

Universidade de São Paulo
Instituto de Matemática e Estatística
Bacharelado em Matemática Aplicada

Daniel Morales

**Uma Interpolação de Volatilidades
Livre de Arbitragem**

São Paulo
Maio de 2019

Uma Interpolação de Volatilidades Livre de Arbitragem

Monografia final da disciplina
MAP2080 – Trabalho de Formatura

Orientador: Prof. Dr. Nelson Mugayar Kuhl

São Paulo
Maio de 2019

Agradecimentos

Agradeço a todos os colegas do estágio na BM&FBovespa, atual B3, que foram os primeiros a ver a apresentação desse tema. Em especial, agradeço a Wilson Nascimento de Freitas, gestor durante o estágio, por propor o artigo e acompanhar seu desenvolvimento.

Agradeço a todos os colegas, professores e funcionários da USP. Aos colegas, pelo companheirismo, pelas discussões de listas, por me inspirarem a ser melhor. Aos professores e funcionários, por me aguentarem durante todo esse tempo.

Agradeço a minha família, em especial a meus pais e avós, que sempre me apoiaram e trabalharam para dar aos filhos as oportunidades que não tiveram quando jovens.

Agradeço às amigas de antes da faculdade, que sobreviveram e me acompanharam durante esse período (além verem uma quantidade razoável do meu cabelo cair, espero que vejam esse cabelo crescer de novo).

Agradeço às minhas professoras de matemática e desenho geométrico do ensino fundamental, Elaine Spisso Barrella e Laura Maria Runau Martins, que me apresentaram às olimpíadas científicas. Agradeço também a todos que participam, organizam e treinam alunos para qualquer olimpíada científica. Acredito fortemente que esse tipo de comunidade faz do mundo um lugar melhor para se viver.

Resumo

Um smile (sorriso) de volatilidades é a visualização de como a hipótese de volatilidade constante do modelo de precificação de opções de Black-Scholes [2] não se verifica na prática: ela depende do prazo e do strike (preço de exercício) da opção. Como construir um smile a partir das poucas opções que o mercado precifica é um problema tratado por diversos modelos (ver [3], [7], [5], [1], [12] e [6], por exemplo), e aqui tratamos de um deles, de Nabil Kahalé [9], que fornece uma interpolação livre de arbitragem para prazo fixado. O modelo de Kahalé, no entanto, nos leva a resolver sistemas de equações não lineares para obtenção de seus parâmetros. Como não foram apontados algoritmos para resolução desses sistemas em seu artigo, usamos os algoritmos do livro de Dennis e Schnabel [4], mesma referência do pacote `nleqslv` para R, com o qual o modelo foi testado nos dados de operações estruturadas de volatilidade da bolsa de valores brasileira antes de ser implementado em VBA no Excel para usuários.

Palavras-chave: interpolação, volatilidade, livre de arbitragem.

Sumário

1	Introdução	6
1.1	Black & Scholes e Volatilidade Implícita	6
1.2	Apreçamento e Não Arbitragem	8
2	Interpolação de Volatilidades de Kahalé	10
2.1	Interpolação C^1	12
2.1.1	Soluções Particulares para Intervalos Extremos	12
2.2	Interpolação C^2	13
2.2.1	Interpolação C^2 Iterativa de Kahalé	14
3	Resolução dos Sistemas de Equações Não Lineares	16
3.1	Método de Newton para Sistemas de Equações Não Lineares .	16
3.2	Modificações Globalmente Convergentes do Método de Newton	17
3.2.1	Busca Linear	18
4	Resultados	20
4.1	Exemplo Didático	20
4.1.1	Interpolação C^1	20
4.1.2	Interpolação C^2 Iterativa, 2 Iterações	21
4.1.3	Interpolação C^2	21
4.2	Escala	21
4.3	Reprodução do Exemplo S&P	25
4.4	Aplicação a Operações Estruturadas de Volatilidade	28
5	Conclusões	31
	Referências	32
A	Códigos	33

1 Introdução

A opção de compra europeia, ou call, a mais simples das opções, é um contrato com as condições: em um momento fixado no futuro, chamado de prazo, o dono, titular ou beneficiário da opção tem a opção de comprar um ativo combinado, conhecido como ativo objeto (uma ação de uma empresa, por exemplo), por um montante combinado, conhecido como preço de exercício ou strike.

A palavra-chave na descrição, opção, implica que o beneficiário desse contrato tem um direito ao invés de um dever. A outra parte do contrato, o vendedor ou lançador, tem um potencial dever de vender o ativo objeto ao preço de exercício se o beneficiário quiser comprá-lo. Por ser um direito sem obrigações ou deveres, uma opção é um contrato com valor, que deve ser pago na abertura do contrato. Esse valor, chamado de prêmio, compensa a potencial obrigação do vendedor ao fim do prazo [14].

Do mesmo modo, uma opção de venda europeia, ou put, é um contrato que dá o direito de vender o ativo objeto pelo preço de exercício no prazo especificado [13]. Pelas similaridades, normalmente tratamos apenas o caso de opções de compra, calls, sem espelhar para o caso de opções de venda, puts.

Como existe uma grande variedade de ações para serem ativo objeto e infinitas combinações de prazo e preço de exercício, para viabilizar a negociação de opções em bolsa de valores (grande escala), faz-se necessário limitar e padronizar essas combinações. Essas opções listadas podem então ser negociadas e precificadas pelo mercado. Como descobrir o valor teórico das opções não listadas com e sem o uso do valor dado pelo mercado é o que trataremos a seguir.

1.1 Black & Scholes e Volatilidade Implícita

Black & Scholes [2] propuseram um modelo para precificar opções que é o mais conhecido até hoje. Ele assume que o retorno do ativo objeto em um curto período de tempo tem distribuição normal com média μ e desvio padrão σ , e que os retornos em períodos de tempo não sobrepostos são independentes [8]. Considerando longos períodos de tempo, essa hipótese implica que o preço do ativo objeto em qualquer tempo no futuro tem distribuição log-normal.

O modelo descreve o prêmio c_{BS} de uma opção de compra europeia como $c_{BS}(S, \tau, K, r, \sigma, q)$, onde

- S é o spot, preço atual do ativo objeto;

- τ é o prazo até o vencimento, em anos;
- K é o preço de exercício;
- r é a taxa de juros livre de risco, anualizada e com composição contínua;
- σ é a volatilidade, que pelas suposições, é o desvio padrão dos retornos anuais com composição contínua;
- q é o dividend yield, rendimento anualizado de dividendos, sempre com composição contínua.

Além disso, a fórmula dada pelo modelo é

$$c_{BS}(S, \tau, K, r, \sigma, q) = e^{-r\tau} [FN(d_1) - KN(d_2)]$$

$$F = Se^{(r-q)\tau}$$

$$d_1 = \frac{\log(F/K)}{\sigma\sqrt{\tau}} + \frac{\sigma\sqrt{\tau}}{2}$$

$$d_2 = d_1 - \sigma\sqrt{\tau}$$

$$N(x) = \int_{-\infty}^x \frac{1}{\sqrt{2\pi}} \exp\left(-\frac{t^2}{2}\right) dt$$

Como um modelo é uma visão simplificada da realidade, o modelo de Black & Scholes não é totalmente exato na realidade. A suposição de normalidade é prática, porém não se verifica em dados reais, e ao usar a fórmula isso fica visível na volatilidade σ . Ela já é a variável mais difícil de se medir, e normalmente usamos os preços de mercado das opções com a inversa da fórmula de Black & Scholes para obter a chamada volatilidade implícita.

Ao calcular a volatilidade implícita para os variados prazos e preços de exercício de opções para o mesmo ativo objeto, essa volatilidade implícita, que deveria ser igual, é diferente, e esse é um modo de verificar a imperfeição do modelo.

Analisando a volatilidade implícita para diferentes preços de exercício e outras variáveis fixadas, o gráfico resultante toma forma característica chamada de smile (sorriso) de volatilidade. Um método de interpolação de preços de opções significa também uma interpolação para formar esse sorriso de volatilidade. Se contarmos a variação no preço de exercício e no prazo, temos uma superfície de volatilidade.

Além de ser uma medida de variação do preço do ativo objeto, a volatilidade implícita é usada em estratégias de trading e uma interpolação como a aqui tratada é importante para calibrar o modelo de Dupire de volatilidade local, possibilitando o apreçamento de opções exóticas.

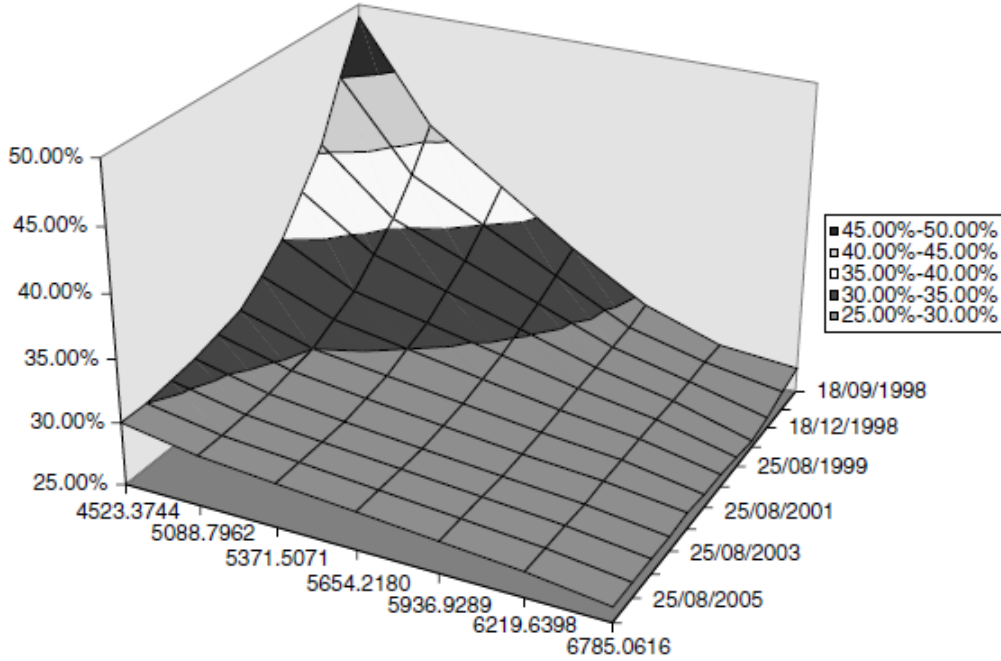


Figura 1: Exemplo de superfície de volatilidade para opções do índice FTSE de 18 de agosto de 1998. Note que a diferença é grande para prazos curtos e vai diminuindo conforme o prazo aumenta. Imagem retirada de [11].

1.2 Apreçamento e Não Arbitragem

Fixado o ativo objeto, o preço c de uma opção depende de seu prazo τ e preço de exercício K , também chamado de strike. O mercado precifica opções apenas para poucos prazos e preços de exercício específicos, possibilitando sua negociação em grande escala. Faz-se necessária, então, uma interpolação para obter $c(\tau, K)$ para todo $\tau \geq 0$ e $K \geq 0$ que siga os valores dados pelo mercado.

Essa interpolação está sujeita às chamadas restrições de não arbitragem, que justamente evitam arbitragem e mantêm o preço coerente, além dos casos limite. São elas:

$$\frac{\partial c}{\partial \tau} < 0$$

$$\frac{\partial c}{\partial K} < 0$$

$$\frac{\partial^2 c}{\partial K^2} > 0$$

E os casos limite:

$$c(\tau, 0) := \lim_{K \rightarrow 0} c(\tau, K) = S$$

$$c(\tau, \infty) := \lim_{K \rightarrow \infty} c(\tau, K) = 0$$

$$c(0, K) := \lim_{\tau \rightarrow 0} c(\tau, K) = S$$

$$c(\infty, K) := \lim_{\tau \rightarrow \infty} c(\tau, K) = 0$$

Uma superfície $c(\tau, K)$ que segue as restrições de não arbitragem é chamada de admissível [11]. A interpolação apresentada a seguir é apenas na dimensão preço de exercício, com o objetivo de produzir smiles (sorrisos) de volatilidade, mas ainda tem como ponto principal seguir as restrições de não arbitragem.

2 Interpolação de Volatilidades de Kahalé

Fixado ativo objeto e prazo, consideramos dados pelo mercado os strikes e prêmios $(k_i, c_i)_{1 \leq i \leq n}$, além do preço atual do ativo objeto (spot), e acrescentamos os limites teóricos a essa sequência com os índices 0 à esquerda e $n+1$ à direita: $k_{n+1} = \infty$, $k_0 = 0$, $c_{n+1} = 0$ e c_0 é o spot, preço do ativo objeto.

Para garantir as condições de limite e que essa sequência é livre de arbitragem, devemos verificar as condições dadas pelo Lema 1 de [9], que são basicamente a versão discreta das condições de não arbitragem escritas de modo prático para verificar no caso discreto:

$$0 = c_{n+1} = k_0 < k_1 < \dots < k_n < k_{n+1} = \infty$$

$$-1 \leq \frac{c_i - c_{i-1}}{k_i - k_{i-1}} \leq \frac{c_{i+1} - c_i}{k_{i+1} - k_i} \leq 0, \text{ para } 1 \leq i \leq n$$

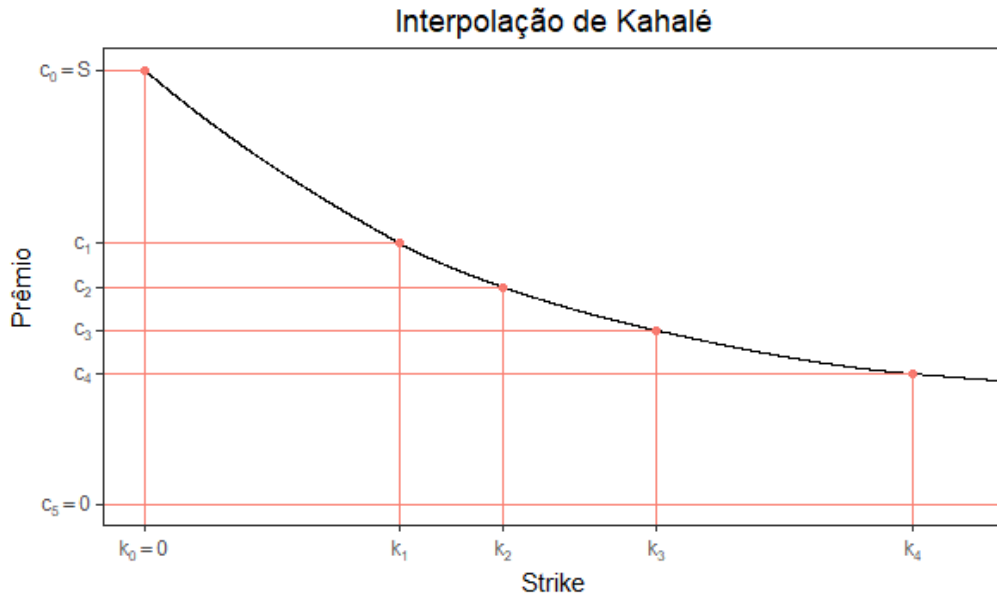


Figura 2: Gráfico exemplo do final de uma interpolação de volatilidades, na forma prêmio x strike, para período fixo.

A curva inspirada por Black & Scholes dada por Kahalé [9] que liga esses pontos é uma função definida por partes ou intervalos, assim como um spline, porém a curva em cada intervalo não é um polinômio: ela se assemelha à fórmula de Black & Scholes sem o fator de desconto e somada a uma reta.

A escolha de uma curva mais elaborada em vez de um método mais estabelecido, como um spline cúbico, acontece justamente para garantir o cumprimento das restrições de não arbitragem.

A cada intervalo $[k_{i-1}, k_i[$, $1 \leq i \leq n+1$, escolhemos parâmetros

- f_i , como um equivalente ao forward de Black & Scholes;
- Σ_i , como um equivalente à volatilidade integrada de Black & Scholes;
- a_i , como um equivalente ao declive (slope) de uma reta;
- b_i , como um equivalente à ordenada na origem (intercept) de uma reta;

e a curva nesse intervalo é dada pela função $c(k; f_i, \Sigma_i, a_i, b_i)$. Trataremos apenas k como variável, logo c' e c'' se referem às derivadas em relação a k . Então, temos

$$c(k; f, \Sigma, a, b) = fN(d_1) - kN(d_2) + ak + b$$

$$c'(k; f, \Sigma, a, b) = -N(d_2) + a$$

$$c''(k; f, \Sigma, a, b) = \frac{N'(d_2)}{k\Sigma}$$

$$d_1 = \frac{\log(f/k)}{\Sigma} + \frac{\Sigma}{2}$$

$$d_2 = d_1 - \Sigma$$

$$N(x) = \int_{-\infty}^x \frac{1}{\sqrt{2\pi}} \exp\left(\frac{-t^2}{2}\right) dt$$

Note que as equações nos dão restrições aos parâmetros, assim como dão informação sobre a ordem de grandeza deles: f e Σ são positivos; pela fórmula de $c'(k; f, \Sigma, a, b)$, temos $-1 \leq a \leq 1$; Σ tem ordem de grandeza parecida com uma volatilidade e perto de a , enquanto f e b têm ordem de grandeza de acordo com o valor do ativo objeto.

Basta agora escolher os parâmetros para cada intervalo. Assim como em um spline, a escolha dos parâmetros é feita de modo a deixar a conexão dos segmentos lisa, ou seja, mais próxima de C^∞ .

2.1 Interpolação C^1

A primeira e mais leve restrição para escolha de parâmetros é de obter uma curva C^1 . Isso significa que devemos escolher $(c'_i)_{0 \leq i \leq n+1}$, ou seja, escolher o valor da primeira derivada em cada ponto de encaixe de intervalo, novamente respeitando as condições limite e de não arbitragem, dadas pelo Teorema 2 de [9], novamente em versão prática para caso discreto:

$$c'_0 = -1, c'_{n+1} = 0$$

$$c'_i < \frac{c_{i+1} - c_i}{k_{i+1} - k_i} < c'_{i+1}, \text{ para } 0 \leq i \leq n$$

e resolver para cada intervalo o sistema

$$\begin{cases} c(k_i; f_i, \Sigma_i, a_i, b_i) = c_i \\ c(k_{i+1}; f_i, \Sigma_i, a_i, b_i) = c_{i+1} \\ c'(k_i; f_i, \Sigma_i, a_i, b_i) = c'_i \\ c'(k_{i+1}; f_i, \Sigma_i, a_i, b_i) = c'_{i+1} \end{cases}$$

Fixada a escolha de $(c'_i)_{0 \leq i \leq n+1}$, é demonstrado no teorema 2 de [9] que a interpolação C^1 existe e é única.

A liberdade de escolher a sequência $(c'_i)_{0 \leq i \leq n+1}$ será importante para a interpolação C^2 iterativa, onde escolheremos essa sequência de modo a gerar uma interpolação C^2 . Quando for de interesse usar apenas a interpolação C^1 , trataremos como padrão a escolha da média entre as inclinações dos intervalos à esquerda e à direita em cada ponto interno, que é usada como escolha inicial na interpolação C^2 iterativa.

2.1.1 Soluções Particulares para Intervalos Extremos

Os intervalos $[k_0, k_1[$ e $[k_n, k_{n+1}[$ usam os limites teóricos da curva, portanto, por conhecermos esses valores, podemos encontrar parâmetros e reduzir o sistema para esses intervalos. No intervalo $[k_0, k_1[$, temos

$$\begin{aligned} \begin{cases} c(0; f_1, \Sigma_1, a_1, b_1) = c_0 = S \\ c(k_1; f_1, \Sigma_1, a_1, b_1) = c_1 \\ c'(0; f_1, \Sigma_1, a_1, b_1) = c'_0 = -1 \\ c'(k_1; f_1, \Sigma_1, a_1, b_1) = c'_1 \end{cases} &\Rightarrow \begin{cases} b_1 = S - f_1 \\ a_1 = 0 \\ -N(d_{2;1}) = c'_1 \end{cases} \\ &\Rightarrow \begin{cases} f_1 (N(d_{1;1}) - 1) + k_1 c'_1 + S = c_1 \\ f_1 = k_1 \exp \left(\Sigma_1 N^{-1}(c'_1) - \frac{\Sigma_1^2}{2} \right) \end{cases} \end{aligned}$$

$$d_{1;1} = \frac{\log(f_1/k_1)}{\Sigma_1} - \frac{\Sigma_1}{2}$$

$$d_{2;1} = d_{1;1} - \Sigma_1$$

que nos leva a resolver numericamente apenas uma equação, para descobrir Σ_1 , e com esse valor encontrar f_1 e b_1 , já que sempre temos $a_1 = 0$.

Já para o intervalo $[k_n, k_{n+1}[$, temos

$$\begin{cases} c(k_n; f_{n+1}, \Sigma_{n+1}, a_{n+1}, b_{n+1}) = c_n \\ c(\infty; f_{n+1}, \Sigma_{n+1}, a_{n+1}, b_{n+1}) = c_{n+1} = 0 \\ c'(k_n; f_{n+1}, \Sigma_{n+1}, a_{n+1}, b_{n+1}) = c'_n \\ c'(\infty; f_{n+1}, \Sigma_{n+1}, a_{n+1}, b_{n+1}) = c'_{n+1} = 0 \end{cases} \Rightarrow \begin{cases} b_{n+1} = 0 \\ -N(d_{2;n}) = c'_n \\ a_{n+1} = 0 \end{cases}$$

$$\Rightarrow \begin{cases} f_{n+1} (N(d_{1;n}) - 1) + k_n c'_n = c_n \\ f_{n+1} = k_n \exp \left(\Sigma_{n+1} N^{-1}(c'_n) - \frac{\Sigma_{n+1}^2}{2} \right) \end{cases}$$

$$d_{1;n} = \frac{\log(f_{n+1}/k_n)}{\Sigma_{n+1}} - \frac{\Sigma_{n+1}}{2}$$

$$d_{2;n} = d_{1;n} - \Sigma_{n+1}$$

que de maneira similar, nos permite encontrar os parâmetros do intervalo com resolução numérica de uma equação, em vez de um sistema com quatro equações.

2.2 Interpolação C^2

A condição mais forte que a escolha de parâmetros dessa interpolação pode fornecer é uma curva C^2 , incluindo as restrições de que a primeira e segunda derivada nos pontos de emenda de intervalos sejam iguais. Essas equações impossibilitam o cálculo dos parâmetros intervalo a intervalo em sistemas de 4 equações, passando para a resolução de um sistema de $4(n+1)$ equações. Para cada k_i , $1 \leq i \leq n$, temos quatro equações

$$\begin{cases} c(k_i; f_i, \Sigma_i, a_i, b_i) = c_i \\ c(k_i; f_{i+1}, \Sigma_{i+1}, a_{i+1}, b_{i+1}) = c_i \\ c'(k_i; f_i, \Sigma_i, a_i, b_i) = c'(k_i; f_{i+1}, \Sigma_{i+1}, a_{i+1}, b_{i+1}) \\ c''(k_i; f_i, \Sigma_i, a_i, b_i) = c''(k_i; f_{i+1}, \Sigma_{i+1}, a_{i+1}, b_{i+1}) \end{cases}$$

Mais duas para $k_0 = 0$

$$\begin{cases} c(0; f_1, \Sigma_1, a_1, b_1) = c_0 = S \\ c'(0; f_1, \Sigma_1, a_1, b_1) = -1 \end{cases}$$

E as últimas duas para $k_{n+1} = \infty$

$$\begin{cases} c(\infty; f_{n+1}, \Sigma_{n+1}, a_{n+1}, b_{n+1}) = c_{n+1} = 0 \\ c'(\infty; f_{n+1}, \Sigma_{n+1}, a_{n+1}, b_{n+1}) = 0 \end{cases}$$

Em seu artigo [9], Nabil Kahalé não explorou essa evolução intuitiva da abordagem C^1 . Apesar da complexidade aumentar quando há muitos pontos disponíveis e recorrermos ao pacote `nleqslv` para solução dos sistemas no R, ainda é uma alternativa válida, e que só é considerada neste texto. Por isso há a divisão entre interpolação C^2 e C^2 iterativa, que em todos os testes levaram ao mesmo resultado, mesmo sendo provada somente a existência, mas não a unicidade, da solução para a interpolação C^2 . Nos resultados, iremos destacar este modo de obter a interpolação.

O único método apresentado no artigo para conseguir uma interpolação C^2 é o iterativo, e este foi possível implementar em VBA, porém pedindo mais pontos iniciais. Vamos a ele a seguir.

2.2.1 Interpolação C^2 Iterativa de Kahalé

O algoritmo apresentado em [9] para obter $(f_i, \Sigma_i, a_i, b_i)_{1 \leq i \leq n+1}$ que geram o spline C^2 consiste em procurar a escolha certa de $(c'_i)_{0 \leq i \leq n+1}$ a ser usada no método de interpolação C^1 .

Para isso, começamos um processo iterativo, atualizando a escolha de $(c'_i)_{0 \leq i \leq n+1}$ a cada iteração.

A sequência pode ser inicializada por $c'_0 = -1$, $c'_{n+1} = \infty$ e $c'_i = \frac{l_i + l_{i+1}}{2}$, para $1 \leq i \leq n$, onde $l_i = \frac{c_i - c_{i+1}}{k_i - k_{i+1}}$, ou seja, usando os limites teóricos nos extremos e a média entre as inclinações dos intervalos à esquerda e à direita em cada ponto interno.

Uma iteração consiste em resolver uma interpolação C^2 para intervalos 2 a 2, ou seja, resolver o sistema

$$\left\{ \begin{array}{l} c(k_{i-1}; f_i, \Sigma_i, a_i, b_i) = c_{i-1} \\ c'(k_{i-1}; f_i, \Sigma_i, a_i, b_i) = c'_{i-1} \\ c(k_i; f_i, \Sigma_i, a_i, b_i) = c_i \\ c'(k_i; f_i, \Sigma_i, a_i, b_i) = c'(k_i; f_{i+1}, \Sigma_{i+1}, a_{i+1}, b_{i+1}) \\ c''(k_i; f_i, \Sigma_i, a_i, b_i) = c''(k_i; f_{i+1}, \Sigma_{i+1}, a_{i+1}, b_{i+1}) \\ c(k_i; f_{i+1}, \Sigma_{i+1}, a_{i+1}, b_{i+1}) = c_i \\ c'(k_{i+1}; f_{i+1}, \Sigma_{i+1}, a_{i+1}, b_{i+1}) = c'_{i+1} \\ c(k_{i+1}; f_{i+1}, \Sigma_{i+1}, a_{i+1}, b_{i+1}) = c_{i+1} \end{array} \right.$$

para $1 \leq i \leq n$, ou seja, para cada ponto interno serão considerados os intervalos à esquerda e à direita. Note que o valor da primeira derivada em k_i não é controlado, apenas restrito para ser igual quando usados os parâmetros f, Σ, a, b dos intervalos à esquerda e à direita. Esse valor é o que substituirá c'_i na próxima iteração, até não haver mais mudança significativa nos valores da sequência. Os valores dos extremos já são os teóricos, e se mantêm.

3 Resolução dos Sistemas de Equações Não Lineares

Reduzido o problema de interpolação à resolução de sistemas de equações não lineares, vamos agora tratar desse tema, seguindo os métodos do livro de Dennis e Schnabel [4] e eventual consulta ao Numerical Recipes [10]. Foi implementado o pseudocódigo completo do apêndice para uso em VBA, e aqui tratamos dos métodos usados para interpolação, apesar do código trazer as outras opções de métodos tratados no livro.

Dada $F : \mathbb{R}^n \rightarrow \mathbb{R}^n$ de classe C^1 , queremos encontrar x^* tal que $F(x^*) = 0$.

3.1 Método de Newton para Sistemas de Equações Não Lineares

Dada $F : \mathbb{R}^n \rightarrow \mathbb{R}^n$ de classe C^1 e $x_0 \in \mathbb{R}^n$, a cada iteração k , resolver

$$\begin{aligned} J(x_k)s_k &= -F(x_k) \\ x_{k+1} &= x_k + s_k \end{aligned}$$

onde J é a matriz jacobiana de F .

O método de Newton é base para todo o algoritmo: não tem apenas qualidades, mas a partir dele iremos tratar suas desvantagens para obter um algoritmo consistente. Idealmente, queremos resolver:

- necessidade de $J(x)$ a cada iteração: relativamente simples, substituindo o jacobiano $J(x_k)$ por uma aproximação A_k tal que cada coluna j , $1 \leq j \leq n$ de A_k seja dada por

$$(A_k)_{\cdot,j} = \frac{F(x_k + he_j) - F(x_k)}{h}$$

para h suficientemente pequeno e e_j vetor da base canônica. Não é necessário no caso da interpolação, onde temos os jacobianos, mas pode ser usado;

- eventual aparecimento de sistemas lineares singulares, já que cada iteração requer a solução de um sistema linear: incluir perturbação nesses sistemas singulares ou próximos de serem singulares, seguindo direção de descida próxima da direção do método de Newton;

- convergência global: o método de Newton não necessariamente é globalmente convergente. Localmente, ele converge a uma velocidade suficientemente boa, mas para isso seria necessário um ponto inicial próximo da solução. Faz-se necessário um método globalmente convergente, que leve pontos distantes da solução para pontos próximos, onde o método de Newton converge. Vamos conseguir isso ao considerar o problema de minimização de uma norma correspondente ao sistema.

Trataremos agora de um método para cobrir esse pontos negativos.

3.2 Modificações Globalmente Convergentes do Método de Newton

Sabemos que o método de Newton é localmente quadraticamente convergente, mas não necessariamente globalmente convergente. Para pontos que não estão próximos de uma solução, outra estratégia deve ser usada. Uma escolha razoável é a de que a cada iteração haja uma diminuição satisfatória na norma do vetor. Escolhendo a norma l_2 , estaríamos voltando nossa atenção para o correspondente problema de minimização

$$\min_{x \in \mathbb{R}^n} f(x) = \min_{x \in \mathbb{R}^n} \frac{1}{2} \|F(x)\|_2^2 = \min_{x \in \mathbb{R}^n} \frac{1}{2} F(x)^T F(x)$$

onde a constante e elevar ao quadrado são adicionados para facilitar futuros cálculos.

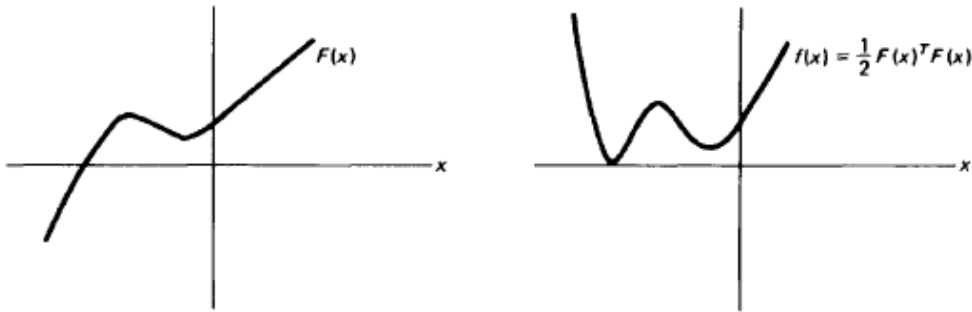


Figura 3: Em uma dimensão é intuitivo e de fácil visualização a dualidade entre a resolução de um sistema de equações não lineares e seu correspondente problema de minimização. Também destacamos um ponto de mínimo local de f que não é solução de F , que virá a ser problemático. Imagem retirada de [4].

Note que toda solução para o sistema é uma solução para o problema de minimização, mas a volta não é válida, já que podem existir minimizadores locais (e nossos problemas acabam por aqui, f não possui pontos de máximo). Logo, não podemos usar uma diretamente uma estratégia de minimização global, mas podemos nos basear em uma.

Aqui há a possibilidade de escolha de método global a ser usado. A referência principal [4] nos proporciona os métodos de busca linear, “hook step” e “double dogleg step”. Como a busca linear produziu os melhores resultados nas aplicações, trataremos apenas dela a seguir.

3.2.1 Busca Linear

A intuição da busca linear é simples: dada uma direção de descida p_k , tomamos um passo nessa direção que produza uma diminuição de f aceitável, isso é, cada iteração será do tipo

$$x_{k+1} = x_k + \lambda_k p_k$$

onde próximo da solução queremos a convergência quadrática da iteração de Newton, com $\lambda_k = 1$ e $p_k = -J(x_k)^{-1}F(x_k)$, mas precisamos entender como é feita a definição de aceitável e as escolhas de λ_k e p_k , principalmente nos casos problemáticos e longe da solução.

Vamos calcular $J(x_k)$ ou uma aproximação como citada anteriormente e aplicar uma técnica de fatoração de matrizes (no nosso caso, usaremos a de fatoração QR) para determinar se $J(x_k)$ está próximo de ser singular. Se não estiver, escolhemos $p_k = -J(x_k)^{-1}F(x_k)$, nossa desejada direção de Newton. Para $J(x_k)$ singular ou próximo de singular, escolhemos $p_k = -(J(x_k) + \mu_k I)^{-1}F(x_k)$, onde I é matriz identidade e $\mu_k > 0$ é suficiente para deixar $J(x_k) + \mu_k I$ inversível.

Seguimos então para a escolha de λ_k aceitável, onde definiremos aceitável como uma condição de Armijo: queremos que o próximo ponto satisfaça

$$f(x_{k+1}) < f(x_k) + 10^{-4} \lambda_k \nabla f(x_k)^T p_k$$

ou seja, queremos uma diminuição razoavelmente significativa em f .

A primeira tentativa é nosso desejado $\lambda_{k,1} = 1$ para completar um passo de Newton. Se esse passo não satisfizer a condição de Armijo, começamos a procurar valores cada vez menores de λ_k na direção p_k , por isso o nome busca linear.

Apesar de existir o método mais básico de fazer essa busca linear, dividindo os candidatos a λ_k por dois sucessivamente, vamos usar aproximações quadráticas e cúbicas, aproveitando os pontos onde já avaliamos o valor de

f . Ou seja, o primeiro candidato a λ_k é $\lambda_{k,1} = 1$, o segundo candidato $\lambda_{k,2}$ é obtido fazendo uma aproximação quadrática, que usa os valores já calculados de $f(x_k)$, $f'(x_k)$ e $f(x_k + \lambda_{k,1}p_k)$ até o limite de $\left[\frac{\lambda_{k,1}}{10}, \frac{\lambda_{k,1}}{2}\right]$. A partir do terceiro candidato, podemos usar aproximações cúbicas com os dois candidatos anteriores, sempre dentro do intervalo limitante.

A busca continua até um passo ser aceito, e sabemos que algum candidato pequeno será aceito, por p_k ser direção de descida.

Há um caso significativo onde esse método pode falhar, e esse é o grande ponto negativo do algoritmo. Isso acontece quando atingimos um mínimo local de f que não é solução de F , e não há solução simples para esse caso, a não ser parar, avisar o acontecimento e recomeçar de outro ponto inicial.

É por esse ponto negativo que usamos vários pontos iniciais para obter uma interpolação consistente em aplicações com novos dados diários, mas ainda assim não é passível de confiança cega, eventualmente podem aparecer dados onde todos os pontos iniciais levam a mínimos locais de f .

4 Resultados

Construído o algoritmo, vamos agora aplicá-lo a dados e visualizar seu funcionamento. Onde for necessário escolher valores de c'_i , usamos como padrão a média das diferenças divididas dos intervalos adjacentes.

A escolha de pontos iniciais também é relevante para os resultados, principalmente pelo uso de mais de um ponto inicial. O padrão usado para a escolha de pontos iniciais foi de combinações de iniciar a_i 's e b_i 's em 0 sempre, iniciar f_i 's baseado em 50% a 200% do valor do spot e iniciar Σ_i 's baseado em valores de volatilidade integrada, de 0,001 a 0,4.

A diferença entre a qualidade da implementação em VBA e no pacote nleqslv fica evidente na quantidade de pontos iniciais necessários: em VBA foram necessários mais pontos iniciais e por isso a interpolação C^2 foi usada somente no R.

4.1 Exemplo Didático

O primeiro teste apresentado é em uma sequência curta e de fácil visualização. Escolhemos $n = 4$, $k = (5, 7, 10, 15)$, $c = (6, 5, 4, 3)$ e spot 10. Note que podemos testar o algoritmo apenas com dados que seguem as restrições de não arbitragem, para depois passarmos a exemplos reais.

4.1.1 Interpolação C^1

i	k_i	c_i	c'_i	c''_i	f_i	Σ_i	a_i	b_i
0	0	10	-1	0	-	-	-	-
1	5	6	-0,65	0,2377	42,8329	1,7228	0	-32,8329
2	7	5	-0,4167	0,0687	4,3708	0,2761	-0,3841	7,6611
3	10	4	-0,2667	0,0135	6,7353	0,7565	-0,0828	3,6849
4	15	3	-0,1	0,0071	21,6273	0,3434	0,7143	-14,7920
5	∞	0	0	0	7,0345	1,63922	0	0

Tabela 1: Dados usados para criar os gráficos na Figura 4 da interpolação C^1 do exemplo didático.

4.1.2 Interpolação C^2 Iterativa, 2 Iterações

i	k_i	c_i	c'_i	c''_i	f_i	Σ_i	a_i	b_i
0	0	10	-1	0	-	-	-	-
1	5	6	-0,5836	0,0831	11,9785	1,1276	0	-1,9785
2	7	5	-0,4178	0,0719	11,0702	0,6774	0,2145	-1,4301
3	10	4	-0,2683	0,0362	6,0587	0,6802	-0,1275	4,5872
4	15	3	-0,1492	0,0123	7,6183	0,8350	-0,0396	2,5950
5	∞	0	0	0	8,9408	1,2568	0	0

Tabela 2: Dados usados para criar os gráficos na Figura 5 da interpolação C^2 iterativa do exemplo didático.

4.1.3 Interpolação C^2

i	k_i	c_i	c'_i	c''_i	f_i	Σ_i	a_i	b_i
0	0	10	-1	0	-	-	-	-
1	5	6	-0,5756	0,0726	11,0033	1,0798	0	-1,0033
2	7	5	-0,4233	0,0763	12,0994	0,6586	0,2687	-2,6485
3	10	4	-0,2639	0,0351	6,2378	0,6578	-0,1162	4,4631
4	15	3	-0,1542	0,0129	6,8521	0,7754	-0,0732	3,4853
5	∞	0	0	0	9,1232	1,2265	0	0

Tabela 3: Dados usados para criar os gráficos na Figura 6 da interpolação C^2 do exemplo didático.

4.2 Escala

O exemplo didático nos mostra um funcionamento do método em dados de magnitude parecida, porém podemos encontrar aplicações onde os valores para f e b são significativamente maiores que Σ e a . Isso acontece em casos onde o valor do ativo objeto já é de grande magnitude, como em índices.

Nesses casos, já vemos de [4] que reescalar os parâmetros é essencial, e o algoritmo do livro já prevê essa necessidade para a função objetivo e seu argumento. No entanto, as equações da função objetivo dependem de outros parâmetros (k_i 's, c_i 's) que também precisam de reescala.

Foi escolhido, então fazer essa reescala fora do algoritmo de [4]. Usamos a ordem de grandeza do spot para reescalar k_i 's, c_i 's, f_i 's e b_i 's antes da resolução para reescalar as variáveis após a solução. Note que a equação é a mesma, multiplicando essas variáveis por qualquer constante, porém a constante usada é conveniente para a resolução numérica.

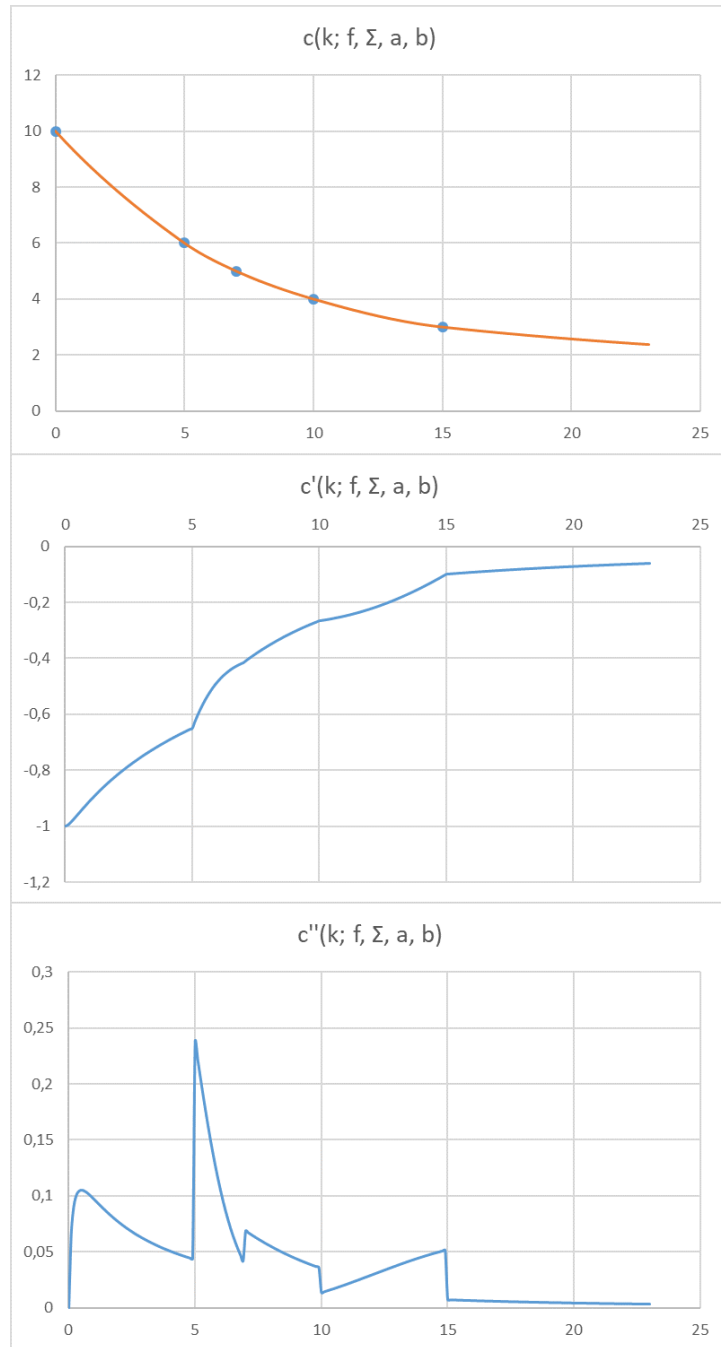


Figura 4: Interpolação C^1 do exemplo didático. Notam-se as “quinas” típicas de pontos onde não há derivada no gráfico da primeira derivada, comprovadas pela descontinuidade no gráfico da segunda derivada, como esperado da interpolação C^1 : primeira derivada contínua, mas não derivável.

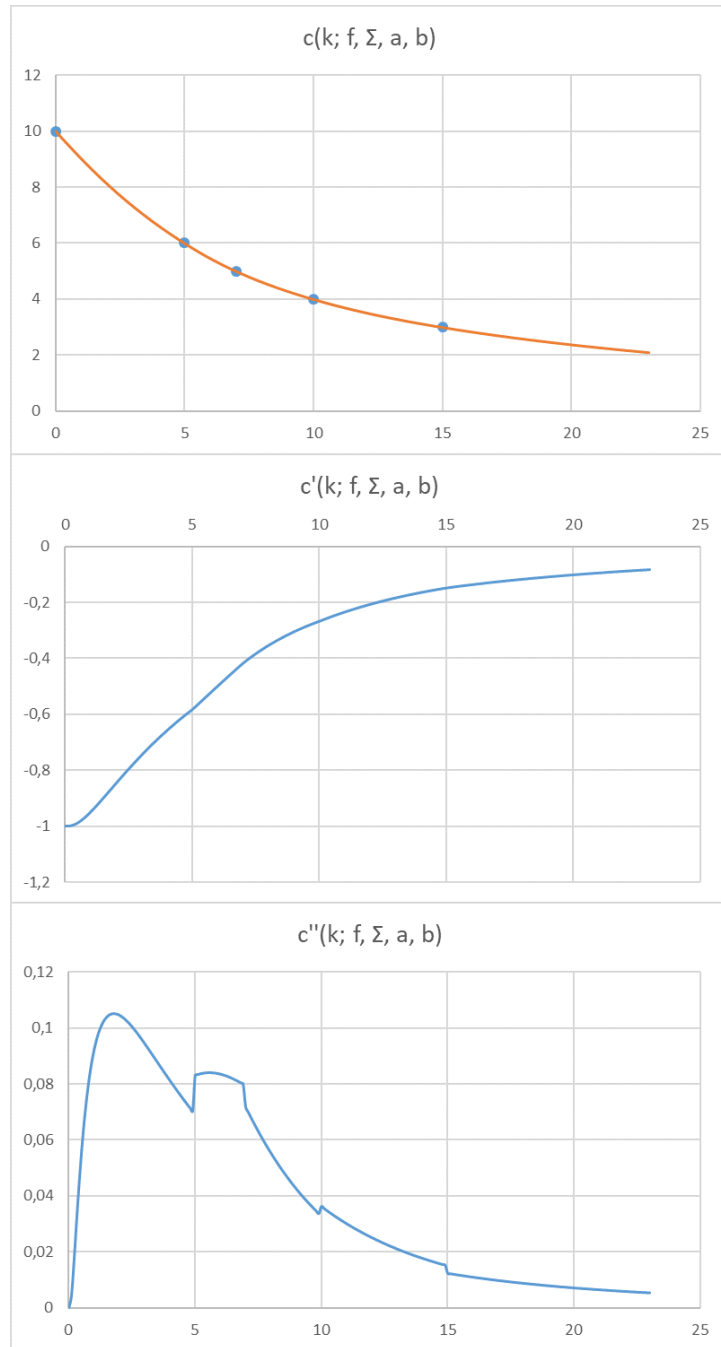


Figura 5: Interpolação C^2 iterativa com duas iterações do exemplo didático. Cada iteração desse método torna as descontinuidades nas emendas de intervalos da segunda derivada menor, até o ponto onde novas iterações trazem alterações pequenas o suficiente para não serem significativas, o que indica ao algoritmo quando parar.

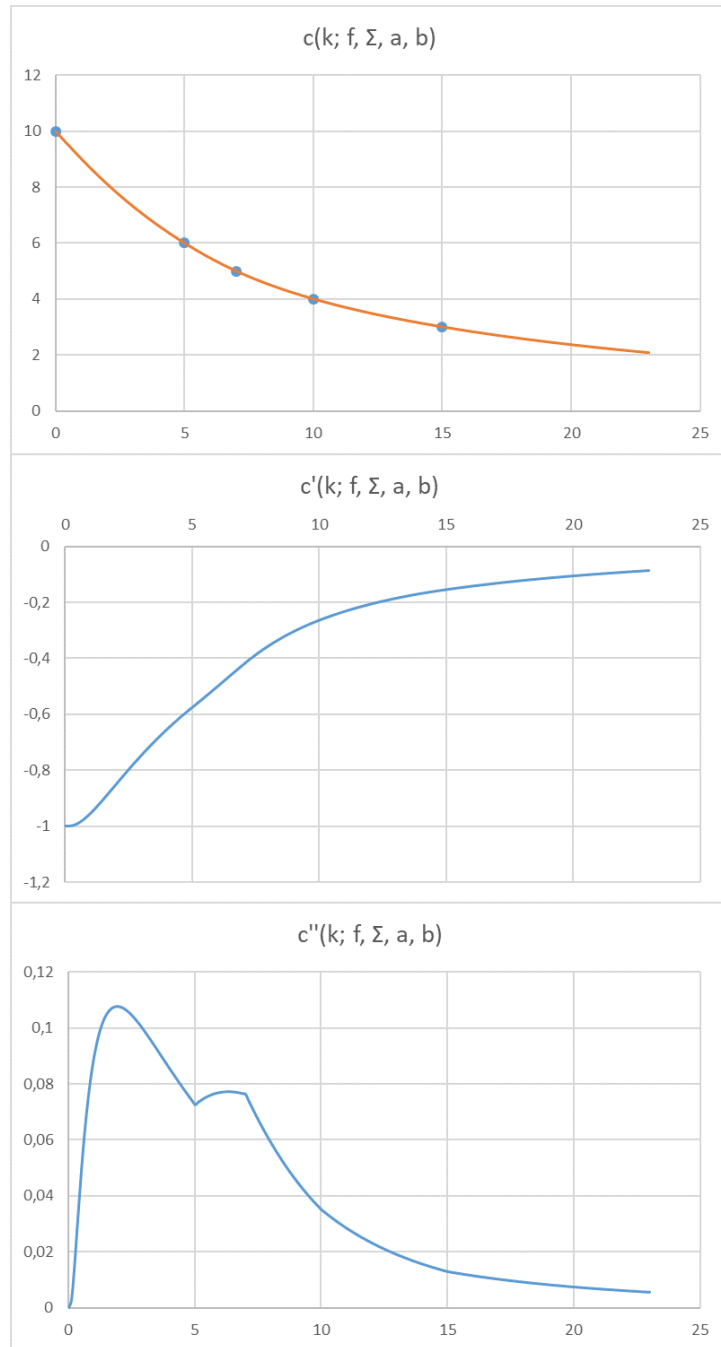


Figura 6: Interpolação C^2 do exemplo didático. As “quinas” da primeira derivada da interpolação C^1 são observadas agora no gráfico da segunda derivada da interpolação C^2 . Agora, como esperado, a segunda derivada é contínua e não derivável.

4.3 Reprodução do Exemplo S&P

A primeira aplicação a dados reais será a reprodução do exemplo do artigo de Kahalé [9], com a matriz de volatilidades do índice S&P 500 em outubro de 1995. Nesse caso, onde os dados estão em volatilidade, já fazemos a comum conversão entre prêmio e volatilidade com a fórmula de Black & Scholes.

$\tau \backslash K$	501,5	531	560,5	590	619,5	649	678,5	708	767	826
0,175	19,0	16,8	13,3	11,3	10,2	9,7	12,0	14,2	16,9	20,0
0,425	17,7	15,5	13,8	12,5	10,9	10,3	10,0	11,4	13,0	15,0
0,695	17,2	15,7	14,4	13,3	11,8	10,4	10,0	10,1	10,8	12,4
0,94	17,1	15,9	14,9	13,7	12,7	11,3	10,6	10,3	10,0	11,0
1	17,1	15,9	15,0	13,8	12,8	11,5	10,7	10,3	09,9	10,8
1,5	16,9	16,0	15,1	14,2	13,3	12,4	11,9	11,3	10,7	10,2
2	16,9	16,1	15,3	14,5	13,7	13,0	12,6	11,9	11,5	11,1
3	16,8	16,1	15,5	14,9	14,3	13,7	13,3	12,8	12,4	12,3
4	16,8	16,2	15,7	15,2	14,8	14,3	13,9	13,5	13,0	12,8
5	16,8	16,4	15,9	15,4	15,1	14,8	14,4	14,0	13,6	13,2

Tabela 4: Dados do exemplo S&P 500 do artigo [9], onde mostramos volatilidade em porcentagem ao longo de prazo (em anos) e preço de exercício. Os dados são de outubro de 2015 e referenciados de outro artigo, mas podemos usá-los sabendo que o spot $S = 590$, a taxa de juros livre de risco $r = 0,06$ e o dividend yield $q = 0,0262$. Também já podemos notar nesses dados a diferença de ordem de grandeza das variáveis que justifica reescalá-las.

Note que há pouca diferença visível entre interpolação C^1 e C^2 ao compararmos as curvas em prêmio das figuras 7 e 9, porém vemos smiles mais suaves ao compararmos as curvas em volatilidade das figuras 8 e 10.

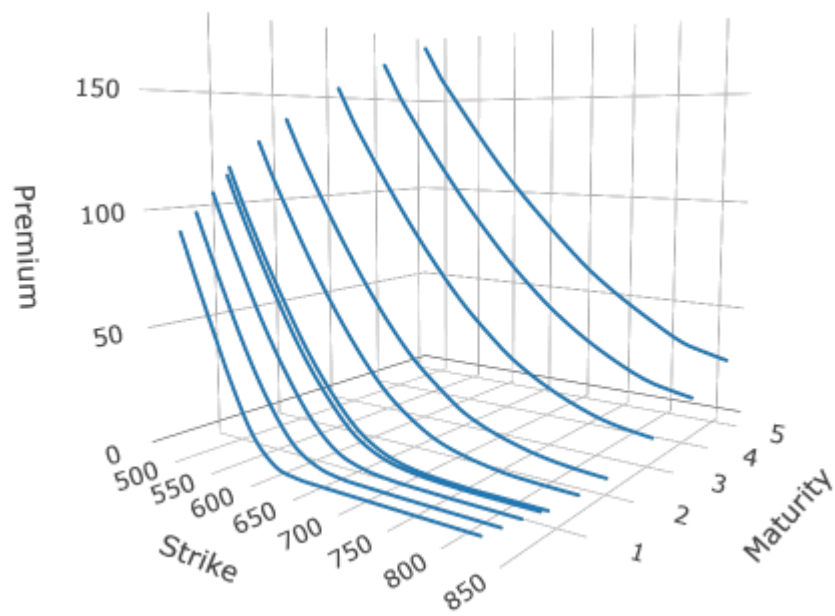


Figura 7: Interpolação C^1 em prêmio do exemplo S&P.

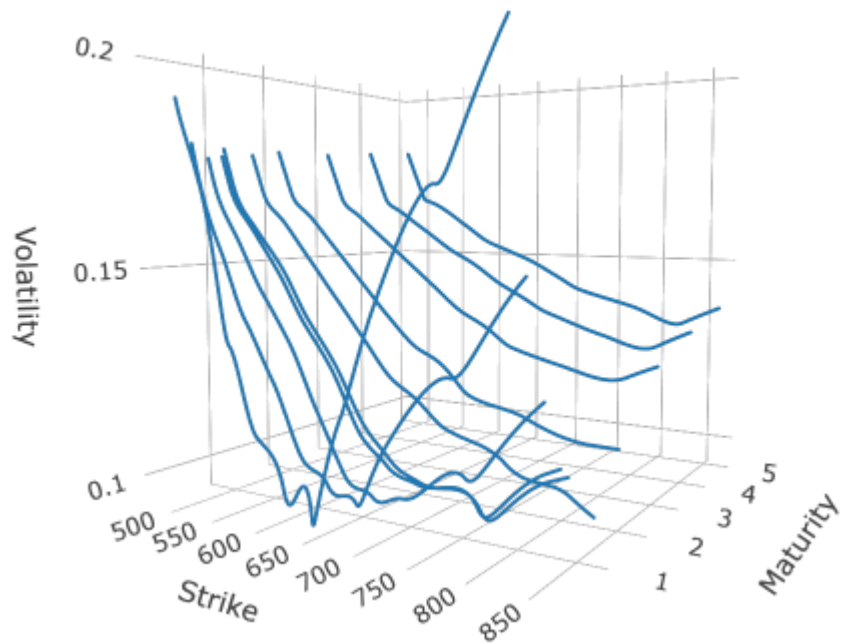


Figura 8: Interpolação C^1 em volatilidade do exemplo S&P.

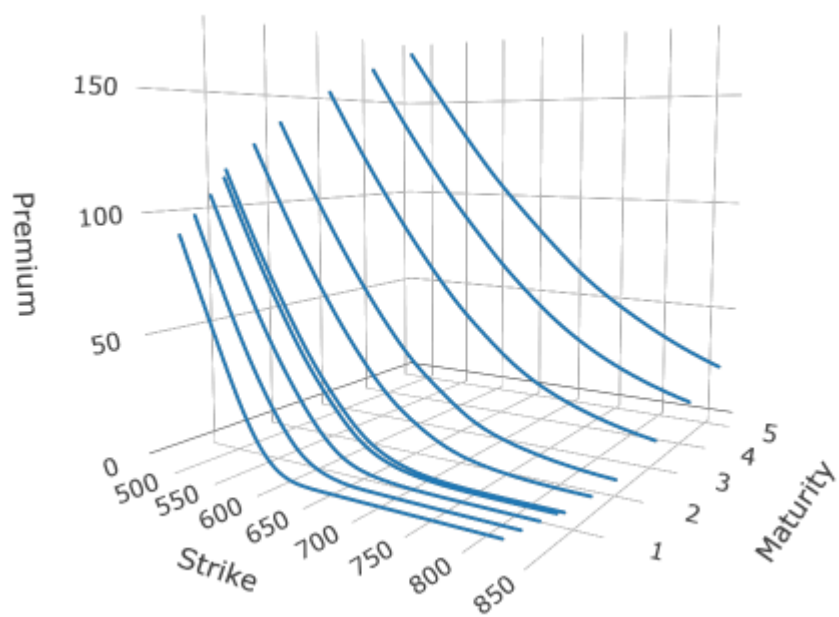


Figura 9: Interpolação C^2 em prêmio do exemplo S&P.

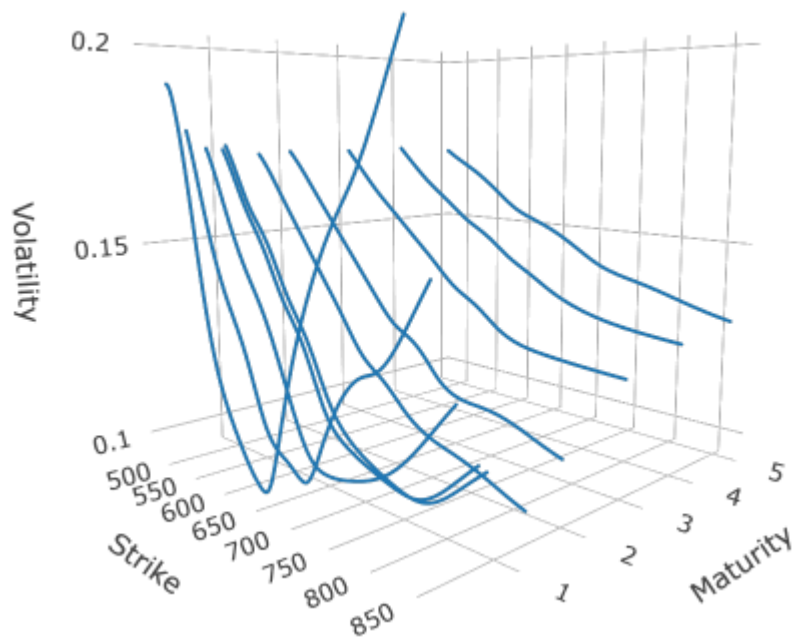


Figura 10: Interpolação C^2 em volatilidade do exemplo S&P.

4.4 Aplicação a Operações Estruturadas de Volatilidade

A motivação para o estudo de interpolações de volatilidade foi como mais uma opção de modelo ao tratar dos smiles de volatilidade de operações estruturadas de volatilidade da bolsa de valores brasileira. São elas:

- VTC - Operação Estruturada de Volatilidade de Taxa de Câmbio
- VOI - Operação Estruturada de Volatilidade de Ibovespa
- VID - Operação Estruturada de Volatilidade de Taxa de Juros à Vista
- VTF - Operação Estruturada de Volatilidade de Taxa de Juro Forward

A interpolação C^1 se mostrou consistente ao longo de testes com diferentes dados e em vários dias, mesmo nos problemáticos, enquanto na interpolação C^2 (iterativa ou não) ocorreram falhas. Os únicos casos com dados reais onde não foi possível aplicar a interpolação C^1 foram aquelas sequências que já não seguem as condições de não arbitragem.

Mostramos agora um caso de VTC de 10/06/2016. Os dados para a interpolação ficam disponíveis com os códigos no apêndice A.

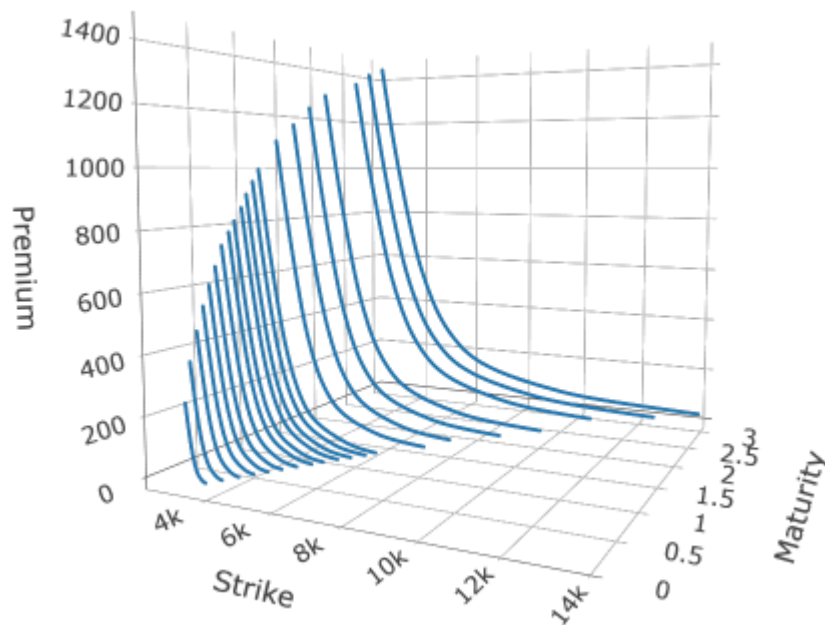


Figura 11: Interpolação C^1 em prêmio de VTC, 10/06/2016.

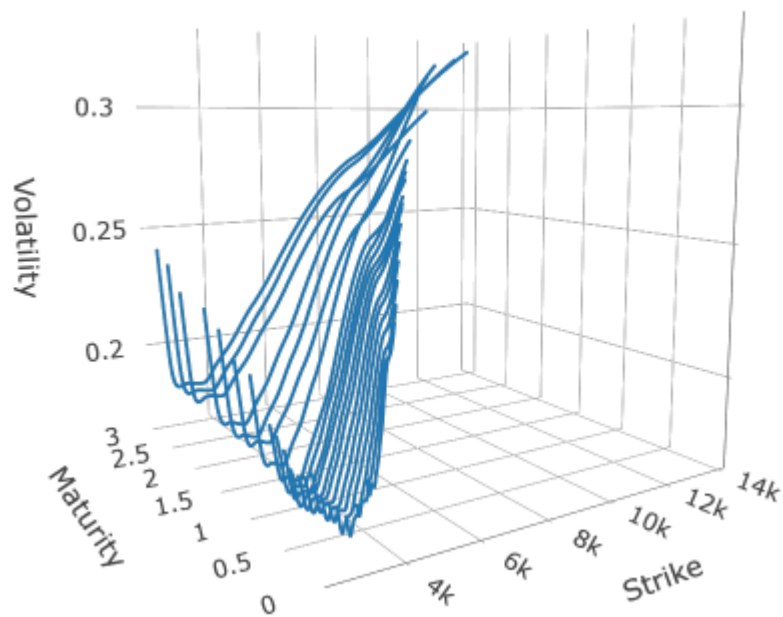


Figura 12: Interpolação C^1 em volatilidade de VTC, 10/06/2016.

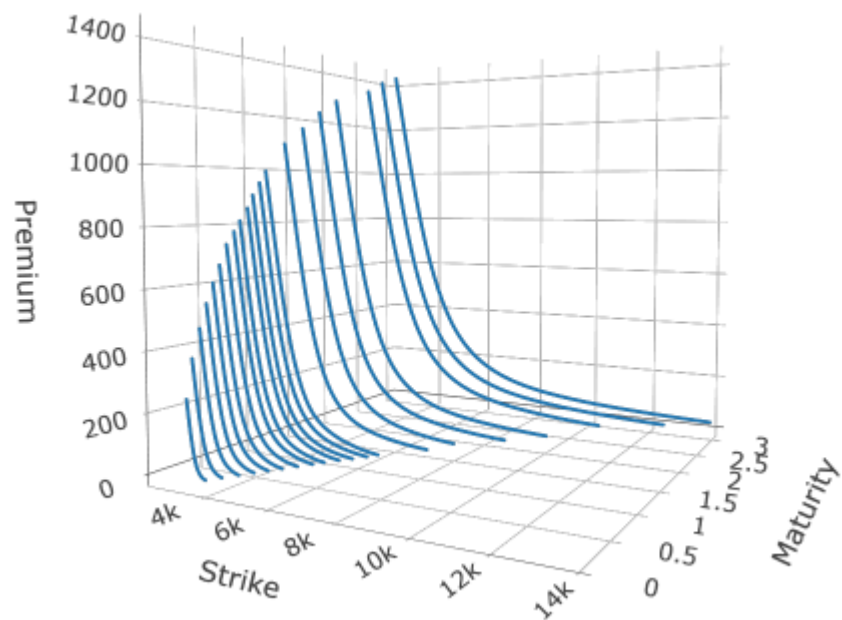


Figura 13: Interpolação C^2 em prêmio de VTC, 10/06/2016.

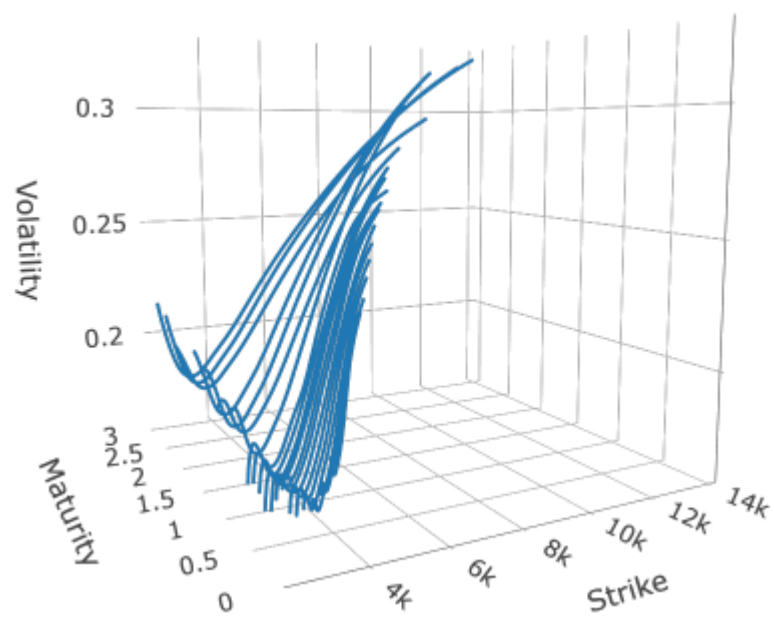


Figura 14: Interpolação C^2 em volatilidade de VTC, 10/06/2016.

5 Conclusões

- É relevante a preocupação com o rápido crescimento do número de equações no sistema gerado pela interpolação C^2 não iterativa, que não foi mencionada no artigo [9], apenas usada aqui. Porém, para a dimensão da nossa aplicação, ainda é viável usá-la, como vimos nos resultados.
- A solução numérica de sistemas de equações não lineares, apesar das várias modificações que melhoram o método de Newton, não garante a solução de qualquer sistema e depende da escolha de pontos iniciais. Para a aplicação em questão, com novos sistemas diariamente, a implementação em VBA não é uma solução consistente a ponto de ficar sem eventual supervisão, acrescentando pontos iniciais, mas é um modelo a ser levado em consideração, principalmente com a implementação em R.
- A demonstração original da convergência do método iterativo em [9] não garante a unicidade da solução, mas podemos considerar o sistema da interpolação C^2 aqui apresentado para novas tentativas de provar unicidade, já que não foram encontrados casos com mais de uma solução.
- Apesar da implementação completa do algoritmo em VBA no Excel, os já conhecidos problemas numéricos do programa ficam evidentes na comparação com o pacote `nleqslv` do R: uma quantidade significativamente maior de pontos iniciais é necessária em VBA para obter resultados parecidos com o R para a interpolação C^1 e C^2 iterativa, além de inviabilizar a interpolação C^2 .
- Para outros estudos, podemos recomendar a busca da aplicação de uma modificação da solução numérica de sistemas de equações não lineares aqui apresentada, que possa por si só sair de mínimos locais do problema de minimização que não são soluções do sistema.

Referências

- [1] J. Andreasen and B. N. Høge. Volatility interpolation. 2010.
- [2] F. Black and M. Scholes. The pricing of options and corporate liabilities. *Journal of political economy*, 81(3):637–654, 1973.
- [3] T. F. Coleman, Y. Li, A. Verma, et al. Reconstructing the unknown local volatility function. *World Scientific Book Chapters*, pages 192–215, 2001.
- [4] J. Dennis and R. Schnabel. *Numerical Methods for Unconstrained Optimization and Nonlinear Equations*. Classics in Applied Mathematics. Society for Industrial and Applied Mathematics, 1996.
- [5] M. R. Fengler. Arbitrage-free smoothing of the implied volatility surface. *Quantitative Finance*, 9(4):417–428, 2009.
- [6] P. Gope and C. P. Fries. Arbitrage-free asset class independent volatility surface interpolation on probability space using normed call prices. *Available at SSRN 1964634*, 2011.
- [7] P. S. Hagan, D. Kumar, A. S. Lesniewski, and D. E. Woodward. Managing smile risk. *The Best of Wilmott*, 1:249–296, 2002.
- [8] J. Hull. *Fundamentals of Futures and Options Markets, Global Edition*. Pearson Education Limited, 2016.
- [9] N. Kahalé. An arbitrage-free interpolation of volatilities. *Risk*, 17(5):102–106, 2004.
- [10] W. H. Press, S. A. Teukolsky, W. T. Vetterling, and B. P. Flannery. *Numerical recipes 3rd edition: The art of scientific computing*. Cambridge university press, 2007.
- [11] R. Rebonato. *Volatility and correlation: the perfect hedger and the fox*. John Wiley & Sons, 2005.
- [12] M. Roper. Arbitrage free implied volatility surfaces. *preprint*, 2010.
- [13] P. Wilmott. *Paul Wilmott introduces quantitative finance*. John Wiley & Sons, 2007.
- [14] P. Wilmott, J. Dewynne, and S. Howison. *Option Pricing: Mathematical Models and Computation*. Oxford Financial Press, 1993.

A Códigos

Para evitar colocar aqui todos os códigos usados e criados, além dos dados de operações estruturadas na seção 4.4 em toda sua precisão, deixamos tudo disponível em https://danielmoralesx.github.io/kahale_interp_vol/, de onde é possível replicar as imagens apresentadas no texto, além de aproveitar outras vistas dos gráficos tridimensionais.