC(i-1,j+1)+ValueOldC(i-1,j-1))/4/ds1/ds2;
            GammaS1C(i,j)=(ValueOldC(i+1,j)-2*ValueOldC(i,j)+ValueOldC(i-1,j))/ds1^2;
            GammaS2C(i,j)=(ValueOldC(i,j+1)-2*ValueOldC(i,j)+ValueOldC(i,j-1))/ds2^2;
        end
    end
%Integracao no tempo
    for i=2:L
        for j=2:M
            Theta(i,j)=-(r-D1)*S1(i)*DeltaS1(i,j)-(r-D2)*S2(j)*DeltaS2(i,j)-1/2*Sigma1^2*S1(i)^2*GammaS1(i,j)-1/2*Sigma2^2*S2(j)^2*GammaS2(i,j)-Rho*Sigma1*Sigma2*S1(i)*S2(j)*CrossGammaS1S2(i,j)+r*ValueOld(i,j);
            ValueNew(i,j)=ValueOld(i,j)-dt*Theta(i,j);
            CostS1(i,j)=kS1*S1(i)*sqrt(2/pi/dtc)*sqrt(Sigma1^2*S1(i)^2*GammaS1C(i,j)+2*Rho*Sigma1*Sigma2*S1(i)*S2(j)*GammaS1C(i,j)*CrossGammaS1S2C(i,j)+Sigma2^2*S2(j)^2*CrossGammaS1S2C(i,j)^2);
            CostS2(i,j)=kS2*S2(j)*sqrt(2/pi/dtc)*sqrt(Sigma1^2*S1(i)^2*CrossGammaS1S2C(i,j)+2*Rho*Sigma1*Sigma2*S1(i)*S2(j)*GammaS2C(i,j)*CrossGammaS1S2C(i,j)+Sigma2^2*S2(j)^2*GammaS2C(i,j)^2);
            ThetaC(i,j)=-(r-D1)*S1(i)*DeltaS1C(i,j)-(r-D2)*S2(j)*DeltaS2C(i,j)-1/2*Sigma1^2*S1(i)^2*GammaS1C(i,j)-1/2*Sigma2^2*S2(j)^2*GammaS2C(i,j)-Rho*Sigma1*Sigma2*S1(i)*S2(j)*CrossGammaS1S2C(i,j)+r*ValueOldC(i,j)+CostS1(i,j)+CostS2(i,j);
            ValueNewC(i,j)=ValueOldC(i,j)-dt*ThetaC(i,j);
        end
    end
ValueOld=ValueNew;
ValueOldC=ValueNewC;
    for j=2:M
        %Condicao de contorno nas extremidades de S1 - Segunda derivada nula
        ValueOld(1,j)=(5*ValueOld(2,j)-4*ValueOld(3,j)+ValueOld(4,j))/2;
        ValueOld(L+1,j)=(5*ValueOld(L,j)-4*ValueOld(L-1,j)+ValueOld(L-2,j))/2;
        ValueOldC(1,j)=(5*ValueOldC(2,j)-4*ValueOldC(3,j)+ValueOldC(4,j))/2;
        ValueOldC(L+1,j)=(5*ValueOldC(L,j)-4*ValueOldC(L-1,j)+ValueOldC(L-2,j))/2;
    end
    for i=2:L
        %Condicao de contorno nas extremidades de S2 - Segunda derivada nula
        ValueOld(i,1)=(5*ValueOld(i,2)-4*ValueOld(i,3)+ValueOld(i,4))/2;
        ValueOld(i,M+1)=(5*ValueOld(i,M)-4*ValueOld(i,M-1)+ValueOld(i,M-2))/2;
        ValueOldC(i,1)=(5*ValueOldC(i,2)-4*ValueOldC(i,3)+ValueOldC(i,4))/2;
        ValueOldC(i,M+1)=(5*ValueOldC(i,M)-4*ValueOldC(i,M-1)+ValueOldC(i,M-2))/2;
    end
ValueOld(1,1)=(5*ValueOld(2,1)-4*ValueOld(3,1)+ValueOld(4,1))+(5*ValueOld(1,2)-4*ValueOld(1,3)+ValueOld(1,4))/4;
ValueOld(L+1,1)=(5*ValueOld(L,1)-4*ValueOld(L-1,1)+ValueOld(L-

```

```

subplot(2,2,1)
surfc(S2(2:M),S1(2:L),GammaS1(2:L,2:M),'EdgeColor','None');
title('\partial^2V/\partials1^2 ','fontsize',tf,'fontname','Verdana','fontangle','Italic',
'fontweight','b','color','k');
xlabel({'s2 ':'Ativo 2 '},'fontsize',af,'fontname','Verdana','fontangle','Italic','font
weight','b','color','k');
ylabel({'s1 ':'Ativo 1 '},'fontsize',af,'fontname','Verdana','fontangle','Italic','font
weight','b','color','k');
zlabel({'\partial^2V/\partials1^2 ':'Gamma Ativo 1 '},'fontsize',af,'fontname','Verdana',
'fontangle','Italic','fontweight','b','color','k');
%axis([Sigma_i Sigma_f 0 Smax min(min(SigmaS)) max(max(SigmaS))])
alpha(.5);
view(-45,25);
subplot(2,2,2)
surfc(S2(2:M),S1(2:L),GammaS2(2:L,2:M),'EdgeColor','None');
title('\partial^2V/\partials2^2 ','fontsize',tf,'fontname','Verdana','fontangle','Italic',
'fontweight','b','color','k');
xlabel({'s2 ':'Ativo 2 '},'fontsize',af,'fontname','Verdana','fontangle','Italic','fontwe
ight','b','color','k');
ylabel({'s1 ':'Ativo 1 '},'fontsize',af,'fontname','Verdana','fontangle','Italic','fontwe
ight','b','color','k');
zlabel({'\partial^2V/\partials2^2 ':'Gamma Ativo 2 '},'fontsize',af,'fontname','Verdana',
'fontangle','Italic','fontweight','b','color','k');
%axis([Sigma_i Sigma_f 0 Smax min(min(SigmaSigma)) max(max(SigmaSigma))])
alpha(.5);
view(-45,25);

figure(4)
set(gcf,'Color',[1,1,1])%Cor do fundo da figura
subplot(2,2,1)
surfc(S2,S1,ValueOldC,'EdgeColor','None');
title('Basket Call com Custo ','fontsize',tf,'fontname','Verdana','fontangle','Italic',
'fontweight','b','color','k');
xlabel({'s2 ':'Ativo 2 '},'fontsize',af,'fontname','Verdana','fontangle','Italic','font
weight','b','color','k');
ylabel({'s1 ':'Ativo 1 '},'fontsize',af,'fontname','Verdana','fontangle','Italic','font
weight','b','color','k');
zlabel({'V ':'Valor Basket Call '},'fontsize',af,'fontname','Verdana','fontangle','Ital
ic','fontweight','b','color','k');
%axis([0 S2max 0 S1max min(min(ValueOld)) max(max(ValueOld))])
alpha(.5);
view(-45,30);
subplot(2,2,2)
surfc(S2(2:M),S1(2:L),DeltaS1C(2:L,2:M),'EdgeColor','None');
title('\partialV/\partials1 ','fontsize',tf,'fontname','Verdana','fontangle','Italic',
'fontweight','b','color','k');
xlabel({'s2 ':'Ativo 2 '},'fontsize',af,'fontname','Verdana','fontangle','Italic','font
weight','b','color','k');
ylabel({'s1 ':'Ativo 1 '},'fontsize',af,'fontname','Verdana','fontangle','Italic','font
weight','b','color','k');
zlabel({'\partialV/\partials1 ':'Delta Ativo 1 '},'fontsize',af,'fontname','Verdana','f
ontangle','Italic','fontweight','b','color','k');
%axis([0 S2max 0 S1max min(min(DeltaS)) max(max(DeltaS))])
alpha(.5);
view(-45,30);
subplot(2,2,3)
surfc(S2(2:M),S1(2:L),DeltaS2C(2:L,2:M),'EdgeColor','None');
title('\partialV/\partials2 ','fontsize',tf,'fontname','Verdana','fontangle','Italic',
'fontweight','b','color','k');
xlabel({'s2 ':'Ativo 2 '},'fontsize',af,'fontname','Verdana','fontangle','Italic','font
weight','b','color','k');
ylabel({'s1 ':'Ativo 1 '},'fontsize',af,'fontname','Verdana','fontangle','Italic','font
weight','b','color','k');
zlabel({'\partialV/\partials2 ':'Delta Ativo 2 '},'fontsize',af,'fontname','Verdana','f
ontangle','Italic','fontweight','b','color','k');
%axis([Sigma_i Sigma_f 0 Smax min(min(DeltaSigma)) max(max(DeltaSigma))])
alpha(.5);
view(-45,30);
subplot(2,2,4)
surfc(S2(2:M),S1(2:L),CrossGammaS1S2C(2:L,2:M),'EdgeColor','None');
title('\partial^2V/\partials1\partials2 ','fontsize',tf,'fontname','Verdana','fontangle

```

```

fprintf('\n com 2 Ativos, Volatilidades Deterministicas, com e sem Custo de Transacao')
fprintf('\n -----')
fprintf('\n Valores usados para este calculo')
fprintf('\n dS1      [intervalo do valor do ativo 1]           : %-.2f',dS1)
fprintf('\n L        [numero de intervalos do ativo 1]          : %-.0f',L)
fprintf('\n dS2      [intervalo do valor do ativo 2]           : %-.4f',dS2)
fprintf('\n M        [numero de intervalos do ativo 2]          : %-.0f',M)
fprintf('\n dt       [intervalo de tempo de integracao]         : %-.6f',dt)
fprintf('\n N        [numero de intervalos de tempo]           : %-.0f',N)
fprintf('\n S10      [valor do ativo 1]                         : %-.2f',S10)
fprintf('\n S20      [valor do ativo 2]                         : %-.2f',S20)
fprintf('\n K        [strike]                                    : %-.2f',K)
fprintf('\n w        [percentual do ativo 1]                    : %-.2f',w)
fprintf('\n S1max     [valor maximo para ativo 1]                : %-.2f',S1max)
fprintf('\n S2max     [valor maximo para ativo 2]                : %-.2f',S2max)
fprintf('\n T        [tempo para o vencimento]                  : %-.2f',T)
fprintf('\n r        [taxa de juros livre de risco]             : %-.2f',r)
fprintf('\n D1        [taxa de dividendo do ativo 1]            : %-.2f',D1)
fprintf('\n D2        [taxa de dividendo do ativo 2]            : %-.2f',D2)
fprintf('\n Rho       [correlacao entre ativos]                 : %-.2f',Rho)
fprintf('\n Sigma1    [volatilidade do ativo 1]                 : %-.2f',Sigma1)
fprintf('\n Sigma2    [volatilidade do ativo 2]                 : %-.2f',Sigma2)
fprintf('\n -----')
fprintf('\n Valores de custo de transacao')
fprintf('\n dtc      [intervalo de tempo de hedge]              : %-.4f',dtc)
fprintf('\n kS1      [custo proporcional ao ativo 1]            : %-.4f',kS1)
fprintf('\n kS2      [custo proporcional ao ativo 2]            : %-.4f',kS2)
fprintf('\n -----')
fprintf('\n Tempo do calculo (com os graficos) : %-.2f [seg]',elapsedTime)
fprintf('\n -----')
%Valores para os dados de saida
fprintf('\n Resultados - Valores Interpolados sem custo')
Valor=[0 vectorInterpS2;vectorInterpS1' interpValue]
fprintf('\n -----')
fprintf('\n Resultados - Valores Interpolados com custo')
ValorC=[0 vectorInterpS2;vectorInterpS1' interpValueC]
fprintf('\n -----')
xlswrite('Call_2Assets_FDExpCost.xlsx',[Valor;ValorC]);

```

```

ValueOld(i,j)=0;%Valor nulo se superar a barreira (Up&Out Call)
CondInic(i,j)=nan;%Valores NaN para manutencao dos zeros apos a barreira
end
end
for i=1:L
    for j=1:M
        if (ValueOld(i+1,j)-ValueOld(i,j))<0
            ValueOld(i,j)=Sb-K;%Substitui os valores maximos pelo valor max da opcao
        end
    end
end
ValueOldC=ValueOld;%Duplica a superficie da condicao inicial para calculos com custo

%Definicao dos graficos das superficies de saida
set(0,'DefaultFigurePosition',[100 100 900 640])%Posicao do pixel inicial e tamanho dos quadros
tf=16;%Tamanho do font para os titulos dos graficos
af=14;%Tamanho do font para os eixos dos graficos

figure(1)
set(gcf,'Color',[1,1,1])%Cor do fundo da figura
surfc(S2,S1,ValueOld,'EdgeColor','None');
title({'Opcao Basket Up&Out Call - Diferencas Finitas ','Condicao Inicial ','fontsize',tf,'fontname','Verdana','fontangle','Italic','fontweight','b','color','k'});
xlabel({'S2 ','Ativo 2 ','fontsize',af,'fontname','Verdana','fontangle','Italic','fontweight','b','color','k'});
ylabel({'S1 ','Ativo 1 ','fontsize',af,'fontname','Verdana','fontangle','Italic','fontweight','b','color','k'});
zlabel({'Payoff ','Condicao Inicial ','fontsize',af,'fontname','Verdana','fontangle','Italic','fontweight','b','color','k'});
alpha(.5);
view(-45,30);

%Calculo do valor da opcao com e sem custo de transacao
for k=N:-1:1%Intervalos de tempo (de tras para frente)
    for i=2:L%Direcao do ativo 2
        for j=2:M%Direcao do tivo 1
            %Calculo da Gregas para integracao explicita
            DeltaS1(i,j)=(ValueOld(i+1,j)-ValueOld(i-1,j))/2/ds1;
            DeltaS2(i,j)=(ValueOld(i,j+1)-ValueOld(i,j-1))/2/ds2;
            CrossGammaS1S2(i,j)=(ValueOld(i+1,j+1)-ValueOld(i+1,j-1)-ValueOld(i-1,j+1)+ValueOld(i-1,j-1))/4/ds1/ds2;
            GammaS1(i,j)=(ValueOld(i+1,j)-2*ValueOld(i,j)+ValueOld(i-1,j))/ds1^2;
            GammaS2(i,j)=(ValueOld(i,j+1)-2*ValueOld(i,j)+ValueOld(i,j-1))/ds2^2;
            DeltaS1C(i,j)=(ValueOldC(i+1,j)-ValueOldC(i-1,j))/2/ds1;
            DeltaS2C(i,j)=(ValueOldC(i,j+1)-ValueOldC(i,j-1))/2/ds2;
            CrossGammaS1S2C(i,j)=(ValueOldC(i+1,j+1)-ValueOldC(i+1,j-1)-ValueOldC(i-1,j+1)+ValueOldC(i-1,j-1))/4/ds1/ds2;
            GammaS1C(i,j)=(ValueOldC(i+1,j)-2*ValueOldC(i,j)+ValueOldC(i-1,j))/ds1^2;
            GammaS2C(i,j)=(ValueOldC(i,j+1)-2*ValueOldC(i,j)+ValueOldC(i,j-1))/ds2^2;
        end
    end
end

%Integracao explicita no tempo
for i=2:L
    for j=2:M
        Theta(i,j)=-(r-D1)*S1(i)*DeltaS1(i,j)-(r-D2)*S2(j)*DeltaS2(i,j)-1/2*Sigma1^2*S1(i)^2*GammaS1(i,j)-1/2*Sigma2^2*S2(j)^2*GammaS2(i,j)-Rho*Sigma1*Sigma2*S1(i)*S2(j)*CrossGammaS1S2(i,j)+r*ValueOld(i,j);
        ValueNew(i,j)=ValueOld(i,j)-dt*Theta(i,j);
        CostS1(i,j)=k*S1(i)*sqrt(2/pi/dtc)*sqrt(Sigma1^2*S1(i)^2*GammaS1C(i,j)^2+2*Rho*Sigma1*Sigma2*S1(i)*S2(j)*GammaS1C(i,j)*CrossGammaS1S2C(i,j)+Sigma2^2*S2(j)^2*CrossGammaS1S2C(i,j)^2);
        CostS2(i,j)=k*S2(j)*sqrt(2/pi/dtc)*sqrt(Sigma1^2*S1(i)^2*CrossGammaS1S2C(i,j)^2+2*Rho*Sigma1*Sigma2*S1(i)*S2(j)*GammaS2C(i,j)*CrossGammaS1S2C(i,j)+Sigma2^2*S2(j)^2*GammaS2C(i,j)^2);
        ThetaC(i,j)=-(r-D1)*S1(i)*DeltaS1C(i,j)-(r-D2)*S2(j)*DeltaS2C(i,j)-1/2*Sigma1^2*S1(i)^2*GammaS1C(i,j)-1/2*Sigma2^2*S2(j)^2*GammaS2C(i,j)-Rho*Sigma1*Sigma2*S1(i)*S2(j)*CrossGammaS1S2C(i,j)+r*ValueOldC(i,j)+CostS1(i,j)+CostS2(i,j);
        ValueNewC(i,j)=ValueOldC(i,j)-dt*ThetaC(i,j);
    end
end
end

```

```

title('\partial^2V/\partials1\partials2 ','fontsize',tf,'fontname','Verdana','fontangle',
'','Italic','fontweight','b','color','k');
xlabel({'s2 ':'Ativo 2 ','fontsize',af,'fontname','Verdana','fontangle','Italic','font
weight','b','color','k');
ylabel({'s1 ':'Ativo 1 ','fontsize',af,'fontname','Verdana','fontangle','Italic','font
weight','b','color','k');
zlabel({'\partial^2V/\partials1\partials2 ':'Cross-Gamma Ativo 12'},'fontsize',af,'fontn
ame','Verdana','fontangle','Italic','fontweight','b','color','k');
%axis([Sigma_i Sigma_f 0 Smax min(min(DeltaSSigma)) max(max(DeltaSSigma))])
alpha(.5);
view(-45,30);

figure(3)
set(gcf,'Color',[1,1,1])%Cor do fundo da figura
subplot(2,2,1)
surfc(S2(2:M),S1(2:L),GammaS1(2:L,2:M),'EdgeColor','None');
title('\partial^2V/\partials1^2 ','fontsize',tf,'fontname','Verdana','fontangle','Itali
c','fontweight','b','color','k');
xlabel({'s2 ':'Ativo 2 ','fontsize',af,'fontname','Verdana','fontangle','Italic','font
weight','b','color','k');
ylabel({'s1 ':'Ativo 1 ','fontsize',af,'fontname','Verdana','fontangle','Italic','font
weight','b','color','k');
zlabel({'\partial^2V/\partials1^2 ':'Gamma Ativo 1'},'fontsize',af,'fontname','Verdana',
'fontangle','Italic','fontweight','b','color','k');
%axis([Sigma_i Sigma_f 0 Smax min(min(SigmaS)) max(max(SigmaS))])
alpha(.5);
view(-45,25);
subplot(2,2,2)
surfc(S2(2:M),S1(2:L),GammaS2(2:L,2:M),'EdgeColor','None');
title('\partial^2V/\partials2^2 ','fontsize',tf,'fontname','Verdana','fontangle','Itali
c','fontweight','b','color','k');
xlabel({'s2 ':'Ativo 2 ','fontsize',af,'fontname','Verdana','fontangle','Italic','font
weight','b','color','k');
ylabel({'s1 ':'Ativo 1 ','fontsize',af,'fontname','Verdana','fontangle','Italic','font
weight','b','color','k');
zlabel({'\partial^2V/\partials2^2 ':'Gamma Ativo 2'},'fontsize',af,'fontname','Verdana',
'fontangle','Italic','fontweight','b','color','k');
%axis([Sigma_i Sigma_f 0 Smax min(min(SigmaSigma)) max(max(SigmaSigma))])
alpha(.5);
view(-45,25);

figure(4)
set(gcf,'Color',[1,1,1])%Cor do fundo da figura
subplot(2,2,1)
surfc(S2,S1,ValueOldC,'EdgeColor','None');
title('Basket Up&Out Call com Custo ','fontsize',tf,'fontname','Verdana','fontangle','I
talic','fontweight','b','color','k');
xlabel({'s2 ':'Ativo 2 ','fontsize',af,'fontname','Verdana','fontangle','Italic','font
weight','b','color','k');
ylabel({'s1 ':'Ativo 1 ','fontsize',af,'fontname','Verdana','fontangle','Italic','font
weight','b','color','k');
zlabel({'V ':'Valor Basket Up&Out Call ','fontsize',af,'fontname','Verdana','fontangle',
'','Italic','fontweight','b','color','k');
%axis([0 S2max 0 S1max min(min(ValueOld)) max(max(ValueOld))])
alpha(.5);
view(-45,30);
subplot(2,2,2)
surfc(S2(2:M),S1(2:L),DeltaS1C(2:L,2:M),'EdgeColor','None');
title('\partialV/\partials1 ','fontsize',tf,'fontname','Verdana','fontangle','Italic',
'fontweight','b','color','k');
xlabel({'s2 ':'Ativo 2 ','fontsize',af,'fontname','Verdana','fontangle','Italic','font
weight','b','color','k');
ylabel({'s1 ':'Ativo 1 ','fontsize',af,'fontname','Verdana','fontangle','Italic','font
weight','b','color','k');
zlabel({'\partialV/\partials1 ':'Delta Ativo 1 ','fontsize',af,'fontname','Verdana','f
ontangle','Italic','fontweight','b','color','k');
%axis([0 S2max 0 S1max min(min(DeltaS)) max(max(DeltaS))])
alpha(.5);
view(-45,30);
subplot(2,2,3)
surfc(S2(2:M),S1(2:L),DeltaS2C(2:L,2:M),'EdgeColor','None');

```

```

title({'Opcao Basket Up&Out Call - Diferencas Finitas ','Volatilidades Deterministicas
e Custo de Transacao'},'fontsize',tf,'fontname','Verdana','fontangle','Italic','fontwei
ght','b','color','k');
xlabel({'s2 ','Ativo 2 ','','fontsize',af,'fontname','Verdana','fontangle','Italic','font
weight','b','color','k');
ylabel({'s1 ','Ativo 1 ','','fontsize',af,'fontname','Verdana','fontangle','Italic','font
weight','b','color','k');
zlabel({'Opcao Basket Up&Out Call sem Custo ','Opcao Basket Up&Out Call com Custo
','','fontsize',af,'fontname','Verdana','fontangle','Italic','fontweight','b','color',
'k');
view(-45,30);

%Mostra os valores calculados com este algoritmo
fprintf('\n -----')
fprintf('\n European Basket Up&Out Call - Metodo de Diferencas Finitas Explicito')
fprintf('\n com 2 Ativos, Volatilidades Deterministicas, com e sem Custo de Transacao')
fprintf('\n -----')
fprintf('\n Valores usados para este calculo')
fprintf('\n dS1      [intervalo do valor do ativo 1]           : %-.2f',dS1)
fprintf('\n L        [numero de intervalos do ativo 1]           : %-.0f',L)
fprintf('\n dS2      [intervalo do valor do ativo 2]           : %-.2f',dS2)
fprintf('\n M        [numero de intervalos do ativo 2]           : %-.0f',M)
fprintf('\n dt       [intervalo de tempo de integracao]           : %-.6f',dt)
fprintf('\n N        [numero de intervalos de tempo]             : %-.0f',N)
fprintf('\n S10      [valor do ativo 1]                           : %-.2f',S10)
fprintf('\n S20      [valor do ativo 2]                           : %-.2f',S20)
fprintf('\n K        [strike]                                       : %-.2f',K)
fprintf('\n w        [percentual do ativo 1]                       : %-.2f',w)
fprintf('\n Sb       [valor da barreira da opcao]                 : %-.2f',Sb)
fprintf('\n S1max    [valor maximo para ativo 1]                   : %-.2f',S1max)
fprintf('\n S2max    [valor maximo para ativo 2]                   : %-.2f',S2max)
fprintf('\n T        [tempo para o vencimento]                   : %-.2f',T)
fprintf('\n r        [taxa de juros livre de risco]               : %-.2f',r)
fprintf('\n D1       [taxa de dividendo do ativo 1]               : %-.2f',D1)
fprintf('\n D2       [taxa de dividendo do ativo 2]               : %-.2f',D2)
fprintf('\n Rho      [correlacao entre os ativos]                 : %-.2f',Rho)
fprintf('\n Sigma1   [volatilidade do ativo 1]                   : %-.2f',Sigma1)
fprintf('\n Sigma2   [volatilidade do ativo 2]                   : %-.2f',Sigma2)
fprintf('\n -----')
fprintf('\n Valores de custo de transacao')
fprintf('\n dtc      [intervalo de tempo de hedge]                : %-.4f',dtc)
fprintf('\n kS1      [custo proporcional ao ativo 1]               : %-.4f',kS1)
fprintf('\n kS2      [custo proporcional ao ativo 2]               : %-.4f',kS2)
fprintf('\n -----')
fprintf('\n -----')
fprintf('\n Tempo do calculo (com os graficos) : %-.2f [seg]',elapsedTime)
fprintf('\n -----')
%Valores para os dados de saida
fprintf('\n Resultados - Valores Interpolados sem custo')
Valor=[0 vectorInterpS2;vectorInterpS1' interpValue]
fprintf('\n -----')
fprintf('\n Resultados - Valores Interpolados com custo')
ValorC=[0 vectorInterpS2;vectorInterpS1' interpValueC]
fprintf('\n -----')
xlswrite('UpOutCall_2Assets_FDExpCost.xlsx',[Valor;ValorC]);

```

```

y = (V-valorOp)/stdv_valorOp;%Numeros gerados pela simulacao
[nn,xx] = hist(y,x);%nn(i) numero de pontos em cada intervalo xx
X = -4:0.001:4;%Pontos para grafico da densidade da normal padrao
Y = exp(-0.5*X.*X)/sqrt(2*pi);%Pontos para PDF da distribuicao normal padrao

figure(1)
set(gcf,'Color',[1,1,1])%Cor do fundo da figura
subplot(2,2,1)
normplot(log(Sts(1:2*nTraj)))%Normplot do logaritmo dos valores dos ativos de uma
simulacao
title({'log ST ','Distribuicao dos valores dos ativos ','fontsize',tf,'fontname','Verd
ana','fontangle','Italic','fontweight','b','color','k'});
xlabel({'nTraj ','Numero de trajetorias por simulacao ','fontsize',af,'fontname','Verd
ana','fontangle','Italic','fontweight','b','color','k'});
ylabel({'percentil log ST ','Valor final do ativo ','fontsize',af,'fontname','Verdana',
'fontangle','Italic','fontweight','b','color','k'});
grid on

subplot(2,2,2)
normplot(V)
title({'V ','Distribuicao dos valores da opcao ','fontsize',tf,'fontname','Verdana','f
ontangle','Italic','fontweight','b','color','k'});
xlabel({'V ','Valor da opcao ','fontsize',af,'fontname','Verdana','fontangle','Italic
','fontweight','b','color','k'});
ylabel({'Percentil V ','Valor da opcao ','fontsize',af,'fontname','Verdana','fontangle
','fontangle','Italic','fontweight','b','color','k'});

subplot(2,2,3)
plot(sort(V))
title('Valor da opcao ','fontsize',tf,'fontname','Verdana','fontangle','Italic','fontwe
ight','b','color','k');
xlabel({'nSimul ','Numero de simulacoes ','fontsize',af,'fontname','Verdana','fontangl
e','fontangle','Italic','fontweight','b','color','k'});
ylabel({'V ','Valor da opcao ','fontsize',af,'fontname','Verdana','fontangle','Italic
','fontweight','b','color','k'});
grid on

str_mu=sprintf('Media= %-.4f',valorOp);
str_sig1=sprintf('IC(95)= %-.4f',lowerCI);
str_sig2=sprintf('IC(95)= %-.4f',upperCI);
subplot(2,2,4)
plot(X+valorOp,Y,'-r',xx+valorOp,nn/(nSimul*dx),'ob')%Grafico PDF normal padrao
%grid(gca,'minor')
title('Normal vs Simulacoes ','fontsize',tf,'fontname','Verdana','fontangle','Italic','
fontweight','b','color','k');
xlabel({'V','Valor da opcao ','fontsize',af,'fontname','Verdana','fontangle','Italic',
'fontweight','b','color','k'});
ylabel({'PDF ','fontsize',af,'fontname','Verdana','fontangle','Italic','fontweight','b',
'color','k'});
hold on;
plot([lowerCI lowerCI],[0 .4],'g',[valorOp valorOp],[0 .4],'k',[upperCI upperCI],[0
.4],'g');
legend('Normal','Pontos Simulados',str_sig1,str_mu,str_sig2,1)
axis([-4+valorOp 4+valorOp 0 .5])

```

```

inferior para o valor da opcao
upperCI=valorOp-tinv(alpha/2,nSimul-1)*stdv_valorOp*sqrt((nSimul-1)/nSimul);%IC
superior para o valor da opcao
[lowerCI valorOp upperCI]%Intervalo de confianca de 95%

elapsedTime=toc(ticID)%Para o timer
%Definicao dos graficos
set(0,'DefaultFigurePosition',[100 100 900 640])
tf=16;%Tamanho do font dos titulos
af=14;%Tamanho do font dos eixos

Sts=sort(St(nInt+1,:));%Valores do ativo no final de cada trajetoria (ordenados)
nZero=1;%nZero e o primeiro valor ao final da trajetoria nao nulo
while Sts(nZero)==0
    nZero=nZero+1;
end

dx = .25;%Tamanho de cada intervalo
x = (-4+dx/2):dx:(4-dx/2);%Centro de cada inetrvalo
y = (V-valorOp)/stdv_valorOp;%Numeros gerados pela simulcao
[nn,xx] = hist(y,x);%nn(i) numero de pontos em cada intervalo xx
X = -4:0.001:4;%Pontos para grafico da densidade da normal padrao
Y = exp(-0.5*X.*X)/sqrt(2*pi);%Pontos para PDF da distribuicao normal padrao

figure(1)
set(gcf,'Color',[1,1,1])%Cor do fundo da figura
subplot(2,2,1)
normplot(log(Sts(nZero:2*nTraj)))%Normplot do logaritmo dos valores dos ativos de uma
simulacao
title({'log ST ':'Distribuicao dos valores dos ativos ','fontsize',tf,'fontname','Verd
ana','fontangle','Italic','fontweight','b','color','k'});
xlabel({'nTraj ':'Numero de trajetorias por simulacao ','fontsize',af,'fontname','Verd
ana','fontangle','Italic','fontweight','b','color','k'});
ylabel({'percentil log ST ':'Valor final do ativo ','fontsize',af,'fontname','Verdana',
'fontangle','Italic','fontweight','b','color','k'});
grid on

subplot(2,2,2)
normplot(V)
title({'V ':'Distribuicao dos valores da opcao ','fontsize',tf,'fontname','Verdana','f
ontangle','Italic','fontweight','b','color','k'});
xlabel({'V ':'Valor da opcao ','fontsize',af,'fontname','Verdana','fontangle','Italic',
'fontweight','b','color','k'});
ylabel({'Percentil V ':'Valor da opcao ','fontsize',af,'fontname','Verdana','fontangle',
'fontweight','b','color','k'});

subplot(2,2,3)
plot(sort(V))
title('Valor da opcao ','fontsize',tf,'fontname','Verdana','fontangle','Italic','fontwe
ight','b','color','k');
xlabel({'nSimul ':'Numero de simulacoes ','fontsize',af,'fontname','Verdana','fontangl
e','Italic','fontweight','b','color','k'});
ylabel({'V ':'Valor da opcao ','fontsize',af,'fontname','Verdana','fontangle','Italic',
'fontweight','b','color','k'});
grid on

str_mu=sprintf('Media= %-.4f',valorOp);
str_sig1=sprintf('IC(95)= %-.4f',lowerCI);
str_sig2=sprintf('IC(95)= %-.4f',upperCI);
subplot(2,2,4)
plot(X+valorOp,Y,'-r',xx+valorOp,nn/(nSimul*dx),'ob')%Grafico PDF normal padrao
%grid(gca,'minor')
title('Normal vs Simulacoes ','fontsize',tf,'fontname','Verdana','fontangle','Italic','
fontweight','b','color','k');
xlabel({'V ':'Valor da opcao ','fontsize',af,'fontname','Verdana','fontangle','Italic',
'fontweight','b','color','k'});
ylabel('PDF ','fontsize',af,'fontname','Verdana','fontangle','Italic','fontweight','b',
'color','k');
hold on;
plot([lowerCI lowerCI],[0 .4],'g',[valorOp valorOp],[0 .4],'k',[upperCI upperCI],[0
.4], 'g');
legend('Normal','Pontos Simulados',str_sig1,str_mu,str_sig2,1)
axis([-4+valorOp 4+valorOp 0 .5])

```

```

af=11;%Tamanho do font dos eixos dos graficos
ticID=tic;%Inicia o timer

%Geracao dos graficos das Brownian Bridges
figure(1)
set(gcf,'Color',[1,1,1])%Cor do fundo da figura
subplot(2,2,1)
plot([0:DeltaT:T],repl,[0 T],[Sb Sb],'k');
title('Trajetorias da Brownian Bridge ','fontsize',tf,'fontname','Verdana','fontangle'
,'Italic','fontweight','b','color','k');
xlabel({'T ':'Periodo de geracao da Brownian Bridge '},'fontsize',af,'fontname','Verd
ana','fontangle','Italic','fontweight','b','color','k');
ylabel({'W(t) ':'Movimento Browniano '},'fontsize',af,'fontname','Verdana','fontangle'
,'Italic','fontweight','b','color','k');
axis([0 T x-5 y+5])
grid on

subplot(2,2,2)
plot(sort(Min),'b');hold on
plot(sort(Max),'r');hold off
title('Brownian Bridge Min-Max ','fontsize',tf,'fontname','Verdana','fontangle','Itali
c','fontweight','b','color','k');
xlabel({'nSimul ':'Simulacoes da Brownian Bridge '},'fontsize',af,'fontname','Verdana'
,'fontangle','Italic','fontweight','b','color','k');
ylabel({'Max ':'Min '},'fontsize',af,'fontname','Verdana','fontangle','Italic','fontw
eight','b','color','k');
grid on

subplot(2,2,3)
plot([0:DeltaT:T],rep2,[0 T],[Sb Sb],'r');
title('Trajetorias da Brownian Bridge ','fontsize',tf,'fontname','Verdana','fontangle'
,'Italic','fontweight','b','color','k');
xlabel({'T ':'Periodo de geracao da Brownian Bridge '},'fontsize',af,'fontname','Verd
ana','fontangle','Italic','fontweight','b','color','k');
ylabel({'W(t) ':'Movimento Browniano '},'fontsize',af,'fontname','Verdana','fontangle'
,'Italic','fontweight','b','color','k');
axis([0 T x-5 y+5])
grid on

subplot(2,2,4)
plot(sort(cut),'g');
title('Probabilidade de corte ','fontsize',tf,'fontname','Verdana','fontangle','Italic'
,'fontweight','b','color','k');
xlabel({'nSimul ':'Simulacoes da Brownian Bridge '},'fontsize',af,'fontname','Verdana'
,'fontangle','Italic','fontweight','b','color','k');
ylabel({'Corte ':'Movimento Browniano '},'fontsize',af,'fontname','Verdana','fontangl
e','Italic','fontweight','b','color','k');
grid on

elapsedTime=toc(ticID);%Para o timer
Tempo_de_Grafico=elapsedTime

```

Referências Bibliográficas

- [1] Maria C. Mariani, Indranil SenGupta, Granville Sewell - *Numerical Methods applied to Option Pricing Models with Transaction Costs and Stochastic Volatility*
- [2] Fabrice D. Rouah, Steven L. Heston - *The Heston Model and its Extensions in Matlab and C#, + Website-Wiley*, John Wiley & Sons, Inc., Hoboken, New Jersey, 2013
- [3] K. J. IN'T Hout e S. Foulon - *ADI Finite Difference Schemes for Options Pricing in the Heston Model with Correlation*, International Journal of Numerical Analysis and Modeling, Volume 7, Number 2, Pages 303-320, Copyright Institute for Scientific Computing and Information 2010
- [4] Domingo A. Tavella - *Quantitative Methods in Derivatives Pricing - An Introduction to Computational Finance*, John Willey & Sons Inc, 2002
- [5] C. S. L. de Graaf - *Finite Difference Methods in Derivatives Pricing under Stochastic Volatility Models*, Master Thesis in Applied Mathematics, Mathematisch Instituut Universiteit Leiden, 2012
- [6] Paul Wilmott - *Paul Wilmott on Quantitative Finance*, 2nd Edition, John Willey & Sons Inc, 2006
- [7] Daniel J. Duffy - *Finite difference methods in financial engineering - A Partial Differential Equation Approach*, Copyright 2006, John Wiley&Sons, The Atrium, Southern Gate, Chichester West Sussex PO19 8SQ, England
- [8] Jürgen Topper - *Financial engineering with finite elements*, Copyright 2005, John Wiley&Sons, The Atrium, Southern Gate, Chichester West Sussex PO19 8SQ, England
- [9] Kyoung-Sook Moon - *Efficient Monte Carlo Algorithm for Pricing Barrier Options*, Korea Mathematics Society 23 (2008), Number 2, pp 285-294
- [10] Espen Gaarder Haug - *The complete guide to Option Pricing Formulas*, 2nd Edition, McGraw-Hill, 2007
- [11] Paulo Brandimare - *Numerical Methods in Finance and Economics - A Matlab®-Based Introduction*, 2nd Edition, John Willey & Sons Inc, 2006
- [12] John C. Tannehill, Dale A. Anderson, Richard H. Pletcher - *Computational Fluid Mechanics and Heat Transfer*, 2nd Edition, Copyright 1997, Taylor and Francis
- [13] Michael Mastro - *Financial Derivative and Energy Market Valuation, Theory and Implementation in Matlab®*, John Willey & Sons Inc, 2013
- [14] Leon Lapidus, George Pinder - *Numerical Solution of Partial Differential Equations in Science and Engineering*, Copyright 1982, John Wiley and Sons, Inc.
- [15] John L. Weatherwax - *Notes and Solutions for: The Mathematics of Financial Derivatives* by Paul Wilmott, Sam Howison, and Jeff Dewynne