PORTFÓLIOS EFICIENTES INCLUINDO OPÇÕES

Cibele Dunder

DISSERTAÇÃO APRESENTADA
AO
INSTITUTO DE MATEMÁTICA E ESTATÍSTICA
DA
UNIVERSIDADE DE SÃO PAULO
PARA OBTENÇÃO DO GRAU DE MESTRE
EM
MATEMÁTICA APLICADA

Área de Concentração: **Matemática Aplicada**
Orientador: **Prof. Dr. Júlio Michael Stern**

Durante a elaboração deste trabalho, o autor recebeu apoio do CNPq (proc. 139188/96-7) e FAPESP (proc. 96/06565-2)

- São Paulo, maio de 1998 -

Portfólios Eficientes Incluindo Opções

Este exemplar corresponde à redação final da dissertação devidamente corrigida e defendida por Cibele Dunder e aprovada pela comissão julgadora.

São Paulo, 20 de junho de 1998

Banca Examinadora:

- **Prof. Dr. Júlio Michael Stern (Orientador) - IME-USP**
- **Prof. Dr. Carlos Humes Jr. - IME-USP**
- **Prof. Dr. Marcos Eugênio da Silva - FEA-USP**

Agradecimentos

Ao Prof. Júlio M. Stern, pela paciência, amizade, confiança, orientação ...
Ao Prof. Carlos Humes Jr., pelas palavras de incentivo
Ao amigo Fábio Nakano, sempre me ajudando nas encrencas computacionais
A todos, que de uma forma ou de outra ajudaram na concretização deste trabalho,

MUITO OBRIGADA.

Aos meus pais

Sumário

Lista de Figuras

Lista de Tabelas

Resumo

Apresentamos um estudo sobre seleção de portfólios eficientes baseado nos três primeiros momentos estatísticos: média, variância e assimetria. Nossa abordagem é uniperiódica e voltada a portfólios formados por ativos fundamentais com retorno (quase) log-normal e opções européias ("calls" e "puts").

Nos primeiros capítulos fazemos uma revisão da teoria básica de portfólios, como o modelo de Markowitz, e definições de ativos e derivativos. Também deduzimos novas expressões de covariância, necessárias pela incorporação das opções, e desenvolvemos algoritmos e implementações para resolver os problemas de quadratura surgidos. Estes procedimentos foram incorporados no software de análise financeira Critical-Point.

Também aprofundamos o modelo padrão de análise de portfólios introduzindo assimetria, dando expressões de co-assimetrias e desenvolvendo algoritmos numéricos eficientes e implementações para resolver os problemas de cubatura que aparecem.

No capítulo final analisamos alguns exemplos de portfólios formados por ativos e derivativos do mercado financeiro brasileiro.

Abstract

We study efficient portfolio selection based on the three first statistical moments: mean, variance and skewness. Our approach is uniperiodic and limited to portfolios made of fundamental assets with (quasi) log-normal return and plain vanilla option (calls and puts).

The first chapters are a review of standard portfolio theory like Markowitz model, and basic derivative assets definitions. We also give the expressions for the covariance components required to include vanilla options, and develop efficient numerical algorithms and implementations to solve the required quadrature problems. Those procedures have been incorporated the financial analysis software Critical-Point.

We also further extend the standard model introducing assimetry, give the expressions for the coskewness components and develop efficient numerical algorithms and implementations to solve the required cubature problems.

In the final chapter we analyze some real scenarios in the Brazilian stock market.

Capítulo 1

Introdução

Todo problema de decisão tem alguns elementos em comum: descrição de alternativas, uma seleção de critérios de escolha dentre estas e, finalmente, a solução do problema. No mercado financeiro as decisões não são fáceis para o investidor, pois há uma gama enorme de ativos à disposição (por exemplo, commodities, moedas, etc), assim como derivativos sobre eles. Combiná-los de forma a obter o maior retorno possível, ou seja, montar um portfólio ótimo, é o objetivo de todo investidor.

Há estudos na Economia que auxiliam os investidores a estruturarem seus problemas de tal forma a minimizar o número de alternativas nestas situações. Parte destes estudos é auxiliado pela análise de média e variância em portfólios, que delinea as propriedades de uma combinação de ativos a partir do comportamento individual destes. Média e variância são ferramentas básicas na análise de risco, assim como medidas de como os retornos de diferentes ativos movem-se juntos (covariância). Mas momentos de ordem superior, como assimetria, podem também nos auxiliar na escolha de carteiras ótimas.

No capítulo 2 familiarizamos o leitor com a teoria de portfólios. Definimos os ativos e derivativos com que trabalharemos, as relações entre os investimentos e como estas relações afetam o risco e o retorno de portfólios. Trazemos algumas considerações sobre análise de portfólios baseada no modelo a dois parâmetros (média-variância), iniciado na modelo de Markowitz. Discorremos sobre as hipóteses básicas do uso desta análise E-S (expected return and standard-deviation) baseadas em restrições à função utilidade e ao comportamento de investidores. Deduzimos as expressões do valor esperado do retorno de carteiras e também de covariâncias, partindo de um modelo contínuo para os preços de ações.

O capítulo 3 amplia as discussões do capítulo anterior introduzindo o terceiro momento central do portfólio. A função utilidade é expandida em termos até terceira ordem e preferência por *skewness* é inserida no perfil do investidor. Do mesmo modo como para as covariâncias, deduzimos as expressões de co-assimetria dos retornos de ações e opções.

Os capítulos 4 e 5 põem em prática a teoria vista nos capítulos anteriores. Calculamos numericamente os momentos até terceira ordem de portfólios formados por ações e opções européias. Discutimos o porquê de certas técnicas de integração numérica terem sido usadas nestes cálculos.

Por fim, apresentamos exemplos de portfólios eficientes nos espaços $retorno \times risco$ e $retorno \times$

$risco \times assimetria$, usando ações e opções da BOVESPA.

Todas as rotinas numéricas foram feitas em linguagem C e encontram-se no apêndice A. Exceção é feita à rotina de cálculo de nós e pesos da integração gaussiana unidimensional, feita em Matlab, e ao problema de programação não linear do capítulo 3, para o qual utilizamos o solver MINOS do GAMS para resolvê-lo (vide apêndice B).

Capítulo 2

Critério Média-Variância para Portfólios

Suponha que temos uma certa quantia W de dinheiro e desejamos investí-la no mercado financeiro (por exemplo, mercado de ações). Há a possibilidade de investirmos todo o montante em ações de uma única companhia. Mas podemos também montar um portfólio, que é uma coleção de ações de várias companhias e/ou outros investimentos, como o mercado de futuros, taxas de juros, entre outros. Os elementos que compõem o portfólio chamaremos de ativos.

Podemos definir um ativo como uma oportunidade de investimento, ou seja, algo que gere dividendos. Também trabalharemos com derivativos (especificamente com contratos de opções[1]), que são instrumentos do mercado cujos valores dependem dos valores de outros ativos. Por exemplo, um contrato futuro sobre o café é um derivativo, cujo valor depende do preço de seu ativo fundamental, o café.

Mas tão importante quanto saber quais são as possibilidades de investimento é saber qual a expectativa de lucro que estas podem oferecer. Se investirmos num ativo sem risco, por exemplo caderneta de poupança, teremos um lucro certo (a porcentagem mensal de rendimento). Já o mercado de ações não pode garantir lucro certo para seu investidor, pois agora estamos falando de *ativos de risco*. Há chances de queda e de alta nos preços destes ativos. Por isso as predições dos analistas destas " chances" de queda ou de alta são de fundamental importância.

Nesta dissertação usaremos predições probabilísticas, ou seja, aquelas que se baseiam nos dados passados dos ativos para quantificar suas possibilidades futuras. Neste caso, duas medidas são de grande interesse: a predição de taxas de retorno e uma medida da incerteza[2] desta taxa (o quanto ela pode afastar-se desde valor predito).

O processo que nos leva à escolha de um portfólio, dentre as inúmeras possibilidades que o mercado oferece, envolve três fases principais:

1. *Análise de ativos e derivativos*: aqui estão incluídas predições sobre as características futuras dos ativos, um modelo (discreto ou contínuo) para o comportamento dos preços

[1] serão dados detalhes sobre estes contratos nas próximas seções
[2] as palavras risco e incerteza neste texto são sinônimas

ou cotações, correlações entre os diferentes ativos e derivativos, as incertezas envolvidas nestes modelos. Enfim, criar um modelo teórico que represente tão bem quanto possível os dados empíricos.

2. *Análise de portfólios*: ocupa-se de predições sobre o comportamento dos portfólios baseados inteiramente nos dados dos ativos da fase 1. Usamos os modelos obtidos para calcular, por exemplo, retorno, variância e assimetria de portfólios.

3. *Seleção de portfólios*: dentre o grande número de portfólios à disposição, resta fazer uma seleção baseada no perfil do investidor. Por exemplo, adaptar diferentes taxas de retorno e de risco de carteiras às diferentes tolerâncias ao risco por parte do investidor.

Para a fase 1, vamos tratar de portfólios formados por ações e contratos de opções européias sobre ações negociadas na Bolsa de Valores. Para abordar as fases 2 e 3, neste capítulo, usaremos o modelo de Markowitz, que utiliza os dois primeiros momentos centrais do retorno do portfólio. No próximo capítulo incluiremos o terceiro momento central (assimetria) em nossa análise.

2.1 Análise de Ativos e Derivativos

2.1.1 Um Modelo de Preços para Ações

Quando investimos num ativo de risco, o que mais gostaríamos de saber é a **expectativa** do retorno deste investimento, já que não sabemos qual será a cotação destes ativos daqui a um período τ qualquer.

Para podermos quantificar esta expectativa precisamos adotar um modelo que represente o comportamento dos preços das ações ao longo do tempo. E o modelo que adotaremos é o da distribuição log-normal dos preços, e é com ele que trabalharemos daqui por diante.

Há dois parâmetros chave para descrever o comportamento dos preços quando a hipótese de log-normalidade é adotada: retorno esperado e desvio-padrão. As estimativas destes parâmetros podem ser feitas com base em séries históricas das cotações dos ativos, através de uma filtragem de dados.

Vamos ver em linhas gerais um exemplo deste processo.

Suponha que estamos na data t_0 e queremos obter uma estimativa do retorno e do desvio padrão do preço do ativo i para daqui a $\tau = T - t_0$ dias ($T > t_0$). Os retornos considerados no modelo são contínuos.

Denotando por $S(t,i)$ e $r(t,i)$ o preço e o retorno logarítmico do ativo em questão, calculamos

$$r(t,i) = \ln\left(\frac{S(t+1,i)}{S(t,i)}\right) \quad (t < t_0)$$

para uma amostra de N preços retroativos à data t_0. Em seguida vem o cálculo do retorno médio

$$med(i) = \frac{1}{N}\sum_t r(t,i)$$

e o desvio padrão

$$dp(i) = \sqrt{\frac{1}{N}\sum_t (r(t,i) - med(i))^2}.$$

Vem então o processo de extrapolação dos valores $med(i)$ e $dp(i)$ para a data T que estamos interessados:

$$\begin{aligned} \mu_i = & \text{ extrapolação de } med(i) = med(i).T \\ \sigma_i = & \text{ extrapolação de } dp(i) = dp(i).\sqrt{T} \end{aligned}$$

Os valores μ_i e σ_i serão os parâmetros da nossa distribuição log-normal para o ativo i, isto é,

$$x_i = \ln\left(\frac{S(T,i)}{S(t_0,i)}\right) \cong N(\mu_i, \sigma_i^2)$$

(x_i, o retorno logarítmico para o período τ, tem, aproximadamente, uma distribuição normal com média μ_i e variância σ_i^2).

Uma outra grandeza essencial é a covariância destes retornos logarítmicos para dois ou mais ativos diferentes.

Seja $cov(i,j)$ a covariância entre o logaritmo dos preços dos ativos i e j e $corr(i,j)$ o coeficiente de correlação entre eles. Temos

$$cov(i,j) = \frac{1}{N}\sum_t (r(t,i) - med(i)) * (r(t,j) - med(j)),$$

$$dp(i) = \sqrt{cov(i,i)}$$

e

$$corr(i,j) = \frac{cov(i,j)}{dp(i) * dp(j)}$$

A respectiva função densidade de probabilidade da variável aleatória x_i, de distribuição normal, é dada por

$$f(x_i) = \frac{1}{\sigma_i\sqrt{2\pi}} \exp\left\{-\frac{(x_i-\mu_i)^2}{2\sigma_i^2}\right\} \quad (-\infty < x_i < +\infty)$$

Para o caso de estarmos trabalhando com dois ou mais ativos diferentes, também temos a função densidade de probabilidade conjunta

$$\begin{aligned} f(x_i, x_j) &= \frac{1}{2\pi\sigma_i\sigma_j\sqrt{1-corr(i,j)^2}} \cdot \\ &\exp\left\{-\frac{1}{2(1-corr(i,j)^2)}\left[\left(\frac{x_i-\mu_i}{\sigma_i}\right)^2 - 2corr(i,j)\frac{x_i-\mu_i}{\sigma_i}\frac{x_j-\mu_j}{\sigma_j} + \left(\frac{x_j-\mu_j}{\sigma_j}\right)^2\right]\right\} \end{aligned}$$

$$(-\infty < x_i, x_j, +\infty)$$

Todos estes dados de média, variância e coeficientes de correlação serão de grande utilidade nas análises da fase 2, pois é a partir deles que geraremos o retorno e o risco de portfólios.

2.1.2 O Mercado de Opções

Um contrato de opção é um instrumento que dá ao seu comprador (titular ou portador) o direito futuro sobre algo, e não uma obrigação; e a seu vendedor uma obrigação futura, se esta for solicitada pelo comprador.

Nas opções de compra (ou "call") o portador tem o direito de comprar algo, numa data futura, pelo preço estipulado no contrato (chamado preço de exercício). Nas opções de venda (ou " put"), o titular tem o direito de vender algo numa data futura, pelo preço de exercício do contrato.

Esse algo que o titular pode comprar ou vender pode ser, a priori, qualquer ativo ou derivativo do mercado; pode ser desde uma ação da Telebrás até contratos futuros ou mesmo um outro contrato de opção.

A data futura em que se pode exercer os direitos de compra e venda dão margem a dois tipos de opções: opção européia e opção americana.

Na opção européia o titular pode exercer o seu direito somente na data de expiração do contrato. A opção americana já dá liberdade ao portador de exercer o seu direito em qualquer data que anteceda a expiração do contrato.

Particularmente, trabalharemos nesta dissertação somente com opções européias que tenham como ativos fundamentais ações negociadas na Bolsa (são as ações-objeto do contrato).

Em resumo, uma opção de compra européia confere ao seu titular o direito de comprar as ações-objeto, ao preço de exercício, na data de expiração do contrato. O lançador da opção de compra assume a obrigação de vender as ações-objeto a que se refere a opção, pelo preço de exercício, se o comprador o requerer. Da mesma forma, uma opção de venda européia dá ao seu titular o direito de vender as ações-objeto, na data de vencimento, pelo preço de exercício do contrato. O lançador assume o compromisso de comprar as ações caso seja exercido o direito do titular.

Em razão dos direitos e obrigações do contrato, o comprador paga ao lançador uma quantia denominada prêmio, que é o preço da opção. Este preço leva em consideração a oferta e demanda do ativo, a diferença entre o preço à vista e o preço de exercício da ação-objeto, a volatilidade dos preços, etc. Os ativos passíveis de negociação em opções são definidos pela instituição que as negocia, assim como o mês de expiração do contrato e os preços de exercício.

Como notação, usaremos:

T :	data de expiração do contrato;
$S(t)$:	preço do ativo básico na data t;
$K_c(t), K_p(t)$:	preço de exercício de uma *call* e uma *put*, respectivamente, na data t;
$P_c(t), P_p(t)$:	prêmio de uma call e uma put, respectivamente, na data t.

Uma opção de compra européia pode ser vista como um derivativo que paga ao portador, na data T, a quantia

$$X(T) = \max(S(T) - K_c(t_0), 0) \quad (\geq 0)$$

onde t_0 é a data de assinatura do contrato.

Se o preço de mercado do ativo básico for maior que o preço especificado no contrato, ou seja, $S(T) > K_c(t_0)$, então o portador do contrato pode comprar as ações pelo preço $K_c(t_0)$, em exercício do seu direito, e vendê-la imediatamente no mercado, lucrando $S(T) - K_c(t_0) > 0$. Claro que este " lucro" pode não compensar o valor pago pela aquisição da opção, o que implicaria numa taxa de retorno negativa.

Se $S(T) \leq K_c(t_0)$, perder-se-á $S(T) - K_c(t_0) \leq 0$ se o direito de compra for exercido. Neste caso, deixa-se de exercer o direito na data T.

Analogamente, uma opção de venda européia pode ser vista como um derivativo cujo valor na data de vencimento é dado por

$$X_T = \max(K_p(t_0) - S(T), 0) \quad (\geq 0)$$

Vamos agora especificar taxas de retorno para o caso de ações e opções européias, com as quais vamos trabalhar até o final desta dissertação.

Daqui por diante omitiremos a palavra européia quando falarmos de opções, ficando implícito que é sobre este tipo de contrato que estamos falando.

Para uma ação, cuja cotação na data t é dada por $S(t)$, a taxa de retorno r_s no período $\tau = T - t$ $(T > t)$ será dada por

$$r_s = \frac{S(T) - S(t)}{S(t)} \tag{2.1}$$

Para as opções de compra e venda, as taxas de retorno serão dadas por

$$r_c = \frac{\max(S(T) - K_c(t_0), 0) - P_c(t_0)}{P_c(t_0)} \quad (\text{“call”}) \tag{2.2}$$

$$r_p = \frac{\max(K_p(t_0) - S(T), 0) - P_p(t_0)}{P_p(t_0)} \quad (\text{“put”}) \tag{2.3}$$

onde t_0 é a data de assinatura dos contratos.

2.1.3 Um Exemplo Didático de Portfólio

Vamos considerar um portfólio bem simples formado por dois ativos básicos, chamados de ativo A e ativo B, e por opções de compra e venda que têm como ações-objetos os próprios ativos A e B. Vamos supor também que as datas de assinatura (t_0) e as datas de expiração (T) são as mesmas para todas as opções.

Denote por S_a e S_b os preços dos ativos A e B, respectivamente. Os retornos logarítmicos de A e B, extrapolados para a data T, são dados por

$$x = ln\left(\frac{S_a(T)}{S_a(t_0)}\right) \cong N(\mu_a, \sigma_a^2) \tag{2.4}$$

e

$$y = ln\left(\frac{S_b(T)}{S_b(t_0)}\right) \cong N(\mu_b, \sigma_b^2) \tag{2.5}$$

e as funções densidade de probabilidade dadas por

$$f(x) = \frac{1}{\sigma_a\sqrt{2\pi}} \exp\left\{-\frac{(x-\mu_a)^2}{2\sigma_a^2}\right\} \tag{2.6}$$

$$f(y) = \frac{1}{\sigma_b\sqrt{2\pi}} \exp\left\{-\frac{(y-\mu_b)^2}{2\sigma_b^2}\right\} \tag{2.7}$$

Sendo ρ o coeficiente de correlação entre os ativos A e B, a função distribuição conjunta é dada por

$$f(x,y) = \frac{1}{2\pi\sigma_a\sigma_b\sqrt{1-\rho^2}} \exp\left\{-\frac{1}{2(1-\rho^2)}\left[\left(\frac{x-\mu_a}{\sigma_a}\right)^2 - 2\rho\frac{x-\mu_a}{\sigma_a}\frac{y-\mu_b}{\sigma_b} + \left(\frac{y-\mu_b}{\sigma_b}\right)^2\right]\right\}$$

$$(-\infty < x, y < +\infty)$$

Precisamos colocar as taxas de retorno dadas em 2.1, 2.2 e 2.3 em função das variáveis x e y, pois é destas variáveis que temos informações sobre a densidade de probabilidade.

O motivo pelo qual não trabalharemos com retornos logarítmicos para opções é a simplificação dos cálculos das covariâncias entre estes retornos, como veremos nas próximas seções.

De 2.4 e 2.5 temos

$$S_a(T) = S_a(t_0)exp(x) \quad \text{e} \quad S_b(T) = S_b(t_0)exp(y) \tag{2.8}$$

Substituindo 2.8 em 2.1 teremos

$$r_{s_a} = exp(x) - 1 \quad \text{e} \quad r_{s_b} = exp(y) - 1 \tag{2.9}$$

onde r_{s_a} e r_{s_b} são as taxas de retorno dos ativos A e B, respectivamente.

Para as opções fixaremos a letra K para denotar preço de exercício do contrato, colocando um índice superior à letra para diferenciar calls de puts, e um índice inferior para indicar a ação-objeto a que se refere. Por exemplo, $K_a^c(t)$ denota o preço de exercício de uma call sobre o ativo A, na data t, e $K_b^p(t)$ indica o preço de exercício de uma put sobre o ativo B, na data t.

Os prêmios das opções de compra serão denotados pela letra C, com um índice inferior indicando o ativo a que se refere. O mesmo vale para as opções de venda, trocando a letra C pela letra P. Por exemplo, $C_a(t)$ indica o prêmio de uma call sobre o ativo A na data t, e $P_b(t)$ indica o prêmio de uma put sobre o ativo B, na data t.

Portanto, as taxas de retorno das opções de compra (assinadas na data t_0) que têm A e B como ações-objeto serão dadas por

$$\begin{aligned} r_{c_a} = &\quad (\max(S_a(t_0)e^x - K_a^c(t_0), 0) - C_a(t_0))/C_a(t_0) \\ r_{c_b} = &\quad (\max(S_b(t_0)e^y - K_b^c(t_0), 0) - C_b(t_0))/C_b(t_0) \end{aligned} \tag{2.10}$$

Do mesmo modo para as puts,

$$\begin{aligned} r_{p_a} = &\quad (\max(K_a^p(t_0) - S_a(t_0)e^x, 0) - P_a(t_0))/P_a(t_0) \\ r_{p_b} = &\quad (\max(K_b^p(t_0) - S_b(t_0)e^y, 0) - P_b(t_0))/P_b(t_0) \end{aligned} \tag{2.11}$$

Quando não houver dúvidas quanto as datas que estamos usando, omitiremos a letra t dos prêmios, dos preços de exercício e das cotações dos ativos para não sobrecarregar a notação.

Uma outra forma de escrever as expressões 2.10 e 2.11 é definir os domínios das v.a. x e y nos quais o termo $\max(S_0 exp(x) - K, 0)$ é positivo.

$$\begin{aligned} \max(S\exp(x) - K, 0) > 0 &\iff x > \ln\left(\tfrac{K}{S}\right) \\ \max(K - S\exp(x), 0) > 0 &\iff x < \ln\left(\tfrac{K}{S}\right) \end{aligned} \tag{2.12}$$

Considerando 2.12 nas expressões 2.10 e 2.11, redefinimos as taxas de retorno das opções por

$$\begin{aligned} r_{c_a} &= \begin{cases} \frac{1}{C_a(t_0)}\left(S_a(t_0)e^x - K_a^c(t_0) - C_a(t_0)\right) & \text{se} \quad \ln\left(\frac{K_a^c(t_0)}{S_a(t_0)}\right) < x < +\infty \\ -1 & \text{c.c.} \end{cases} \\ r_{c_b} &= \begin{cases} \frac{1}{C_b(t_0)}\left(S_b(t_0)e^y - K_b^c(t_0) - C_b(t_0)\right) & \text{se} \quad \ln\left(\frac{K_b^c(t_0)}{S_b(t_0)}\right) < y < +\infty \\ -1 & \text{c.c.} \end{cases} \\ r_{p_a} &= \begin{cases} \frac{1}{P_a(t_0)}\left(K_a^p(t_0) - S_a(t_0)e^x - P_a(t_0)\right) & \text{se} \quad -\infty < x < \ln\left(\frac{K_a^p(t_0)}{S_a(t_0)}\right) \\ -1 & \text{c.c.} \end{cases} \\ r_{p_b} &= \begin{cases} \frac{1}{P_b(t_0)}\left(K_b^p(t_0) - S_b(t_0)e^y - P_b(t_0)\right) & \text{se} \quad -\infty < y < \ln\left(\frac{K_b^p(t_0)}{S_b(t_0)}\right) \\ -1 & \text{c.c.} \end{cases} \end{aligned} \tag{2.13}$$

Sendo ρ o coeficiente de correlação entre os preços dos ativos A e B, $\rho \neq \pm 1$, a função densidade de probabilidade conjunta de x e y é dada por

$$f(x,y) = \frac{1}{2\pi\sigma_a\sigma_b\sqrt{1-\rho^2}} \exp\left\{-\frac{1}{2(1-\rho^2)}\left[\left(\frac{x-\mu_a}{\sigma_a}\right)^2 - 2\rho\frac{x-\mu_a}{\sigma_a}\frac{y-\mu_b}{\sigma_b} + \left(\frac{y-\mu_b}{\sigma_b}\right)^2\right]\right\}$$

Nas próximas seções derivaremos as expressões dos *valores esperados* das taxas dadas em 2.13 e das covariâncias entre elas, para o caso deste portfólio exemplo. Isto será usado no cômputo do retorno esperado (R_p) e da variância (σ_p) do portfólio.

Mas, antes, vamos abordar o modelo de Markowitz para análise de portfólios e inserir R_p e σ_p neste contexto.

2.2 O Modelo de Markowitz

Quantificar uma medida para o risco de um investimento é tarefa perseguida há muito tempo, mas foi nos anos 50 que Harry M. Markowitz definiu um retorno esperado e uma medida para o risco de um portfólio. Ele mostrou que a variância do retorno era uma medida significativa do risco do portfólio, sob determinadas hipóteses. Também definiu fórmulas para o cálculo desta variância.

As hipóteses básicas nas quais o modelo está baseado referem-se ao comportamento do investidor.

2.2.1 Preferências de Risco e Função Utilidade

Quando as pessoas investem em ações ou outros ativos de risco, elas esperam obter um retorno maior do que elas teriam se tivessem investido num ativo livre de risco.

Investimentos que ofereçam uma alta taxa de retorno vêm sempre acompanhados de uma alta taxa de risco. E nem todas as pessoas estão dispostos a correr altos riscos.

De um modo geral, podemos caracterizar os níveis de propensão ao risco por parte do investidor de três formas:

- aversão ao risco: o indivíduo prefere um retorno menor, porém mais garantido do que arriscar-se muito para ter chance de altos retornos;
- propensão ao risco: o indivíduo está mais interessado em maximizar os lucros, não se importando muito com as chances de perdas;
- neutralidade ao risco: o indivíduo não se importa muito em obter um resultado certo ou participar de um jogo, desde que os dois ofereçam o mesmo retorno.

O conceito de *utilidade* serve para descrever estas preferências dos indivíduos. Utilidade é o nível de satisfação que uma pessoa (no caso o investidor) tem ao fazer um certo investimento (ou consumir um bem ou exercer uma atividade). Já a *função utilidade* mede numericamente esta satisfação.

No caso das 3 categorias citadas acima, uma pessoa que tenha aversão ao risco terá uma função utilidade côncava em relação à origem, no espaço $utilidade \times retorno$ (fig2.1 (a)). Já para o indivíduo propenso ao risco, a função será côncava em relação à origem (fig2.1 (b)). E para o neutro ao risco, a função será linear (fig2.1 (c)).

Uma hipótese importante na teoria de portfólios é esperar que o investidor queira maximizar os retornos de seus investimentos, para uma dada taxa de risco. E considerar também que os ativos e derivativos formadores da carteira interagem entre si, para a partir daí quantificarmos o risco total do portfólio.

O modelo de Markowitz baseia-se ainda nas seguintes hipóteses:

1. investidores consideram cada investimento representado por uma distribuição de probabilidade dos retornos esperados para um *único período*;

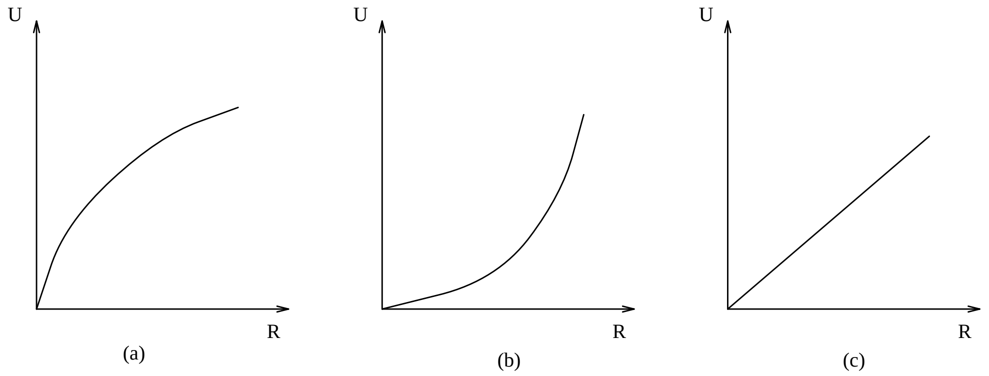

Figura 2.1: utilidade

2. investidores maximizam a função utilidade esperada;
3. investidores estimam o risco do portfólio baseados na variabilidade dos retornos;
4. as curvas utilidade são funções do retorno esperado e da variância do portfólio;
5. para um dado nível de risco, investidores preferem altos retornos a baixos. Analogamente, para um dado nível de retorno esperado, há preferências por menores riscos.

Sob estas considerações, um portfólio é considerado *eficiente* se não há outro que ofereça um retorno esperado maior com a mesma taxa de risco; ou se não há outro que ofereça menor risco à uma mesma taxa de retorno esperado.

Não especificaremos qual a função utilidade U do retorno a ser usada na seleção de portfólios (isto fica a critério do investidor). Mas como partiremos do princípio que estamos trabalhando com investidores aversos ao risco, podemos especificar algumas propriedades de U:

(a) $U^{'}(R_p) > 0$ (utilidade marginal da riqueza[3] é positiva)

(b) $U^{''}(R_p) < 0$ (utilidade marginal da riqueza decresce com um crescimento da riqueza)

(c) $\frac{d}{dR_p}[-U^{''}(R_p)/U^{'}(R_p)] \leq 0$ (aversão ao risco absoluta marginal decrescendo com o crescimento da riqueza)

Portanto, vamos considerar um investidor que procure maximizar o valor esperado de alguma função utilidade $U = U(R_p)$, onde R_p é o retorno do portfólio no período considerado.

Markowitz usou uma aproximação quadrática da função utilidade para gerar um conjunto de portfólios eficientes no espaço $retorno\ esperado \times risco$ ($E[R_p] \times \sigma_p$). A aproximação é baseada na expansão em série de Taylor de U em torno do retorno esperado ($E_p = E[R_p]$) do portfólio[4]:

[3]Neste caso, a riqueza é o retorno obtido do portfólio
[4]veremos adiante como calcular a esperança do retorno no caso do portfólio didático.

$$U(R_p) \cong U(E_p) + U^{'}(E_p)(R_p - E_p) + \frac{1}{2}U^{''}(E_p)(R_p - E_p)^2 \tag{2.14}$$

e portanto,

$$E[U(R_p)] \cong U(E_p) + \frac{1}{2}U^{''}(E_p)\sigma_p^2 \tag{2.15}$$

onde E é o operador esperança.

Neste caso, a utilidade esperada do portfólio foi aproximada por uma função que só depende do retorno esperado e da variância do portfólio. Se considerarmos que a aproximação 2.15 é razoável [5], é mais conveniente determinar os portfólios eficientes no espaço $E[R_p] \times \sigma_p$ do que no espaço $U(R_p) \times \sigma_p$. Passaremos a trabalhar com a função

$$f(E_p, \sigma_p^2) = \eta E_p - \sigma_p^2 \tag{2.16}$$

Um portfólio eficiente, obtido pela maximização de f, que minimiza a variância para um dado retorno esperado fornece uma boa aproximação para a máxima utilidade esperada $E[U(R_p)]$.

A constante η é interpretada como um parâmetro de ponderação entre o retorno esperado e o risco do portfólio. É um parâmetro de propensão ao risco. Quanto mais alto o valor de η, mais importante é o retorno esperado em comparação à variância do portfólio para o investidor.

Há convenientes em se trabalhar com a análise $E[R_p] \times \sigma_p$ no lugar de maximizar diretamente a função utilidade:

1. facilidade na obtenção de máximos e mínimos de funções convexas;
2. generalidade do modelo, no sentido deste adaptar-se a mais de um tipo de função utilidade (quando esta função utilidade pode ser aproximada razoavelmente por uma função quadrática).

A figura 2.2 ilustra diferentes graus de aversão ao risco, mas todos os gráficos referem-se a indivíduos aversos ao risco.

As retas dos gráficos (a), (b) e (c) representam curvas[6] de indiferença, ao longo das quais o mesmo nível de utilidade (ou satisfação) é obtido. Mas quanto mais à esquerda está a reta, maior é a utilidade que ela representa. O gráfico (a) representa uma aversão ao risco maior do que os gráficos (b) e (c). Mas em todos eles o objetivo do investidor é sempre maximizar a utilidade, ou seja, minimizar os valores de λ_i.

Para um ponto $(e, v) \in (E_p \times \sigma_p^2)$

$$v - \lambda_i = \frac{1}{\tan(\alpha_j)} = \eta_j e \qquad (0 < \alpha_j < \frac{\pi}{2})$$

[5] isto vai depender de quão bem comportada for a função utilidade para admitir uma aproximação quadrática suficientemente boa.

[6] No espaço $E_p \times \sigma_p^2$

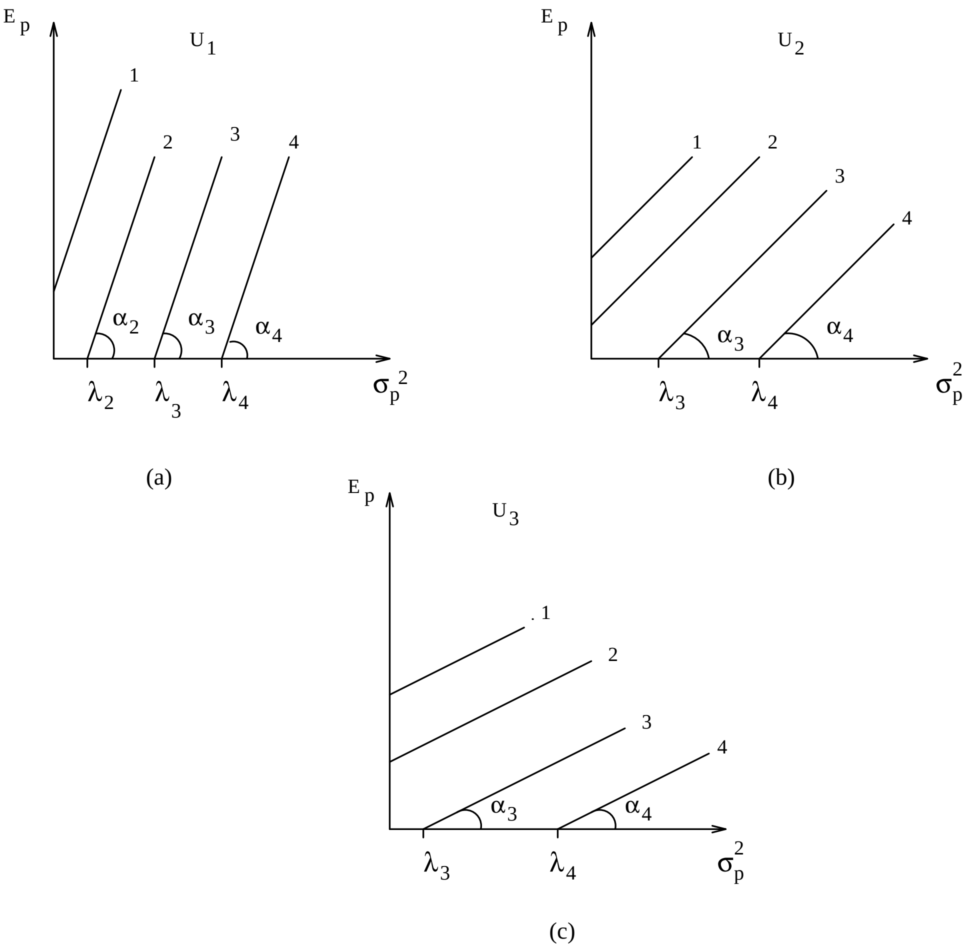

Figura 2.2: curvas (linhas) de indiferença no espaço $\sigma_p^2 \times E_p$

ou ainda

$$\lambda_i = v - \eta_j e$$

onde $\eta_j = \frac{1}{\tan(\alpha_j)}$ $\qquad (0 < \eta_j < +\infty)$.

Portanto, o problema de encontrar portfólios eficientes no espaço $E[R_p] \times \sigma_p^2$ resume-se a minimizar os valores de λ, isto é,

$$\min\{-\eta e + v\}$$

ou

$$\max\{\eta E_p - \sigma_p^2\}$$

2.2.2 Risco e Retorno de Portfólios

Seja P um portfólio formado por n ativos (aqui o ativo pode ser uma ação, uma opção,etc). Denotemos por w_i a fração do total do investimento W do portfólio atribuido ao ativo i. Temos, então, a restrição

$$\sum_{i=1}^{n} w_i = 1 \tag{2.17}$$

Denote por R_p o retorno do portfólio e por R_i a taxa de retorno do i-ésimo ativo.

R_p será a média ponderada dos retornos dos ativos

$$R_p = \sum_{i=1}^{n} w_i R_i$$

Denote por σ_p^2 a variância do portfólio P. Vamos supor que conhecemos as covariâncias (σ_{ij}) entre as taxas de retorno dos ativos i e j $(i, j = 1..n)$.

Da mesma forma que calculamos a variância dos ativos pela expressão

$$\sigma_i^2 = E[(R_i - E(R_i))^2]$$

podemos calcular

$$\sigma_p^2 = E[(R_p - E(R_p))^2]$$

Denote $E(R_i)$ por $\bar{R}_i$.

De 2.17 e das propriedades do operador E temos

$$\sigma_p^2 = E[(\sum_{i=1}^{n} w_i R_i - E(\sum_{i=1}^{n} w_i R_i))^2] = E[(\sum_{i=1}^{n} w_i R_i - \sum_{i=1}^{n} w_i \bar{R}_i)^2] = E[(\sum_{i=1}^{n} w_i (R_i - \bar{R}_i))^2]$$

Para o caso particular $n = 2$ temos

$$\sigma_p^2 = \sum_{i=1}^{2} w_i^2 \sigma_i^2 + 2 \sum_{i=1}^{2} \sum_{j=1, j \neq i}^{2} w_i w_j \sigma_{ij}$$

Por indução, para um portfólio com n ativos, sua variância é dada por

$$\sigma_p^2 = \sum_{i=1}^{n} w_i^2 \sigma_i^2 + 2 \sum_{i=1}^{n} \sum_{j=1, j \neq i}^{n} w_i w_j \sigma_{ij} \tag{2.18}$$

Podemos expressar 2.18 de forma matricial. Denote por $\mathbf{w}$ o vetor de pesos do portfólio, $\bar{\mathbf{z}}$ o vetor de retornos esperados de cada ativo do portfólio e $\mathbf{\Sigma}$ a matriz de covariância destes ativos. Desta forma, o retorno esperado do portfólio e sua variância pode ser escrito como

$$\begin{aligned} \bar{R}_p &= \mathbf{w}'\bar{\mathbf{z}} \\ \sigma_p^2 &= \mathbf{w}'\mathbf{\Sigma}\mathbf{w} \end{aligned} \tag{2.19}$$

onde $\mathbf{w}, \bar{\mathbf{z}} \in \Re^n, \mathbf{\Sigma} \in \Re^{n \times n}$ e o símbolo ($'$) indica operação de transposição.

Sendo $\mathbf{1}$ o vetor $n \times 1$ onde cada elemento do vetor é o número 1, a restrição 2.17 é dada por

$$\mathbf{w}'\mathbf{1} = 1$$

2.2.3 Seleção de Portfólios

Agora já temos todas as informações necessárias para que o investidor, de acordo com o seu perfil, possa selecionar um portfólio. Temos variáveis de decisão $w_1..w_n$, a função objetivo a ser minimizada, dada por

$$-\eta \sum_{i=1}^{n} w_i \bar{R}_i + \sum_{i=1}^{n} \sum_{j=1,j}^{n} w_i w_j \sigma_{ij}$$

e a restrição linear

$$\sum_{i=1}^{n} w_i = 1$$

Podemos ter outras restrições lineares, por exemplo geradas por limitações de aquisição de ativos impostas pelo próprio mercado financeiro, da forma

$$Ew = e \qquad \text{e} \qquad Dw \leq d$$

onde d e e são vetores de limitações de investimento para determinadas combinações de ativos, dadas pelas matrizes E e D.

Sem perda de generalidade, admitiremos sempre quantidades não negativas de cada ativo. Assim, o problema de programação matemática que obtemos neste modelo é o seguinte:

$$\begin{cases} \min & -\eta\mathbf{w}'\bar{\mathbf{z}} + \mathbf{w}'\mathbf{\Sigma}\mathbf{w} \\ s.a & D\mathbf{w} \leq d \\ & E\mathbf{w} = e \\ & \mathbf{w}'\mathbf{1} = 1 \\ & \mathbf{w} \geq 0 \end{cases} \tag{2.20}$$

Este é um problema quadrático paramétrico e cada solução ótima representa um portfólio cujo retorno esperado e variância são dados por $R_p(\eta) = w(\eta)'\bar{\mathbf{z}}$ e $\sigma_p^2(\eta) = w(\eta)'\mathbf{\Sigma}w(\eta)$. A curva $\bar{R}_p(\eta)$ versus $\sigma_p(\eta)$ é denominada *fronteira eficiente* do modelo.

Obter os valores $\bar{\mathbf{z}}$ e σ_p^2 não é tarefa fácil no caso de portfólios mistos (ativos e derivativos quaisquer).

Mas para o nosso portfólio didático formado por ações e opções européias, calcularemos as expressões da esperança das taxas dadas em 2.13 e suas respectivas covariâncias. É o que faremos a seguir.

2.2.4 Expressões do Valor Esperado das Taxas de Retorno e Covariâncias de Ações e Opções

Vamos voltar ao portfólio didático da seção 2.1.3, formado pelos ativos A e B e por opções sobre estes ativos. Manteremos a notação introduzida naquela seção, com a exceção de que omitiremos o termo t_0 (data de assinatura das calls e das puts). Portanto os preços das ações serão dados por S_a e S_b, os prêmios das opções por C_a, C_b, P_a e P_b e os preços de exercício por K_a^c, K_a^p, K_b^c e K_b^p. Recomenda-se voltar àquela seção para rever as definições e notações usadas .

Vale lembrar ainda que dada uma v.a contínua x em $\Re$ e sua respectiva função densidade de probabilidade $p(x)$, a esperança de uma função $g(x)$ é dada por

$$E[g(x)] = \int_{-\infty}^{+\infty} g(x)p(x)\,dx$$

Em virtude das repetições dos cálculos algébricos que envolvem a dedução das expressões das covariâncias, detalhamos esses cálculos em algumas delas. As demais usam raciocínio análogo.

Esperança das taxas de retorno

Notação: $\mathrm{erf(a)} = \int_a^{+\infty} \exp(-\psi^2)\,d\psi$

$$\begin{aligned}
\mu_{sa} &= E[r_{s_a}] = \int_{-\infty}^{+\infty} \exp(x) f(x)\,dx - 1 \\
&= \frac{1}{\sigma_a\sqrt{2\pi}} \int_{-\infty}^{+\infty} \exp\left\{x - \frac{(x-\mu_a)^2}{2\sigma_a^2}\right\} dx - 1 \\
&\overset{1''}{=} \frac{\exp\left(\frac{1}{2}\sigma_a^2 + \mu_a\right)}{\sqrt{\pi}} \int_{-\infty}^{+\infty} \exp\left(-t^2\right)\,dt - 1 \\
&= \exp\left(\mu_a + \frac{1}{2}\sigma_a^2\right) - 1
\end{aligned}$$

$$
\begin{aligned}
\mu_{ca} &= E[r_{c_a}] = \frac{1}{C_a}\int_{\ln\left(\frac{K_a^c}{S_a}\right)}^{+\infty} (S_a \exp(x) - K_a^c) f(x)\, dx - 1 \\
&= \frac{S_a}{C_a\sigma_a\sqrt{2\pi}}\int_{\ln\left(\frac{K_a^c}{S_a}\right)}^{+\infty} \exp\left\{x - \frac{(x-\mu_a)^2}{2\sigma_a^2}\right\} dx \\
&\quad -\frac{K_a^c}{C_a\sigma_a\sqrt{2\pi}}\int_{\ln\left(\frac{K_a^c}{S_a}\right)}^{+\infty} \exp\left\{\frac{(x-\mu_a)^2}{2\sigma_a^2}\right\} dx - 1 \\
&\overset{1''}{=} \frac{S_a}{2C_a}\exp\left(\mu_a + \frac{1}{2}\sigma_a^2\right)\left\{1 - erf\left[\frac{1}{\sigma_a\sqrt{2}}\left(\ln\left(\frac{K_a^c}{S_a}\right) - \mu_a - \sigma_a^2\right)\right]\right\} \\
&\quad -\frac{1}{2C_a}(K_a^c)\left\{1 - erf\left[\frac{1}{\sigma_a\sqrt{2}}\left(\ln\left(\frac{K_a^c}{S_a}\right) - \sigma_a^2\right)\right]\right\} - 1
\end{aligned}
$$

Mudança de variável usada:

$$
1'' : \begin{cases} t &= \frac{1}{\sigma_a\sqrt{2}}(x - \mu_a - \sigma_a^2) \\ \\ w &= \frac{1}{\sigma_a\sqrt{2}}(x - \sigma_a) \end{cases}
$$

$$
\begin{aligned}
\mu_{pa} &= E[r_{p_a}] = \frac{1}{P_a}\int_{-\infty}^{\ln\left(\frac{K_a^p}{S_a}\right)} (K_a^p - S_a \exp(x)) f(x)\, dx - 1 \\
&\overset{1''}{=} \frac{1}{2P_a}(K_a^p)\exp\left(\mu_a + \frac{1}{2}\sigma_a^2\right)\left\{1 + erf\left[\frac{1}{\sigma_a\sqrt{2}}\left(\ln\left(\frac{K_a^p}{S_a}\right) - \mu_a - \sigma_a^2\right)\right]\right\} \\
&\quad -\frac{S_a}{2P_a}\left\{1 + erf\left[\frac{1}{\sigma_a\sqrt{2}}\left(\ln\left(\frac{K_a^p}{S_a}\right) - \sigma_a^2\right)\right]\right\} - 1
\end{aligned}
$$

Covariâncias

No lugar de calcularmos as expressões de coriâncias para as taxas de retorno simples dadas em 2.13, deduziremos expressões para o retorno simples ($= \frac{\text{preço final}}{\text{preço inicial}}$) de ações e opções, sendo a diferença entre as duas grandezas apenas uma constante, isto é,

Taxa de retorno simples = Retorno simples - 1.

O resultado numérico de covariâncias não muda, visto que para duas v.a. X e Y e constantes k_1 e K_2 vale a igualdade

$$cov(X + k_1, Y + k_2) = cov(X, Y).$$

- $Cov(r_{s_a}, r_{s_b})$

$$\begin{aligned} cov(r_{s_a}, r_{s_a}) &= var(r_{s_a}) \\ &= E_{2x} - E_x^2 \end{aligned}$$

$$\begin{aligned} E_{2x} &= \int_{-\infty}^{+\infty} exp(2x) f(x)\, dx \\ &= \frac{1}{\sigma_a \sqrt{2\pi}} \int_{-\infty}^{+\infty} \exp\left\{ 2x - \frac{(x-\mu_a)^2}{2\sigma_a^2} \right\} dx \\ &\overset{2''}{=} \frac{1}{\sqrt{\pi}} \exp\left(2\mu_a + 2\sigma_a^2\right) \int_{-\infty}^{+\infty} \exp(-\psi^2)\, d\psi \\ &= \exp\left(2\mu_a + 2\sigma_a^2\right) \end{aligned}$$

$$E_x = \exp\left(\mu_a + 0.5\sigma_a^2\right)$$

onde 2" indica a mudança de variável

$$\psi = \frac{x - \mu_a - 2\sigma_a^2}{\sigma_a \sqrt{2}}$$

Logo,

$$var(r_{s_a}) = \exp\left(2\mu_a + \sigma_a^2\right)\left(\exp(\sigma_a^2) - 1\right)$$

- $Cov(r_{s_a}, r_{s_b})$

$$\begin{aligned} cov(r_{s_a}, r_{s_b}) &= E[r_{s_a} r_{s_b}] - E[r_{s_a}]E[r_{s_b}] \\ &= E_{xy} - E_x E_y \end{aligned}$$

sendo

$$\begin{aligned}
E_{xy} &= \int_{-\infty}^{+\infty}\int_{-\infty}^{+\infty} f(x,y)\exp(x+y)\,dy\,dx \\
&= \frac{1}{2\pi\sigma_a\sigma_b\sqrt{1-\rho^2}}\int_{-\infty}^{+\infty}\int_{-\infty}^{+\infty}\exp(x+y) \\
&\quad \cdot\exp\left\{-\frac{1}{2(1-\rho^2)}\left[(\frac{x-\mu_a}{\sigma_a})^2-2\rho\frac{x-\mu_a}{\sigma_a}\frac{y-\mu_b}{\sigma_b}+(\frac{y-\mu_b}{\sigma_b})^2\right]\right\}dy\,dx \\
&\overset{3''}{=} \frac{\exp(\mu_a+\mu_b)}{2\pi\sqrt{1-\rho^2}}\int_{-\infty}^{+\infty}\int_{-\infty}^{+\infty}\exp\left\{\sigma_a u+\sigma_b v-\frac{1}{2(1-\rho^2)}(u^2-2\rho uv+v^2)\right\}dv\,du \\
&\overset{4''}{=} \frac{1}{\pi\sqrt{2}}\exp\left(\mu_a+\mu_b+\frac{1}{2}\sigma_b^2(1-\rho^2)\right) \\
&\quad \cdot\int_{-\infty}^{+\infty}\left(\int_{-\infty}^{+\infty}\exp(-\psi^2)\,d\psi\right)\exp(-\frac{1}{2}(u^2-2u(\sigma_a+\sigma_b\rho)))\,du \\
&\overset{5''}{=} \frac{1}{\sqrt{\pi}}\exp\left(\mu_a+\mu_b+\frac{1}{2}\sigma_b^2(1-\rho^2)+\frac{1}{2}(\sigma_a+\sigma_b\rho)^2\right)\int_{-\infty}^{+\infty}\exp(-\psi^2)\,d\psi \\
&= \exp\left(\mu_a+\mu_b+\frac{1}{2}(\sigma_b^2+\sigma_a^2)+\sigma_a\sigma_b\rho\right)
\end{aligned}$$

$$\begin{aligned}
E_x &= \int_{-\infty}^{+\infty}\int_{-\infty}^{+\infty} f(x,y)\exp(x)\,dy\,dx \\
&\overset{3''}{=} \frac{\exp(\mu_a)}{2\pi\sqrt{1-\rho^2}}\int_{-\infty}^{+\infty}\int_{-\infty}^{+\infty}\exp\left\{\sigma_a u-\frac{1}{2(1-\rho^2)}(u^2-\rho uv+v^2)\right\}dv\,du \\
&\overset{6''}{=} \frac{1}{\pi\sqrt{2}}exp(\mu_a)\int_{-\infty}^{+\infty}\left(\int_{-\infty}^{+\infty}\exp(-\psi^2)\,d\psi\right)\exp(-\frac{1}{2}(u^2-2u\sigma_a))\,du \\
&\overset{7''}{=} \frac{1}{\sqrt{\pi}}\exp\left(\mu_a+\frac{1}{2}\sigma_a^2\right)\int_{-\infty}^{+\infty}\exp(-\psi^2)\,d\psi \\
&= \exp\left(\mu_a+\frac{1}{2}(\sigma_a^2)\right)
\end{aligned}$$

$$E_y = \exp(\mu_b+\frac{1}{2}\sigma_b^2)$$

Mudanças de variável:

$$3'': \begin{cases} u & = (x-\mu_a)/\sigma_a \\ \\ v & = (y-\mu_b)/\sigma_b \end{cases}$$

$$4'': \quad \psi = \frac{1}{\sqrt{2(1-\rho^2)}}(v - u\rho - \sigma_b(1-\rho^2))$$

$$5'': \quad w = \frac{1}{\sqrt{2}}(u - \sigma_a - \sigma_b\rho)$$

$$6'': \quad \psi = \frac{1}{\sqrt{2(1-\rho^2)}}(v - u\rho)$$

$$7'': \quad w = \frac{1}{\sqrt{2}}(u - \sigma_a)$$

Portanto,

$$cov(r_{s_a}, r_{s_b}) = \exp(\mu_a + \mu_b + \frac{1}{2}(\sigma_a^2 + \sigma_v^2))\,(\exp(\sigma_a\sigma_b\rho) - 1))$$

- $Cov(r_{s_a}, r_{c_a})$

$$cov(r_{s_a}, r_{c_a}) \;\; = \;\; \frac{1}{C_a}\left\{S_a E_{2x} - K_a^c E_x\right\} - \mu_{sa}\mu_{ca}$$

onde

$$E_{2x} \;\; = \;\; \frac{1}{2}\exp(2\mu_a + 2\sigma_a^2)\left\{1 - erf\left[\frac{1}{\sigma_a\sqrt{2}}\left(\ln\left(\frac{K_a^c}{S_a}\right) - \mu_a - 2\sigma_a^2\right)\right]\right\}$$

$$E_x \;\; = \;\; \frac{1}{2}\exp(\mu_a + \frac{1}{2}\sigma_a^2)\left\{1 - erf\left[\frac{1}{\sigma_a\sqrt{2}}\left(\ln\left(\frac{K_a^c}{S_a}\right) - \mu_a - \sigma_a^2\right)\right]\right\}$$

- $Cov(r_{s_a}, r_{p_a})$

$$cov(r_{s_a}, r_{p_a}) = \frac{1}{P_a}\{-S_a E_{2x} + K_a^p E_x\} - \mu_{sa}\mu_{pa}$$

onde

$$E_{2x} = \frac{1}{2}\exp(2\mu_a + 2\sigma_a^2)\left\{1 + erf\left[\frac{1}{\sigma_a\sqrt{2}}\left(\ln\left(\frac{K_a^p}{S_a}\right) - \mu_a - 2\sigma_a^2\right)\right]\right\}$$

$$E_x = \frac{1}{2}\exp(\mu_a + \frac{1}{2}\sigma_a^2)\left\{1 + erf\left[\frac{1}{\sigma_a\sqrt{2}}\left(\ln\left(\frac{K_a^p}{S_a}\right) - \mu_a - \sigma_a^2\right)\right]\right\}$$

- $Cov(r_{s_a}, r_{c_b})$

$$cov(r_{s_a}, r_{c_b}) = \frac{1}{C_b}\{S_b E_{xy} - E_x K_b^c\} - \mu_{cb}\bar{E}_x - \frac{\mu_{sa}}{C_b}\{S_b E_y - K_b^c E_1\} + \mu_{sa}\mu_{cb}$$

onde

$$R_b^c = \frac{1}{\sigma_b}(\ln\left(\frac{K_b^c}{S_b}\right) - \mu_b)$$

$$E_{xy} = \frac{1}{2}\exp(\mu_a + \mu_b + \frac{1}{2}(\sigma_a^2 + \sigma_b^2) + \rho\sigma_a\sigma_b)\left\{1 - erf\left[\frac{1}{\sqrt{2}}(R_b^c - \sigma_b - \sigma_a\rho)\right]\right\}$$

$$E_x = \frac{1}{2}\exp(\mu_a + \frac{1}{2}\sigma_a^2)\left\{1 - erf\left[\frac{1}{\sqrt{2}}\left(R_b^c - \rho\sigma_a^2\right)\right]\right\}$$

$$E_y = \frac{1}{2}\exp(\mu_b + \frac{1}{2}\sigma_b^2)\left\{1 - erf\left[\frac{1}{\sqrt{2}}\left(R_b^c - \sigma_b\right)\right]\right\}$$

$$\bar{E}_x = \frac{1}{2}\exp(\mu_a + \frac{1}{2}\sigma_a^2)$$

$$E_1 = \frac{1}{2}\left\{1 - erf\left[\frac{1}{\sqrt{2}}\left(R_b^c\right)\right]\right\}$$

- $Cov(r_{s_a}, r_{p_b})$

$$cov(r_{s_a}, r_{p_b}) = \frac{1}{P_b}\left\{-S_b E_{xy} + E_x K_b^p\right\} - \mu_{pb}\bar{E}_x - \frac{\mu_{sa}}{P_b}\left\{-S_b E_y + K_b^c E_1\right\} + \mu_{sa}\mu_{pb}$$

onde

$$R_b^p = \frac{1}{\sigma_b}(\ln\left(\frac{K_b^p}{S_b}\right) - \mu_b)$$

$$E_{xy} = \frac{1}{2}\exp(\mu_a + \mu_b + \frac{1}{2}(\sigma_a^2 + \sigma_b^2) + \rho\sigma_a\sigma_b)\left\{1 + erf\left[\frac{1}{\sqrt{2}}(R_b^c - \sigma_b - \sigma_a\rho)\right]\right\}$$

$$E_x = \frac{1}{2}\exp(\mu_a + \frac{1}{2}\sigma_a^2)\left\{1 + erf\left[\frac{1}{\sqrt{2}}\left(R_b^p - \rho\sigma_a\right)\right]\right\}$$

$$E_y = \frac{1}{2}\exp(\mu_b + \frac{1}{2}\sigma_b^2)\left\{1 + erf\left[\frac{1}{\sqrt{2}}\left(R_b^p - \sigma_b\right)\right]\right\}$$

$$\bar{E}_x = \frac{1}{2}\exp(\mu_a + \frac{1}{2}\sigma_a^2)$$

$$E_1 = \frac{1}{2}\left\{1 + erf\left[\frac{1}{\sqrt{2}}\left(R_b^c\right)\right]\right\}$$

- $Cov(r_{c_a}, r_{p_a})$

$$cov(r_{c_a}, r_{p_a}) = \begin{cases} \frac{1}{C_a P_a}\left\{-S_a^2 E_{2x} + S_a E_x(K_a^c + K_a^p) - K_a^c K_a^p E_1\right\} - \mu_{ca}\mu_{pa}, & \text{se} \quad K_a^c > K_a^p \\ -\mu_{ca}\mu_{pa}, & \text{se} \quad K_a^c \leq K_a^p \end{cases}$$

onde

$$
\begin{aligned}
E_{2x} &= \frac{1}{2}\exp(2\mu_a + 2\sigma_a^2) \\
&\cdot \left\{ erf\left[\frac{1}{\sigma_a\sqrt{2}}\left(\ln\left(\frac{K_a^p}{S_a}\right) - \mu_a - 2\sigma_a^2\right)\right] - erf\left[\frac{1}{\sigma_a\sqrt{2}}\left(\ln\left(\frac{K_a^c}{S_a}\right) - \mu_a - 2\sigma_a^2\right)\right]\right\}
\end{aligned}
$$

$$
\begin{aligned}
E_{x} &= \frac{1}{2}\exp(\mu_a + \frac{1}{2}\sigma_a^2) \\
&\cdot \left\{ erf\left[\frac{1}{\sigma_a\sqrt{2}}\left(\ln\left(\frac{K_a^p}{S_a}\right) - \mu_a - \sigma_a^2\right)\right] - erf\left[\frac{1}{\sigma_a\sqrt{2}}\left(\ln\left(\frac{K_a^c}{S_a}\right) - \mu_a - \sigma_a^2\right)\right]\right\}
\end{aligned}
$$

$$
E_1 = \frac{1}{2}\left\{ erf\left[\frac{1}{\sigma_a\sqrt{2}}\left(\ln\left(\frac{K_a^p}{S_a}\right) - \mu_a\right)\right] - erf\left[\frac{1}{\sigma_a\sqrt{2}}\left(\ln\left(\frac{K_a^c}{S_a}\right) - \mu_a\right)\right]\right\}
$$

- $Cov(r_{c_a}, r_{c_b})$

$$
\begin{aligned}
cov(r_{c_a}, r_{c_b}) &= \frac{1}{C_a C_b}\left\{S_a S_b E_{xy} - S_a E_x K_b^c - S_b E_y K_a^c\right\} \\
&- \frac{\mu_{cb}}{C_a}(S_a \bar{E}_x - K_a^c \bar{E}_1) - \frac{\mu_{ca}}{C_b}(S_b \bar{\bar{E}}_y - K_b^c \bar{\bar{E}}_1) + \mu_{ca}\mu_{cb}
\end{aligned}
$$

onde

$$
R_a^c = \frac{1}{\sigma_b}(\ln\left(\frac{K_a^c}{S_b}\right) - \mu_a)
$$

$$
R_b^c = \frac{1}{\sigma_b}(\ln\left(\frac{K_b^c}{S_b}\right) - \mu_b)
$$

$$
\begin{aligned}
E_{xy} &= \frac{1}{2\sqrt{2\pi}}\exp(\mu_a + \mu_b + \frac{1}{2}(\sigma_a^2 + \sigma_b^2) + \rho\sigma_a\sigma_b) \\
&\cdot \int_{R_a^c}^{+\infty}\left\{1 - erf\left[\frac{1}{\sqrt{2(1-\rho^2)}}(R_b^c - \sigma_b(1-\rho^2) - \rho u)\right]\right\} \\
&\cdot exp(-\frac{1}{2}(u - \sigma_a - \rho\sigma_b)^2)\, du
\end{aligned}
$$

$$
E_x = \frac{1}{2\sqrt{2\pi}}\exp(\mu_a + \frac{1}{2}\sigma_a^2)
$$

$$
\begin{aligned}
& \cdot \int_{R_a^c}^{+\infty} \left\{ 1 - erf \left[\frac{1}{\sqrt{2(1-\rho^2)}} \left(R_b^c - \rho u \right) \right] \right\} \\
& \cdot \exp(-\frac{1}{2}(u-\sigma_a)^2)\, du
\end{aligned}
$$

$$
\begin{aligned}
E_y &= \frac{1}{2\sqrt{2\pi}} \exp(\mu_b + \frac{1}{2}\sigma_b^2) \\
& \cdot \int_{R_a^c}^{+\infty} \left\{ 1 - erf \left[\frac{1}{\sqrt{2(1-\rho^2)}} \left(R_b^c - \rho u - \sigma_b(1-\rho^2) \right) \right] \right\} \\
& \cdot \exp(-\frac{1}{2}(u-\sigma_b \rho)^2)\, du
\end{aligned}
$$

$$
E_1 = \frac{1}{2\sqrt{2\pi}} \int_{R_a^c}^{+\infty} \left\{ 1 - erf \left[\frac{1}{\sqrt{2(1-\rho^2)}} \left(R_b^c - \rho u \right) \right] \right\} \exp(-\frac{1}{2}u^2)\, du
$$

$$
\bar{E}_x = \frac{1}{2} \exp(\mu_a + \frac{1}{2}\sigma_a^2) \left\{ 1 - erf \left[\frac{1}{\sqrt{2}} \left(R_a^c - \sigma_a \right) \right] \right\}
$$

$$
\bar{E}_1 = \frac{1}{2} \left\{ 1 - erf \left[\frac{R_a^c}{\sqrt{2}} \right] \right\}
$$

$$
\bar{\bar{E}}_y = \frac{1}{2} \exp(\mu_b + \frac{1}{2}\sigma_b^2) \left\{ 1 - erf \left[\frac{1}{\sqrt{2}} \left(R_b^c - \sigma_b \right) \right] \right\}
$$

$$
\bar{\bar{E}}_1 = \frac{1}{2} \left\{ 1 - erf \left[\frac{R_b^c}{\sqrt{2}} \right] \right\}
$$

- $Cov(r_{c_a}, r_{p_b})$

$$
\begin{aligned}
cov(r_{c_a}, r_{p_b}) &= \frac{1}{C_a P_b} \left\{ -S_a S_b E_{xy} + S_a E_x K_b^p + S_b E_y K_a^c - K_a^c K_b^p E_1 \right\} \\
& -\frac{\mu_{pb}}{C_a}(S_a \bar{E}_x - K_a^c \bar{E}_1) - \frac{\mu_{ca}}{P_b}(-S_b \bar{\bar{E}}_y + K_b^p \bar{\bar{E}}_1) + \mu_{ca}\mu_{pb}
\end{aligned}
$$

onde

$$
R_a^c = \frac{1}{\sigma_b}(\ln \left(\frac{K_a^c}{S_b} \right) - \mu_a)
$$

$$R_b^p = \frac{1}{\sigma_b}(\ln\left(\frac{K_b^p}{S_b}\right) - \mu_b)$$

Os valores de E_{xy}, E_x, E_y , E_1, $\bar{E}_x$,$\bar{E}_1$,$\bar{\bar{E}}_x$ e $\bar{\bar{E}}_1$ são iguais às expressões de $cov(r_{c_a}, r_{c_b})$, trocando o sinal da função erf e trocando também R_b^c por R_b^p.

- $Cov(r_{p_a}, r_{p_b})$

$$\begin{aligned} cov(r_{p_a}, r_{p_b}) &= \frac{1}{P_a P_b}\left\{S_a S_b E_{xy} - S_a E_x K_b^p - S_b E_y K_a^p + K_a^p K_b^p E_1\right\} \\ &+ \frac{\mu_{pb}}{P_a}(S_a \bar{E}_x - K_a^p \bar{E}_1) + \frac{\mu_{pa}}{P_b}(S_b \bar{\bar{E}}_y - K_b^p \bar{\bar{E}}_1) + \mu_{pa}\mu_{pb} \end{aligned}$$

onde

$$R_a^p = \frac{1}{\sigma_b}(\ln\left(\frac{K_a^p}{S_a}\right) - \mu_a)$$

$$R_b^p = \frac{1}{\sigma_b}(\ln\left(\frac{K_b^p}{S_b}\right) - \mu_b)$$

$$\begin{aligned} E_{xy} &= \frac{1}{2\sqrt{2\pi}} \exp(\mu_a + \mu_b + \frac{1}{2}(\sigma_a^2 + \sigma_b^2) + \rho\sigma_a\sigma_b) \\ &\cdot \int_{-\infty}^{R_a^p} \left\{1 + erf\left[\frac{1}{\sqrt{2(1-\rho^2)}}(R_b^p - \sigma_b(1-\rho^2) - \rho u)\right]\right\} \\ &\cdot exp(-\frac{1}{2}(u - \sigma_a - \rho\sigma_b)^2)\, du \end{aligned}$$

$$\begin{aligned} E_x &= \frac{1}{2\sqrt{2\pi}} \exp(\mu_a + \frac{1}{2}\sigma_a^2) \\ &\cdot \int_{-\infty}^{R_a^p} \left\{1 + erf\left[\frac{1}{\sqrt{2(1-\rho^2)}}\left(R_b^p - \rho u\right)\right]\right\} \\ &\cdot \exp(-\frac{1}{2}(u - \sigma_a)^2)\, du \end{aligned}$$

$$E_y = \frac{1}{2\sqrt{2\pi}} \exp(\mu_b + \frac{1}{2}\sigma_b^2)$$

$$
\cdot \int_{-\infty}^{R_a^p} \left\{ 1 + erf \left[\frac{1}{\sqrt{2(1-\rho^2)}} \left(R_b^p - \rho u - \sigma_b (1-\rho^2) \right) \right] \right\}
$$
$$
\cdot \exp(-\frac{1}{2}(u - \sigma_b \rho)^2)\, du
$$

$$
E_1 = \frac{1}{2\sqrt{2\pi}} \int_{-\infty}^{R_a^p} \left\{ 1 + erf \left[\frac{1}{\sqrt{2(1-\rho^2)}} \left(R_b^p - \rho u \right) \exp(-\frac{1}{2} u^2) \right] \right\} du
$$

$$
\bar{E}_x = \frac{1}{2} \exp(\mu_a + \frac{1}{2}\sigma_a^2) \left\{ 1 + erf \left[\frac{1}{\sqrt{2}} \left(R_a^p - \sigma_a \right) \right] \right\}
$$

$$
\bar{E}_1 = \frac{1}{2} \left\{ 1 + erf \left[\frac{R_a^p}{\sqrt{2}} \right] \right\}
$$

$$
\bar{\bar{E}}_y = \frac{1}{2} \exp(\mu_b + \frac{1}{2}\sigma_b^2) \left\{ 1 + erf \left[\frac{1}{\sqrt{2}} \left(R_b^p - \sigma_b \right) \right] \right\}
$$

$$
\bar{\bar{E}}_1 = \frac{1}{2} \left\{ 1 + erf \left[\frac{R_b^p}{\sqrt{2}} \right] \right\}
$$

- $Cov(r_{c_{a1}}, r_{c_{a2}})$

  Neste caso, vamos calcular a covariância entre as taxas de retorno de duas opções de compra sobre o mesmo ativo A, mesma data de expiração, mas com preços de exercício diferentes e, portanto, prêmios diferentes.

  Sejam K_1^c e K_2^c os preços de exercício e C_1 e C_2 os prêmios das calls.

$$
cov(r_{c_{a1}}, r_{c_{a2}}) = \frac{1}{C_1 C_2} \left\{ S_a^2 E_{2x} - S_a E_x (K_1^c + K_2^c) + K_1^c K_2^c E_1 \right\} - \mu_{c1}\mu_{c2}
$$

$$
K = \max\{K_1^c, K_2^c\}
$$

onde

$$
E_{2x} = \frac{1}{2} \exp(2\mu_a + 2\sigma_a^2) \left\{ 1 - erf \left[\frac{1}{\sigma_a \sqrt{2}} \left(\ln\left(\frac{K}{S_a}\right) - \mu_a - 2\sigma_a^2 \right) \right] \right\}
$$

$$E_x = \frac{1}{2}\exp(\mu_a + \frac{1}{2}\sigma_a^2)\left\{1 - erf\left[\frac{1}{\sigma_a\sqrt{2}}\left(\ln\left(\frac{K}{S_a}\right) - \mu_a - \sigma_a^2\right)\right]\right\}$$

$$E_1 = \frac{1}{2}\left\{1 - erf\left[\frac{1}{\sigma_a\sqrt{2}}\left(\ln\left(\frac{K}{S_a}\right) - \mu_a\right)\right]\right\}$$

- $Cov(r_{p_{a1}}, r_{p_{a2}})$

  Aqui calculamos a covariância entre as taxas de retorno de duas opções de venda sobre o mesmo ativo A, mesma data de expiração, mas com preços de exercício e prêmios diferentes. Sejam K_1^p e K_2^p os preços de exercício e P_1 e P_2 os prêmios das calls.

$$cov(r_{p_{a1}}, r_{p_{a2}}) = \frac{1}{P_1 P_2}\left\{S_a^2 E_{2x} - S_a E_x(K_1^p + K_2^p) + K_1^p K_2^p E_1\right\} - \mu_{p1}\mu_{p2}$$

$$K = \min\{K_1^p, K_2^p\}$$

onde

$$E_{2x} = \frac{1}{2}\exp(2\mu_a + 2\sigma_a^2)\left\{1 + erf\left[\frac{1}{\sigma_a\sqrt{2}}\left(\ln\left(\frac{K}{S_a}\right) - \mu_a - 2\sigma_a^2\right)\right]\right\}$$

$$E_x = \frac{1}{2}\exp(\mu_a + \frac{1}{2}\sigma_a^2)\left\{1 + erf\left[\frac{1}{\sigma_a\sqrt{2}}\left(\ln\left(\frac{K}{S_a}\right) - \mu_a - \sigma_a^2\right)\right]\right\}$$

$$E_1 = \frac{1}{2}\left\{1 + erf\left[\frac{1}{\sigma_a\sqrt{2}}\left(\ln\left(\frac{K}{S_a}\right) - \mu_a\right)\right]\right\}$$

- $Cov(r_{c_{aT}}, r_{c_{a\tau}})$

  Neste caso calculamos a covariância entre duas opções de compra sobre o mesmo ativo, mas com datas de expiração diferentes (dadas por T e τ, $\tau < T$).

  Suponha que temos a informação

$$x_T = \ln\frac{S_a(T)}{S_a(t_0)} \cong N(\mu, \sigma^2)$$

Denote por x_τ o retorno logarítimico do ativo no período t_0 a τ e $x_{T-\tau}$ o retorno logarítimico do ativo no período τ a T. Então

$$x_\tau = \ln \frac{S_a(\tau)}{S_a(t_0)} \cong N(\mu_\tau, \sigma_\tau^2)$$

$$x_{T-\tau} = \ln \frac{S_a(T-\tau)}{S_a(\tau)} \cong N(\mu_{T-\tau}, \sigma_{T-\tau}^2)$$

onde

$$\begin{aligned}
\mu_\tau &= \mu_T \frac{\tau - t_0}{T - t_0} \\
\sigma_\tau^2 &= \sigma_T^2 \frac{\tau - t_0}{T - t_0} \\
\mu_{T-\tau} &= \mu_T \frac{T - \tau}{T - t_0} \\
\sigma_{T-\tau}^2 &= \sigma_T^2 \frac{T - \tau}{T - t_0}
\end{aligned}$$

E a correlação entre x_τ e x_T é dada por

$$corr(x_T, x_\tau) = \frac{\sigma_T^2 + \sigma_\tau^2 - \sigma_{T-\tau}^2}{2\sigma_T \sigma_\tau}$$

Desta forma, podemos calcular a covariância entre as duas opções usando a mesma expressão da $Cov(r_{c_a}, r_{c_b})$, considerando $x = x_T$, $y = x_\tau$ e $\rho = corr(x_T, x_\tau)$.

- $Cov(r_{p_{aT}}, r_{p_{a\tau}})$

  Idem ao caso anterior, onde a expressão usada para o cálculo da covariância é a dada por $Cov(r_{p_a}, r_{p_b})$.

- $Cov(r_{c_{aT}}, r_{p_{a\tau}})$

  Idem aos casos anteriores, usando a expressão dada em $Cov(r_{c_a}, r_{p_b})$ para o cálculo da covariância.

Exemplos de portfólios eficientes gerados pela resolução do problema 2.20 serão dados no capítulo 6. O tratamento numérico das integrais envolvidas nas expressões de covariância, dadas acima, será feito no capítulo 4.

Capítulo 3

Assimetria de Portfólios

No capítulo anterior vimos como otimizar portfólios com critérios baseados em média e variância, considerando certas hipóteses sobre o comportamento de um investidor que objetiva maximizar o valor esperado de alguma função utilidade $U(R)$, onde R é a taxa de retorno do portfólio em 1 período. Também utilizamos uma aproximação quadrática para a função utilidade e supomos um distribuição lognormal para o retorno de ações de risco do mercado.

Nosso objetivo agora é estender estes resultados usando, além do valor esperado do retorno do portfólio e de sua variância, o seu terceiro momento central, a assimetria do retorno do portfólio em relação ao seu primeiro momento.

Veremos neste capítulo como calcular assimetria e co-assimetria para o caso de portfólios mistos de ações e opções européias de forma exata, como no caso das covariâncias do capítulo 2. No capítulo 6 veremos exemplos de portfólios eficientes no espaço $retorno \times risco \times assimetria$.

3.1 Seleção de Portfólios e Momentos de Terceira Ordem

Continuaremos a trabalhar com a teoria de utilidade sob incerteza (isto é, o comportamento do investidor segue regras probabilísticas) e considerar que o indivíduo queira maximizar a utilidade esperada da riqueza.

Vamos ver como assimetria enquadra-se no comportamento do investidor e qual a sua influência na utilidade esperada.

Para um portfólio formado por n ativos de risco, os três primeiros momentos do retorno R_p do portfólio são dados por:

$$\bar{R}_p = E[R_p] = \sum_{i=1}^{n} w_i \bar{R}_i$$

$$\sigma_p^2 = E[(R_p - \bar{R}_p)^2] = \sum_{i=1}^{n} \sum_{j=1}^{n} w_i w_j \sigma_{ij}$$

$$m_p^3 = E[(R_p - \bar{R}_p)^3] = \sum_{i=1}^{n} \sum_{j=1}^{n} \sum_{k=1}^{n} w_i w_j w_k m_{ijk}$$

onde

$\bar{R}_i$ = retorno esperado do i-ésimo ativo

σ_{ij} = covariância entre o retorno do i-ésimo e do j-ésimo ativo

m_{ijk} = co-assimetria[2] entre o retorno do i-ésimo, j-ésimo e o k-ésimo ativo

w_i = fração do total do investimento atribuido ao i-ésimo ativo.

A expansão da função utilidade U em série de potências em torno do retorno esperado do portfólio é dada por

$$\begin{aligned} U(R_p) &= U(\bar{R}_p) + U'(\bar{R}_p)(R_p - \bar{R}_p) + \frac{U''(\bar{R}_p)}{2!}(R_p - \bar{R}_p)^2 + \frac{U'''(\bar{R}_p)}{3!}(R_p - \bar{R}_p)^3 \\ &+ \cdots + \frac{U^{(n-1)}(\bar{R}_p)}{(n-1)!}(R_p - \bar{R}_p)^{n-1} + r_n \end{aligned} \tag{3.1}$$

onde

$$r_n = \frac{U^n(\xi)}{n!}(R_p - \bar{R}_p)^n \qquad \xi \in (R_p, \bar{R}_p)$$

Desta forma, a função utilidade torna-se um polinômio de grau n em R_p. Este número n pode variar até obtermos precisão suficiente na aproximação. Como as funções utilidade que melhor representam o comportamento de um investidor averso ao risco são funções do tipo exponencial negativa $U(y) = K(1 - e^{-\alpha y})$ e a logarítmica $U(y) = log(y)$, que são funções suaves, usar aproximações com $n \leq 3$ na expansão de Taylor é satisfatório. Mas se não restringirmos o comportamento destas funções utilidade, aproximações mais precisas são necessárias. Isto envolve cálculos de momentos de ordem maior ou igual a 3, e como veremos no cálculo de assimetrias, esta tarefa pode ser bastante difícil.

Aproximando U por um polinômio de terceira ordem, a utilidade esperada do portfólio será dada por

$$\begin{aligned} E[U(R_p)] &\cong E[U(\bar{R}_p)] + \frac{U''(\bar{R}_p)}{2!}E[(R_p - \bar{R}_p)^2] + \frac{U'''(\bar{R}_p)}{3!}E[(R_p - \bar{R}_p)^3] \\ &\cong U(\bar{R}_p) + \frac{U''(\bar{R}_p)}{2!}\sigma_p^2 + \frac{U'''(\bar{R}_p)}{3!}m_p^3 \end{aligned} \tag{3.2}$$

[2]Aqui assimetria (*skewness*) é definido como o terceiro momento central não normalizado

De acordo com as hipóteses sobre o comportamento de um investidor averso ao risco, a propriedade (c) (capítulo 2) é satisfeita se

$$
\begin{aligned}
\frac{d}{dR_p}[-U^{''}/U^{'}] &= (-U^{'}U^{'''}+U^{''2})/U^{'2} \leq 0 \\
&\Longleftrightarrow -U^{'}U^{'''}+U^{''2} \leq 0 \\
&\Longleftrightarrow -U^{'''} \leq (U^{''2}/-U^{'}) < 0 \\
&\Longleftrightarrow U^{'''} > 0
\end{aligned}
$$

Ou seja, para uma pessoa com decrescente ou constante aversão absoluta ao risco, $U^{'''}$ deve ser positivo, o que implica numa preferência por *skewness* positivo ($\frac{\partial U}{\partial m_p^3} > 0$).

A equação 3.2 mostra que a influência do skewness m_p^3 sobre a utilidade esperada é positiva.

Logo, supondo que as preferências do investidor são dadas pela função utilidade $U = U(\bar{R}_p, \sigma_p^2, m_p^3)$, o conjunto de portfólios ótimos deverá

- maximizar m_p^3 para $\bar{R}_p$ e σ_p^2 fixos
- maximizar $\bar{R}_p$ para m_p^3 e σ_p^2 fixos
- minimizar σ_p^2 para $\bar{R}_p$ e m_p^3 fixos.

Para quantificar as preferências de um investidor quanto à propensão ao risco, nossa função objetivo a ser maximizada será dada por

$$
\left\{
\begin{array}{ll}
\max & \eta\bar{R}_p - \sigma_p^2 + \alpha m_p^3 \\
= \max & \eta\sum_i w_i\bar{z}_i - \sum_i\sum_j w_iw_j\sigma_{ij} + \alpha\sum_i\sum_j\sum_k w_iw_jw_km_{ijk} \\
s.a & \sum_i w_i = 1 \\
 & w_i \geq 0 \qquad (i = 1..n)
\end{array}
\right.
\tag{3.3}
$$

Os parâmetros η e α ponderam, respectivamente, a taxa de retorno esperada e assimetria do portfólio.

Agora nossa função objetivo é um problema de programação não linear. Para resolvê-lo, fixaremos η e usaremos o solver MINOS do GAMS, e cada solução ótima representa um portfólio cujo retorno esperado, variância e assimetria serão dados por $\bar{R}_p = w(\alpha)\bar{z}$, $m_p^3 = \sum_i\sum_j\sum_k w_i(\alpha)w_j(\alpha)w_k(\alpha)m_{ijk}$ e $\sigma_p^2 = w(\alpha)\Sigma w(\alpha)$. Observe que, variando η, teremos uma superfície eficiente no espaço **retorno** $\times$ **risco** $\times$ **assimetria**.

Podemos ainda defender o uso do terceiro momento na otimização de portfólios pelo fato de que os ativos e derivativos que compõem as carteiras têm retornos assimétricos. Para os preços das ações supomos um comportamento log-normal e o retorno de opções (fig3.1) tem um comportamento assimétrico.

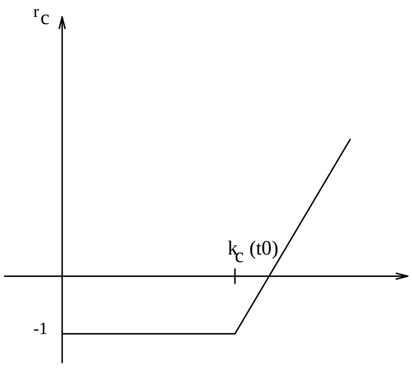

Figura 3.1: taxa de retorno de uma call

Vamos agora deduzir as expressões de co-assimetria para o caso de portfólios formados por ações e opções.

3.2 Expressões de Co-assimetrias

Vamos agora voltar ao exemplo didático de portfólio da seção 2.1.3 e ampliá-lo para conter três tipos de ativos básicos, ações A, B e C, e opções européias sobre estas ações. Os retornos logarítmicos das ações são dados por

$$\begin{aligned} x &= ln\left(\frac{S_a(T)}{S_a(t_0)}\right) \cong N(\mu_a, \sigma_a^2) \\ y &= ln\left(\frac{S_b(T)}{S_b(t_0)}\right) \cong N(\mu_b, \sigma_b^2) \\ z &= ln\left(\frac{S_c(T)}{S_c(t_0)}\right) \cong N(\mu_c, \sigma_c^2) \end{aligned}$$

e função densidade de probabilidade conjunta dada por

$$\begin{aligned} f(x,y,z) &= \frac{1}{\sigma_a\sigma_b\sigma_c\sqrt{8\pi^3\gamma}} \exp\left[\left(\frac{x-\mu_a}{\sigma_a}\right)^2(1-\rho_{bc}^2) + \left(\frac{x-\mu_b}{\sigma_b}\right)^2(1-\rho_{ac}^2) + \left(\frac{x-\mu_c}{\sigma_c}\right)^2(1-\rho_{ab}^2)\right]\cdot \\ &\exp\left[2\frac{x-\mu_a}{\sigma_a}\frac{y-\mu_b}{\sigma_b}(\rho_{ac}\rho_{bc}-\rho_{ab}) + 2\frac{x-\mu_a}{\sigma_a}\frac{z-\mu_c}{\sigma_c}(\rho_{ab}\rho_{bc}-\rho_{ac}) + 2\frac{y-\mu_b}{\sigma_a}\frac{z-\mu_c}{\sigma_b}(\rho_{ac}\rho_{ab}-\rho_{bc})\right] \end{aligned}$$

$$(-\infty < x, y, z < +\infty)$$

onde

$$\gamma = 1 + 2\rho_{ab}\rho_{bc}\rho_{ac} - \rho_{ab}^2 - \rho_{bc}^2 - \rho_{ac}^2$$

$$\rho_{ab}, \rho_{ac}, \rho_{bc} = \text{coeficiente de correlação entre } x \text{ e } y, \text{ entre } x \text{ e } z \text{ e entre } y \text{ e } z, \text{ respectivamente.}$$

Vamos denotar assimetria pela letra m. Por exemplo,

$$m(r_{s_a}, r_{s_b}, r_{s_c}) = E[(r_{s_a} - E(r_{s_a}))(r_{s_b} - E(r_{s_b}))(r_{s_c} - E(r_{s_c}))]$$

é a co-assimetria entre os retornos r_{s_a}, r_{s_b} e r_{s_c} dos ativos A, B e C.

Aqui estamos supondo que os ativos A, B e C são distintos entre si. Abaixo estão as expressões de alguns casos de co-assimetria. Pela extensiva repetição de cálculos simbólicos na derivação das expressões abaixo, os detalhes das mudanças de variáveis usadas foram explicitadas em alguns casos. Os outros seguem raciocínio análogo.

Do mesmo modo que fizemos com as covariâncias, as expressões de assimetria serão feitas usando-se retornos simples de ações e opções, e não as taxas de retorno definidadas em 2.13.

- $m(r_{s_a}, r_{s_a}, r_{s_a})$

$$m(r_{s_a}, r_{s_a}, r_{s_a}) = m_{s_a}^3 = E_{3x} - 3E_{2x}E_x + 2E_x^3$$

onde

$$\begin{aligned} E_{3x} &= \exp\left(3\mu_a + 4.5\sigma_a^2\right) \\ E_{2x} &= \exp\left(2\mu_a + 2\sigma_a^2\right) \\ E_x &= \exp\left(\mu_a + 0.5\sigma_a^2\right) \end{aligned}$$

- $m(r_{s_a}, r_{s_a}, r_{c_a})$

$$m(r_{s_a}, r_{s_a}, r_{c_a}) = \frac{1}{C_a}(S_aE_{3x} - K_a^cE_{2x}) - \frac{2\mu_{sa}}{C_a}(S_aE_{2x} - K_a^cE_x) - \mu_{ca}I_{2x} + 2\mu_{sa}^2\mu_{ca}$$

onde

$$
\begin{aligned}
g &= \frac{1}{\sigma_a\sqrt{2}}(\ln\frac{K_a^c}{S_a} - \mu_a) \\
E_{3x} &= 0.5exp(3\mu_a + 4.5\sigma_a^2)[1 - erf(g - \frac{3\sigma_a}{\sqrt{2}})] \\
E_{2x} &= 0.5exp(2\mu_a + 2\sigma_a^2)[1 - erf(g - \frac{2\sigma_a}{\sqrt{2}})] \\
E_x &= 0.5exp(\mu_a + 0.5\sigma_a^2)[1 - erf(g - \frac{\sigma_a}{\sqrt{2}})] \\
I_{2x} &= exp(2\mu_a + 2\sigma_a^2)
\end{aligned}
$$

- $m(r_{s_a}, r_{s_a}, r_{p_a})$

$$
m(r_{s_a}, r_{s_a}, r_{p_a}) = \frac{1}{P_a}(-S_aE_{3x} + K_a^pE_{2x}) + \frac{2\mu_{sa}}{P_a}(S_aE_{2x} - K_a^pE_x) - \mu_{pa}I_{2x} + 2\mu_{sa}^2\mu_{pa}
$$

onde

$g = \frac{1}{\sigma_a\sqrt{2}}(\ln\frac{K_a^p}{S_a} - \mu_a)$

e as expressões de E_{3x}, E_{2x}, E_x e I_{2x} são iguais as de $m(r_{s_a}, r_{s_a}, r_{c_a})$, trocando o sinal da função erf.

- $m(r_{s_a}, r_{c_a1}, r_{c_a2})$

$$
\begin{aligned}
m(r_{s_a}, r_{c_a1}, r_{c_a2}) = {} & \frac{1}{C_1C_2}(S_a^2E_{3x} - S_aE_{2x}(K_1^c + K_c^2) + K_1^cK_2^cE_x) \\
& -\frac{\mu_{sa}}{C_1C_2}(S_a^2E_{2x} - S_aE_x(K_1^c + K_c^2) + K_1^cK_2^cE_1) \\
& -\frac{\mu_{ca1}}{C_2}(S_a\bar{E}_{2x} - K_1^c\bar{E}_x) - \frac{\mu_{ca2}}{C_2}(S_a\bar{\bar{E}}_{2x} - K_2^c\bar{\bar{E}}_x) \\
& +2\mu_{sa}\mu_{ca1}\mu_{ca2}
\end{aligned}
$$

onde

$K = \max\{K_1^c, K_2^c\}$

$g = \frac{1}{\sigma_a\sqrt{2}}(\ln\frac{K}{S_a} - \mu_a)$

$$\begin{aligned}
g_1 &= \frac{1}{\sigma_a\sqrt{2}}(\ln\frac{K_1^c}{S_a} - \mu_a) \\
g_2 &= \frac{1}{\sigma_a\sqrt{2}}(\ln\frac{K_2^c}{S_a} - \mu_a) \\
\bar{E}_{2x} &= 0.5exp(2\mu_a + 2\sigma_a^2)[1 - erf(g_1 - \frac{2\sigma_a}{\sqrt{2}})] \\
\bar{E}_{x} &= 0.5exp(\mu_a + 0.5\sigma_a^2)[1 - erf(g_1 - \frac{\sigma_a}{\sqrt{2}})] \\
\bar{\bar{E}}_{2x} &= 0.5exp(2\mu_a + 2\sigma_a^2)[1 - erf(g_2 - \frac{2\sigma_a}{\sqrt{2}})] \\
\bar{\bar{E}}_{x} &= 0.5exp(\mu_a + 0.5\sigma_a^2)[1 - erf(g_2 - \frac{\sigma_a}{\sqrt{2}})]
\end{aligned}$$

e as expressões de E_{3x}, E_{2x} e E_x são iguais as de $m(r_{s_a}, r_{s_a}, r_{c_a})$.

- $m(r_{s_a}, r_{p_a1}, r_{p_a2})$

$$\begin{aligned}
m(r_{s_a}, r_{p_a1}, r_{p_a2}) = & \frac{1}{P_1P_2}(S_a^2E_{3x} - S_aE_{2x}(K_1^p + K_p^2) + K_1^pK_2^pE_x) \\
& -\frac{\mu_{sa}}{P_1P_2}(S_a^2E_{2x} - S_aE_x(K_1^p + K_p^2) + K_1^pK_2^pE_1) \\
& +\frac{\mu_{pa2}}{P_2}(S_a\bar{E}_{2x} - K_1^p\bar{E}_x) + \frac{\mu_{pa1}}{P_2}(S_a\bar{\bar{E}}_{2x} - K_2^p\bar{\bar{E}}_x) \\
& +2\mu_{sa}\mu_{pa1}\mu_{pa2}
\end{aligned}$$

onde

$K = \min\{K_1^p, K_2^p\}$
$g = \frac{1}{\sigma_a\sqrt{2}}(\ln\frac{K}{S_a} - \mu_a)$

$$\begin{aligned}
g_1 &= \frac{1}{\sigma_a\sqrt{2}}(\ln\frac{K_1^p}{S_a} - \mu_a) \\
g_2 &= \frac{1}{\sigma_a\sqrt{2}}(\ln\frac{K_2^p}{S_a} - \mu_a) \\
\bar{E}_{2x} &= 0.5exp(2\mu_a + 2\sigma_a^2)[1 + erf(g_1 - \frac{2\sigma_a}{\sqrt{2}})]
\end{aligned}$$

$$\begin{aligned}
\bar{E}_x &= 0.5exp(\mu_a + 0.5\sigma_a^2)[1 + erf(g_1 - \frac{\sigma_a}{\sqrt{2}})] \\
\bar{\bar{E}}_{2x} &= 0.5exp(2\mu_a + 2\sigma_a^2)[1 + erf(g_2 - \frac{2\sigma_a}{\sqrt{2}})] \\
\bar{\bar{E}}_x &= 0.5exp(\mu_a + 0.5\sigma_a^2)[1 + erf(g_2 - \frac{\sigma_a}{\sqrt{2}})]
\end{aligned}$$

e as expressões de E_{3x}, E_{2x} e E_x são iguais as de $m(r_{s_a}, r_{s_a}, r_{p_a})$.

- $m(r_{s_a}, r_{c_a}, r_{p_a})$

$$\begin{aligned}
m(r_{s_a}, r_{c_a}, r_{p_a}) &= 2\mu_{sa}\mu_{ca}\mu_{pa} + \frac{\mu_{ca}}{P_a}(S_a E_{2x}^p - K_a^p E_x^p) \\
&\quad - \frac{\mu_{pa}}{C_a}(S_a E_{2x}^c - K_a^c E_x^c) + L
\end{aligned}$$

onde

$$L = \begin{cases} \frac{1}{C_a P_a}(-S_a^2 E_{3x} + S_a E_{2x}(K_a^c + K_a^p) - K_a^p K_a^c E_x) \\ -\frac{\mu_{sa}}{C_a P_a}(-S_a^2 E_{2x} + S_a E_x(K_a^c + K_a^p) - K_a^p K_a^c E_1) & \text{se} \quad K_a^c < K_a^p \\ \\ 0 & \text{c.c.} \end{cases}$$

$$\begin{aligned}
g &= \frac{1}{\sigma_a\sqrt{2}}(\ln\frac{K_a^c}{S_a} - \mu_a) \\
E_{3x} &= 0.5exp(3\mu_a + 4.5\sigma_a^2)[erf(h - \frac{3\sigma_a}{\sqrt{2}}) - erf(g - \frac{3\sigma_a}{\sqrt{2}})] \\
E_{2x} &= 0.5exp(2\mu_a + 2\sigma_a^2)[erf(h - \frac{2\sigma_a}{\sqrt{2}}) - erf(g - \frac{2\sigma_a}{\sqrt{2}})] \\
E_x &= 0.5exp(\mu_a + 0.5\sigma_a^2)[erf(h - \frac{\sigma_a}{\sqrt{2}}) - erf(g - \frac{\sigma_a}{\sqrt{2}})] \\
E_1 &= 0.5[erf(g) - erf(h)]
\end{aligned}$$

As expressões de E_{2x}^c e E_x^c são iguais as dadas em $m(r_{sa}, r_{ca1}, r_{ca2})$, trocando K_1^c em g_1 por K_a^c.

O mesmo vale para E_{2x}^p e E_x^p, trocando K_1^c em g_1 por K_a^p e também trocando o sinal de erf.

- $m(r_{s_a}, r_{s_a}, r_{s_b})$

$$m(r_{s_a}, r_{s_a}, r_{s_b}) = E_{2xy} - 2\mu_{sa}E_{xy} - \mu_{sb}E_{2x} + \mu_{sa}^2 E_y + 2\mu_{sa}^2 E_y - \mu_{sa}^2 \mu_{sb}$$

onde

$$\begin{aligned}
E_{2xy} &= \exp\left(2\mu_a + \mu_b + 2\sigma_a^2 + \frac{1}{2}\sigma_b^2 + 2\sigma_a\sigma_b\rho_{ab}\right) \\
E_{xy} &= \exp\left(\mu_a + \mu_b + \frac{1}{2}(\sigma_a^2 + \sigma_b^2) + \sigma_a\sigma_b\rho_{ab}\right) \\
E_{2x} &= \exp\left(2\mu_a + \mu_b + 2\sigma_a^2\right) \\
E_x &= \exp\left(\mu_a + \sigma_a^2\right) \\
E_y &= \exp\left(\mu_b + \sigma_b^2\right)
\end{aligned}$$

- $m(r_{s_a}, r_{s_a}, r_{c_b})$

$$\begin{aligned}
m(r_{s_a}, r_{s_a}, r_{c_b}) &= \frac{1}{C_b}(S_b E_{2xy} - E_{2x}K_b^c) - \frac{2\mu_{sa}}{C_b}(S_b E_{xy} - E_x K_b^c) \\
&\quad -\mu_{cb}I_{2x} + 2\mu_{sa}\mu_{cb}I_x + \frac{2\mu_{sa}^2}{C_b}(S_b E_x - E_1 K_b^c) - \mu_{sa}^2\mu_{cb}
\end{aligned}$$

onde

$$\begin{aligned}
R_b^c &= \frac{1}{\sigma_b}(ln\frac{K_b^c}{S_b} - \mu_b) \\
E_{2xy} &= 0.5\exp\left(2\mu_a + \mu_b + 2\sigma_a^2 + \frac{1}{2}\sigma_b^2 + 2\sigma_a\sigma_b\rho_{ab}\right) \\
&\quad \cdot[1 - erf(\frac{1}{\sqrt{2}}(R_b^c - \sigma_b - 2\sigma_a\rho_{ab}))] \\
E_{xy} &= 0.5\exp\left(\mu_a + \mu_b + \frac{1}{2}(\sigma_a^2 + \sigma_b^2) + \sigma_a\sigma_b\rho_{ab}\right)[1 - erf(\frac{1}{\sqrt{2}}(R_b^c - \sigma_b - \sigma_a\rho_{ab}))]
\end{aligned}$$

$$\begin{aligned}
E_{2x} &= 0.5\exp\left(2\mu_a + 2\sigma_a^2\right)[1 - erf(\frac{1}{\sqrt{2}}(R_b^c - 2\sigma_a\rho_{ab}))] \\
E_x &= 0.5\exp\left(\mu_a + \sigma_a^2\right)[1 - erf(\frac{1}{\sqrt{2}}(R_b^c - \sigma_a\rho_{ab}))] \\
E_y &= 0.5\exp\left(\mu_b + \sigma_b^2\right)[1 - erf(\frac{1}{\sqrt{2}}(R_b^c - \sigma_b))] \\
E_1 &= 0.5[1 - erf(\frac{1}{\sqrt{2}}(R_b^c))] \\
I_{2x} &= \exp\left(2\mu_a + 2\sigma_a^2\right) \\
I_x &= \exp\left(\mu_a + 0.5\sigma_a^2\right)
\end{aligned}$$

- $m(r_{s_a}, r_{s_a}, r_{p_b})$

$$\begin{aligned}
m(r_{s_a}, r_{s_a}, r_{p_b}) &= -\frac{1}{P_b}(S_bE_{2xy} - E_{2x}K_b^p) + \frac{2\mu_{sa}}{P_b}(S_bE_{xy} - E_xK_b^p) \\
&\quad -\mu_{pb}I_{2x} + 2\mu_{sa}\mu_{pb}I_x - \frac{2\mu_{sa}^2}{P_b}(S_bE_x - E_1K_b^p) - \mu_{sa}^2\mu_{pb}
\end{aligned}$$

As expressões de E_{3x}, E_{2x}, E_x e E_1 são iguais as de $m(r_{s_a}, r_{s_a}, r_{cb})$, trocando o sinal da função erf e trocando também R_b^c por $R_b^p = \frac{1}{\sigma_b}(ln\frac{K_b^p}{S_b} - \mu_b)$.

- $m(r_{s_a}, r_{c_a}, r_{c_b})$

$$\begin{aligned}
m(r_{s_a}, r_{c_a}, r_{c_b}) &= \frac{1}{C_aC_b}(S_aS_bE_{2xy} - S_aK_b^cE_{2x} - S_bK_a^cE_{xy} + K_a^cK_b^cE_x) \\
&\quad -\frac{\mu_{sa}}{C_aC_b}(S_aS_bE_{xy} - S_aK_b^cE_x - S_bK_a^cE_y + K_a^cK_b^cE_1) \\
&\quad -\frac{\mu_{ca}}{C_b}(S_bE_{xy}^b - K_b^cE_x^b) - \frac{\mu_{cb}}{C_a}(S_aE_{2x}^a - K_a^cE_x^a) \\
&\quad +\frac{\mu_{sa}\mu_{ca}}{C_b}(S_bE_y^b - K_b^cE_1^b) + \frac{\mu_{sa}\mu_{cb}}{C_a}(S_aE_x^a - K_a^cE_1^a) \\
&\quad +\mu_{ca}\mu_{cb}I_x - \mu_{sa}\mu_{ca}\mu_{cb}
\end{aligned}$$

onde

$$R_a^c = \frac{1}{\sigma_b}(ln\frac{K_a^c}{S_a} - \mu_a)$$

$$R_b^c = \frac{1}{\sigma_b}(ln\frac{K_b^c}{S_b} - \mu_b)$$

$$\begin{aligned} E_{2xy} &= \exp\left(2\mu_a + \mu_b + 2\sigma_a^2 + \frac{1}{2}\sigma_b^2 + 2\sigma_a\sigma_b\rho_{ab}\right)\int_{R_a^c}^{+\infty}\exp[-0.5(u-2\sigma_a-\sigma_b\rho_{ab})^2] \\ &\cdot\left[1 - erf\left(\frac{1}{\sqrt{2(1-\rho_{ab})}}(R_a^b - u\rho_{ab} - \sigma_b(1-\rho_{ab}^2))\right)\right] du \end{aligned}$$

$$\begin{aligned} E_{xy} &= \exp\left(\mu_a + \mu_b + \frac{1}{2}(\sigma_a^2 + \sigma_b^2) + \sigma_a\sigma_b\rho_{ab}\right)\int_{R_a^c}^{+\infty}\exp[-0.5(u-\sigma_a-\sigma_b\rho_{ab})^2] \\ &\cdot\left[1 - erf\left(\frac{1}{\sqrt{2(1-\rho_{ab})}}(R_a^b - u\rho_{ab} - \sigma_b(1-\rho_{ab}^2))\right)\right] du \end{aligned}$$

$$\begin{aligned} E_{2x} &= \exp\left(2\mu_a + \mu_b + 2\sigma_a^2\right)\int_{R_a^c}^{+\infty}\exp[-0.5(u-2\sigma_a)^2] \\ &\cdot\left[1 - erf\left(\frac{1}{\sqrt{2(1-\rho_{ab})}}(R_a^b - u\rho_{ab}))\right)\right] du \end{aligned}$$

$$\begin{aligned} E_{x} &= \exp\left(\mu_a + \sigma_a^2\right)\int_{R_a^c}^{+\infty}\exp[-0.5(u-\sigma_a)^2] \\ &\cdot\left[1 - erf\left(\frac{1}{\sqrt{2(1-\rho_{ab})}}(R_a^b - u\rho_{ab}))\right)\right] du \end{aligned}$$

$$\begin{aligned} E_{y} &= \exp\left(\mu_b + \sigma_b^2)\right)\int_{R_a^c}^{+\infty}\exp[-0.5(u-\sigma_b\rho_{ab})^2] \\ &\cdot\left[1 - erf\left(\frac{1}{\sqrt{2(1-\rho_{ab})}}(R_a^b - u\rho_{ab} - \sigma_b(1-\rho_{ab}^2))\right)\right] du \end{aligned}$$

$$E_1 = \frac{1}{2\sqrt{2\pi}}\int_{R_a^c}^{+\infty}\exp[-0.5u^2]\left[1 - erf\left(\frac{1}{\sqrt{2(1-\rho_{ab})}}(R_a^b - u\rho_{ab})\right)\right] du$$

$$E_{xy}^b = 0.5\exp\left(\mu_a + \mu_b + \frac{1}{2}(\sigma_a^2 + \sigma_b^2) + \sigma_a\sigma_b\rho_{ab}\right)(1 - erf\left(\frac{1}{\sqrt{2}}(R_b^c - \sigma_b - \rho_{ab}\sigma_a)\right))$$

$$
\begin{aligned}
E_x^b &= 0.5\exp\left(\mu_a + 0.5\sigma_a^2\right)(1 - erf\left(\frac{1}{\sqrt{2}}(R_b^c - \rho_{ab}\sigma_a)\right))\\
E_{2x}^a &= 0.5\exp\left(2\mu_a + 2\sigma_a^2\right)(1 - erf\left(\frac{1}{\sqrt{2}}(R_a^c - 2\sigma_a)\right))\\
E_x^a &= 0.5\exp\left(\mu_a + 0.5\sigma_a^2\right)(1 - erf\left(\frac{1}{\sqrt{2}}(R_a^c - \sigma_a)\right))\\
E_y^b &= 0.5\exp\left(\mu_b + 0.5\sigma_b^2\right)(1 - erf\left(\frac{1}{\sqrt{2}}(R_b^c - \sigma_b)\right))\\
E_1^b &= 0.5(1 - erf\left(\frac{1}{\sqrt{2}}(R_b^c)\right))\\
E_1^a &= 0.5(1 - erf\left(\frac{1}{\sqrt{2}}(R_a^c)\right))\\
I_x &= \exp\left(\mu_a + 0.5\sigma_a^2\right)
\end{aligned}
$$

- $m(r_{c_a}, r_{c_b}, r_{c_c})$

$$
\begin{aligned}
m(r_{c_a}, r_{c_b}, r_{c_c}) = {} & \frac{1}{C_aC_bC_c}(S_aS_bS_cE_{xyz} - S_aS_bK_c^cE_{xy} - S_aS_cK_b^cE_{xz} - S_bS_cK_a^cE_{yz}\\
& + S_aK_b^cK_c^cE_x + S_bK_a^cK_c^cE_y + S_cK_a^cK_b^cE_z - K_a^cK_b^cK_c^cE_1)\\
& - \frac{\mu_{ca}}{C_aC_b}(S_bS_cE_{yz}^1 - S_bK_c^cE_y^1 - S_cK_b^cE_z^1 + K_b^cK_b^cE_1^1)\\
& - \frac{\mu_{cb}}{C_aC_c}(S_aS_cE_{xz}^2 - S_aK_b^cE_x^2 - S_cK_a^cE_z^2 + K_a^cK_c^cE_1^2)\\
& - \frac{\mu_{cc}}{C_aC_b}(S_aS_bE_{xy}^3 - S_aK_b^cE_x^3 - S_bK_a^cE_x^3 + K_a^cK_b^cE_1^3)\\
& + \frac{\mu_{ca}\mu_{cb}}{C_c}(S_cE_z^c - K_c^cE_1^c) + \frac{\mu_{ca}\mu_{cc}}{C_b}(S_bE_y^b - K_b^cE_1^b)\\
& + \frac{\mu_{cb}\mu_{cc}}{C_a}(S_aE_x^a - K_a^cE_1^a) - \mu_{ca}\mu_{cb}\mu_{cc}
\end{aligned}
$$

Notação:

$$
\begin{aligned}
R_a^c &= \frac{1}{\sigma_a}(ln\frac{K_a^c}{S_a} - \mu_a) \\
R_b^c &= \frac{1}{\sigma_b}(ln\frac{K_b^c}{S_b} - \mu_b) \\
R_c^c &= \frac{1}{\sigma_c}(ln\frac{K_c^c}{S_c} - \mu_c) \\
a &= 1 - \rho_{ab}^2 \\
b &= 1 - \rho_{ac}^2 \\
c &= 1 - \rho_{bc}^2 \\
A &= \rho_{ac}\rho_{bc} - \rho_{ab} \\
B &= \rho_{ab}\rho_{bc} - \rho_{ac} \\
C &= \rho_{ab}\rho_{ac} - \rho_{bc} \\
T &= \frac{1}{c}(\sigma_c B - \sigma_a c + \rho_{ab}(\sigma_c C - sigma_b c)) \\
L &= \frac{1}{\sqrt{2}}(R_a^c + T)
\end{aligned}
$$

$$
\begin{aligned}
E_{xy}^3 &= \frac{1}{2\sqrt{2\pi}} \exp\left(\mu_a + \mu_b + 0.5(\sigma_a^2 + \sigma_b^2) + \sigma_a\sigma_b\rho_{ab}\right) \int_{R_a^c}^{+\infty} \exp[-0.5(u - \sigma_a - \sigma_b\rho_{ab})^2] \\
&\cdot \left[1 - erf\left(\frac{1}{\sqrt{2(1-\rho_{ab})}}(R_b^c - u\rho_{ab} - \sigma_b(1-\rho_{ab}^2))\right)\right] du
\end{aligned}
$$

$$
\begin{aligned}
E_x^3 &= \frac{1}{2\sqrt{2\pi}} \exp\left(\mu_a + 0.5\sigma_a^2\right) \int_{R_a^c}^{+\infty} \exp[-0.5(u - \sigma_a)^2] \\
&\cdot \left[1 - erf\left(\frac{1}{\sqrt{2(1-\rho_{ab})}}(R_b^c - u\rho_{ab}))\right)\right] du
\end{aligned}
$$

$$
\begin{aligned}
E_y^3 &= \frac{1}{2\sqrt{2\pi}} \exp\left(\mu_b + 0.5\sigma_b^2\right) \int_{R_a^c}^{+\infty} \exp[-0.5(u - \sigma_a - \sigma_b\rho_{ab})^2] \\
&\cdot \left[1 - erf\left(\frac{1}{\sqrt{2(1-\rho_{ab})}}(R_b^c - u\rho_{ab} - \sigma_b(1-\rho_{ab}^2))\right)\right] du
\end{aligned}
$$

$$E_1^3 \;=\; \frac{1}{2\sqrt{2\pi}} \int_{R_a^c}^{+\infty} \exp[-0.5u^2] \left[1 - erf\left(\frac{1}{\sqrt{2(1-\rho_{ab})}} (R_b^c - u\rho_{ab})) \right) \right] du$$

$$\begin{aligned} E_{yz}^1 \;=\; & \exp \frac{1}{2\sqrt{2\pi}} \left(\mu_b + \mu_c + 0.5(\sigma_b^2 + \sigma_c^2) + \sigma_b \sigma_c \rho_{bc} \right) \int_{R_c^c}^{+\infty} \exp[-0.5(u - \sigma_c - \sigma_b \rho_{bc})^2] \\ & \cdot \left[1 - erf\left(\frac{1}{\sqrt{2(1-\rho_{bc})}} (R_b^c - u\rho_{bc} - \sigma_b(1 - \rho_{bc}^2)) \right) \right] du \end{aligned}$$

$$\begin{aligned} E_z^1 \;=\; & \frac{1}{2\sqrt{2\pi}} \exp\left(\mu_c + 0.5\sigma_c^2 \right) \int_{R_c^c}^{+\infty} \exp[-0.5(u - \sigma_c)^2] \\ & \cdot \left[1 - erf\left(\frac{1}{\sqrt{2(1-\rho_{bc})}} (R_b^c - u\rho_{bc})) \right) \right] du \end{aligned}$$

$$\begin{aligned} E_y^1 \;=\; & \frac{1}{2\sqrt{2\pi}} \exp\left(\mu_b + 0.5\sigma_b^2 \right) \int_{R_c^c}^{+\infty} \exp[-0.5(u - \sigma_b \rho_{bc})^2] \\ & \cdot \left[1 - erf\left(\frac{1}{\sqrt{2(1-\rho_{ab})}} (R_b^c - u\rho_{bc} - \sigma_b(1 - \rho_{bc}^2)) \right) \right] du \end{aligned}$$

$$E_1^3 \;=\; \frac{1}{2\sqrt{2\pi}} \int_{R_c^c}^{+\infty} \exp[-0.5u^2] \left[1 - erf\left(\frac{1}{\sqrt{2(1-\rho_{bc})}} (R_b^c - u\rho_{bc})) \right) \right] du$$

$$\begin{aligned} E_{xz}^2 \;=\; & \frac{1}{2\sqrt{2\pi}} \exp\left(\mu_a + \mu_c + 0.5(\sigma_a^2 + \sigma_c^2) + \sigma_a \sigma_c \rho_{ac} \right) \int_{R_c^c}^{+\infty} \exp[-0.5(u - \sigma_c - \sigma_a \rho_{ac})^2] \\ & \cdot \left[1 - erf\left(\frac{1}{\sqrt{2(1-\rho_{ac})}} (R_a^c - u\rho_{ac} - \sigma_a(1 - \rho_{ac}^2)) \right) \right] du \end{aligned}$$

$$\begin{aligned} E_z^2 \;=\; & \frac{1}{2\sqrt{2\pi}} \exp\left(\mu_c + 0.5\sigma_c^2 \right) \int_{R_c^c}^{+\infty} \exp[-0.5(u - \sigma_c)^2] \\ & \cdot \left[1 - erf\left(\frac{1}{\sqrt{2(1-\rho_{ac})}} (R_a^c - u\rho_{ac})) \right) \right] du \end{aligned}$$

$$
\begin{aligned}
E_x^2 &= \frac{1}{2\sqrt{2\pi}} \exp\left(\mu_a + 0.5\sigma_a^2\right) \int_{R_c^c}^{+\infty} \exp[-0.5(u - \sigma_a \rho_{ac})^2] \\
&\quad \cdot \left[1 - erf\left(\frac{1}{\sqrt{2(1-\rho_{ac})}}(R_a^c - u\rho_{ac} - \sigma_a(1-\rho_{ac}^2))\right)\right] du
\end{aligned}
$$

$$
E_1^2 = \frac{1}{2\sqrt{2\pi}} \int_{R_c^c}^{+\infty} \exp[-0.5u^2] \left[1 - erf\left(\frac{1}{\sqrt{2(1-\rho_{ac})}}(R_a^c - u\rho_{ac})\right)\right] du
$$

$$
E_z^c = 0.5 \exp\left(\mu_c + 0.5\sigma_c^2\right)(1 - erf\left(\frac{1}{\sqrt{2}}(R_c^c - \sigma_c)\right))
$$

$$
E_1^c = 0.5(1 - erf\left(\frac{R_c^c}{\sqrt{2}}\right))
$$

$$
E_y^b = 0.5 \exp\left(\mu_b + 0.5\sigma_b^2\right)(1 - erf\left(\frac{1}{\sqrt{2}}(R_b^c - \sigma_b)\right))
$$

$$
E_1^b = 0.5(1 - erf\left(\frac{R_b^c}{\sqrt{2}}\right))
$$

$$
E_x^a = 0.5 \exp\left(\mu_a + 0.5\sigma_a^2\right)(1 - erf\left(\frac{1}{\sqrt{2}}(R_a^c - \sigma_a)\right))
$$

$$
E_1^a = 0.5(1 - erf\left(\frac{R_a^c}{\sqrt{2}}\right))
$$

$$
\begin{aligned}
E(e^{x+y+z}) &= \int_{ln\frac{K_a^c}{S_a}}^{+\infty} \int_{ln\frac{K_b^c}{S_b}}^{+\infty} \int_{ln\frac{K_c^c}{S_c}}^{+\infty} f(x,y,z) exp(x+y+z)\, dz\, dy\, dx \\
&\overset{1''}{=} \frac{exp(\mu_a + \mu_b + \mu_c)}{\sqrt{8R\pi^3}} \int_{R_a^c}^{+\infty} \int_{R_b^c}^{+\infty} \int_{R_c^c}^{+\infty} exp\{-\frac{1}{2R}[au^2 + bv^2 + cw^2 \\
&\qquad +2uvA + 2uwB + 2vwC - 2R\sigma_a u - 2R\sigma_b v - 2R\sigma_c w]\}\, dw\, dv\, du
\end{aligned}
$$

$$
\begin{aligned}
&\overset{2''}{=} \frac{1}{2\pi\sqrt{c\pi}} exp(\mu_a + \mu_b + \mu_c + \frac{1}{2c}\sigma_c^2 R) \int_{R_a^c}^{+\infty} \int_{R_b^c}^{+\infty} \left(\int_{F(u,v)}^{+\infty} e^{-\zeta^2}\, d\zeta \right) \\
&\quad \cdot exp\{-\frac{1}{2R}[u^2(a - \frac{B^2}{c}) + v^2(b - \frac{C^2}{c}) + 2v(u(A - \frac{BC}{c}) - R\sigma_b + \frac{RC\sigma_c}{c}) \\
&\qquad + 2uR(\sigma\frac{B}{c} - \sigma_a)]\}\, dv\, du \\
&= \frac{1}{4\pi\sqrt{c}} exp(\mu_a + \mu_b + \mu_c + \frac{1}{2c}\sigma_c^2 R) \int_{R_a^c}^{+\infty} \int_{R_b^c}^{+\infty} [1 - erf(F(u,v))] \\
&\quad \cdot exp\{-\frac{1}{2c}[u^2 + v^2 + 2v(-u\rho_{ab} - c\sigma_b + C\sigma_c) + 2u(\sigma_c B - \sigma_a c)]\}\, dv\, du \\
&\overset{3''}{=} \frac{1}{2\pi\sqrt{2}} exp(\mu_a + \mu_b + \mu_c + \frac{1}{2c}\sigma_c^2 R + \frac{1}{2c}(\sigma_c C - c\sigma_b)^2) \\
&\quad \cdot \int_{R_a^c}^{+\infty} \int_{G(u)}^{+\infty} [1 - erf(F(u,\xi))] \\
&\quad \cdot exp\{-\frac{1}{2c}[u^2 c + v^2 + 2u(\sigma_c B - \sigma_a c + \rho_{ab}(\sigma_c C - \sigma_b c))] - \xi^2\}\, d\xi\, du \\
&= \frac{1}{2\pi\sqrt{2}} exp(\mu_a + \mu_b + \mu_c + \frac{1}{2c}\sigma_c^2 R + \frac{1}{2c}(\sigma_c C - c\sigma_b)^2) \\
&\quad \cdot \int_{R_a^c}^{+\infty} \int_{G(u)}^{+\infty} [1 - erf(F(u,\xi))] exp\{-\frac{1}{2}(u^2 + 2uT) - \xi^2\}\, d\xi\, du \\
&\overset{4''}{=} \frac{1}{2\pi\sqrt{2}} exp(\mu_a + \mu_b + \mu_c + \frac{1}{2c}\sigma_c^2 R + \frac{1}{2c}(\sigma_c C - c\sigma_b)^2 + 0.5T^2) \\
&\quad \cdot \int_{L}^{+\infty} \int_{G(\psi)}^{+\infty} [1 - erf(F(\psi,\xi))] exp\{-\psi^2 - \xi^2\}\, d\xi\, d\psi \\
&\overset{5''}{=} \frac{1}{2\pi} exp(\mu_a + \mu_b + \mu_c + 0.5(\sigma_a^2 + \sigma_b^2 + \sigma_c^2) + \sigma_a\sigma_b\rho_{ab} + \sigma_a\sigma_c\rho_{ac} + \sigma_b\sigma_c\rho_{bc} - L^2) \\
&\quad \cdot \int_0^{+\infty} H_1(\varphi) exp(-\varphi^2)\, d\varphi
\end{aligned}
$$

onde

$$
\begin{aligned}
H_1(\varphi) &= exp(-2L\varphi) \int_{G_1(\varphi)}^{+\infty} [1 - erf(F_1(\varphi,\xi))] exp(-\xi^2)\, d\xi \\
G_1(\varphi) &= \frac{1}{\sqrt{2c}}(R_b^c - c\sigma_b - C\sigma_c - \rho_{ab}(\sqrt{2}\varphi + L) + \rho_{ab}T)
\end{aligned}
$$

$$F_1(\varphi,\xi) = \sqrt{\frac{c}{2R}}[R_c^c + \frac{1}{c}(\varphi\sqrt{2}(B+\rho_{ac}C) + (\sqrt{2}L - T)(B + C\rho_{ab}) + \xi C\sqrt{2c} + Cc\sigma_b - C^2\sigma_c) - \sigma_c R]$$

As mudanças de variáveis indicadas acima são as seguintes:

$$1'' \quad \begin{cases} u = \frac{1}{\sigma_a}(x-\mu_a) \\ v = \frac{1}{\sigma_b}(y-\mu_b) \\ w = \frac{1}{\sigma_c}(z-\mu_c) \end{cases}$$

$$2'' \quad \left\{ \zeta = \sqrt{\tfrac{c}{2R}}(w + \tfrac{1}{c}(uB + vC - \sigma_c R)) \right.$$

$$3'' \quad \left\{ \xi = \tfrac{1}{\sqrt{2c}}(v - c\sigma_b + C\sigma_c - u\rho_{ab}) \right.$$

$$4'' \quad \left\{ \psi = \tfrac{1}{\sqrt{2}}(u+T) \right.$$

$$5'' \quad \left\{ \varphi = \psi - L \right.$$

$$E(e^{x+y}) = \frac{1}{2\pi}\exp\left(\mu_a + \mu_b + \frac{1}{2}(\sigma_a^2 + \sigma_b^2) + \sigma_a\sigma_b\rho_{ab}\right)\int_0^{+\infty} H_2(\varphi)exp(-\varphi^2)\,d\varphi$$

onde

$$H_2(\varphi) = exp(-2L_2\varphi)\int_{G_2(\varphi)}^{+\infty}[1 - erf(F_2(\varphi,\xi))]exp(-\xi^2)\,d\xi$$

$$G_2(\varphi) = G_1(\varphi) - \frac{1}{\sqrt{2c}}\sigma_c C$$

$$F_2(\varphi,\xi) = F_1(\varphi,\xi) + \sqrt{\frac{c}{2R}}\sigma_c(R + \frac{C^2}{c})$$

$$T_2 = -\sigma_a - \sigma_b\rho_{ab}$$

$$L_2 = \frac{1}{\sqrt{2}}(R_a^c + T_2)$$

$$E(e^{x+z}) = \frac{1}{2\pi}\exp\left(\mu_a + \mu_c + \frac{1}{2}(\sigma_a^2 + \sigma_c^2) + \sigma_a\sigma_c\rho_{ac}\right)\int_0^{+\infty} H_3(\varphi)exp(-\varphi^2)\,d\varphi$$

onde

$$
\begin{aligned}
H_3(\varphi) &= exp(-2L_2\varphi)\int_{G_3(\varphi)}^{+\infty}[1-erf(F_3(\varphi,\xi))]exp(-\xi^2)\,d\xi \\
G_3(\varphi) &= G_1(\varphi)+\frac{1}{\sqrt{2c}}\sigma_b C \\
F_3(\varphi,\xi) &= F_1(\varphi,\xi)+\sqrt{\frac{c}{2R}}\sigma_b C \\
T_3 &= T_1+\sigma_b\rho_{ab} \\
L_3 &= \frac{1}{\sqrt{2}}(R_a^c+T_3)
\end{aligned}
$$

$$
E(e^{y+z}) = \frac{1}{2\pi}\exp\left(\mu_b+\mu_c+\frac{1}{2}(\sigma_b^2+\sigma_c^2)+\sigma_b\sigma_c\rho_{bc}\right)\int_0^{+\infty}H_4(\varphi)exp(-\varphi^2)\,d\varphi
$$

onde

$$
\begin{aligned}
H_4(\varphi) &= exp(-2L_4\varphi)\int_{G_4(\varphi)}^{+\infty}[1-erf(F_4(\varphi,\xi))]exp(-\xi^2)\,d\xi \\
G_4(\varphi) &= G_1(\varphi) \\
F_4(\varphi,\xi) &= F_1(\varphi,\xi) \\
T_4 &= T_1+\sigma_a \\
L_4 &= \frac{1}{\sqrt{2}}(R_a^c+T_4)
\end{aligned}
$$

$$
E(e^x) = \frac{1}{2\pi}\exp(\mu_a+\frac{1}{2}\sigma_a^2-L_5^2)\int_0^{+\infty}H_5(\varphi)exp(-\varphi^2)\,d\varphi
$$

onde

$$
\begin{aligned}
H_5(\varphi) &= exp(-2L_5\varphi)\int_{G_5(\varphi)}^{+\infty}[1-erf(F_5(\varphi,\xi))]exp(-\xi^2)\,d\xi \\
G_5(\varphi) &= G_1(\varphi)-\frac{1}{\sqrt{2c}}(\sigma_c C-\sigma_b c)
\end{aligned}
$$

$$
\begin{aligned}
F_5(\varphi,\xi) &= F_1(\varphi,\xi) - \sqrt{\frac{c}{2R}}[\frac{1}{c}(\sigma_b C c - C^2\sigma_c) - R\sigma_c] \\
T_5 &= -\sigma_a \\
L_5 &= \frac{1}{\sqrt{2}}(R_a^c + T_5)
\end{aligned}
$$

$$
E(e^y) = \frac{1}{2\pi}\exp(\mu_b + \frac{1}{2}\sigma_b^2 - L_6^2)\int_0^{+\infty} H_6(\varphi)exp(-\varphi^2)\,d\varphi
$$

onde

$$
\begin{aligned}
H_6(\varphi) &= exp(-2L_6\varphi)\int_{G_6(\varphi)}^{+\infty}[1 - erf(F_6(\varphi,\xi))]exp(-\xi^2)\,d\xi \\
G_6(\varphi) &= G_1(\varphi) - \frac{1}{\sqrt{2c}}\sigma_c C \\
F_6(\varphi,\xi) &= F_1(\varphi,\xi) + \sqrt{\frac{c}{2R}}\sigma_c(\frac{C^2}{c} - R) \\
T_6 &= -\sigma_b\rho_{ab} \\
L_6 &= \frac{1}{\sqrt{2}}(R_a^c + T_6)
\end{aligned}
$$

$$
E(e^z) = \frac{1}{2\pi}\exp(\mu_c + \frac{1}{2}\sigma_c^2 - L_7^2)\int_0^{+\infty} H_7(\varphi)exp(-\varphi^2)\,d\varphi
$$

onde

$$
\begin{aligned}
H_7(\varphi) &= exp(-2L_7\varphi)\int_{G_7(\varphi)}^{+\infty}[1 - erf(F_7(\varphi,\xi))]exp(-\xi^2)\,d\xi \\
G_7(\varphi) &= G_1(\varphi) - \frac{1}{\sqrt{2c}}\sigma_b C \\
F_7(\varphi,\xi) &= F_1(\varphi,\xi) + \sqrt{\frac{c}{2R}}\sigma_b C \\
T_7 &= \frac{1}{c}(\sigma_c B + C\sigma_c\rho_{ab} \\
L_7 &= \frac{1}{\sqrt{2}}(R_a^c + T_7)
\end{aligned}
$$

$$E(1) = \frac{1}{2\pi}\exp(-L_8^2)\int_0^{+\infty} H_8(\varphi) exp(-\varphi^2)\, d\varphi$$

onde

$$\begin{aligned}
H_8(\varphi) &= exp(-2L_8\varphi)\int_{G_8(\varphi)}^{+\infty} [1 - erf(F_8(\varphi,\xi))] exp(-\xi^2)\, d\xi \\
G_8(\varphi) &= G_1(\varphi) - \frac{1}{\sqrt{2c}}(\sigma_c C - \sigma_b c) \\
F_8(\varphi,\xi) &= F_1(\varphi,\xi) - \sqrt{\frac{c}{2R}}(C\sigma_b - \sigma_c(\frac{C^2}{c} - R)) \\
T_8 &= 0 \\
L_8 &= \frac{1}{\sqrt{2}}(R_a^c)
\end{aligned}$$

Capítulo 4

Cálculo Numérico de Covariâncias

Definidas as expressões das covariâncias do capítulo 2, o trabalho agora é computá-las de forma eficiente.

Genericamente, as integrais que apareceram nas expressões de covariância são do tipo

$$\int_a^{+\infty} \exp\left(-(bx+c)^2\right) erf(g(x))dx \tag{4.1}$$

onde a, b, c são constantes, g(x) é função linear e

$$erf(g(x)) = \int_{g(x)}^{+\infty} e^{-t^2}\, dt$$

.

4.1 Método de Quadratura Numérica

O método que usaremos para calcular integrais do tipo 4.1 consiste em aplicar o método dos trapézios em intervalos $I_n^k \subset \Re$ de amplitudes constantes, mas cada um com partição diferente. Esta partição não homogênea é para favorecer os intervalos que necessitam de um maior número de pontos (nos quais o valor da função a ser integrada necessita ser calculado).

Entenda-se por I_n^k o k-ésimo subintervalo da n-ésima aproximação da integral.

Posteriormente aplicaremos extrapolação de Richardson para acelerar a convergência das integrais (adiante explicaremos o que é extrapolação e método dos trapézios).

Estes subintervalos I_n^k's serão obtidos da seguinte forma:

- fixa-se uma amplitude inicial $\Delta \in \Re$. Uma primeira aproximação da integral pode ser dada pelo uso da regra do trapézio no intervalo $I_1^0 = [a, a+\Delta]$.
- a cada refinamento de I_0 aumenta-se de um o número de intervalos I_n^k.

Na n-ésima partição da integral, haverá n subintervalos I_n^k, cada um com partições diferentes:

- $n = 1 : [a, a + \Delta]$
- $n = 2 : [a, a + \Delta], [a + \Delta, a + 2\Delta]$

  $\vdots$
- $n = k : [a, a + \Delta], [a + \Delta, a + 2\Delta], \cdots, [a + (k - 1)\Delta, a + k\Delta]$

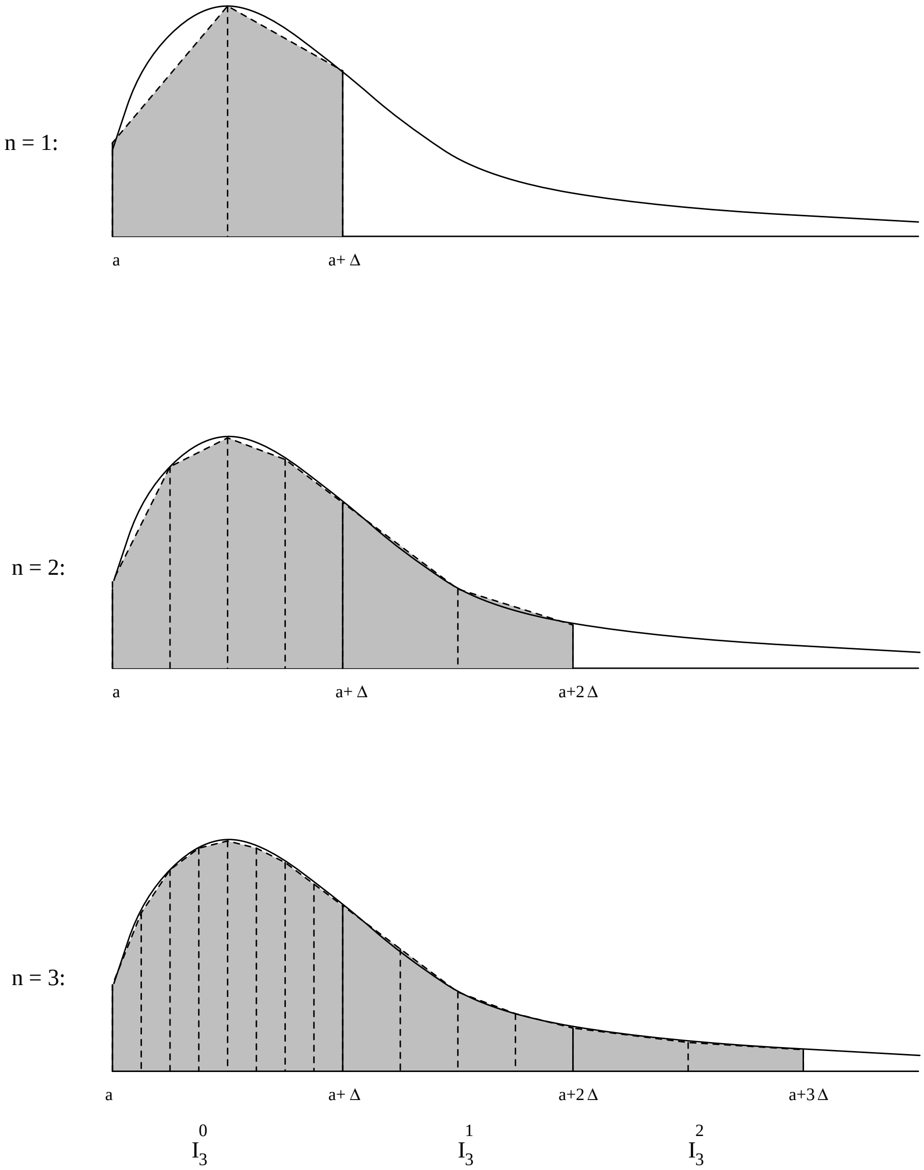

Figura 4.1: partições para a regra do trapézio

Cada intervalo I_n^k possui as seguintes características:

$$\left\{\begin{array}{rl} x_{0k} = & a + k\Delta \\ x_k = & a + (k+1)\Delta \\ h_k = & \frac{\Delta}{2^{n-k}} \\ N = & 2^{n-k} + 1 \end{array}\right. \qquad \left(\begin{array}{rl} k = & 0, 1, 2, \cdots, n-1 \\ n = & 1, 2, \cdots \end{array}\right)$$

onde

N é o número de abcissas, x_{0k} e x_k são extremos de intervalo em I_n^k.

- $n = 1 : I_1^0$
- $n = 2 : I_2^0, I_2^1$

  $\vdots$
- $n = k : I_k^0, I_k^1, I_k^2, \cdots, I_k^{k-1}$

A figura 4.1 ilustra sucessivas aproximações para a integral 4.1 e como as partições acima são feitas.

A n-ésima aproximação da integral será dada por

$$\int_a^{+\infty} f(x)dx \quad = \quad \sum_{k=0}^{n-1} \int_{I_n^k} f(x)dx \tag{4.2}$$

e cada integral desta somatória será calculada pela regra do trapésio.

O método dos trapézios consiste em aproximar a área abaixo de uma curva $y = f(x)$ em $[a, b]$ pela soma de N trapezóides de altura h,

$$h = x_i - x_{i-1} = \frac{b-a}{N}$$

e bases

$$f(x_i) \quad \text{e} \quad f(x_{i-1})$$

.

Com esta notação, o método fornece-nos a aproximação

$$\int_a^b f(x)\, dx \cong h \sum_{i=0}^{N} f_i - 0.5h(f_0 + f_N) \tag{4.3}$$

onde $f_i = f(x_i)$ (veja figura 4.2).

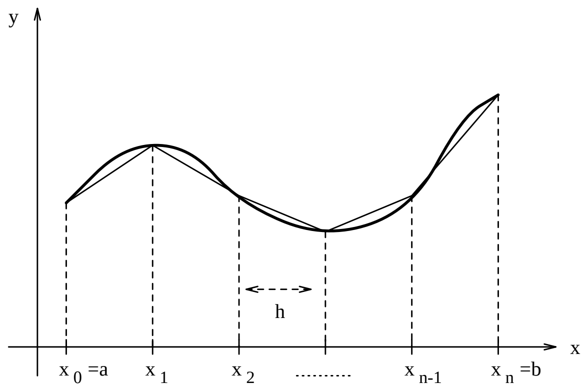

Figura 4.2: regra do trapézio

O erro desta aproximação é dado por

$$E = -\frac{1}{12}(b-a)h^2 f''(\eta) \qquad \eta \in (a,b)$$

A convergência é, portanto, lenta. Uma melhor aproximação é obtida ao custo de uma diminuição do valor de h, o que aumenta o número de pontos x_i usados.

Mas podemos melhorar este resultado.

Considere aproximações da integral 4.3 para 2 valores diferentes de N (N_1 e N_2):

$$\begin{aligned} I &= I_1 - \frac{1}{12}(b-a)h_1^2 f''(\eta_1) \\ I &= I_2 - \frac{1}{12}(b-a)h_2^2 f''(\eta_2) \end{aligned}$$

$\eta_1, \eta_2 \in (a,b)$.

Supondo $f''(\eta_1) \cong f''(\eta_2)$ teremos

$$I \cong I_2 + \frac{h_2^2}{h_1^2 - h_2^2}(I_2 - I_1)$$

Este tipo de procedimento, no qual geramos uma terceira aproximação a partir de outras duas, e que espera-se ser melhor, é chamado *Extrapolação de Richardson*.

Vamos supor que o erro de um procedimento numérico com passo fixo h possa ser escrito na forma

$$E = \sum_{j=k}^{\infty} a_j h^j \qquad a_k \neq 0$$

onde a_j não depende de h.

Sejam

$$I = I_1 + \sum_{j=k}^{\infty} a_j h_1^j \tag{4.4}$$

$$I = I_2 + \sum_{j=k}^{\infty} a_j h_2^j \tag{4.5}$$

Multiplique 4.4 por h_2^k e 4.5 por h_1^k e subtraia uma da outra. Assim teremos

$$I = \frac{1}{h_1^k - h_2^k}(I_2 h_1^k - I_1 h_2^k) + \sum_{j=k}^{\infty} a_j \frac{1}{h_1^k - h_2^k}(h_2^j h_1^k - h_1^j h_2^k)$$

Se $h_1 = 2h_2$ e $h_2 = h$

$$I = \frac{1}{2^k - 1}(I_2 2^k - I_1) + \sum_{j=k+1}^{\infty} a_j h^j (2^k - 2^j) \tag{4.6}$$

O efeito do processo acima foi aumentar a menor potência de h na fórmula do erro.

Veremos agora como podemos usar estas idéias para acelerar a convergência das integrais pelo método dos trapézios. O método que une regra do trapézios e extrapolação de Richardson é chamado **Método de Romberg**.

4.2 Tratamento de Erros

O objetivo aqui é expressar o erro cometido no processo de integração numérica anterior (dado em 4.2) em termos de potência de h (a chamada fórmula de Euler-Maclaurin), a fim de posteriormente aplicarmos extrapolação de Richardson e acelerar a convergência das integrais.

4.2.1 Fórmula de Euler-Maclaurin

1. **Definição 4.2.1.** *Os polinômios de Bernoulli são dados por*

$$\frac{t(exp(xt) - 1)}{exp(1) - 1} = \sum_{k=0}^{+\infty} B_k(x) \frac{t^k}{k!}$$

e

$$\frac{t}{exp(t) - 1} = \sum_{k=0}^{+\infty} B_k \frac{t^k}{k!}$$

2. Propriedades:

   (a) $B_k(0) = 0 \quad (k \geq 0)$

   (b) $B_k(1) = 0 \quad (k \neq 1)$

   (c) $B_{2k+1} = 0 \quad (k > 0)$

   (d) $B'_{2k}(x) = 2kB_{2k-1}(x) \quad (k > 1)$

   (e) $B'_{2k+1}(x) = (2k+1)[B_{2k}(x) + B_{2k}] \quad (k \geq 0)$

dem de d) e e):

$$\begin{aligned}
\frac{d}{dx}\left[\frac{t(\exp(tx)-1)}{\exp(t)-1}\right] &= \frac{t^2 \exp tx}{exp(t)-1} \\
&= \sum_{k=0}^{+\infty} B'_k(x)\frac{t^k}{k!} \\
&= \sum_{k=0}^{+\infty} B'_{2k}(x)\frac{t^{2k}}{(2k)^2} + \sum_{k=0}^{+\infty} B'_{2k+1}(x)\frac{t^{2k+1}}{(2k+1)!} \qquad (4.7)
\end{aligned}$$

$$\begin{aligned}
\frac{t^2\exp(tx)}{exp(t)-1} &= \frac{t^2(\exp((tx)-1))}{\exp(t)-1} + \frac{t^2}{\exp(t)-1} \\
&= t\sum_{k=0}^{+\infty} B_k(x)\frac{t^k}{k!} + t\sum_{k=0}^{+\infty} B_k\frac{t^k}{k!} \\
&= \sum_{k=0}^{+\infty} B_{2k}(x)\frac{t^{2k+1}}{(2k)!} + \sum_{k=1}^{+\infty} B_{2k-1}(x)\frac{t^{2k}}{(2k-1)!} + \sum_{k=0}^{+\infty} B_{2k}\frac{t^{2k+1}}{(2k)!} + \sum_{k=1}^{+\infty} B_{2k-1}\frac{t^{2k}}{(2k-1)!} \\
&= \sum_{k=0}^{+\infty} \left(B_{2k}(x) + B_{2k}\right)(2k+1)\frac{t^{2k+1}}{(2k+1)!} + \sum_{k=1}^{+\infty} 2k\left(B_{2k-1}(x) + B_{2k-1}\right)\frac{t^{2k}}{(2k)!} \qquad (4.8)
\end{aligned}$$

Por 4.7 e 4.8 seguem as propriedades desejadas. □

Definamos agora

$$X_k = \frac{1}{(2k)!}\int_0^h B_{2k}\left(\frac{y}{h}\right) f^{(2k)}(x+y)dy \tag{4.9}$$

onde $f(x)$ é a função integrando de 4.2.

Integrando por partes 4.9,

$$\begin{aligned}
X_k &= \frac{1}{(2k)!}\left[B_{2k}\left(\frac{y}{h}\right) f^{(2k-1)}(x+y)\right]_0^h - \frac{1}{h(2k)!}\int_0^h B_{2k}'\left(\frac{y}{h}\right) f^{(2k-1)}(x+y)dy \\
&= -\frac{2k}{h(2k)!}\int_0^h B_{2k-1}\left(\frac{y}{h}\right) f^{(2k-1)}(x+y)dy \\
&= \frac{1}{h^2(2k-1)!}\int_0^h B_{2k-1}'\left(\frac{y}{h}\right) f^{(2k-2)}(x+y)dy \\
&= \frac{2k-1}{h^2(2k-2)!}\int_0^h B_{2k-2}\left(\frac{y}{h}\right) f^{(2k-2)}(x+y)dy \\
&= \frac{1}{h^2(2k-2)!}\left[\int_0^h B_{2k-2}\left(\frac{y}{h}\right) f^{(2k-2)}(x+y)dy + \left[B_{(2k-2)} f^{(2k-3)}(x+y)\right]_0^h\right] \\
&= \frac{1}{h^2}X_{k-1} + \frac{1}{h^2}\frac{B_{2k-2}}{(2k-2)!}\left(f^{(2k-3)}(x+h) - f^{(2k-3)}(x)\right) \qquad (k>1)
\end{aligned}$$

Logo,

$$h^2 X_k - X_{k-1} = \frac{B_{2k-2}}{(2k-2)!}\left(f^{(2k-3)}(x+h) - f^{(2k-3)}(x)\right) \tag{4.10}$$

Quando $k=1$ temos,

$$\begin{aligned}
X_1 &= \frac{1}{2}\int_0^h B_2\left(\frac{y}{h}\right) f''(x+y)dy \\
&= \frac{1}{2}\int_0^h \left(\frac{y^2}{h^2} - \frac{y}{h}\right) f''(x+y)dy \\
&= \left[\frac{1}{2}\int_0^h \left(\frac{y^2}{h^2} - \frac{y}{h}\right) f'(x+y)\right]_0^h - \frac{1}{2}\int_0^h \left(\frac{y^2}{h^2} - \frac{y}{h}\right) f'(x+y)dy \\
&= -\left[\left(\frac{y}{h^2} - \frac{1}{2h}\right) f(x+y)\right]_0^h + \frac{1}{h^2}\int_0^h f(x+y)dy \\
&= \frac{1}{2h}\left[f(x+y) + f(x)\right] + \frac{1}{h^2}\int_0^{x+h} f(y)dy \qquad (4.11)
\end{aligned}$$

Usando 4.10 e 4.11 temos

$$\begin{aligned}
\frac{1}{2}\left[f(x+h)-f(x)\right] &= \frac{1}{h}\int_0^{x+h} f(y)dy + hX_1 \\
&= \frac{1}{h}\int_0^{x+h} f(y)dy + h^3X_2 + h\frac{B_2}{2!}\left[f^{'}(x+h)-f^{'}(x)\right] \\
&= \frac{1}{h}\int_0^{x+h} f(y)dy + h\frac{B_2}{2!}\left[f^{'}(x+h)-f^{'}(x)\right] \\
&\quad +h^3\frac{B_4}{4!}\left[f^{'''}(x+h)-f^{'''}(x)\right] - h^5X_3 \\
&= \cdots \\
&= \frac{1}{h}\int_0^{x+h} f(y)dy + \sum_{k=1}^{m}\frac{B_{2k}}{(2k)!}h^{2k-1}\left[f^{2k-1}(x+h)-f^{2k-1}(x)\right] \\
&\quad -h^{2m+1}X_{m+1}
\end{aligned}$$

Podemos mudar os limites de integração em 4.9 para $[jh,(j+1)h]$.Repetindo a derivação de 4.11 para $j=1,2,\cdots,n-1$ e substituindo x por x_0 obtemos

$$\begin{aligned}
\sum_{j=0}^{n} f(x_0+jh) &= \frac{1}{h}\int_{x_0}^{x_0+nh} f(y)dy + \frac{1}{2}\left(f(x_0+h)-f(x_0)\right) \\
&= \sum_{k=1}^{m}\frac{B_{2k}}{(2k)!}h^{2k-1}\left[f^{2k-1}(x_0+h)-f^{2k-1}(x_0)\right] + E_m
\end{aligned} \tag{4.12}$$

onde

$$E_m = -\frac{h^{2m+1}}{(2m+2)!}\int_0^{nh} B_{2m+2}\left(\frac{y}{h}\right) f^{(2m+2)}(x_0+y)dy$$

Usando as propriedades dos polinômios de Bernoulli, temos

$$E_m = nh^{2m+2}\frac{B_{2m+2}}{(2m+2)!}f^{(2m+2)}(\xi), \qquad \xi\in(x_{0k},x_k)$$

Aplicando 4.12 obtemos a fórmula de integração numérica

$$\begin{aligned}
\int_{I_k^n} f(y)dy &= \int_{x_{0k}}^{x_0} f(y)dy \\
&= h_k \sum_{j=0}^{N-1} f(x_{0k} + jh_k) - \frac{h_k}{2}\left[f(x_k) - f(k_{0k})\right] + E_k
\end{aligned}$$

Logo,

$$\begin{aligned}
\int_a^{+\infty} f(y)dy &= \sum_{k=0}^{n-1} \int_{I_n^k} f(y)dy + E_r \\
&= \sum_{k=0}^{n-1} \left\{ \frac{\Delta}{2^{n-k}} \sum_{j=0}^{2^{n-k}} f(x_{0k} + jh_k) - \frac{h_k}{2}[f(x_{0k}) + f(x_k)] + E_k \right\} + E_r \\
&= \sum_{k=0}^{n-1} \left\{ \frac{\Delta}{2^{n-k}} \sum_{j=0}^{2^{n-k}} f(x_{0k} + jh_k) - \frac{h_k}{2}[f(x_{0k}) + f(x_k)] \right\} + E_t
\end{aligned}$$

onde

$$E_t = -\sum_{k=0}^{n-1} \sum_{l=1}^{+\infty} \frac{B_{2l}}{(2l)!} \left(\frac{\Delta}{2^{n-k}}\right)^{2l} \left[f^{(2l-1)}(x_k) - f^{(2l-1)}(x_{0k})\right] + \int_{a+n\Delta}^{+\infty} f(y)dy$$

4.2.2 Extrapolação de Richardson

Defina $T_{0,i}$ como sendo a aproximação da integral referente ao i-ésimo passo do processo numérico. A fim de acelerar a convergência da integral, queremos construir as aproximações $T_{k,j}$ da integral, referentes à k-ésima aproximação do j-ésimo passo, aplicando *Extrapolação de Richardson.*

$$\begin{array}{llllll}
n = 1: & T_{0,1} & & & & \\
n = 2: & T_{0,2} & T_{1,1} & & & \\
n = 3: & T_{0,3} & T_{1,2} & T_{2,1} & & \\
\vdots & \vdots & \vdots & \vdots & \ddots & \\
n = j: & T_{0,j} & T_{1,j-1} & T_{2,j-2} & \cdots & T_{j-1,1}
\end{array}$$

Denotando

$$J = \int_a^{+\infty} f(y)dy$$

temos,

$$T_{0,n} = J + \sum_{l=1}^{+\infty} \frac{B_{2l}}{(2l!)} \left(\frac{\Delta}{2^n}\right)^{2l} \sum_{k=0}^{n-1} 4^{kl} \left[f^{(2l-1)}(x_k) - f^{(2l-1)}(x_{0k})\right] - \int_{a+n\Delta}^{+\infty} f(y)dy$$

$$\begin{aligned} T_{0,n+1} = {} & J + \sum_{l=1}^{+\infty} \frac{B_{2l}}{(2l!)} \left(\frac{\Delta}{2^n}\right)^{2l} \frac{1}{2^{2l}} \sum_{k=0}^{n-1} 4^{kl} \left[f^{(2l-1)}(x_k) - f^{(2l-1)}(x_{0k})\right] \\ & + \sum_{l=1}^{+\infty} \frac{B_{2l}}{(2l!)} \left(\frac{\Delta}{2^{n+1}}\right)^{2l} 4^{kl} \left[f^{(2l-1)}(x_k) - f^{(2l-1)}(x_{0k})\right] \\ & - \int_{a+n\Delta}^{+\infty} f(y)dy + \frac{4}{3} \int_{a+n\Delta}^{a+(n+1)\Delta} f(y)dy \end{aligned}$$

Definindo

$$T_{1,n} = \frac{4T_{0,n+1} - T_{0,n}}{3}$$

temos

$$\begin{aligned} T_{1,n} = {} & J + \sum_{l=2}^{+\infty} \frac{B_{2l}}{(2l!)} \left(\frac{\Delta}{2^n}\right)^{2l} \left(\frac{4}{2^{2l}} - 1\right) \sum_{k=0}^{n-1} 4^{kl} \left[f^{(2l-1)}(x_k) - f^{(2l-1)}(x_{0k})\right] \\ & + \sum_{l=1}^{+\infty} \frac{B_{2l}}{(2l!)} \left(\frac{\Delta}{2^{n+1}}\right)^{2l} 4^{nl} \left[f^{(2l-1)}(x_k) - f^{(2l-1)}(x_{0k})\right] \\ & - \int_{a+(n+1)\Delta}^{+\infty} f(y)dy + \frac{1}{3} \int_{a+n\Delta}^{a+(n+1)\Delta} f(y)dy \end{aligned}$$

Observe que essa convergência pode não existir dependendo do comportamento da função $f(.)$. Mas no caso $f(x) = \exp(-x^2)$, $f \in C^\infty(\Re)$ e seu decaimento para zero quando $|x| \to \infty$ é rápido o suficiente para compensar o crescimento dos números de Bernoulli.

Genericamente,

$$T_{j,n} = \frac{1}{4^j - 1} \left[4^j T_{j-1,n+1} - T_{j-1,n}\right]$$

Dessa forma, os primeiros termos da somatória do erro que envolvem potências de h vão sendo eliminados.

Exemplo

O método acima nos gera as seguintes aproximações para a integral

$$\int_0^{\infty} (1 - erf(3x-2)) \exp(-x^2) dx$$

onde

$$erf(x) = \frac{2}{\sqrt{\pi}} \int_0^x \exp(-x^2) dx$$

A primeira aproximação da integral sem o uso da extrapolação é dada por:

$$\begin{array}{ll} n=1: & 1.\underline{1}05380599 \\ n=2: & 1.\underline{1}21862111 \\ n=3: & 1.1\underline{1}7203034 \\ n=4: & 1.11\underline{5}603036 \\ n=5: & 1.11\underline{5}176011 \\ n=6: & 1.115\underline{0}67668 \\ n=7: & 1.115\underline{0}40485 \\ n=8: & 1.1150\underline{3}3683 \end{array}$$

Observe que a convergência das aproximações da integral é lenta se utilizarmos somente a regra do trapézio. Mas, extrapolando os valores acima obtemos

$1.\underline{1}27355948$	$\downarrow$					
$1.11\underline{5}650007$	$1.1\underline{1}4869611$					
$1.115\underline{0}69704$	$1.11503\underline{1}017$	$1.1150\underline{3}3579$				
$1.1150\underline{3}3669$	$1.11503\underline{1}266$	$1.11503\underline{1}270$	$1.11503\underline{1}261$			
$1.11503\underline{1}553$	$1.1150314\underline{1}2$	$1.1150314\underline{1}4$	$1.11503141\underline{5}$	$1.11503141\underline{5}$		
$1.115031\underline{4}24$	$1.11503141\underline{5}$	$1.11503141\underline{5}$	$1.11503141\underline{5}$	$1.11503141\underline{5}$	$1.11503141\underline{5}$	
$1.11503141\underline{5}$	$1.11503141\underline{5}$	$1.11503141\underline{5}$	$1.11503141\underline{5}$	$1.11503141\underline{5}$	$1.11503141\underline{5}$	$1.11503141\underline{5}$

Cada dígito sublinhado é o último algarismo correto na aproximação. Neste caso, 1.115031415 é o valor aproximado da integral.

Podemos aqui fixar uma tolerância ϵ para a aproximação da integral e limitar o número de passos na extrapolação, por exemplo definindo um critério de parada da forma $|T_{j,n+1} - T_{j,n}| < \epsilon$ e $|T_{j+1,n} - T_{j,n}| < \epsilon$.

O exemplo anterior é de caráter ilustrativo, visto que poderíamos ter parado as iterações já na segunda coluna (apontada pela flechinha), de acordo com o critério de parada estipulado.

Capítulo 5

Cálculo Exato de Assimetria e Co-assimetria

Observando as expressões das assimetrias e co-assimetrias da seção 3.2, deparamo-nos com o problema de avaliar numericamente integrais duplas e triplas em domínios ilimitados.

Nosso problema agora é avaliar expressões do tipo:

$$\int_{a}^{+\infty}\int_{g(x)}^{+\infty} \exp\left(-x^2-y^2\right)F(x,y)dydx \tag{5.1}$$

onde a, b e c são constantes, $g(x)$ é função linear em y e

$$F(x,y) = erf(H(x,y)) = \int_{H(x,y)}^{+\infty} exp(-z^2)dz$$

5.1 Cubatura Numérica

Para resolver as integrais do tipo 5.1 precisamos de um método que ofereça boa precisão juntamente com o menor número de nós possível nas variáveis x e y.

Podemos redefinir 5.1 por

$$\int_{a}^{+\infty} H(t)exp(-t^2)dt \tag{5.2}$$

onde

$$H(t) = \int_{g(t)}^{+\infty} F(t,u)exp(-u^2)du \tag{5.3}$$

Usaremos o método de quadratura Gaussiana com função peso $exp(-x^2)$ no intervalo $[a, +\infty)$ para avaliar a integrai 5.2 e o método de Romberg para avaliar 5.3.

O método de Gauss para avaliar integrais do tipo

$$\int_a^b f(x)w(x)dx \tag{5.4}$$

consiste em encontrar pesos $A_i's$ e nós $x_i's$ ($x_i \in [a,b]$) específicos para aproximar a integral acima pela soma

$$\int_a^b f(x)w(x)dx \cong Q(f;w;a,b) = \sum_i A_i f(x_i) \tag{5.5}$$

$w(x) > 0$ é chamada função peso (no nosso caso, $w(x) = \exp(-x^2)$), e deve ser integrável em $[a,b]$.

Vamos encontrar (n+1) pontos ($x_i, y_i = f(x_i)$) e pesos A_i, n arbitrário, impondo a restrição de que a aproximação 5.5 seja exata para polinômios de grau menor ou igual a 2n-1. Isto é, que seja exata nos casos em que $f(x)$ é um polinômio de grau menor ou igual a $2n-1$.

Podemos determinar um polinômio

$$Q_n(x) = x^n + \sum_{k=0}^{n-1} q_k x^k \tag{5.6}$$

tal que Q_n é ortogonal[1] a todos os monômios x^j ($0 \le j \le n-1$) em $[a,b]$, isto é,

$$\int_a^b Q_n(x)x^j \exp(-x^2)dx = 0 \qquad (j = 0..n-1)$$

De 5.6, obtemos o seguinte sistema linear em q_k:

$$\sum_{k=0}^{n-1} q_k \int_a^b x^{k+j} exp(-x^2)dx = -\int_a^b x^{n+j} exp(-x^2)dx \tag{5.7}$$

Expressando $P(x)$, grau($P(x)$)≤2n-1, por

$$P(x) = S(x)Q_n(x) + R(x) \tag{5.8}$$

onde grau($S(x)$) e grau($R(x)$) são menores ou iguais a n-1, teremos $P(x_i) = R(x_i)$ **se** x_i **forem os zeros de** $Q_n(x)$.

Como grau($R(x)$) ≤n-1, por interpolação Lagrangiana temos

$$R(x) = \sum_{i=1}^{n} R(x_i)L_i(x)$$

[1]Segundo o produto interno $< f, g >= \int_a^b f(x)g(x)w(x)\,dx$

com

$$L_i(x) = \prod_{j=1, j\neq i}^{n} \frac{x - x_j}{x_i - x_j}$$

Como também grau($S(x)$) $\leq$n-1, por ortogonalidade temos

$$\int_a^b S(x)Q_n(x)exp(-x^2)dx = \sum_{j=0}^{n-1} s_j \int_a^b x^j Q_n(x)exp(-x^2)dx = 0$$

Então, de 5.8,

$$\int_a^b P(x)exp(-x^2)dx = \int_a^b R(x)exp(-x^2)dx = \sum_{i=1}^{n} R(x_i) \int_a^b L_i(x)exp(-x^2)dx = \sum_{i=1}^{n} A_i P(x_i)$$

onde

$$A_i = \int_a^b L_i(x)exp(-x^2)dx \tag{5.9}$$

Com isso determinamos valores ótimos de nós e pesos da quadratura Gaussiana.

Resumindo, os passos para encontrar os valores dos nós e pesos ótimos são os seguintes:

1. Resolva o sistema linear 5.7 e ache os valores de q_k

2. Ache as raízes x_i (i=1..n) de $Q_n(x)$ dado em 5.6

3. Os pesos A_i (i=1..n) são encontrados por 5.9.

Observe que o intervalo $[a, b]$ só foi usado como limites de integração e, portanto, nada impede que todo o procedimento acima seja feito com a=$+\infty$ ou b=$-\infty$.

A tabela abaixo contém os valores de nós e pesos para o caso a=0 e b=$+\infty$.

n=16	
x_i	A_i
3.375541041407145E-03	1.869170292281806E-02
5.804615541291068E-02	9.002873675900949E-02
1.820547674925986E-01	1.513016842298416E-01
3.679542616521208E-01	1.867836831554472E-01
6.067329515918165E-01	1.814697491540716E-01
8.897071867487791E-01	1.370740042987264E-01
1.209623943058248E+00	7.788177757936561E-02
1.561080047413430E+00	3.198902321781417E-02
1.940565510629326E+00	9.092778869377570E-03
2.346403257792925E+00	1.702422581625416E-03
2.778776657707638E+00	1.977389463502662E-04
3.240017969804344E+00	1.316842547890586E-05
3.735450317676748E+00	4.489296733529744E-07
4.275562412865718E+00	6.537390919551276E-09
4.882323348658508E+00	2.904141450987963E-11
5.615285661142339E+00	1.709992622046885E-14

Aplicando integração Gaussiana ao nosso problema inicial 5.1 , a aproximação da integral dupla, após mudanças apropriadas de variáveis, será dada por[2]:

$$\int_a^{+\infty}\int_{g(x)}^{+\infty} exp(-x^2-y^2)F(x,y)dydx \cong \sum_{i=1}^{n}\sum_{j=1}^{m} A_iB_jF(x_i,y_j)$$

onde x_i, A_i são os nós e os pesos na direção x e y_i, B_i são os nós e os pesos na direção y.

Uma estimativa do erro da integração Gaussiana é dada pelo teorema (ver [13]):

Teorema 1. *Se* $f \in C^{2n}[a,b]$, *então*

$$\int_a^b w(x)f(x)dx - \sum_{i=1}^{n} A_if(x_i) = \frac{f^{(2n)}(\xi)}{(2n)!} < p_n, p_n >$$

para algum $\xi \in (a,b)$ *e*

$$< p_n, p_n >= \int_a^b w(x)p_n^2 dx$$

$$p_n(x) = \prod_{i=1}^{n}(x-x_i)$$

[2]Esta expressão não corresponde, na prática, ao método que usamos para avaliar integrais do tipo 5.1. Isto será explicado mais adiante.

5.2 Comparação entre métodos de integração

Outros tipos de quadratura Gaussiana poderiam ter sido usados para resolver integrais do tipo 4.1 e 5.1.

Por exemplo, os pesos e nós das quadraturas de Gauss-Hermite e Gauss-Laguerre, referentes, respectivamente, às integrais

$$\int_{-\infty}^{+\infty} f(x)e^{-x^2}\,dx \qquad \text{e} \qquad \int_{0}^{+\infty} f(x)e^{-x}\,dx$$

estão tabelas para diversos valores de n (n é o número de nós).

No nosso caso estamos interessados em integrais com função peso $w(x) = e^{-x^2}$ no domínio $(0,\infty)$ (vide 4.1), para os quais não temos valores tabelados. O procedimento para gerar os nós e os pesos para este caso, visto na seção anterior, envolve cálculos com sistemas extremamente mal condicionados. Isto impediu-nos de obter um número de nós suficientemente bom que proporcionasse um grau de precisão da mesma ordem que o obtido com outros métodos (por exemplo, com o método de Romberg do capítulo 4).

Certamente o cômputo desses nós e pesos (vide apêndice B) pode ser melhorado. E consequentemente um número maior de nós ($n > 16$) e maior precisão na aproximação das integrais pode ser obtido. A tabela abaixo mostra o comportamento das aproximações da integral

$$I = \int_{0}^{+\infty} (1 - erf(3x-2))e^{-x-x^2}\,dx$$

para n crescente, até o ponto em que conseguimos obter os nós (n=16):

n	I
2	0.892830244
4	0.820232479
8	0.824555848
12	0.824184939
14	0.824149912
15	0.824155619
16	0.824156806

A tabela acima sugere que n=20, por exemplo,já geraria um resultado satisfatório.

Em vista das dificuldades surgidas, o método que usamos para computar integrais do tipo 5.1 consiste em combinar o método de Romberg (da seção 4.1) com os dados do método de Gauss para n=16, com função peso $w(x) = e^{-x^2}$ em $(0,+\infty)$. Usamos Gauss para avaliar as integrais iteradas do tipo 5.2 e Romberg para avaliar 5.3. Por isso, quando falarmos no método Romberg+Gauss é sobre esta combinação de métodos que estamos falando.

Abaixo estão alguns exemplos de aproximações obtidas por alguns métodos de integração. Observe que apesar do insatisfatório número de nós obtido pelo processo da seção anterior (n=16), o resultado ainda é melhor do que o obtido com quadratura de Gauss-Hermite e Gauss-Laguerre.

Nestes exemplos também testamos o comportamento do método de Romberg em um intervalo com partição homogênea [a,b]. Quando falarmos de Romberg em $(0,\infty)$ significa que usamos o método da seção 4.2 (com partição representada pela figura 4.1). E quando falarmos de Romberg em (a,b) para calcular a integral $\int_0^{+\infty} f(x)e^{-x^2}\,dx$ significa que pegamos um certo valor de b, $b > a$, e aplicamos o método de Romberg usual em (a,b) para aproximar o valor da integral.

1.

$$\int_0^{+\infty} (1 - erf(3x-2))e^{-x-x^2}\,dx$$

Método	Valor da Aproximação
Romberg em (a,b)	0.8241563130
Romberg em (0,+∞)	0.8241563128
Gauss (n=16) $w(x) = e^{-x^2}$ em (0,+∞)	0.8241568063
Gauss-Hermite (n=20)	0.8018970972
Gauss-Laguerre (n=24)	0.8189798236
Maple	0.8241563129

2.

$$\int_0^{+\infty}\int_0^{+\infty} (1 - erf(3y-x-2))\exp(-x-y-x^2-y^2)\,dx\,dy$$

Método	Aproximação
Gauss + Romberg	0.4922038787
Maple	0.4922038788

3.

$$\int_0^{+\infty}\int_0^{+\infty} (1 - erf(y+x+2))\exp(-x-y-x^2-y^2)\,dx\,dy$$

Método	Aproximação
Gauss + Romberg	0.0001242206589
Maple	0.0001242206580

4.

$$\int_{-\infty}^{0}\int_{0}^{+\infty}(1 - erf(y + x + 2))\exp(-x - y - x^2 - y^2)\,dx\,dy$$

Método	Aproximação
Gauss + Romberg	0.08375908383
Maple	0.08375908378

Comparando as várias regras de integração numérica, como a de Romberg utilizada no capítulo 4, vemos que para um mesmo esforço computacional, a quadratura Gaussiana gera melhores resultados. Como não podemos prever exatamente qual o número de nós usar para obter uma estimativa da integral dentro de uma precisão desejada, a quadratura Gaussiana tem uma certa desvantagem com relação aos métodos baseados em extrapolação, visto que os nós e os valores da função integrando nestes nós na n-ésima aproximação não poderão ser aproveitados numa n+1-ésima aproximação.

5.3 Efeitos do uso de diferentes métodos de integração sobre fronteiras eficientes

Até agora analisamos a questão de aproximação numérica de integrais do tipo 5.3 sem considerarmos qual a influência destes resultados sobre o resultado final do problema de otimização, isto é, sobre portfólios eficientes.

O que se observou pelos exemplos numéricos da seção anterior foi um resultado melhor, em termos de precisão, para o método de Romberg, mas a um custo computacional maior. Ou seja, ele aproxima o valor da integral usando um número de nós muito maior do que o número usado na quadratura Gaussiana (cerca de 2^n nós (n-ésima aproximação), contra os 16 em Gauss).

Mas alguns testes revelaram que a influência do uso destes dois métodos de aproximação de integrais para gerar matrizes de covariância (no caso de portfólios com opções) é pequena e, portanto, a solução do problema de otimização pela abordagem MV (média-variância) não é afetada neste caso.

Sejam INEC16 e INEC17 opções de compra tendo como ações-objeto INEP4 e OTC19 e OTC24 também opções de compra, mas sobre TEL4.

Abaixo seguem as covariâncias entre retornos das opções acima usando-se o método de Romberg e também os nós e pesos da Quadratura Gaussiana (n=16).

Cov-Gauss	$OTC19$	$OTC24$
$INEC16$	1.2406466672	1.3343696501
$INEC16$	1.0490361321	1.1248742353

Cov-Romberg	$OTC19$	$OTC24$
$INEC16$	1.2406466541	1.3343696566
$INEC17$	1.0490361115	1.1248742573

Os valores das covariâncias das tabelas acima são muito próximos, não afetando o resultado em termos de portfólios eficientes. Mais testes são necessários a fim de verificar se este mesmo comportamento vale para o caso de aproximarmos integrais duplas pelo método de gauss e obtermos poucas mudancas em termos de assimetria e co-assimetria (para o caso de otimização de portfólios pelo critério MVA media-variância-assimetria).

Capítulo 6

Exemplos de Portfólios Eficientes

Vamos agora ver alguns exemplos de portfólios eficientes usando os critérios de escolha vistos nos capítulos 2 e 3, baseados em média-variância (MV) e média-variância-assimetria (MVA).

6.1 Primeiro exemplo

Suponha que um investidor queira saber qual a melhor maneira de combinar num portfólio os oito ativos abaixo:

- BB4, PET4, TEL4, WHM3, VAL4
- Opções de compra européias OTC16 e OTC17[1]
- Opções de venda européias OTP19, OTP24

onde TEL4-PN é a ação-objeto das opções.

Aqui a única restrição sobre o montante investido em cada ativo/derivativo é a restrição básica das frações w_i do montante total W a serem investidas em cada ativo, dada por $\sum_i w_i = 1$. Isto significa que não colocaremos restrições de aquisição em nenhum dos ativos e derivativos disponíveis.

Portanto, nossos problemas de otimização de carteiras mistas pelos critérios MV e MVA serão dados por

[1]Estas siglas para nomear opções são de caráter ilustrativo, não sendo necessariamente as mesmas usadas pela BOVESPA.

$$(PMV)\quad \begin{cases} \max & \eta \sum_i w_i \bar{z}_i - \sum_i \sum_j w_i w_j \sigma_{ij} \\ s.a & \sum_i w_i = 1 \\ & w_i \geq 0 \qquad (i = 1..n) \end{cases}$$

$$(PMVA)\quad \begin{cases} \max & \eta \sum_i w_i \bar{z}_i - \sum_i \sum_j w_i w_j \sigma_{ij} + \alpha \sum_i \sum_j \sum_k w_i w_j w_k m_{ijk} \\ s.a & \sum_i w_i = 1 \\ & w_i \geq 0 \qquad (i = 1..n) \end{cases}$$

onde n é o número de elementos (ativos + derivativos) do portfólio.

Vamos considerar 3 exemplos de carteiras (C1, C2 e C3), formadas por:

1. C1 = { BB4, PET4, TEL4, VAL4, OTC16, OTC17, OTP19, OTP24 }
2. C2 = { BB4, PET4, VAL4, WHM3, OTC16, OTC17, OTP19, OTP24 }
3. C3 = { BB4, PET4, TEL4, VAL4 }

Para estes exemplos, os retornos logarítmicos das ações foram feitos baseados numa amostra retroativa de 90 dias, com data final para 03/06/96 e extrapolados para 55 dias (para o dia 19/08/96, data de expiração das opções).

Os retornos logarítmicos das ações e os respectivos desvios padrões estão na tabela 6.1. Os coeficientes de correlação entre estes retornos, obtidos pelo *Critical Point*, estão na tabela 6.2.

Os dados referentes às opções, prêmios e preços de exercício, estão na tabela 6.3.

O preço dos lotes das ações nas datas de início e de expiração das opções estão na tabela 6.4.

Observe que o número de ações por lote não influencia os cálculos da taxa de retorno para os ativos e derivativos (dadas pelas expressões 2.9 e 2.13) por estas serem medidas percentuais (adimensionais).

Para o exemplo 1 (conjunto C1), será feita a análise de otimização pelos dois critérios (MV e MVA). Para os outros dois portfólios, será feita a análise usando somente o critério média-variância.

Todas as expressões de retorno, variância-covariância, assimetria e co-assimetria dos portfólios foram computadas usando os métodos vistos nos capítulos 4 e 5 (cujas rotinas implementadas estão listadas nos apêndices A e B).

6.1.1 Otimização de Portfólios pelo Critério Média-Variância

As taxas de retorno esperadas para os componentes das carteiras dos exemplos (taxas estas dadas pelas expressões 2.9 e 2.13, capítulo 2) e os respectivos desvios padrões estão na tabela 6.5.

Os coeficientes de correlação entre estas taxas estão na tabela 6.6 e 6.7.

As fronteiras eficientes obtidas pela resolução do sistema (PMV) estão no final deste capítulo.

Observe que a inclusão de opções no portfólio expandiu nosso horizonte de possibilidades de investimento, comparando as fronteiras de C1, C2 e C3. Com opções, temos uma grande gama de portfólios eficientes para diferentes valores de variância e retorno esperado num domínio bem maior do que o obtido com carteiras formados apenas por ativos básicos.

Os portfólios destacados por bolinhas na figura da fronteira eficiente, para o caso do exemplo 1 (conj. C1), estão relacionados na tabela 6.8. Nesta tabela, a letra "e" significa retorno esperado do portfólio e a letra "v" significa variância.

As carteiras C1 e C2 diferem somente em um dos ativos fundamentais, sendo que C1 possui o ativo TEL4 no lugar de WHM3, presente em C2. As taxas de retorno destes dois ativos e suas volatilidades não diferem muito (com relação ao retorno, aproximadamente 11% para WHM3 e 12% para TEL4). Por isso os resultados das tabelas 6.8 e 6.9 são bem próximos. Vale destacar que a presença da TEL4 em C1 não substitui a presença das opções *plain vanilla* sobre ela, quando uma taxa de retorno maior é desejada no portfólio. Observe ainda a presença das opções OTP19 e OTP24 nos portfólios A e B dessas tabelas, apesar destes derivativos oferecerem uma expectativa de retorno negativa, como podemos ver pela tabela 6.5. Interpretamos a presença destas opções nestes portfólios eficientes como uma tentativa de *hedge*, em vista da alta volatilidade do retorno das opções de compra e venda.

Observe que um portfólio eficiente pode não conter todos os ativos disponíveis.

6.1.2 Otimização de Portfólios pelo Critério Média-Variância-Assimetria

Vamos agora relacionar alguns portfólios eficientes pela resolução do problema 3.3 (capítulo 3).

As co-assimetrias entre as taxas de retorno dos elementos do portfólio 2 (dado por C2) estão descritas na tabela 6.12. Observe que os valores das co-assimetrias que envolvem opções são bem mais significativas do que as que envolvem somente as ações. Isto devido à maior assimetria do retorno de opções, como já foi mencionado no capítulo 3.

A tabela 6.12 sugere-nos que a utilização do critério MVA para seleção de carteiras eficientes formadas somente por fundamentais (no caso do exemplo, ações da BOVESPA) gerará resultados muito próximos aos obtidos usando o critério MV.Isto porque a parcela de contribuição dos termos de assimetria das ações não é tão significativa para a função utilidade esperada.

Agora vamos tomar os valores de η dos portfólios eficientes da tabela 6.9, relacionados pelas letras A, B, C, D, E e F (no gráfico da fronteira eficiente, estas letras correspondem aos portfólios destacados por "bolinhas" da esquerda para a direita, em movimento ascendente). Para cada um destes valores de η vamos resolver o problema 3.3 para alguns valores de α (coeficiente de ponderação do termo da assimetria na função utilidade esperada do portfólio). Os portfólios eficientes assim obtidos estão relacionados na tabela 6.11.

Os gráficos 6.1, 6.2 e 6.3 descrevem o comportamento do *skewness* dos portfólios versus retorno esperado, desvio padrão $\times$ retorno esperado e desvio padrão $\times$ retorno esperado $\times$ assimetria para os valores $\eta = 1$ e $0 < \alpha < 1$, para o exemplo C2.

Comparando os resultados obtidos pelos dois critérios, dados nas tabelas 6.9 e 6.11, e considerando que o preço do lote da TEL4 teve uma alta considerável no período de vigência do contrato (03/06/96 à 19/08/96), as melhores carteiras foram obtidas levando-se em conta a assimetria dos ativos. Mas isto ocorreu devido a maior assimetria do retorno das opções, isto é, devido a estarmos otimizando portfólios mistos de ações e opções.

Por exemplo, para $\eta = 0.105$ e $\alpha = 0.100$ o modelo MV nos gerou um portfólio eficiente contendo opções de venda sobre TEL4, o que não ocorreu com o modelo MVA. Como era previsto, à medida que aumentamos o valor do parâmetro α, para um valor fixo de η, o modelo MVA fornece-nos portfólios eficientes formados por percentuais crescentes em opções de compra. Esta característica deve-se a não existência de restrições de aquisição de derivativos e pela alta taxa de retorno e assimetria que elas nos oferecem.

Queremos ainda salientar que o estudo de estruturação ótima de carteiras mistas, formadas por ativos fundamentais e derivativos com retornos assimétricos, foi feito aqui a nível exploratório. A utilidade do terceiro momento (assimetria) no caso destes portfólios mistos é inicialmente defendida, já que podemos estender o modelo MV incluindo na função utilidade esperada a assimetria do retorno de portfólios.

Pelas observações que fizemos no capítulo 5 com relação a precisão no cálculo numérico do tensor de terceira ordem, é necessário ainda um aperfeiçoamento das técnicas de integração utilizadas. Com isto teremos condições de aprimorar o modelo MVA para inclusão de outras opções (como as exóticas), cujas expressões de rentabilidade são mais complicadas do que as expressões das opções européias. E também ampliar as técnicas de visualização e análise de portfólios para estes casos, ressaltando assim a importância do *skewness* de portfólios formados por opções.

6.2 Segundo exemplo

Queremos aqui obter carteiras eficientes considerando os seguintes investimentos:

- CGAS4, INEP4, MAKR3, PETR4, TELB4, TLSP4
- Opção de compra TELBH10
- Opção de venda TELBT10

Foram analisadas 81 cotações das ações acima, retroativas ao dia 30/06/98. Como as opções vencem em agosto, extrapolamos os retornos logarítmicos para 36 dias de pregão após esta data. Também calculamos as taxas de retorno simples esperadas (tabela 6.12) para as ações e opções, assim como o desvio padrão de cada taxa.

A tabela 6.13 contém a matriz de correlação entre as taxas de retorno para todos os investimentos considerados acima.

Para as opções, temos

- preços de exercício: R$ 140,00 para TELBT10 e R$ 160,00 para TELBH10
- prêmios: R$ 17,00 para TELBT10 e R$ 0,50 para TELBH10

Resolvemos o problema (PMV) para um portfólio formado por todos os ativos e opções acima e também para um carteira formada só pelas ações (sem incluir as opções TELBH10 e TELBT10).

Observando as fronteiras eficientes para os dois casos (veja adiante), fixado um nível de risco, ou desvio padrão, a fronteira eficiente que considera as opções fornece-nos um portfólio com retorno esperado maior do que o obtido pela fronteira que considera só as ações. Isso se dá pelo fato de as opções servirem como meio de proteção a maiores oscilações no preço das ações (no caso, ações da TELEBRÁS).

ativo	Retorno logarítmico	desvio-padrão do retorno
BB4	-1.782941E-01	3.728419E-01
PET4	1.517448E-01	1.478579E-01
TEL4	1.064671E-01	1.368356E-01
VAL4	3.902896E-03	1.504271E-01
WHM3	9.710318E-02	1.481048E-01

Tabela 6.1: Retorno logarítimico e respectivo desvio padrão das ações.

corr	BB4	PET4	TEL4	VAL4	WHM3
BB4	1.00000E+00	1.88931E-01	1.63584E-01	9.22659E-02	1.78502E-01
PET4	1.88931E-01	1.00000E+00	5.17330E-01	6.31955E-01	2.22423E-01
TEL4	1.63584E-01	5.17330E-01	1.00000E+00	5.86087E-01	1.99405E-01
VAL4	9.22659E-02	6.31955E-01	5.86087E-01	1.00000E+00	1.63505E-01
WHM3	1.78502E-01	2.22423E-01	1.99405E-01	1.63505E-01	1.00000E+00

Tabela 6.2: Coeficientes de correlação entre os retornos logarítmicos das ações

opção	preço-exercício (R$)	prêmio (R$)
OTC16	56.00	10.90
OTC17	60.00	6.70
OTP19	64.00	6.70
OTP24	72.00	10.90

Tabela 6.3: Dados das opções

ação	03/06/96 R$	19/08/96 R$	lote
BB4	903,00	1.020,00	100000
PET4	1.215,00	1.254,80	10000
VAL4	2.071,42	1.848,42	100
WHM3	1.266,29	1.361,72	100000
TEL4	63,20	75,60	1000

Tabela 6.4: Cotação das ações

ativo	taxa de retorno	desvio-padrão
BB4	-0.1030799567	0.3463747222
PET4	0.1766551319	0.1749329755
TEL4	0.1228039581	0.1543615526
VAL4	0.0153334211	0.1536017754
WHM3	0.1141265068	0.1659164932
OTC16	0.3859981177	0.8704101868
OTC17	0.7067222457	1.3475513486
OTP19	-0.8211298158	0.4213573091
OTP24	-0.5926258435	0.5133684347

Tabela 6.5: Taxa de retorno esperada e desvio-padrão das ações e opções.

corr	BB4	PET4	TEL4	VAL4	WHM3
BB4	1.0000000000	0.1823564539	0.1578521444	0.0888047461	0.1722390250
PET4	0.1823564539	1.0000000000	0.5147879950	0.6293648575	0.2205331658
TEL4	0.1578521444	0.5147879950	1.0000000000	0.5835617107	0.1977843120
VAL4	0.0888047461	0.6293648575	0.5835617107	1.0000000000	0.1619850356
WHM3	0.1722390250	0.2205331658	0.1977843120	0.1619850356	1.0000000000
OTC16	0.1568451573	0.5118030420	0.9963686397	0.5804297367	0.1962824088
OTC17	0.1547705883	0.5054410381	0.9869885907	0.5735589167	0.1933646829
OTP19	-0.1073096454	-0.3461757989	-0.6444407927	-0.3892180077	-0.1374425047
OTP24	-0.1383966747	-0.4485565937	-0.8502748970	-0.5061052322	-0.1755751299

Tabela 6.6: Coeficientes de correlação entre as taxas de retorno das ações e opções

corr	OTC16	OTC17	OTP19	OTP24
BB4	0.1568451573	0.1547705883	-0.1073096454	-0.1383966747
PET4	0.5118030420	0.5054410381	-0.3461757989	-0.4485565937
TEL4	0.9963686397	0.9869885907	-0.6444407927	-0.8502748970
VAL4	0.5804297367	0.5735589167	-0.3892180077	-0.5061052322
WHM3	0.1962824088	0.1933646829	-0.1374425047	-0.1755751299
OTC16	1.0000000000	0.9953744913	-0.5969581835	-0.8280633925
OTC17	0.9953744913	1.0000000000	-0.5247821186	-0.7875704685
OTP19	-0.5969581835	-0.5247821186	1.0000000000	0.8654714267
OTP24	-0.8280633925	-0.7875704685	0.8654714267	1.0000000000

Tabela 6.7: Continuação da tabela 6.6.

Portfólio	η	e	v	composição
A	0.023	0.014	0.005	0.184 PET4
				0.651 TEL4
				0.165 OTP24
B	0.096	0.151	0.002	0.564 PET4
				0.433 TEL4
				0.003 OTP24
C	0.467	0.229	0.064	0.901 PET4
				0.099 OTC17
D	0.970	0.317	0.190	0.735 PET4
				0.265 OTC17
E	1.380	0.388	0.357	0.601 PET4
				0.399 OTC17
F	1.820	0.465	0.603	0.456 PET4
				0.544 OTC17

Tabela 6.8: Portfólios eficientes pelo critério média-variância para o exemplo 1 (C1)

.

Portfólio	η	e	v	composição
A	0.043	0.106	0.014	0.477 PET4
				0.071 VAL4
				0.409 WHM3
				0.043 OTP24
B	0.206	0.171	0.025	0.704 PET4
				0.274 WHM3
				0.022 OTC17
C	0.467	0.229	0.064	0.899 PET4
				0.002 WHM3
				0.099 OTC17
D	0.970	0.317	0.190	0.682 PET4
				0.318 OTC17
E	1.380	0.388	0.357	0.558 PET4
				0.442 OTC17
F	1.820	0.466	0.606	0.454 PET4
				0.546 OTC17

Tabela 6.9: Portfólios eficientes pelo critério média-variância para C2.

Portfólio	η	e	v	composição
A	0.042	0.147	0.020	0.457 PET4
				0.542 TEL4
				0.001 BB4
B	0.120	0.156	0.0217	0.616 PET4
				0.384 TEL4
C	0.221	0.167	0.025	0.821 PET4
				0.179 TEL4
D	0.304	0.176	0.030	0.988 PET4
				0.012 TEL4

Tabela 6.10: Portfólios eficientes pelo critério média-variância para C3.

η	α	composição do portfólio	utilidade esperada
0.0230	1.000	0.3636 PET4	-0.0056
		0.4999 TEL4	
		0.1061 VAL4	
	2.800	0.4992 PET4	-0.0037
		0.5008 TEL4	
	4.000	1.0000 OTC17	6.6346
0.0965	1.000	0.6312 PET4	0.0055
		0.3688 TEL4	
	1.400	0.6721 PET4	0.0060
		0.3279 TEL4	
	1.800	0.7314 PET4	0.0065
		0.2686 TEL4	
	2.600	1.0000 OTC17	4.0524
0.4670	0.400	0.8494 PET4	0.0751
		0.1506 OTC17	
	0.800	1.0000 OTC17	0.9274
0.9700	0.100	0.6969 PET4	0.2189
		0.3031 OTC17	
	0.200	0.6244 PET4	0.2288
		0.3756 OTC17	
	0.300	1.0000 OTC17	0.3420
1.3800	0.100	0.5122 PET4	0.3768
		0.4878 OTC17	
	0.200	1.0000 OTC17	0.4436
1.8200	0.100	0.2726 PET4	0.5951
		0.7274 OTC17	
	0.200	1.0000 OTC17	0.7546

Tabela 6.11: Portfólios eficientes pelo critério média-variância-assimetria para o conjunto C1.

ativo	taxa de retorno	desvio-padrão
CGAS4	-0.03983	0.14406
INEP4	0.05236	0.17938
MAKR3	-0.04346	0.12435
PETR4	0.036153	0.11118
TELB4	-0.02251	0.12497
TLSP4	0.07066	0.13210
TELBH10	-0.80473	2.3380
TELBT10	0.15042	0.78972

Tabela 6.12: Taxa de retorno esperada e desvio-padrão para os investimentos do segundo exemplo

corr	CGAS4	INEP4	MAKR3	PETR4	TELB4
CGAS4	1.00000	0.327278	0.0742597	0.293132	0.34164
INEP4	0.327278	1.000000	-0.0129062	0.601057	0.61465
MAKR3	0.0742597	-0.0129062	1.000000	-0.140435	-0.183826
PETR4	0.293132	0.601057	-0.140435	1.000000	0.784221
TELB4	0.34164	0.61465	-0.183826	0.784221	1.000000
TLSP4	0.374945	0.554344	-0.23277	0.618559	0.66399
TELBH10	0.0877064	0.168615	-0.0430102	0.211345	0.20723
TELBT10	-0.334617	-0.597389	0.181918	-0.763793	-0.741684

Tabela 6.13: Correlação entre as taxas de retorno das ações e opções do segundo exemplo

corr	TLSP4	TELBH10	TELBT10
CGAS4	0.374945	0.0877064	-0.334617
INEP4	0.554344	0.168615	-0.597389
MAKR3	-0.23277	-0.0430102	0.181918
PETR4	0.618559	0.211345	-0.763793
TELB4	0.66399	0.20723	-0.741684
TLSP4	1.00000	0.177058	-0.647495
TELBH10	0.177058	1.000000	-0.121658
TELBT10	-0.647495	-0.121658	1.000000

Tabela 6.14: Continuação da tabela 6.13.

	BB4	PET4	TEL4	VAL4	OTC16	OTC17	OTP19	OTP24
BB4.BB4	0.0505385252	0.0030752561	0.0023308007	0.0012892656	0.0132785857	0.0207538627	-0.3311915830	-0.3386035243
BB4.PET4	0.0030752561	0.0007138585	0.0003400919	0.0002965267	0.0022198392	0.0040654380	0.0013325650	0.0008032973
BB4.TEL4	0.0023308007	0.0003400919	0.0004391309	0.0002133898	0.0029733847	0.0056692507	0.0024797822	0.0018084445
BB4.VAL4	0.0012892656	0.0002965267	0.0002133898	0.0002450382	0.0015037447	0.0029612989	0.0015394342	0.0011557139
BB4.OTC16	0.0132785857	0.0022198392	0.0029733847	0.0015037447	0.0188913176	0.0347165842	0.0114986329	0.0081968353
BB4.OTC17	0.0207538627	0.0040654380	0.0056692507	0.0029612989	0.0347165842	0.0609494097	0.0134423026	0.0087850325
BB4.OTP19	-0.3311915830	0.0013325650	0.0024797822	0.0015394342	0.0114986329	0.0134423026	-0.0157566302	-0.0138419035
BB4.OTP24	-0.3386035243	0.0008032973	0.0018084445	0.0011557139	0.0081968353	0.0087850325	-0.0138419035	-0.0127156809
PET4.BB4	0.0030752561	0.0007138585	0.0003400919	0.0002965267	0.0022198392	0.0040654380	0.0013325650	0.0008032973
PET4.PET4	0.0007138585	0.0024051754	0.0008989431	0.0011675090	0.0055552581	0.0096056800	-0.4871690301	-0.5089445838
PET4.TEL4	0.0003400919	0.0008989431	0.0007573136	0.0006039625	0.0050792918	0.0096074894	0.0040030421	0.0028953468
PET4.VAL4	0.0002965267	0.0011675090	0.0006039625	0.0010345408	0.0038907043	0.0070424573	0.0020847373	0.0012027831
PET4.OTC16	0.0022198392	0.0055552581	0.0050792918	0.0038907043	0.0321063469	0.0586652091	0.0185536755	0.0130896705
PET4.OTC17	0.0040654380	0.0096056800	0.0096074894	0.0070424573	0.0586652091	0.1026386252	0.0216346313	0.0139114253
PET4.OTP19	0.0013325650	-0.4871690301	0.0040030421	0.0020847373	0.0185536755	0.0216346313	-0.0255747717	-0.0224587425
PET4.OTP24	0.0008032973	-0.5089445838	0.0028953468	0.0012027831	0.0130896705	0.0139114253	-0.0224587425	-0.0206404453
TEL4.BB4	0.0023308007	0.0003400919	0.0004391309	0.0002133898	0.0029733847	0.0056692507	0.0024797822	0.0018084445
TEL4.PET4	0.0003400919	0.0008989431	0.0007573136	0.0006039625	0.0050792918	0.0096074894	0.0040030421	0.0028953468
TEL4.TEL4	0.0004391309	0.0007573136	0.0015265161	0.0007793724	0.0099321075	0.0182933718	0.0063348778	0.0044177621
TEL4.VAL4	0.0002133898	0.0006039625	0.0007793724	0.0008174455	0.0051925756	0.0097663174	0.0039242142	0.0028199591
TEL4.OTC16	0.0029733847	0.0050792918	0.0099321075	0.0051925756	0.0617328400	0.1106294470	0.0293001546	0.0197347840
TEL4.OTC17	0.0056692507	0.0096074894	0.0182933718	0.0097663174	0.1106294470	0.1912471487	0.0337895302	0.0201606941
TEL4.OTP19	0.0024797822	0.0040030421	0.0063348778	0.0039242142	0.0293001546	0.0337895302	-0.0414146218	-0.0363098170
TEL4.OTP24	0.0018084445	0.0028953468	0.0044177621	0.0028199591	0.0197347840	0.0201606941	-0.0363098170	-0.0334200513
VAL4.BB4	0.0012892656	0.0002965267	0.0002133898	0.0002450382	0.0015037447	0.0029612989	0.0015394342	0.0011557139
VAL4.PET4	0.0002965267	0.0011675090	0.0006039625	0.0010345408	0.0038907043	0.0070424573	0.0020847373	0.0012027831
VAL4.TEL4	0.0002133898	0.0006039625	0.0007793724	0.0008174455	0.0051925756	0.0097663174	0.0039242142	0.0028199591
VAL4.VAL4	0.0002450382	0.0010345408	0.0008174455	0.0016572883	0.0050854663	0.0088699060	-0.3547914660	-0.3784112515
VAL4.OTC16	0.0015037447	0.0038907043	0.0051925756	0.0050854663	0.0327037764	0.0595133146	0.0181822160	0.0127229373
VAL4.OTC17	0.0029612989	0.0070424573	0.0097663174	0.0088699060	0.0595133146	0.1038630354	0.0211585210	0.0134302029
VAL4.OTP19	0.0015394342	0.0020847373	0.0039242142	-0.3547914660	0.0181822160	0.0211585210	-0.0251774405	-0.0221034374
VAL4.OTP24	0.0011557139	0.0012027831	0.0028199591	-0.3784112515	0.0127229373	0.0134302029	-0.0221034374	-0.0203203229
OTC16.BB4	0.0132785857	0.0022198392	0.0029733847	0.0015037447	0.0188913176	0.0347165842	0.0114986329	0.0081968353
OTC16.PET4	0.0022198392	0.0055552581	0.0050792918	0.0038907043	0.0321063469	0.0586652091	0.0185536755	0.0130896705
OTC16.TEL4	0.0029733847	0.0050792918	0.0099321075	0.0051925756	0.0617328400	0.1106294470	0.0293001546	0.0197347840
OTC16.VAL4	0.0015037447	0.0038907043	0.0051925756	0.0050854663	0.0327037764	0.0595133146	0.0181822160	0.0127229373
OTC16.OTC16	0.0188913176	0.0321063469	0.0617328400	0.0327037764	0.3735381203	0.6575132436	0.1394270743	0.0907463435
OTC16.OTC17	0.0347165842	0.0586652091	0.1106294470	0.0595133146	0.6575132436	1.1235934363	0.1587891402	0.0889325628
OTC16.OTP19	0.0114986329	0.0185536755	0.0293001546	0.0181822160	0.1394270743	0.1587891402	-0.1964613828	-0.1742178881
OTC16.OTP24	0.0081968353	0.0130896705	0.0197347840	0.0127229373	0.0907463435	0.0889325628	-0.1742178881	-0.1644250366
OTC17.BB4	0.0207538627	0.0040654380	0.0056692507	0.0029612989	0.0347165842	0.0609494097	0.0134423026	0.0087850325
OTC17.PET4	0.0040654380	0.0096056800	0.0096074894	0.0070424573	0.0586652091	0.1026386252	0.0216346313	0.0139114253
OTC17.TEL4	0.0056692507	0.0096074894	0.0182933718	0.0097663174	0.1106294470	0.1912471487	0.0337895302	0.0201606941
OTC17.VAL4	0.0029612989	0.0070424573	0.0097663174	0.0088699060	0.0595133146	0.1038630354	0.0211585210	0.0134302029
OTC17.OTC16	0.0347165842	0.0586652091	0.1106294470	0.0595133146	0.6575132436	1.1235934363	0.1587891402	0.0889325628
OTC17.OTC17	0.0609494097	0.1026386252	0.1912471487	0.1038630354	1.1235934363	1.8815840333	0.1735543668	0.0738254268
OTC17.OTP19	0.0134423026	0.0216346313	0.0337895302	0.0211585210	0.1587891402	0.1735543668	-0.2489452686	-0.2183983394
OTC17.OTP24	0.0087850325	0.0139114253	0.0201606941	0.0134302029	0.0889325628	0.0738254268	-0.2183983394	-0.2098481702
OTP19.BB4	-0.3311915830	0.0013325650	0.0024797822	0.0015394342	0.0114986329	0.0134423026	-0.0157566302	-0.0138419035
OTP19.PET4	0.0013325650	-0.4871690301	0.0040030421	0.0020847373	0.0185536755	0.0216346313	-0.0255747717	-0.0224587425
OTP19.TEL4	0.0024797822	0.0040030421	0.0063348778	0.0039242142	0.0293001546	0.0337895302	-0.0414146218	-0.0363098170
OTP19.VAL4	0.0015394342	0.0020847373	0.0039242142	-0.3547914660	0.0181822160	0.0211585210	-0.0251774405	-0.0221034374
OTP19.OTC16	0.0114986329	0.0185536755	0.0293001546	0.0181822160	0.1394270743	0.1587891402	-0.1964613828	-0.1742178881
OTP19.OTC17	0.0134423026	0.0216346313	0.0337895302	0.0211585210	0.1587891402	0.1735543668	-0.2489452686	-0.2183983394
OTP19.OTP19	-0.0157566302	-0.0255747717	-0.0414146218	-0.0251774405	-0.1964613828	-0.2489452686	0.2134013931	0.1947074869
OTP19.OTP24	-0.0138419035	-0.0224587425	-0.0363098170	-0.0221034374	-0.1742178881	-0.2183983394	0.1947074869	0.1748466011
OTP24.BB4	-0.3386035243	0.0008032973	0.0018084445	0.0011557139	0.0081968353	0.0087850325	-0.0138419035	-0.0127156809
OTP24.PET4	0.0008032973	-0.5089445838	0.0028953468	0.0012027831	0.0130896705	0.0139114253	-0.0224587425	-0.0206404453
OTP24.TEL4	0.0018084445	0.0028953468	0.0044177621	0.0028199591	0.0197347840	0.0201606941	-0.0363098170	-0.0334200513
OTP24.VAL4	0.0011557139	0.0012027831	0.0028199591	-0.3784112515	0.0127229373	0.0134302029	-0.0221034374	-0.0203203229
OTP24.OTC16	0.0081968353	0.0130896705	0.0197347840	0.0127229373	0.0907463435	0.0889325628	-0.1742178881	-0.1644250366
OTP24.OTC17	0.0087850325	0.0139114253	0.0201606941	0.0134302029	0.0889325628	0.0738254268	-0.2183983394	-0.2098481702
OTP24.OTP19	-0.0138419035	-0.0224587425	-0.0363098170	-0.0221034374	-0.1742178881	-0.2183983394	0.1947074869	0.1748466011
OTP24.OTP24	-0.0127156809	-0.0206404453	-0.0334200513	-0.0203203229	-0.1644250366	-0.2098481702	0.1748466011	0.1633187616

Tabela 6.15: Co-assimetrias entre os retornos dos elementos do portfólio C1.

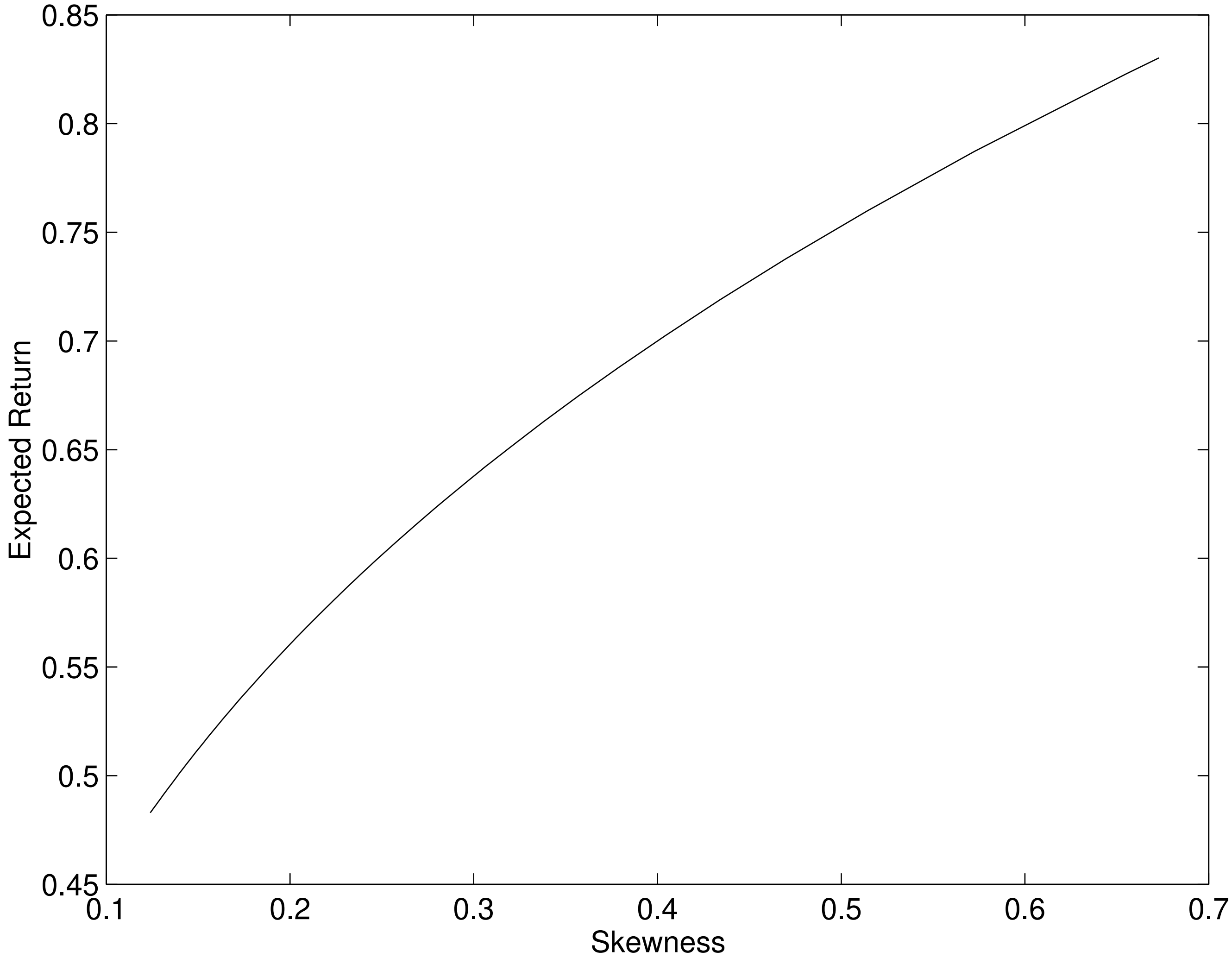

Figura 6.1: Assimetria $\times$ Retorno Esperado para $\eta = 1$ e $0 < \alpha < 1$

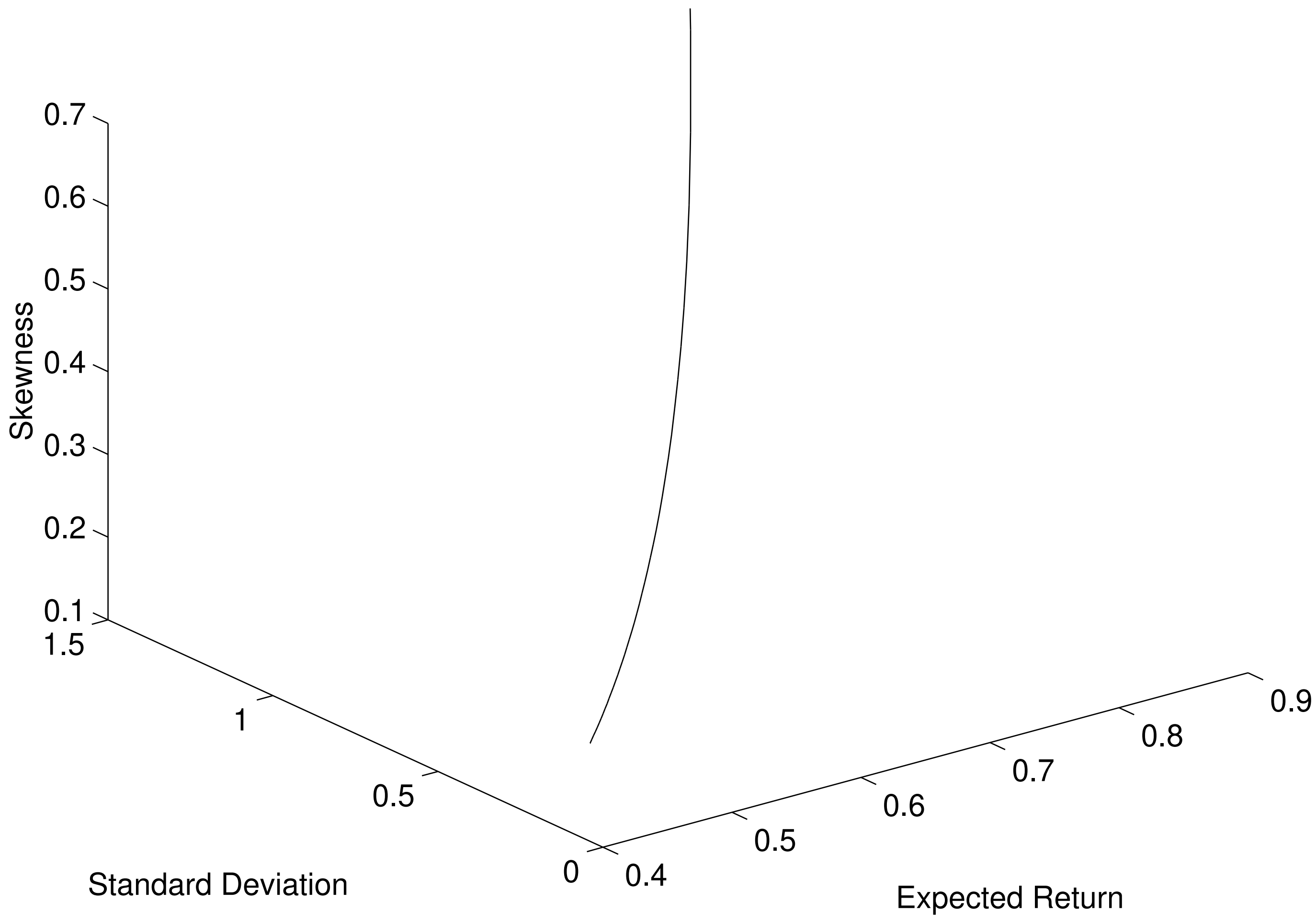

Figura 6.2: Retorno Esperado $\times$ Desvio Padrão $\times$ Assimetria para $\eta = 1$ e $0 < \alpha < 1$

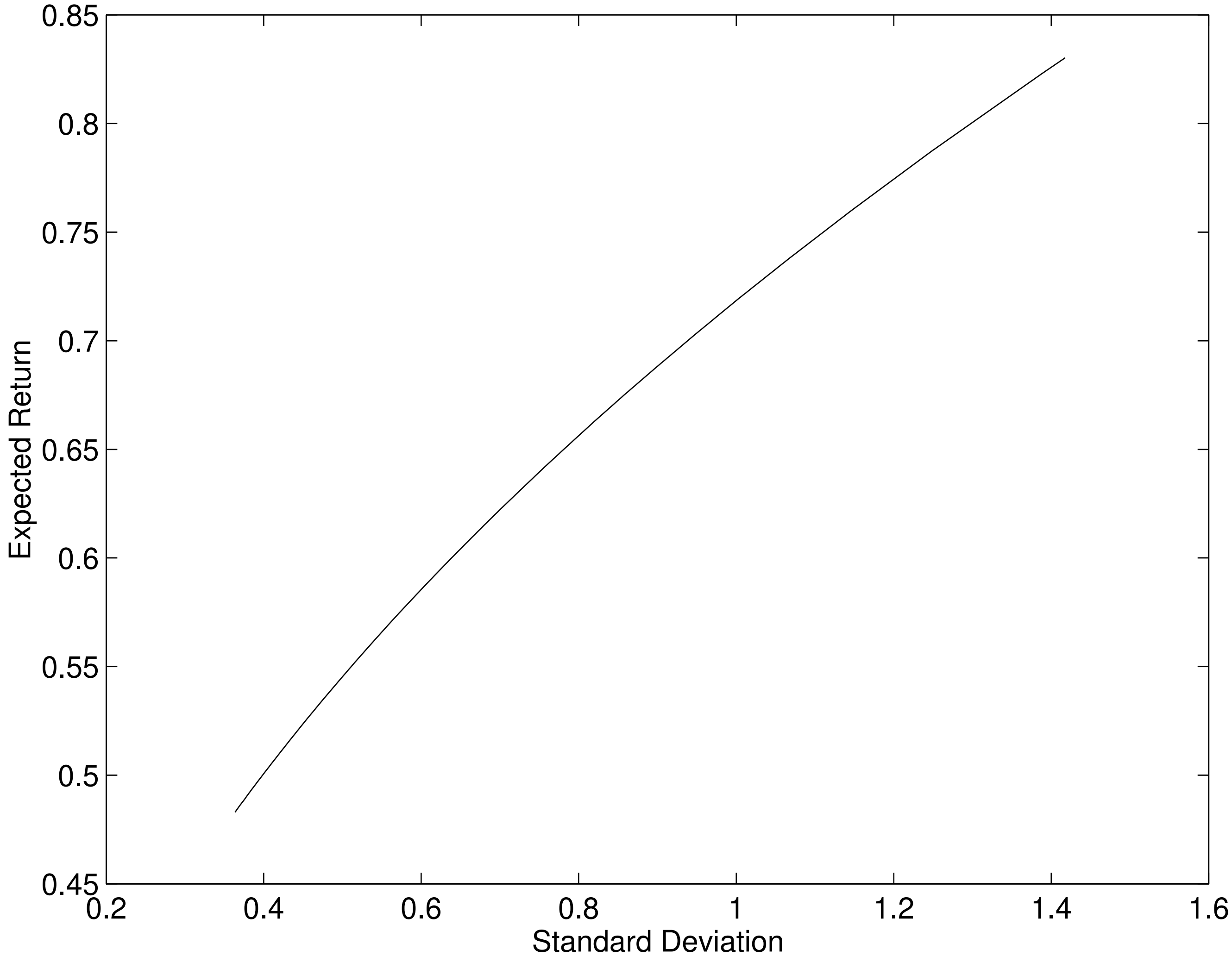

Figura 6.3: Desvio Padrão $\times$ Retorno Esperado para $\eta = 1$ e $0 < \alpha < 1$

Capítulo 7

Conclusão

Neste trabalho estudamos uma metodologia para cálculo de matrizes de covariância entre retornos de ações-objetos e opções européias. A abordagem, em termos de representação de momentos de segunda e terceira ordem do retorno de portfólios através de operadores integrais, permite uma avaliação de retornos esperados de opções baseada inteiramente no comportamento dos ativos-objeto, e não baseados em séries históricas de preços de opções.

O método usado no mercado para análise de opções baseia-se em aproximações lineares e fórmula de Black-Scholes (gregos), o que dificulta a análise de portfólios mistos de ações e opções uma vez que a sensibilidade da matriz de covariância frente a ações e opções sobre o mesmo ativo-objeto não é captada usando-se aproximações lineares. O cálculo exato destas matrizes permite que analisemos portfólios grandes sem correr o risco de obter correlação 1 entre opções e ações-objeto, e possibilitando assim o uso de técnicas de otimização para obtenção de fronteiras eficientes.

A viabilidade do uso desta metologia em termos práticos depende de métodos computacionalmente eficientes para a aproximação numérica das integrais em questão. Este assunto foi discutido nos capítulos 4 e 5.

Inserimos também o uso do terceiro momento central do retorno de portfólios para obtenção de carteiras eficientes. O modelo de terceira ordem é uma extensão natural do modelo média-variância já que estamos interessados em estudar portfólios formados por derivativos com retornos assimétricos, como é o caso das opções européias. O paralelo entre o modelo MV (média-variância) e MVA (média-variância-assimetria) mostrou-nos a necessidade do uso do terceiro momento quando trabalhamos com opções, e também mostrou que a aproximação de segunda ordem é bastante boa (comparada com a de terceira ordem) para portfólios formados apenas por ações-objeto.

O desenvolvimento de técnicas de visualização de superfícies eficientes geradas pelo problema de otimização paramétrico do modelo MVA são necessárias para melhor aproveitamento de informação e interpretação de dados.

Apêndice A

Implementações em linguagem C

A.1 Rotinas de Covariâncias

As expressões de covariância dadas a seguir estão descritas no capítulo 2, onde as variáveis e constantes nelas inseridas estão definidas.

```
double cov_riri(double medi, double stdi)
/***********************************************************************/
/* variância da taxa de retorno do ativo básico  i                     */
/* medi = retorno logarítimico esperado do ativo i                     */
/* stdi = desvio padrão do retorno log. do ativo i.                    */
/***********************************************************************/
{
        double resp;

        resp=exp(2.0*medi+stdi*stdi)*(exp(stdi*stdi)-1.0);
        return(resp);
}
```

```
double cov_rirj(double medi, double medj, double stdi, double stdj,
                double corij)
/***********************************************************************/
/* covariância entre a taxa de retorno dos ativos básicos i e j;       */
/* medi, medj = retorno logarítimico esperado dos ativos i e j;        */
/* stdi, stdj = desvio padrão do retorno log. dos ativos i e j;        */
/* corij = coef. de correlacao entre os retornos log. dos ativos i e j */
/***********************************************************************/
{
        double resp;
```

```
        resp=exp(medi+medj+0.5*(stdi*stdi+stdj*stdj))*
                (exp(corij*stdi*stdj)-1.0);
        return(resp);
}


double cov_rici(double medi, double stdi, double Si, double Kci, double Ci,
                double ms, double mc)
/**************************************************************************/
/* covariância entre a taxa de retorno do ativo básico i e da call        */
/* sobre o mesmo ativo i;                                                 */
/* medi = retorno logarítimico esperado do ativo i;                       */
/* stdi = desvio padrão do retorno log. do ativo i;                       */
/* Si = cotacao do ativo básico i na data de abertura do contrato t;      */
/* Kci = preco de exercicio da call na data de expiracao T;               */
/* Ci = premio da call na data t;                                         */
/* ms = esperanca da taxa de retorno do ativo fundamental;                */
/* mc = esperanca da taxa de retorno da call;                             */
/**************************************************************************/
{

        double R, E2x, Ex, E1, resp;

        R=(log(Kci/Si)-medi)/stdi;
        E2x=0.5*exp(2.0*medi+2.0*sqr(stdi))
               *(1.0-erff((R-2.0*stdi)/sqrt(2.0)));
        Ex=0.5*exp(medi+0.5*sqr(stdi))
              *(1.0-erff((R-stdi)/sqrt(2.0)));
        resp=(Si*E2x-Kci*Ex)/Ci-ms*mc;
        return(resp);
}



double cov_ricj(double medi, double medj, double stdi, double stdj,
                double corij, double Sj, double Kcj, double Cj, double ms,
                double mc)
/**************************************************************************/
/* covariância entre a taxa de retorno do ativo básico i e da call sobre o */
/* ativo básico j;                                                        */
/* medi, medj = retorno logarítimico esperado dos ativos i e j;           */
/* stdi, stdj = desvio padrão do retorno log. dos ativos i e j;           */
/* corij = coef. de correlacao entre os retornos log. dos ativos i e j;   */
/* Sj = cotacao do ativo básico j na data de abertura do contrato t;      */
/* Kcj = preco de exercicio da call na data de expiracao T;               */
/* Cj = premio da call na data t;                                         */
```

```
/* ms = esperanca da taxa de retorno do ativo i;                          */
/* mc = esperanca da taxa de retorno da call;                             */
/*************************************************************************/
{

        double R, Exy, Ex, Ey, E1, resp;

        R=(log(Kcj/Sj)-medj)/stdj;
        Exy=0.5*exp(medi+medj+0.5*(sqr(stdi)+sqr(stdj))+corij*stdi*stdj)
                  *(1.0-erff((R-stdj-corij*stdi)/sqrt(2.0)));
        Ex=0.5*exp(medi+0.5*sqr(stdi))
              *(1.0-erff((R-corij*stdi)/sqrt(2.0)));
        Ey=0.5*exp(medj+0.5*sqr(stdj))*(1.0-erff((R-stdj)/sqrt(2.0)));
        E1=0.5*(1.0-erff(R/sqrt(2.0)));
        Exi=exp(medi+0.5*stdi*stdi);
        resp=(Sj*Exy-Kcj*Ex-ms*(Sj*Ey-Kcj*E1))/Cj;
        resp+=ms*mc-mc*Exi;
        return(resp);

}


double cov_ripi(double medi, double stdi, double Si, double Kpi, double Pi,
                double ms, double mp)
/*************************************************************************/
/* covariância entre a taxa de retorno do ativo básico i e da put sobre o */
/* mesmo ativo i;                                                         */
/* medi = retorno logarítimico esperado do ativo i;                       */
/* stdi = desvio padrão do retorno log. do ativo i;                       */
/* Si = cotacao do ativo básico i na data de abertura do contrato t;      */
/* Kpi = preco de exercicio da put na data de expiracao T;                */
/* Pi = premio da put na data t.                                          */
/* ms = taxa de retorno esperada do ativo fundamental;                    */
/* mp = taxa de retorno esperada da put;                                  */
/*************************************************************************/
{

        double R, E2x, Ex, E1, resp;

        R=(log(Kpi/Si)-medi)/stdi;
        E2x=0.5*exp(2.0*medi+2.0*sqr(stdi))
             *(1.0+erff((R-2.0*stdi)/sqrt(2.0)));
        Ex=0.5*exp(medi+0.5*sqr(stdi))*(1.0+erff((R-stdi)/sqrt(2.0)));
        resp=(-Si*E2x+Kpi*Ex)/Pi-ms*mp;
        return(resp);
```

```
}


double cov_ripj(double medi, double medj, double stdi, double stdj,
                double corij, double Sj, double Kpj, double Pj,
                double ms, double mp)
/***********************************************************************/
/* covariância entre a taxa de retorno do ativo básico i e da put sobre o  */
/* ativo básico j;                                                         */
/* medi, medj = retorno logarítimico esperado dos ativos i e j;            */
/* stdi, stdj = desvio padrão do retorno log. dos ativos i e j;            */
/* corij = coef. de correlacao entre os retornos log. dos ativos i e j;    */
/* Sj = cotacao do ativo básico j na data de abertura do contrato t;       */
/* Kpj = preco de exercicio da put na data de expiracao T;                 */
/* Pj = premio da put na data t;                                           */
/* ms = taxa de retorno esperada do ativo;                                 */
/* mp = taxa de retorno esperada da put;                                   */
/***********************************************************************/
{

        double R, Exy, Ex, Ey, E1, resp;;

        R=(log(Kpj/Sj)-medj)/stdj;
        Exy=0.5*exp(medi+medj+0.5*(sqr(stdi)+sqr(stdj))+corij*stdi*stdj)
                  *(1.0+erff((R-stdj-corij*stdi)/sqrt(2.0)));
        Ex=0.5*exp(medi+0.5*sqr(stdi))
              *(1.0+erff((R-corij*stdi)/sqrt(2.0)));
        Ey=0.5*exp(medj+0.5*sqr(stdj))*(1.0+erff((R-stdj)/sqrt(2.0)));
        E1=0.5*(1.0+erff(R/sqrt(2.0)));
        Exi=exp(medi+0.5*stdi*stdi);
        resp=(-Sj*Exy+Kpj*Ex-ms*(-Sj*Ey+Kpj*E1))/Pj;
        resp+=ms*mp-mp*Exi;
        return(resp);

}



double cov_cici(double medi, double stdi, double Si, double Kc1, double Kc2,
                double C1, double C2, double mc1, double mc2)
/***********************************************************************/
/* covariância da taxa de retorno de 2 calls sobre o mesmo ativo básico i  */
/* (mas podem ter strikes diferentes e, portanto, premios diferentes);     */
/* medi = retorno logarítimico esperado do ativo i;                        */
/* stdi = desvio padrão do retorno log. do ativo i;                        */
/* Si = cotacao do ativo básico i na data de abertura do contrato t;       */
```

```
/* Kc1,Kc2 = precos de exercicio das calls na data de expiracao T;         */
/* C1,C2 = premios das calls na data t;                                     */
/* mc1, mc2 = taxas de retorno esperadas das calls;                         */
/***************************************************************************/
{
        double E2x, Ex, E1, resp, R, K;

        K=dmax(Kc1,Kc2);
        R=(log(K/Si)-medi)/stdi;
        E2x=0.5*exp(2.0*medi+2.0*sqr(stdi))
              *(1.0-erff((R-2.0*stdi)/sqrt(2.0)));
        Ex=0.5*exp(medi+0.5*sqr(stdi))
              *(1.0-erff((R-stdi)/sqrt(2.0)));
        E1=0.5*(1.0-erff(R/sqrt(2.0)));
        resp=(Si*Si*E2x-Ex*Si*(Kc1+Kc2)+Kc1*Kc2*E1)/(C1*C2)-mc1*mc2;
        return(resp);
}


double cov_pipi(double medi, double stdi, double Si, double Kp1, double Kp2,
                double P1, double P2)
/***************************************************************************/
/* covariância da taxa de retorno de 2 puts sobre o mesmo ativo básico i   */
/* (mas podem ter strikes diferentes e, portanto, premios diferentes);     */
/* medi = retorno logarítimico esperado do ativo i;                        */
/* stdi = desvio padrão do retorno log. do ativo i;                        */
/* Si = cotacao do ativo básico i na data de abertura do contrato t;       */
/* Kp1,Kp2 = precos de exercicio das puts na data de expiracao T;          */
/* P1,P2 = premios das puts na data t;                                     */
/***************************************************************************/
{
        double E2x, Ex, E1, resp, R, K;

        K=dmin(Kp1,Kp2);
        R=(log(K/Si)-medi)/stdi;
        E2x=0.5*exp(2.0*medi+2.0*sqr(stdi))
              *(1.0+erff((R-2.0*stdi)/sqrt(2.0)));
        Ex=0.5*exp(medi+0.5*sqr(stdi))
              *(1.0+erff((R-stdi)/sqrt(2.0)));
        E1=0.5*(1.0+erff(R/sqrt(2.0)));
        resp=(Si*Si*E2x-Ex*Si*(Kp1+Kp2)+Kp1*Kp2*E1)/(P1*P2)-mp1*mp2;
        return(resp);
}


double cov_cipi(double medi, double stdi, double Si, double Kci,
```

```
               double Kpi, double Ci, double Pi, double mc, double mp)
/****************************************************************************/
/* covariância entre uma call e uma put sobre o mesmo ativo básico i;      */
/* medi = retorno logarítimico esperado do ativo i;                         */
/* stdi = desvio padrão do retorno log. do ativo i;                         */
/* Si = cotacao do ativo básico i na data de abertura do contrato t;        */
/* Kci,Kpi = precos de exercicio da call e da put na data T;                */
/* Ci, Pi = premios da call e da put na data t;                             */
/* mc, mp = taxas de retorno esperadas da call e da put;                    */
/****************************************************************************/
{
        double E2x, Ex, E1, resp, Rc, Rp;

        if (Kci < Kpi){
                Rc=(log(Kci/Si)-medi)/stdi;
                Rp=(log(Kpi/Si)-medi)/stdi;
                E2x=0.5*exp(2.0*(medi+stdi*stdi))*
                    (erff((Rp-2.0*stdi)/sqrt(2.0))
                    -erff((Rc-2.0*stdi)/sqrt(2.0)));
                Ex=0.5*exp(medi+0.5*stdi*stdi)*
                    (erff((Rp-stdi)/sqrt(2.0))
                    -erff((Rc-stdi)/sqrt(2.0)));
                E1=0.5*(erff(Rp/sqrt(2.0))
                       -erff(Rc/sqrt(2.0)));
                resp=(-Si*Si*E2x+Si*Ex*(Kci+Kpi)-Kci*Kpi*E1)/(Ci*Pi);
                resp-=mc*mp;
                return(resp);
        }
        else return(-mc*mp);
}


double cov_cicj(double medi, double medj, double stdi, double stdj,
                double corij, double Si, double Sj, double Kci,
                double Kcj, double Ci, double Cj, double mci, double mcj)
/****************************************************************************/
/* covariância entre 2 calls sobre ativos básicos diferentes (i e j);      */
/* medi, medj = retorno logarítimico esperado dos ativos i e j;             */
/* stdi, stdj = desvio padrão do retorno log. dos ativos i e j;             */
/* corij = coef. de correlacao entre os retornos log. dos ativos i e j;    */
/* Si,Sj = cotacao dos ativos básicos i e j na data de abertura dos        */
/* contratos;                                                               */
/* Kci,Kcj = preco de exercicio da call na data de expiracao dos contratos;*/
/* Ci,Cj = premios das calls na data t;                                     */
/* mci, mcj = taxas de retorno esperadas das calls;                         */
/****************************************************************************/
```

```
{
        double resp, Exy, Ex, Ey, E1, Ri, Rj, a;

        Ri=(log(Kci/Si)-medi)/stdi;
        Rj=(log(Kcj/Sj)-medj)/stdj;
        a=stdi+stdj*corij;
        Exy=I_cov(1,Ri,1.0,-1.0,a,medi,medj,stdi,stdj,corij,Rj);
        Exy*=exp(medi+medj+0.5*(stdi*stdi+stdj*stdj)+corij*stdi*stdj);
        Exy/=(2.0*sqrt(2.0*pi));
        a=stdi;
        Ex=I_cov(2,Ri,1.0,-1.0,a,medi,medj,stdi,stdj,corij,Rj);
        Ex*=exp(medi+0.5*stdi*stdi)/(2.0*sqrt(2.0*pi));
        a=stdj*corij;
        Ey=I_cov(3,Ri,1.0,-1.0,a,medi,medj,stdi,stdj,corij,Rj);
        Ey*=exp(medj+0.5*stdj*stdj)/(2.0*sqrt(2.0*pi));
        a=0.0;
        E1=I_cov(4,Ri,1.0,-1.0,a,medi,medj,stdi,stdj,corij,Rj);
        E1/=(2.0*sqrt(2.0*pi));
        Exa=0.5*exp(medi+0.5*stdi*stdi)
                *(1.0-erff((Ri-stdi)/sqrt(2.0)));
        E1a=(1.0-erff(Ri/sqrt(2.0)));
        Eyb=0.5*exp(medj+0.5*stdj*stdj)
                *(1.0-erff((Rj-stdj)/sqrt(2.0)));
        E1b=(1.0-erff(Rj/sqrt(2.0)));
        resp=(Si*Sj*Exy-Si*Kcj*Ex-Sj*Kci*Ey+Kci*Kcj*E1)/(Ci*Cj);
        resp-=mcj*(Exa*Si-Kci*E1a)/Ci;
        resp-=mci*(Eyb*Sj-Kcj*E1b)/Cj;
        resp+=mci*mcj;
        return(resp);
}


double cov_cipj(double medi, double medj, double stdi, double stdj,
                double corij, double Si, double Sj, double Kci,
                double Kpj, double Ci, double Pj, double mc, double mp)
/***********************************************************************/
/* covariância entre uma call sobre o ativo básico i, e uma put sobre o    */
/* ativo básico j;                                                         */
/* medi, medj = retorno logarítimico esperado dos ativos i e j;            */
/* stdi, stdj = desvio padrão do retorno log. dos ativos i e j;            */
/* corij = coef. de correlacao entre os retornos log. dos ativos i e j;    */
/* Si,Sj = cotacao dos ativos básicos i e j na data de abertura dos        */
/* contratos;                                                              */
/* Kci,Kpj = precos de exercicio da call e da put na data de expiracao dos */
/* contratos;                                                              */
/* Ci,Pj = premios da call e da put na data t;                             */
```

```
/*  mc, mp = taxas de retorno esperada da call e da put;                  */
/************************************************************************/
{
        double resp, Exy, Ex, Ey, E1, Ri, Rj, a;

        Ri=(log(Kci/Si)-medi)/stdi;
        Rj=(log(Kpj/Sj)-medj)/stdj;
        a=stdi+stdj*corij;
        Exy=I_cov(1,Ri,1.0,1.0,a,medi,medj,stdi,stdj,corij,Rj);
        Exy*=exp(medi+medj+0.5*(stdi*stdi+stdj*stdj)+corij*stdi*stdj);
        Exy/=(2.0*sqrt(2.0*pi));
        a=stdi;
        Ex=I_cov(2,Ri,1.0,1.0,a,medi,medj,stdi,stdj,corij,Rj);
        Ex*=exp(medi+0.5*stdi*stdi)/(2.0*sqrt(2.0*pi));
        a=stdj*corij;
        Ey=I_cov(3,Ri,1.0,1.0,a,medi,medj,stdi,stdj,corij,Rj);
        Ey*=exp(medj+0.5*stdj*stdj)/(2.0*sqrt(2.0*pi));
        a=0.0;
        E1=I_cov(4,Ri,1.0,1.0,a,medi,medj,stdi,stdj,corij,Rj);
        E1/=(2.0*sqrt(2.0*pi));
        Exa=0.5*exp(medi+0.5*stdi*stdi)
                *(1.0-erff((Ri-stdi)/sqrt(2.0)));
        E1a=(1.0-erff(Ri/sqrt(2.0)));
        Eyb=0.5*exp(medj+0.5*stdj*stdj)
                *(1.0+erff((Rj-stdj)/sqrt(2.0)));
        E1b=(1.0+erff(Rj/sqrt(2.0)));
        resp=(-Si*Sj*Exy+Si*Kpj*Ex+Sj*Kci*Ey-Kci*Kpj*E1)/(Ci*Pj);
        resp-=mp*(Exa*Si-Kci*E1a)/Ci;
        resp-=mc*(-Eyb*Sj+Kpj*E1b)/Pj;
        resp+=mc*mp;
        return(resp);
}


double cov_pipj(double medi, double medj, double stdi, double stdj,
                double corij, double Si, double Sj, double Kpi,
                double Kpj, double Pi, double Pj, double mpi,
                double mpj)
/************************************************************************/
/* covariância entre puts sobre ativos básico diferentes ( i e j);        */
/* medi, medj = retorno logarítimico esperado dos ativos i e j;           */
/* stdi, stdj = desvio padrão do retorno log. dos ativos i e j;           */
/* corij = coef. de correlacao entre os retornos log. dos ativos i e j;   */
/* Si,Sj = cotacao dos ativos básicos i e j na data de abertura dos       */
/* contratos;                                                             */
/* Kpi,Kpj = precos de exercicio das puts na data de expiracao dos        */
```

```
/* contratos;                                                           */
/* Pi,Pj = premios das puts na data t;                                  */
/* mpi, mpj = taxas de retorno esperadas das puts;                      */
/************************************************************************/
{
        double resp, Exy, Ex, Ey, E1, Ri, Rj, a;

        Ri=(log(Kpi/Si)-medi)/stdi;
        Rj=(log(Kpj/Sj)-medj)/stdj;
        a=stdi+stdj*corij;
        Exy=I_cov(1,Ri,-1.0,1.0,a,medi,medj,stdi,stdj,corij,Rj);
        Exy*=exp(medi+medj+0.5*(stdi*stdi+stdj*stdj)+corij*stdi*stdj);
        Exy/=(2.0*sqrt(2.0*pi));
        a=stdi;
        Ex=I_cov(2,Ri,-1.0,1.0,a,medi,medj,stdi,stdj,corij,Rj);
        Ex*=exp(medi+0.5*stdi*stdi)/(2.0*sqrt(2.0*pi));
        a=stdj*corij;
        Ey=I_cov(3,Ri,-1.0,1.0,a,medi,medj,stdi,stdj,corij,Rj);
        Ey*=exp(medj+0.5*stdj*stdj)/(2.0*sqrt(2.0*pi));
        a=0.0;
        E1=I_cov(4,Ri,-1.0,1.0,a,medi,medj,stdi,stdj,corij,Rj);
        E1/=(2.0*sqrt(2.0*pi));
        Exa=0.5*exp(medi+0.5*stdi*stdi)
                *(1.0+erff((Ri-stdi)/sqrt(2.0)));
        E1a=(1.0+erff(Ri/sqrt(2.0)));
        Eyb=0.5*exp(medj+0.5*stdj*stdj)
                *(1.0+erff((Rj-stdj)/sqrt(2.0)));
        E1b=(1.0+erff(Rj/sqrt(2.0)));
        resp=(Si*Sj*Exy-Si*Kpj*Ex-Sj*Kpi*Ey+Kpi*Kpj*E1)/(Pi*Pj);
        resp+=mpj*(Exa*Si-Kpi*E1a)/Pi;
        resp+=mpi*(Eyb*Sj-Kpj*E1b)/Pj;
        resp+=mpi*mpj;
        return(resp);
}


double E_ri(double medi, double stdi)
/* esperanca da taxa de retorno do ativo i. */
{
        return(exp(medi+0.5*sqr(stdi)));
}


double E_ci(double medi, double stdi, double Si, double Kci, double Ci)
/************************************************************************/
/* experanca da taxa de retorno da call = (max(Kci(T)-Si(t),0)-Ci(t))/Ci(t)*/
```

```
/* t = data do inicio do contrato;                                      */
/* T = data de expiracao do contrato;                                   */
/***********************************************************************/
{
        double R, Ex, E1, resp;

        R=(log(Kci/Si)-medi)/(stdi);
        Ex=(1.0-erff((R-stdi)/sqrt(2.0)))*exp(medi+0.5*sqr(stdi))*0.5;
        E1=0.5*(1.0-erff(R/sqrt(2.0)));
        resp=(Si*Ex-Kci*E1)/Ci;
        return(resp);
}


double E_pi(double medi, double stdi, double Si, double Kpi, double Pi)
/***********************************************************************/
/* experanca da taxa de retorno da put = (max(-Kpi(T)+Si(t),0)-Pi(t))/Pi(t)*/
/* t = data do inicio do contrato;                                      */
/* T = data de expiracao do contrato;                                   */
/***********************************************************************/
{
        double R, Ex, E1, resp;

        R=(log(Kpi/Si)-medi)/(stdi);
        Ex=(1.0+erff((R-stdi)/sqrt(2.0)))*exp(medi+0.5*sqr(stdi))*0.5;
        E1=0.5*(1.0+erff(R/sqrt(2.0)));
        resp=(-Si*Ex+Kpi*E1)/Pi;
        return(resp);
}


double ss(int i, int j, double *med, double *std, double **cor, int *atv)
/***********************************************************************/
/* chamada da covariância que envolve somente 2 ativos básicos;         */
/* a funcao de covariância apropriada vai depender se i e j referem-se  */
/* ao mesmo ativo básico.                                               */
/***********************************************************************/
{
        double resp=1.0E+308;


        if (atv[i] == atv[j]){
                resp=cov_riri(med[atv[i]],std[atv[i]]);
        }
        else{
                resp=cov_rirj(med[atv[i]],med[atv[j]],std[atv[i]],
```

```
                              std[atv[j]],cor[atv[i]][atv[j]]);
        }
        return(resp);
}


double sc(int i, int j, double *med, double *std, double **cor, double *S,
          double *K, double *P, int *atv, double *esp)
/*************************************************************************/
/* chamada da covariância que envolve um ativo básico e uma call;        */
/* a funcao de covariância apropriada vai depender se i e j referem-se   */
/* ao mesmo ativo básico.                                                */
/*************************************************************************/
{
        double resp=1.0E+308;

        if (atv[i] == atv[j]){
                resp=cov_rici(med[atv[i]],std[atv[i]],S[j],K[j],P[j],esp[i],
                              esp[j]);
        }
        else{
                resp=cov_ricj(med[atv[i]],med[atv[j]],std[atv[i]],
                               std[atv[j]], cor[atv[i]][atv[j]],S[j],
                               K[j],P[j],esp[i],esp[j]);
        }
        return(resp);
}


double sp(int i, int j, double *med, double *std, double **cor, double *S,
          double *K, double *P, int *atv, double *esp)
/*************************************************************************/
/* chamada da covariância que envolve um ativo básico e uma put;         */
/* a funcao de covariância apropriada vai depender se i e j referem-se   */
/* ao mesmo ativo básico.                                                */
/*************************************************************************/
{
        double resp=1.0E+308;

        if (atv[i] == atv[j]){
                resp=cov_ripi(med[atv[i]],std[atv[i]],S[j],K[j],P[j]
                              esp[i],esp[j]);
        }
        else{
                resp=cov_ripj(med[atv[i]],med[atv[j]],std[atv[i]],std[atv[j]],
                              cor[atv[i]][atv[j]],S[j],K[j],P[j],
```

```
                            esp[i],esp[j]);
        }
        return(resp);
}


double cc(int i, int j, double *med, double *std, double **cor, double *S,
          double *K, double *P, int *atv, double *esp)
/*****************************************************************************/
/* chamada da covariância que envolve 2 calls;                               */
/* a funcao de covariância apropriada vai depender se i e j referem-se     */
/* ao mesmo ativo básico.                                                    */
/*****************************************************************************/
{
        double resp=1.0E+308;

        if (atv[i] == atv[j]){
                resp=cov_cici(med[atv[i]],std[atv[i]],S[i],K[i],K[j],P[i],P[j]
                              esp[i],esp[j]);
        }
        else{
                resp=cov_cicj(med[atv[i]],med[atv[j]],std[atv[i]],std[atv[j]],
                              cor[atv[i]][atv[j]],S[i],S[j],K[i],K[j],P[i],P[j],
                              esp[i],esp[j]);
        }
        return(resp);
}


double cp(int i, int j, double *med, double *std, double **cor, double *S,
          double *K, double *P, int *atv, double *esp)
/*****************************************************************************/
/* chamada da covariância que envolve uma call e uma put;                   */
/* a funcao de covariância apropriada vai depender se i e j referem-se     */
/* ao mesmo ativo básico.                                                    */
/*****************************************************************************/
{
        double resp=1.0E+308;

        if (atv[i] == atv[j]){
                resp=cov_cipi(med[atv[i]],std[atv[i]],S[i],K[i],K[j],P[i],P[j],
                              esp[i],esp[j]);
        }
        else{
                resp=cov_cipj(med[atv[i]],med[atv[j]],std[atv[i]],std[atv[j]],
                              cor[atv[i]][atv[j]],S[i],S[j],K[i],K[j],P[i],P[j],
```

```
                              esp[i],esp[j]);
        }
        return(resp);
}


double pp(int i, int j, double *med, double *std, double **cor, double *S,
          double *K, double *P, int *atv, double *esp)
/*****************************************************************************/
/* chamada da covariância que envolve 2 puts;                                */
/* a funcao de covariância apropriada vai depender se i e j referem-se      */
/* ao mesmo ativo básico.                                                    */
/*****************************************************************************/
{
        double resp=1.0E+308;

        if (atv[i] == atv[j]){
                resp=cov_pipi(med[atv[i]],std[atv[i]],S[i],K[i],K[j],P[i],P[j],
                              esp[i],esp[j]);
        }
        else{
                resp=cov_pipj(med[atv[i]],med[atv[j]],std[atv[i]],std[atv[j]],
                              cor[atv[i]][atv[j]],S[i],S[j],K[i],K[j],P[i],P[j],
                              esp[i],esp[j]);
        }
        return(resp);
}


/*****************************************************************************/
/********        Rotinas para o cálculo numérico das integrais       *******/
/*****************************************************************************/

#define EPS 3.0e-7
#define ITMAX 100
#define FPMIN 1.0e-30
#define FUNC(x) ((*func)(x))
#define EPS1 1.0e-5
#define JMAX 20
#define NMAX 20
#define JMAx 40
#define MAXIT 40
#define UNUSED (-1.11e30)
#define SIGN(a,b) ((b) >=0.0 ? fabs(a) : -fabs(a))
#define pi 3.141592653589793
```

```
void nrerror(char error_text[])
{

        fprintf(stderr,"Numerical Recipes run-time error...\n");
        fprintf(stderr,"%s\n",error_text);
        fprintf(stderr,"...now exiting to system...\n");
        exit(1);
}


double gammln(double xx)
{
        double x,y,tmp,ser;
        static double cof[6]={76.18009172947146,-86.50532032941677,
                              24.01409824083091,-1.231739572450155,
                              0.1208650973866179e-2,-0.5395239384953e-5};
        int j;

        y=x=xx;
        tmp=x+5.5;
        tmp-=(x+0.5)*log(tmp);
        ser=1.000000000190015;
        for (j=0;j<=5;j++) ser+=cof[j]/++y;
        return -tmp+log(2.5066282746310005*ser/x);
}


void gser(double *gamser, double a, double x, double *gln)
{
        double gammln(double xx);
        void nrerror(char error_text[]);
        int n;
        double sum,del,ap;

        *gln=gammln(a);
        if (x <= 0.0){
                if (x < 0.0) nrerror(" x less than 0 in routine gser ");
                *gamser=0.0;
                return;
        }
        else{
                ap=a;
                del=sum=1.0/a;
                for (n=1;n<=ITMAX;n++){
                        ++ap;
```

```
                        del*=x/ap;
                        sum+=del;
                        if (fabs(del) < fabs(sum)*EPS){
                                *gamser=sum*exp(-x+a*log(x)-(*gln));
                                return;
                        }
                }
                nrerror("a too large, ITMAX too small in routine gser");
                return;
        }
}

void gcf(double *gammcf, double a, double x, double *gln)
{
        double gammln(double xx);
        void nrerror(char error_text[]);
        int i;
        double an,b,c,d,del,h;

        *gln=gammln(a);
        b=x+1.0-a;
        c=1.0/FPMIN;
        d=1.0/b;
        h=d;
        for (i=1;i<=ITMAX; i++){
                an=-i*(i-a);
                b+=2.0;
                d=an*d+b;
                if (fabs(d) < FPMIN) d=FPMIN;
                c=b+an/c;
                if (fabs(c) < FPMIN) c=FPMIN;
                d=1.0/d;
                del=d*c;
                h*=del;
                if (fabs(del-1.0) < EPS) break;
        }
        if (i > ITMAX) nrerror("a too large, ITMAX too small in gcf");
        *gammcf=exp(-x+a*log(x)-(*gln))*h;
}

double gammp(double a, double x)
{
        void gcf(double *gammcf, double a, double x, double *gln);
        void gser(double *gamser, double a, double x, double *gln);
        void nerror(char error_text[]);
        double gamser, gammcf, gln;
```

```
        if (x < 0.0 || a <= 0.0) nrerror("Invalid arguments in routine gammp");
        if (x < (a+1.0)){
                gser(&gamser,a,x,&gln);
                return gamser;
        }
        else{
                gcf(&gammcf,a,x,&gln);
                return 1.0-gammcf;
        }
}


double gammq(double a, double x)
{
        void gcf(double *gammcf, double a, double x, double *gln);
        void gser(double *gamser, double a, double x, double *gln);
        void nrerror(char error_text[]);
        double gamser, gammcf, gln;

        if (x < 0.0 || a <= 0.0) nrerror("Invalid arguments in routine gammq");
        if (x < (a+1.0)){
                gser(&gamser,a,x,&gln);
                return 1.0-gamser;
        }
        else{
                gcf(&gammcf,a,x,&gln);
                return gammcf;
        }
}


double erff(double x)
{
        double gammp(double a, double x);
        return x < 0.0 ? -gammp(0.5,x*x) : gammp(0.5,x*x);
}

double erffc(double x)
{
        double gammp(double a, double x);
        double gammq(double a, double x);

        return x < 0.0 ? 1.0+gammp(0.5,x*x) : gammq(0.5,x*x);
}
```

```
double func(char opcao, double y, double sgn, double a, double medi,
            double medj, double stdi, double stdj, double corij, double Rj)
/**********************************************************************/
/* a = elemento do termo "exp{(y-a)^2}"                               */
/* y = no'da integracao numerica                                      */
/* sgn = sinal de erff                                                */
/**********************************************************************/
{
        double resp;

        switch(opcao){
                case 1:
                  resp=(Rj-y*corij-stdj*(1.0-corij*corij))/
                               sqrt(2.0*(1.0-corij*corij));
                  resp=(1.0+sgn*erff(resp))*exp(-0.5*sqr(y-a));
                  return(resp);

                case 2:   /* relativo a Ex */
                  resp=(Rj-y*corij)/sqrt(2.0*(1.0-corij*corij));
                  resp=(1.0+sgn*erff(resp))*exp(-0.5*sqr(y-a));
                  return(resp);

                case 3:  /* relativo a Ey */
                  resp=(Rj-y*corij-stdj*(1.0-corij*corij))/
                               sqrt(2.0*(1.0-corij*corij));
                  resp=(1.0+sgn*erff(resp))*exp(-0.5*sqr(y-a));
                  return(resp);

                case 4: /* relativo a E1 */
                  resp=(Rj-y*corij)/sqrt(2.0*(1.0-corij*corij));
                  resp=(1.0+sgn*erff(resp))*exp(-0.5*sqr(y-a));
                  return(resp);

                case 9:
                        return((1.0-erff(3.0*y-2.0))*exp(-y-y*y));
                        /* esta e' para testes... */

        }
        printf(" \n deu erro !! ");
        return(1.0E308);
}


double trapezio(double *F, int n, double delta)
{
```

```
        int i,k,l,ini, fim;
        double s;

        ini=0;
        s=0.0;
        for (k=0;k<n;k++){
                for (l=1,i=1;i<=n-k;i++) l<<=1;
                fim=ini+l;
                for (i=ini; i<=fim; i++) s+=F[i]*delta/l;
                s-=delta*(F[ini]+F[fim])/(2*l);
                ini=fim;
        }
        return s;
}


double extrap(double *T, int n)
/*****************************************************************************/
/* faz o processo de extrapolação no vetor T[1..n], onde T[i] e' a         */
/* i-ésima aproximação da integral.                                          */
/*****************************************************************************/
{
        int j,k,m;

        T[0]=-1.0;
        for (k=1;k<n;k++){
                for (m=4,j=1;j<k;j++) m<<=2;
                for (j=1;j<=(n-k);j++){
                        T[j]=(m*T[j+1]-T[j])/(m-1);
                        if (fabs(T[j]-T[j-1])<0.000000001)
                           return(T[j]); /* criterio de parada */
                }
        }
        return(T[1]);
}


double integral(char opcao, double lim, double sgn, double L, double a,
                double medi, double medj, double stdi, double stdj,
                double corij, double Rj)
/*****************************************************************************/
/* calcula a integral pelo metodo do trapezio e aplica extrapolacao de     */
/* Richardson                                                                */
/* opcao = indica qual funcao a ser usada como integrando na chamada de    */
/*         "func";                                                           */
/* lim = +1 se estivermos integrando de L ate'  +infinito                   */
```

```
/*      -1 se estivermos integrando de -infinito ate'  L                */
/* sgn = sinal da funcao erff na chamada de "func";                     */
/* a : faz parte do corpo da funcao a ser integrada (exp[-(x-a)^2])...  */
/* medi,medj,stdi,stdj,corij,Rj = constantes necessarias para montar a  */
/* funcao                                                               */
/* integrando na chamada de "func".                                     */
/***********************************************************************/
{

        double *F=NULL, *X=NULL, *T=NULL;
        double quad, delta, erro;
        int nf,k, n;
        long m, Nf,i, j,l,N;

        delta=2.0;
        X=aloc_vet(3);
        F=aloc_vet(3);
        T=aloc_vet(3);
        N=3;
        n=1;
        erro=1.0;
        while (( (erro > 0.0001) | (erro < -0.0001)) | (n <= 8)){
                if (n>1){
                        for (m=2,j=1;j<n;j++) m<<=1;
                        Nf=(m-1)*2+1;
                        X=realoca(X,Nf);
                        F=realoca(F,Nf);
                        T=realoca(T,n);
                        for (i=N-1;i>0;i--){
                                         X[2*i]=X[i];
                                         F[2*i]=F[i];
                        }
                        m=0;
                        j=0;
                        for (k=0;k<n-1;k++){
                                for (l=1,i=1;i<=n-k;i++) l<<=1;
                                m+=l;
                                for (i=j;i< m/2;i++){
                                       X[2*i+1]=X[2*i]+lim*delta/l;
                                       F[2*i+1]=func(opcao,X[2*i+1],sgn,a,
                                                    medi,medj,stdi,stdj,
                                                    corij,Rj);
                                }
                                j=i;
                        }
                        for (i=Nf-2;i<Nf;i++){
```

```
                                X[i]=X[i-1]+lim*delta/2;
                                F[i]=func(opcao,X[i],sgn,a,medi,medj,stdi,
                                        stdj,corij,Rj);
                        }
                        N=Nf;
                }
                else for (i=0;i<=2;i++){
                                X[i]=L+lim*i*delta/2;
                                F[i]=func(opcao,X[i],sgn,a,medi,medj,stdi,
                                        stdj,corij,Rj);
                     }
                T[n]=trapezio(F,n,delta);
                if (n > 1) erro = T[n]-T[n-1];
                n++;
        }
        n--;
        nf=n;
        quad=extrap(T,nf);
        free(X);
        free(F);
        free(T);
        return(quad);
}

double romb_cov(char opcao, double a, double b, double sgn, double medi,
                double medj, double stdi, double stdj, double corij,
                double x0, double Rj)
/***************************************************************************/
/* método de romberg para integracao em dominios finitos (a,b)            */
/* x0 = termo da funcao que leva a exponencial : exp[(x-L)^2/2]            */
/***************************************************************************/
{
        double *T=NULL, *F=NULL, *X=NULL;
        double h,resp;
        int i,j,n,fim,l;


        X=aloc_vet(1);
        T=aloc_vet(1);
        F=aloc_vet(1);
        n=1;
        fim=0;
        T[0]=-1;
        while ((!fim) && (n<9)){
                for (l=2,j=1;j<n;j++) l<<=1;   /* l=2^n */
                T=realoca(T,n+1);
```

```
                F=realoca(F,l+1);
                X=realoca(X,l+1);
                T[n]=0.0;
                h=(b-a)/l;
                if (n>1){
                        for (j=l/2;j>=1;j--){
                                X[2*j]=X[j];
                                F[2*j]=F[j];
                        }
                        for (j=1;j<=l/2;j++){
                                i=2*j-1;
                                X[i]=X[i-1]+h;
                                F[i]=func(opcao,X[i],sgn,x0,medi,medj,stdi,
                                          stdj,corij,Rj);
                        }
                        for (i=0;i<l;i++) T[n]+=F[i]+F[i+1];
                        T[n]*=0.5*h;
                        if ((fabs(T[n]-T[n-1]) < 0.00001) | n==8) fim=1;
                        else n++;
                }
                else{
                        for (i=0;i<=l;i++){
                                X[i]=a+i*h;
                                F[i]=func(opcao,X[i],sgn,x0,medi,medj,stdi,
                                          stdj,corij,Rj);
                        }
                        for (i=0;i<l;i++) T[n]+=F[i]+F[i+1];
                        T[n]*=0.5*h;
                        n++;
                }
        }
        resp=extrap(T,n);
        free(T);
        free(X);
        free(F);
        return(resp);
}

double I_cov(char opcao, double L, double lim, double sgn, double a,
             double medi, double medj, double stdi, double stdj,
             double corij, double Rj)
/*****************************************************************************/
/* calculo da integral de f(x)*exp{(x-a)^2} (o corpo de f(x) e'dado pela   */
/* "opcao"),                                                                 */
/* de L ate' (+-) infinito;                                                  */
/* lim = +1 se estamos integrando de L ate' +infinito;                       */
```

```
/*       -1 se         "          de -infinito ate' L;                    */
/* sgn = sinal multiplicando a funcao erff na "opcao";                    */
/**************************************************************************/
{
        double I, delta;

        delta=1.0;
        if (fabs(a-L) < 0.5*delta){
                I=integral(opcao,lim,sgn,L,a,medi,medj,stdi,stdj,corij,Rj);
                return(I);
        }
        else{
                if ((L-a)*lim > 0.0){
                        I=integral(opcao,lim,sgn,L,a,medi,medj,stdi,stdj,
                                    corij,Rj);
                        return(I);
                }
                else{
                        I=integral(opcao,lim,sgn,a,a,medi,medj,stdi,stdj,
                                   corij,Rj);
                        if (L > a)
                          I+=romb_cov(opcao,a,L,sgn,medi,medj,stdi,stdj,
                                      corij,a,Rj);
                        else
                          I+=romb_cov(opcao,L,a,sgn,medi,medj,stdi,stdj,
                                      corij,a,Rj);
                        return(I);
                }
        }
}

/**************************************************************************/
/*****  Funções de alocação, leitura e escrita de vetores e matrizes   *****/
/**************************************************************************/


double dmax(double a, double b)
{
        if (a >= b) return(a);
        else return(b);
}

double dmin(double a, double b)
{
        if (a >= b) return(b);
        else return(a);
```

```
}


double sqr(double y)
{
        return(y*y);
}


double **aloc_mat(int n)
/* aloca uma matriz */
{
        double **R=NULL;
        int i;

        if ((R = (double**)malloc(sizeof(double*)*(n))) == NULL){
                printf(" memoria insuficiente!");
                exit(1);
        }
        for (i=0;i<n;i++)
                if ((R[i] = (double*)malloc(sizeof(double)*(n))) == NULL){
                        printf(" memoria insuficiente!");
                        exit(1);
                }
        return(R);
}

double  *aloc_vet(int n)
/* aloca um vetor */
{
        double *R=NULL;

        if ((R=(double*)malloc(sizeof(double)*(n)))==NULL){
                printf("Memoria Insuficiente para alocar X!!");
                exit(1);
        }
        return(R);
}


int *aloc_vet_int(int n)
/* aloca um vetor de inteiros */
{
        int *R=NULL;

        if ((R=(int*)malloc(sizeof(int)*(n)))==NULL){
```

```
                printf("Memoria Insuficiente para alocar X!!");
                exit(1);
        }
        return(R);
}

int *le_vet_int(char *arq, int n)
/*
leitura de um arquivo que contém os dados do vetor de media,desvio padrão, etc.
*/
{
        int i;
        int *R;
        FILE *fp;


        if ((fp = fopen(arq,"r")) == NULL){
                printf(" Erro na abertura do arquivo ");
                exit(1);
        }
        R=aloc_vet_int(n);
        i=0;
        while (!feof(fp) && (i<n)){
                fscanf(fp,"%d ", &R[i]);
                i++;
        }
        fclose(fp);
        return(R);
}



double *le_vet(char *arq, int n)
/*
leitura de um arquivo que contém os dados do vetor de media,desvio padrão,etc.
*/
{
        int i;
        double *R;
        FILE *fp;

        if ((fp = fopen(arq,"r")) == NULL){
                printf(" Erro na abertura do arquivo ");
                exit(1);
        }
        R=aloc_vet(n);
```

```
        i=0;
        while (!feof(fp) && (i<n)){
                fscanf(fp,"%lf ", &R[i]);
                i++;
        }
        fclose(fp);
        return(R);
}

double **le_mat(char *arq, int *n)
/* funcao de leitura da matriz dos coef. de correlacao */
{
        int i,j;
        double **R;
        FILE *fp;

        if ((fp = fopen(arq,"r")) == NULL){
                printf(" Erro na abertura do arquivo ");
                exit(1);
        }
        fscanf(fp," %d ",n);
        R=aloc_mat(*n);
        i=0;
        while (!feof(fp) && (i<*n)){
                for (j=0;j<*n;j++) fscanf(fp,"%lf ", &R[i][j]);
                i++;
        }
        fclose(fp);
        return(R);
}


double *realoca(double *Y, int Nf)
/* realoca um vetor */
{
        double *R=NULL;

        if ((R=(double*)realloc(Y,sizeof(double)*(Nf+3)))==NULL){
                printf("Memoria Insuficiente para alocar X!!");
                exit(1);
        }
        return(R);
}


void escreve_vet(char *arq, double *v, char *str[], int n)
```

```
/* escreve em arquivo o conteudo do vetor v */
{
        int i;
        FILE *fp=NULL;
        if ((fp=fopen(arq,"w+")) == NULL){
                printf(" erro na abertura do arquivo %s",arq);
                exit(1);
        }
        fprintf(fp," vet={");
        for (i=0;i<n-1;i++) fprintf(fp," %.10lf@%s,  ",v[i],str[i]);
        fprintf(fp," %.10lf@%s };",v[i],str[i]);
        fclose(fp);
}


void escreve_mat(char *arq,double **m, int n)
/* escreve em arquivo o conteudo da matriz m */
{

        int i,j;
        FILE *fp=NULL;

        if ((fp=fopen(arq,"w+")) == NULL){
                printf("Erro na abertura do arquivo %s",arq);
                exit(1);
        }
        fprintf(fp, " mat = [  \n");
        for (i=0;i<n;i++){
                for (j=0;j<n-1;j++){
                        fprintf(fp," %.10lf, ",m[i][j]);
                }
                fprintf(fp," %.10lf;\n ",m[i][j]);
        }
        fprintf(fp,"] \n");
        fclose(fp);
}


void escreve_mat3d(char *arq,double ***m, int n)
/* escreve em arquivo o conteudo da matriz m */
{

        int i,j,k;
        FILE *fp=NULL;

        if ((fp=fopen(arq,"w+")) == NULL){
```

```
                printf("Erro na abertura do arquivo %s",arq);
                exit(1);
        }
        for (k=0;k<n;k++){
        fprintf(fp, " k=%d : [  \n",k);
        for (i=0;i<n;i++){
                for (j=0;j<n-1;j++){
                        fprintf(fp," %.10f, ",m[i][j][k]);
                }
                fprintf(fp," %.10f;\n ",m[i][j][k]);
        }
        fprintf(fp,"] \n");
        }
        fclose(fp);
}

double ***aloc_mat3d(int n)
/* aloca uma matriz tri-dimensional nxnxn*/
{
        double ***R=NULL;
        int i,j;

        if ((R = (double***)malloc(sizeof(double*)*(n))) == NULL){
                printf(" memoria insuficiente!");
                exit(1);
        }
        for (i=0;i<n;i++){
                if ((R[i] = (double**)malloc(sizeof(double)*(n))) == NULL){
                        printf(" memoria insuficiente!");
                        exit(1);
                }
                for (j=0;j<n;j++){
                        if ((R[i][j] =
                             (double*)malloc(sizeof(double)*(n))) == NULL){
                                 printf(" memoria insuficiente!");
                                 exit(1);
                        }
                }
        }
        return(R);
}
```

A.2 Rotinas de Co-assimetrias

```
/*************************************************************************/
/* expressões de assimetria que só envolvem integral simples (função erff) */
/*************************************************************************/


double m_ririri(double medi, double stdi)
/* terceiro momento da taxa de retorno ri */
{
        double Ex,E2x,E3x,resp;

        Ex=exp(medi+0.5*stdi*stdi);
        E2x=exp(2.0*(medi+stdi*stdi));
        E3x=exp(0.5*(6.0*medi+9.0*stdi*stdi));
        resp=E3x-3.0*E2x*Ex+2.0*Ex*Ex*Ex;
        return(resp);
}


double m_ririci(double medi, double stdi, double Si, double Kci, double Ci,
                double ms, double mc)
/*************************************************************************/
/* coskewnes entre a taxa de retorno da call (sobre o i-ésimo ativo) com o */
/* proprio ativo;                                                          */
/* medi, stdi = media e desvio p. do ativo i;                              */
/* Kci = strike price da call;                                             */
/* Ci = premio da call;                                                    */
/* Si = cotacao do ativo i na data t=0.                                    */
/* ms = taxa de retorno esperada do ativo objeto;                          */
/* mp = taxa de retorno esperada da put;                                   */
/*************************************************************************/
{
        double Ex, E2x, E3x,E2xi,resp,f;

        f=(log(Kci/Si)-medi)/stdi);
        E3x=0.5*exp(3.0*medi+4.5*stdi*stdi)*(1.0-erff((f-3.0*stdi)/sqrt(2.0)));
        E2x=0.5*exp(2.0*medi+2.0*stdi*stdi)*(1.0-erff((f-2.0*stdi)/sqrt(2.0)));
        Ex=0.5*exp(medi+0.5*stdi*stdi)*(1.0-erff((f-stdi)/sqrt(2.0)));
        E2xi=exp(2.0*medi+2.0*stdi*stdi);
        resp=(Si*E3x-Kci*E2x)/Ci-2.0*ms*(Si*E2x-Kci*Ex)/Ci;
        resp+=-ms*E2xi+2.0*ms*ms*mc
        return(resp);
}
```

```
double m_riripi(double medi, double stdi, double Si, double Kpi, double Pi
                double ms, double mp)
/***********************************************************************/
/* coskewnes entre a taxa de retorno da put (sobre o i-ésimo ativo) com o  */
/* proprio ativo;                                                        */
/* medi, stdi = media e desvio p. do ativo i;                            */
/* Kpi = strike price da put;                                            */
/* Pi = premio da put;                                                   */
/* Si = cotacao do ativo i na data t=0.                                  */
/* ms = taxa de retorno esperada do ativo objeto;                        */
/* mp = taxa de retorno esperada da put;                                 */
/***********************************************************************/
{
        double Ex, E2x, E3x,E2xi,resp,f;

        f=(log(Kpi/Si)-medi)/stdi);
        E3x=0.5*exp(3.0*medi+4.5*stdi*stdi)*(1.0+erff((f-3.0*stdi)/sqrt(2.0)));
        E2x=0.5*exp(2.0*medi+2.0*stdi*stdi)*(1.0+erff((f-2.0*stdi)/sqrt(2.0)));
        Ex=0.5*exp(medi+0.5*stdi*stdi)*(1.0+erff((f-stdi)/sqrt(2.0)));
        E2xi=exp(2.0*medi+2.0*stdi*stdi);
        resp=(-Si*E3x+Kpi*E2x)/Pi-2.0*ms*(-Si*E2x+Kpi*Ex)/Pi;
        resp+=-mp*E2xi+2.0*ms*ms*mp
        return(resp);

}



double m_ricici(double medi, double stdi, double Si, double Kc1, double Kc2,
                double C1, double C2, double ms, double mc1, double mc2)
/***********************************************************************/
/* coskewnes entre a taxa de retorno das calls sobre o i-ésimo ativo com o */
/* proprio ativo;                                                        */
/* medi, stdi = media e desvio p. do ativo i;                            */
/* Kc1,Kc2 = strike price das calls;                                     */
/* C1,C2 = premios das calls;                                            */
/* Si = cotacao do ativo i na data t=0.                                  */
/* ms = taxa de retorno esperada do ativo objeto;                        */
/* mc1, mc2 = taxas de retorno esperadas das calls;                      */
/***********************************************************************/
{
        double Ex, E2x, E3x,E1,E2x1,Ex1,E2x2,Ex2,resp,K,f;

        K=dmax(Kc1,Kc2);
```

```
        f=(log(K/Si)-medi)/stdi);
        E3x=0.5*exp(3.0*medi+4.5*stdi*stdi)*(1.0-erff((f-3.0*stdi)/sqrt(2.0)));
        E2x=0.5*exp(2.0*medi+2.0*stdi*stdi)*(1.0-erff((f-2.0*stdi)/sqrt(2.0)));
        Ex=0.5*exp(medi+0.5*stdi*stdi)*(1.0-erff((f-stdi)/sqrt(2.0)));
        E1=0.5*(1.0-erff(f));
        f=(log(Kc1/Si)-medi)/stdi);
        E2x1=0.5*exp(2.0*medi+2.0*stdi*stdi)*(1.0-erff((f-2.0*stdi)/sqrt(2.0)));
        Ex1=0.5*exp(medi+0.5*stdi*stdi)*(1.0-erff((f-stdi)/sqrt(2.0)));
        f=(log(Kc2/Si)-medi)/stdi);
        E2x2=0.5*exp(2.0*medi+2.0*stdi*stdi)*(1.0-erff((f-2.0*stdi)/sqrt(2.0)));
        Ex2=0.5*exp(medi+0.5*stdi*stdi)*(1.0-erff((f-stdi)/sqrt(2.0)));
        resp=(Si*Si*E3x-Si*E2x*(Kc1+Kc2)+Kc1*Kc2*Ex)/(C1*C2);
        resp-=ms*(Si*Si*E2x-Si*Ex*(Kc1+Kc2)+Kc1*Kc2*E1)/(C1*C2);
        resp-=mc1*(Si*E2x2-Kc2*Ex2)/C2;
        resp-=mc2*(Si*E2x1-Kc1*Ex1)/C1;
        resp+=2.0*ms*mc2*mc1;
        return(resp);
}


double m_ripipi(double medi, double stdi, double Si, double Kp1, double Kp2,
                double P1, double P2, double ms, double mp1, double mp2)
/**********************************************************************/
/* coskewnes entre a taxa de retorno das puts sobre o i-ésimo ativo com o  */
/* proprio ativo;                                                      */
/* medi, stdi = media e desvio p. do ativo i;                          */
/* Kp1,Kp2 = strike price das puts;                                    */
/* P1,P2 = premios das puts;                                           */
/* Si = cotacao do ativo i na data t=0.                                */
/* ms = taxa de retorno esperada do ativo objeto                       */
/* mp1, mp2 = taxas de retorno esperadas das puts                      */
/**********************************************************************/
{
        double Ex, E2x, E3x,E1,E2x1,Ex1,E2x2,Ex2,resp,K,f;

        K=dmin(Kp1,Kp2);
        f=(log(K/Si)-medi)/stdi;
        E3x=0.5*exp(3.0*medi+4.5*stdi*stdi)*(1.0+erff((f-3.0*stdi)/sqrt(2.0)));
        E2x=0.5*exp(2.0*medi+2.0*stdi*stdi)*(1.0+erff((f-2.0*stdi)/sqrt(2.0)));
        Ex=0.5*exp(medi+0.5*stdi*stdi)*(1.0+erff((f-stdi)/sqrt(2.0)));
        E1=0.5*(1.0+erff(f));
        f=(log(Kp1/Si)-medi)/stdi);
        E2x1=0.5*exp(2.0*medi+2.0*stdi*stdi)*(1.0+erff((f-2.0*stdi)/sqrt(2.0)));
        Ex1=0.5*exp(medi+0.5*stdi*stdi)*(1.0+erff((f-stdi)/sqrt(2.0)));
        f=(log(Kp2/Si)-medi)/stdi);
        E2x2=0.5*exp(2.0*medi+2.0*stdi*stdi)*(1.0+erff((f-2.0*stdi)/sqrt(2.0)));
```

```
        Ex2=0.5*exp(medi+0.5*stdi*stdi)*(1.0+erff((f-stdi)/sqrt(2.0)));
        resp=(Si*Si*E3x-Si*E2x*(Kp1+Kp2)+Kp1*Kp2*Ex)/(P1*P2);
        resp-=ms*(Si*Si*E2x-Si*Ex*(Kp1+Kp2)+Kp1*Kp2*E1)/(P1*P2);
        resp-=mp1*(-Si*E2x2+Kp2*Ex2)/P2;
        resp-=mp2*(-Si*E2x1+Kp1*Ex1)/P1;
        resp+=2.0*ms*mp2*mp1;
        return(resp);
}


double m_ricipi(double medi, double stdi, double Si, double Kci, double Kpi,
                double Ci, double Pi, double ms, double mc, double mp)
/**************************************************************************/
/* coskewnes entre a taxa de retorno da put e da call sobre o i-ésimo ativo*/
/* com o proprio ativo;                                                   */
/* medi, stdi = media e desvio p. do ativo i;                             */
/* Kci,Kpi = strike price da call e da put;                               */
/* Ci,Pi = premios da call e da put;                                      */
/* Si = cotacao do ativo i na data t=0.                                   */
/* ms = taxa de retorno esperada do ativo objeto                          */
/* mc, mp = taxas de retorno esperadas da call e da put                   */
/**************************************************************************/
{
        double Ex, E2x, E3x,E1,resp,fc,fp;

        fc=(log(Kci/Si)-medi)/stdi;
        E2xc=0.5*exp(2.0*medi+2.0*stdi*stdi)*
             (1.0-erff((fc-2.0*stdi)/sqrt(2.0));
        Exc=0.5*exp(medi+0.5*stdi*stdi)*
             (1.0-erff((fc-2.0*stdi)/sqrt(2.0)));
        fp=(log(Kpi/Si)-medi)/stdi;
        E2xp=0.5*exp(2.0*medi+2.0*stdi*stdi)*
             (1.0+erff((fp-2.0*stdi)/sqrt(2.0));
        Exp=0.5*exp(medi+0.5*stdi*stdi)*
             (1.0+erff((fp-stdi)/sqrt(2.0)));
        resp=-mc*(-Si*E2xp+Kpi*Exp)/Pi-mp*(Si*E2xc-Kci*Exc)/Ci;
        if (Kci < Kpi){
                E3x=0.5*exp(3.0*medi+4.5*stdi*stdi)*
                        (erff((fp-3.0*stdi)/sqrt(2.0))-
                         erff((fc-3.0*stdi)/sqrt(2.0)));
                E2x=0.5*exp(2.0*medi+2.0*stdi*stdi)*
                        (erff((fp-2.0*stdi)/sqrt(2.0))-
                         erff((fc-2.0*stdi)/sqrt(2.0)));
                Ex=0.5*exp(medi+0.5*stdi*stdi)*
                       (erff((fp-stdi)/sqrt(2.0))-
                        erff((fc-stdi)/sqrt(2.0)));
```

```
                E1=0.5*(erff(fp)-erff(fc));
                resp+=(-Si*Si*E3x+Si*E2x*(Kci+Kpi)-Kci*Kpi*Ex)/(Ci*Pi);
                resp-=ms*(-Si*Si*E2x+Si*Ex*(Kci+Kpi)-Kci*Kpi*E1)/(Ci*Pi);
                return(resp);
        }
        resp+=2.0*ms*mc*mp;
        return(resp);
}


double m_cicici(double medi, double stdi, double Si, double Kc1, double Kc2,
                double Kc3, double C1, double C2, double C3, double mc1,
                double mc2, double mc3)
/**********************************************************************/
/* coskewnes entre a taxa de retorno das calls sobre o i-ésimo ativo      */
/* medi, stdi = media e desvio p. do ativo i                              */
/* Kc1,Kc2,Kc3 = strike price das calls                                   */
/* C1,C2,C3 = premios das calls                                           */
/* Si = cotacao do ativo i na data t=0                                    */
/* mc1, mc2, mc3 = taxas de retorno esperadas das calls                   */
/**********************************************************************/
{
        double Ex, E2x, E3x,E1,resp,K,f,
               E2x12,Ex12,E112, E2x13,Ex13,E113, E2x23,Ex23,E123;

        K=dmax(Kc1,Kc2);
        K=dmax(K,Kc3);
        f=(log(K/Si)-medi)/stdi);
        E3x=0.5*exp(3.0*medi+4.5*stdi*stdi)*(1.0-erff((f-3.0*stdi)/sqrt(2.0)));
        E2x=0.5*exp(2.0*medi+2.0*stdi*stdi)*(1.0-erff((f-2.0*stdi)/sqrt(2.0)));
        Ex=0.5*exp(medi+0.5*stdi*stdi)*(1.0-erff((f-stdi)/sqrt(2.0)));
        E1=0.5*(1.0-erff(f));

        K=dmax(Kc1,Kc2);
        f=(log(K/Si)-medi)/stdi);
        E2x12=0.5*exp(2.0*medi+2.0*stdi*stdi)*(1.0-erff((f-2.0*stdi)/sqrt(2.0)));
        Ex12=0.5*exp(medi+0.5*stdi*stdi)*(1.0-erff((f-stdi)/sqrt(2.0)));
        E112=0.5*(1.0-erff(f));

        K=dmax(Kc1,Kc3);
        f=(log(K/Si)-medi)/stdi);
        E2x13=0.5*exp(2.0*medi+2.0*stdi*stdi)*(1.0-erff((f-2.0*stdi)/sqrt(2.0)));
        Ex13=0.5*exp(medi+0.5*stdi*stdi)*(1.0-erff((f-stdi)/sqrt(2.0)));
        E113=0.5*(1.0-erff(f));

        K=dmax(Kc3,Kc2);
```

```
        f=(log(K/Si)-medi)/stdi);
        E2x23=0.5*exp(2.0*medi+2.0*stdi*stdi)*(1.0-erff((f-2.0*stdi)/sqrt(2.0)));
        Ex23=0.5*exp(medi+0.5*stdi*stdi)*(1.0-erff((f-stdi)/sqrt(2.0)));
        E123=0.5*(1.0-erff(f));

        resp=(Si*Si*Si*E3x-Si*Si*E2x*(Kc1+Kc2+Kc3)
              +Si*Ex*(Kc1*Kc2+Kc1*Kc3+Kc2*Kc3)-Kc1*Kc2*Kc3*E1)/(C1*C2*C3);
        resp-=mc1*(Si*Si*E2x23-Si*Ex23*(Kc2+Kc3)+Kc2*Kc3*E123)/(C3*C2);
        resp-=mc2*(Si*Si*E2x13-Si*Ex13*(Kc1+Kc3)+Kc1*Kc3*E113)/(C3*C1);
        resp-=mc3*(Si*Si*E2x12-Si*Ex12*(Kc2+Kc1)+Kc2*Kc1*E112)/(C1*C2);
        resp+=2.0*mc1*mc2*mc3;
        return(resp);
}


double m_pipipi(double medi, double stdi, double Si, double Kp1, double Kp2,
                double Kp3, double P1, double P2, double P3, double mp1,
                double mp2, double mp3)
/***********************************************************************/
/* coskewnes entre a taxa de retorno das puts sobre o i-ésimo ativo      */
/* medi, stdi = media e desvio p. do ativo i                             */
/* Kp1,Kp2,Kp3 = strike price das puts                                   */
/* C1,C2,C3 = premios das puts                                           */
/* Si = cotacao do ativo i na data t=0                                   */
/* mp1, mp2, mp3 = taxas de retorno esperadas das puts                   */
/***********************************************************************/
{
        double Ex, E2x, E3x,E1,resp,K,f,
               E2x12,Ex12,E112, E2x13,Ex13,E113, E2x23,Ex23,E123;

        K=dmin(Kp1,Kp2);
        K=dmin(K,Kp3);
        f=(log(K/Si)-medi)/stdi);
        E3x=0.5*exp(3.0*medi+4.5*stdi*stdi)*(1.0+erff((f-3.0*stdi)/sqrt(2.0)));
        E2x=0.5*exp(2.0*medi+2.0*stdi*stdi)*(1.0+erff((f-2.0*stdi)/sqrt(2.0)));
        Ex=0.5*exp(medi+0.5*stdi*stdi)*(1.0+erff((f-stdi)/sqrt(2.0)));
        E1=0.5*(1.0+erff(f));

        K=dmin(Kp1,Kp2);
        f=(log(K/Si)-medi)/stdi);
        E2x12=0.5*exp(2.0*medi+2.0*stdi*stdi)*(1.0+erff((f-2.0*stdi)/sqrt(2.0)));
        Ex12=0.5*exp(medi+0.5*stdi*stdi)*(1.0+erff((f-stdi)/sqrt(2.0)));
        E112=0.5*(1.0+erff(f));

        K=dmin(Kp1,Kp3);
        f=(log(K/Si)-medi)/stdi);
```

```
        E2x13=0.5*exp(2.0*medi+2.0*stdi*stdi)*(1.0+erff((f-2.0*stdi)/sqrt(2.0)));
        Ex13=0.5*exp(medi+0.5*stdi*stdi)*(1.0+erff((f-stdi)/sqrt(2.0)));
        E113=0.5*(1.0+erff(f));

        K=dmin(Kp3,Kp2);
        f=(log(K/Si)-medi)/stdi);
        E2x=0.5*exp(2.0*medi+2.0*stdi*stdi)*(1.0+erff((f-2.0*stdi)/sqrt(2.0)));
        Ex=0.5*exp(medi+0.5*stdi*stdi)*(1.0+erff((f-stdi)/sqrt(2.0)));
        E1=0.5*(1.0+erff(f));

        resp=-(Si*Si*Si*E3x-Si*Si*E2x*(Kp1+Kp2+Kp3)
              +Si*Ex*(Kp1*Kp2+Kp1*Kp3+Kp2*Kp3)-Kp1*Kp2*Kp3*E1)/(P1*P2*P3);
        resp-=mp1*(Si*Si*E2x23-Si*Ex23*(Kp2+Kp3)+Kp2*Kp3*E123)/(P3*P2);
        resp-=mp2*(Si*Si*E2x13-Si*Ex13*(Kp1+Kp3)+Kp1*Kp3*E113)/(P3*P1);
        resp-=mp3*(Si*Si*E2x12-Si*Ex12*(Kp2+Kp1)+Kp2*Kp1*E112)/(P1*P2);
        resp+=2.0*mp1*mp2*mp3;
        return(resp);

}


double m_cicipi(double medi, double stdi, double Si, double Kc1, double Kc2,
                double Kpi, double C1, double C2, double Pi, double mc1,
                double mc2, double mp)
/****************************************************************************/
/* coskewnes entre a taxa de retorno das calls e put sobre o i-ésimo ativo */
/* medi, stdi = media e desvio p. do ativo i                                */
/* Kc1,Kc2 = strike price das calls                                         */
/* Kpi = strike da put                                                      */
/* C1,C2,Pi = premios das calss e da put                                    */
/* Si = cotacao do ativo i na data t=0                                      */
/****************************************************************************/
{
        double Ex, E2x, E3x,E1,resp,K,fc,fp;


        resp=0.0;
        fp=(log(Kpi/Si)-medi)/(sqrt(2.0)*stdi);
        K=dmax(Kc1,Kc2);
        if (K < Kpi){
                fc=(log(K/Si)-medi)/(sqrt(2.0)*stdi);
                E3x=0.5*exp(3.0*medi+4.5*stdi*stdi)*
                        (erff(fp-3.0*stdi/sqrt(2.0))-
                         erff(fc-3.0*stdi/sqrt(2.0)));
                E2x=0.5*exp(2.0*medi+2.0*stdi*stdi)*
                        (erff(fp-2.0*stdi/sqrt(2.0))-
```

```
                  erff(fc-2.0*stdi/sqrt(2.0)));
        Ex=0.5*exp(medi+0.5*stdi*stdi)*
                (erff(fp-stdi/sqrt(2.0))-
                erff(fc-stdi/sqrt(2.0)));
        E1=0.5*(erff(fp)-erff(fc));
        resp+=(-Si*Si*Si*E3x+Si*Si*E2x*(Kc1+Kc2+Kp)
             -Si*Ex*(Kc1*Kc2+Kc1*Kpi+Kc2*Kpi)+Kc1*Kc2*Kpi*E1)/(C1*C2*Pi);

    }
    if (Kc2 < Kpi){
        fc=(log(Kc2/Si)-medi)/(sqrt(2.0)*stdi);
        E2x=0.5*exp(2.0*medi+2.0*stdi*stdi)*
                (erff(fp-2.0*stdi/sqrt(2.0))-
                 erff(fc-2.0*stdi/sqrt(2.0)));
        Ex=0.5*exp(medi+0.5*stdi*stdi)*
                (erff(fp-stdi/sqrt(2.0))-
                erff(fc-stdi/sqrt(2.0)));
        E1=0.5*(erff(fp)-erff(fc));
        resp-=mc1*(-Si*Si*E2x+Si*Ex*(Kc2+Kpi)-E1*Kc2*Kpi)/(C2*Pi);

    }
    if (Kc1 < Kpi){
        fc=(log(Kc1/Si)-medi)/(sqrt(2.0)*stdi);
        fp=(log(Kpi/Si)-medi)/(sqrt(2.0)*stdi);
        E2x=0.5*exp(2.0*medi+2.0*stdi*stdi)*
                (erff(fp-2.0*stdi/sqrt(2.0))-
                 erff(fc-2.0*stdi/sqrt(2.0)));
        Ex=0.5*exp(medi+0.5*stdi*stdi)*
                (erff(fp-stdi/sqrt(2.0))-
                erff(fc-stdi/sqrt(2.0)));
        E1=0.5*(erff(fp)-erff(fc));
        resp-=mc2*(-Si*Si*E2x+Si*Ex*(Kc1+Kpi)-E1*Kc1*Kpi)/(C1*Pi);

    }
    K=dmax(Kc1,Kc2);
    fc=(log(K/Si)-medi)/(sqrt(2.0)*stdi);
    E2x=0.5*exp(2.0*medi+2.0*stdi*stdi)*
           (1.0-erff(fc-2.0*stdi/sqrt(2.0)));
    Ex=0.5*exp(medi+0.5*stdi*stdi)*
           (1.0-erff(fc-stdi/sqrt(2.0)));
    E1=0.5*(1.0-erff(fc));
    resp-=mp*(Si*Si*E2x-Si*Ex*(Kc1+Kc2)+E1*Kc1*Kc2)/(C1*C2);
    resp+=2.0*mc1*mc2*mp;
    return(resp);
}
```

```
double m_cipipi(double medi, double stdi, double Si, double Kci, double Kp1,
                double Kp2, double Ci, double P1, double P2, double mc,
                double mp1, double mp2)
/************************************************************************/
/* coskewnes entre a taxa de retorno da call e das puts sobre o i-ésimo    */
/* ativo;                                                                */
/* medi, stdi = media e desvio p. do ativo i;                            */
/* Kp1,Kp2 = strike price das puts;                                      */
/* Kci = strike da call;                                                 */
/* P1,P2,Ci = premios das puts e da call;                                */
/* Si = cotacao do ativo i na data t=0                                   */
/************************************************************************/
{
        double Ex, E2x, E3x,E1,resp,K,fc,fp;


        resp=0.0;
        K=dmin(Kp1,Kp2);
        fc=(log(Kci/Si)-medi)/(sqrt(2.0)*stdi);
        if (Kci < K){
                fp=(log(K/Si)-medi)/(sqrt(2.0)*stdi);
                E3x=0.5*exp(3.0*medi+4.5*stdi*stdi)*
                        (erff(fp-3.0*stdi/sqrt(2.0))-
                         erff(fc-3.0*stdi/sqrt(2.0)));
                E2x=0.5*exp(2.0*medi+2.0*stdi*stdi)*
                        (erff(fp-2.0*stdi/sqrt(2.0))-
                         erff(fc-2.0*stdi/sqrt(2.0)));
                Ex=0.5*exp(medi+0.5*stdi*stdi)*
                        (erff(fp-stdi/sqrt(2.0))-
                        erff(fc-stdi/sqrt(2.0)));
                E1=0.5*(erff(fp)-erff(fc));
                resp+=(Si*Si*Si*E3x-Si*Si*E2x*(Kp1+Kp2+Kci)
                     +Si*Ex*(Kp1*Kp2+Kci*Kp1+Kp2*Kci)-Kp1*Kp2*Kci*E1)/(P1*P2*Ci);

        }
        fc=(log(Kci/Si)-medi)/(sqrt(2.0)*stdi);
        if (Kci < Kp2){
                fp=(log(Kp2/Si)-medi)/(sqrt(2.0)*stdi);
                E2x=0.5*exp(2.0*medi+2.0*stdi*stdi)*
                        (erff(fp-2.0*stdi/sqrt(2.0))-
                         erff(fc-2.0*stdi/sqrt(2.0)));
                Ex=0.5*exp(medi+0.5*stdi*stdi)*
                        (erff(fp-stdi/sqrt(2.0))-
                        erff(fc-stdi/sqrt(2.0)));
                E1=0.5*(erff(fp)-erff(fc));
```

```
                resp-=mp1*(-Si*Si*E2x+Si*Ex*(Kp2+Kci)-E1*Kp2*Kci)/(Ci*P2);

        }
        if (Kci < Kp1){
                fp=(log(Kp1/Si)-medi)/(sqrt(2.0)*stdi);
                E2x=0.5*exp(2.0*medi+2.0*stdi*stdi)*
                        (erff(fp-2.0*stdi/sqrt(2.0))-
                         erff(fc-2.0*stdi/sqrt(2.0)));
                Ex=0.5*exp(medi+0.5*stdi*stdi)*
                        (erff(fp-stdi/sqrt(2.0))-
                        erff(fc-stdi/sqrt(2.0)));
                E1=0.5*(erff(fp)-erff(fc));
                resp-=mp2*(-Si*Si*E2x+Si*Ex*(Kci+Kp1)-E1*Kci*Kp1)/(Ci*P1);

        }
        K=dmin(Kp1,Kp2);
        fp=(log(K/Si)-medi)/(sqrt(2.0)*stdi);
        E2x=0.5*exp(2.0*medi+2.0*stdi*stdi)*
               (1.0+erff(fp-2.0*stdi/sqrt(2.0)));
        Ex=0.5*exp(medi+0.5*stdi*stdi)*
               (1.0+erff(fp-stdi/sqrt(2.0)));
        E1=0.5*(1.0+erff(fp));
        resp-=mc*(Si*Si*E2x-Si*Ex*(Kp1+Kp2)+E1*Kp1*Kp2)/(P1*P2);
        resp+=2.0*mp1*mp2*mc;
        return(resp);

}


double m_ririrj(double medi, double medj, double stdi, double stdj,
                double corij)
/***************************************************************************/
/* coskewness entre as taxas de retorno de 2 ativos básicos                */
/* corij = coef. de correlacao entre os ativos i e j.                      */
/***************************************************************************/
{
        double Exy, Ex, E2xy, Ey, E2x, resp;

        E2xy=exp(2.0*medi+medj+2.0*stdi*stdi+0.5*stdj*stdj+
                  2.0*stdi*stdj*corij);
        Exy=exp(medi+medj+0.5*(stdi*stdi+stdj*stdj)+stdi*stdj*corij);
        E2x=exp(2.0*(medi+stdi*stdi));
        Ex=exp(medi+0.5*stdi*stdi);
        Ey=exp(medj+0.5*stdj*stdj);
        resp=E2xy-2.0*Exy*Ex-Ey*E2x+2.0*Ex*Ey*Ex;
        return(resp);
```

```
}


double m_riricj(double medi, double medj, double stdi, double stdj,
                double corij, double Sj, double Kcj, double Cj, double ms,
                double mc)
/*************************************************************************/
/* coskewness entre uma call sobre um ativo j e ativo básico i           */
/* corij = coef. de correlacao entre os ativos i e j.                    */
/* Sj = cotacao do ativo j em t=0;                                       */
/* Cj = premio da call                                                   */
/* corij = correl. entre os ativos i e j                                 */
/* Kcj = strike da call                                                  */
/* ms = taxa de retorno esperada do ativo objeto                         */
/* mc = taxa de retorno esperada da call                                 */
/*************************************************************************/
{
        double Exy, E2xy, Ex, Ey, E1, E2x, E2xi, Exi, R, resp;

        R=(log(Kcj/Sj)-medj)/stdj;
        Exy=0.5*exp(medi+medj+0.5*(stdi*stdi+stdj*stdj)
                    +stdi*stdj*corij)*
            (1.0-erff((R-stdi*corij-stdj)/sqrt(2.0)));
        E2xy=0.5*exp(2.0*medi+medj+2.0*stdi*stdi+0.5*stdj*stdj+
                     2.0*stdi*stdj*corij)*
             (1.0-erff((R-2.0*stdi*corij-stdj)/sqrt(2.0)));
        E2x=0.5*exp(2.0*medi+2.0*stdi*stdi)*
             (1.0-erff((R-2.0*stdi*corij)/sqrt(2.0)));
        Ex=0.5*exp(medi+0.5*stdi*stdi)*
            (1.0-erff((R-stdi*corij)/sqrt(2.0)));
        Ey=0.5*exp(medj+0.5*stdj*stdj)*
            (1.0-erff((R-stdj)/sqrt(2.0)));
        E1=0.5*(1.0-erff(R/sqrt(2.0)));
        E2xi=exp(2.0*medi+2.0*stdi*stdi);
        Exi=exp(medi+0.5*stdi*stdi);
        resp=(Sj*E2xy-E2x*Kcj)/Cj-2.0*ms*(Sj*Exy-Kcj*Ex)/Cj-mc*E2xi
              +2.0*ms*mc*Exi+ms*ms*(Sj*Ey-Kcj*E1)/Cj-ms*ms*mc;
        return(resp);
}


double m_riripj(double medi, double medj, double stdi, double stdj,
                double corij, double Sj, double Kpj, double Pj,
                double ms, double mp)
/*************************************************************************/
/* coskewness entre uma put sobre um ativo j e o ativo básico i          */
```

```
/* corij = coef. de correlacao entre os ativos i e j.                  */
/* Sj = cotacao do ativo j em t=0;                                      */
/* Pj = premio da put                                                   */
/* Kpj = strike da put                                                  */
/***********************************************************************/
{
        double Exy, E2xy, Ex, E2x, Ey, E1, E2xi, Exi, R, resp;

        R=(log(Kpj/Sj)-medj)/stdj;
        Exy=0.5*exp(medi+medj+0.5*(stdi*stdi+stdj*stdj)
                    +stdi*stdj*corij)*
            (1.0+erff((R-stdi*corij-stdj)/sqrt(2.0)));
        E2xy=0.5*exp(2.0*medi+medj+2.0*stdi*stdi+0.5*stdj*stdj+
                     2.0*stdi*stdj*corij)*
             (1.0+erff((R-2.0*stdi*corij-stdj)/sqrt(2.0)));
        E2x=0.5*exp(2.0*medi+2.0*stdi*stdi)*
             (1.0+erff((R-2.0*stdi*corij-stdj)/sqrt(2.0)));
        Ex=0.5*exp(medi+0.5*stdi*stdi)*
           (1.0+erff((R-stdi*corij)/sqrt(2.0)));
        Ey=0.5*exp(medj+0.5*stdj*stdj)*
           (1.0+erff((R-stdj)/sqrt(2.0)));
        E1=0.5*(1.0+erff(R/sqrt(2.0)));
        E2xi=exp(2.0*medi+2.0*stdi*stdi);
        Exi=exp(medi+0.5*stdi*stdi);
        resp=(-Sj*E2xy+E2x*Kpj)/Pj-2.0*ms*(-Sj*Exy+Kcj*Ex)/Pj;
        resp+=-mp*E2xi+2.0*ms*mp*Exi+ms*ms*(-Sj*Ey+Kcj*E1)/Pj-ms*ms*mp;
        return(resp);
}


double m_ricirj(double medi, double medj, double stdi, double stdj,
                double corij, double Si, double Kci, double Ci,
                double msi, double mc, double msj)
/***********************************************************************/
/* coskewness entre uma call sobre um ativo i e os ativos básicos i e j  */
/* corij = coef. de correlacao entre os ativos i e j.                  */
/* S = cotacao do ativo j em t=0;                                       */
/* C = premio da call                                                   */
/* /* corij = correl. entre os ativos i e j                              */
K = strike da call                                                      */
/***********************************************************************/
{
        double Exy, E2xy, Ex, E2x, Ey, E1, Exyi, Exi, Eyi, R, resp;

        R=(log(Kci/Si)-medi)/stdi;
        Exy=0.5*exp(medi+medj+0.5*(stdi*stdi+stdj*stdj)
```

```
                    +stdi*stdj*corij)*
                (1.0-erff((R-stdi-corij*stdj)/sqrt(2.0)));
        E2xy=0.5*exp(2.0*medi+medj+2.0*stdi*stdi+0.5*stdj*stdj+
                     2.0*stdi*stdj*corij)*
                 (1.0-erff((R-2.0*stdi-corij*stdj)/sqrt(2.0)));
        Ex=0.5*exp(medi+0.5*stdi*stdi)*
           (1.0-erff((R-stdi)/sqrt(2.0)));
        Ey=0.5*exp(medj+0.5*stdj*stdj)*
           (1.0-erff((R-stdj*corij)/sqrt(2.0)));
        E1=0.5*(1.0-erff(R/sqrt(2.0)));
        E2x=0.5*exp(2.0*(medi+stdi*stdi))*
            (1.0-erff((R-2.0*stdi)/sqrt(2.0)));
        Exyi=exp(medi+medj+0.5*(stdi*stdi+stdj*stdj)+corij*stdi*stdj);
        Exi=exp(medi+0.5*stdi*stdi);
        Eyi=exp(medj+0.5*stdj*stdj);
        resp=(Si*E2xy-Kci*Exy)/Ci-msi*(Si*Exy-Kci*Ey)/Ci-mc*Exyi;
        resp+=-msj*(Si*E2x-Kci*Ex)/Ci+msi*mc*Eyi+mc*msj*Exi;
        resp+=msi*msj*(Si*Ex-Kci*E1)/Ci-msi*msj*mc;
        return(resp);
}


double m_ripirj(double medi, double medj, double stdi, double stdj,
                double corij, double Si, double Kpi, double Pi,
                double msi, double mp, double msj)
/*************************************************************************/
/* coskewness entre uma put sobre o i-ésimo ativo e os ativos básicos i e j*/
/* corij = coef. de correlacao entre os ativos i e j.                    */
/* Si = cotacao do ativo j em t=0;                                       */
/* Pi = premio da put                                                    */
/* corij = correl. entre os ativos i e j                                 */
/* Kpi = strike da put                                                   */
/*************************************************************************/
{
        double Exy, E2xy, Ex, E2x, Ey, E1, Exyi, Exi, Eyi, R, resp;

        R=(log(Kpi/Si)-medi)/stdi;
        Exy=0.5*exp(medi+medj+0.5*(stdi*stdi+stdj*stdj)
            +stdi*stdj*corij)*
            (1.0+erff((R-stdi-corij*stdj)/sqrt(2.0)));
        E2xy=0.5*exp(2.0*medi+medj+2.0*stdi*stdi+0.5*stdj*stdj+
                     2.0*stdi*stdj*corij)*
             (1.0+erff((R-2.0*stdi-corij*stdj)/sqrt(2.0)));
        Ex=0.5*exp(medi+0.5*stdi*stdi)*
           (1.0+erff((R-stdi)/sqrt(2.0)));
        Ey=0.5*exp(medj+0.5*stdj*stdj)*
```

```
           (1.0+erff((R-stdj*corij)/sqrt(2.0)));
        E1=0.5*(1.0+erff(R/sqrt(2.0)));
        E2x=0.5*exp(2.0*(medi+stdi*stdi))*
            (1.0+erff((R-2.0*stdi)/sqrt(2.0)));
        Exyi=exp(medi+medj+0.5*(stdi*stdi+stdj*stdj)+corij*stdi*stdj);
        Exi=exp(medi+0.5*stdi*stdi);
        Eyi=exp(medj+0.5*stdj*stdj);
        resp=(-Si*E2xy+Kpi*Exy)/Pi-msi*(-Si*Exy+Kpi*Ey)/Pi-mp*Exyi;
        resp+=-msj*(-Si*E2x+Kpi*Ex)/Pi+msi*mp*Eyi+mp*msj*Exi;
        resp+=msi*msj*(-Si*Ex+Kpi*E1)/Pi-msi*msj*mp;
        return(resp);
}


double m_cicirj(double medi, double medj, double stdi, double stdj,
                double corij, double Si, double Kc1, double Kc2, double C1,
                double C2, double mc1, double mc2, double ms)
/************************************************************************/
/* coskewness entre calls sobre um ativo i e o ativo básico j          */
/* corij = coef. de correlacao entre os ativos i e j.                   */
/* Si = cotacao do ativo i em t=0;                                      */
/* C1,C2 = premio das calls                                             */
/* Kc1,Kc2 = strike das calls                                           */
/************************************************************************/
{
        double E2x, Exy, E2xy, Ex, Ey, E1, R, Exy2b, Ey2b, Ex2b, E12b,
               Exy1b, Ey1b, Ex1b, E11b, Eyi, resp,K;

        K=dmax(Kc1,Kc2);
        R=(log(K/Si)-medi)/stdi;
        Exy=0.5*exp(medi+medj+0.5*(stdi*stdi+stdj*stdj)+
                    stdi*stdj*corij)*
            (1.0-erff((R-stdi-corij*stdj)/sqrt(2.0)));
        E2xy=0.5*exp(2.0*medi+medj+2.0*stdi*stdi+0.5*stdj*stdj+
                     2.0*stdi*stdj*corij)*
             (1.0-erff((R-2.0*stdi-corij*stdj)/sqrt(2.0)));
        Ex=0.5*exp(medi+0.5*stdi*stdi)*
           (1.0-erff((R-stdi)/sqrt(2.0)));
        Ey=0.5*exp(medj+0.5*stdj*stdj)*
           (1.0-erff((R-stdj*corij)/sqrt(2.0)));
        E1=0.5*(1.0-erff(R/sqrt(2.0)));
        E2x=0.5*exp(2.0*(medi+stdi*stdi))*
            (1.0-erff((R-2.0*stdi)/sqrt(2.0)));
        R=(log(Kc2/Si)-medi)/stdi;
        Exy2b=0.5*exp(medi+medj+0.5*(stdi*stdi+stdj*stdj)+
                    stdi*stdj*corij)*
```

```
            (1.0-erff((R-stdi-corij*stdj)/sqrt(2.0)));
        Ex2b=0.5*exp(medi+0.5*stdi*stdi)*
           (1.0-erff((R-stdi)/sqrt(2.0)));
        Ey2b=0.5*exp(medj+0.5*stdj*stdj)*
           (1.0-erff((R-stdj*corij)/sqrt(2.0)));
        E12b=0.5*(1.0-erff(R/sqrt(2.0)));
        R=(log(Kc1/Si)-medi)/stdi;
        Exy1b=0.5*exp(medi+medj+0.5*(stdi*stdi+stdj*stdj)+
                    stdi*stdj*corij)*
            (1.0-erff((R-stdi-corij*stdj)/sqrt(2.0)));
        Ex1b=0.5*exp(medi+0.5*stdi*stdi)*
           (1.0-erff((R-stdi)/sqrt(2.0)));
        Ey1b=0.5*exp(medj+0.5*stdj*stdj)*
           (1.0-erff((R-stdj*corij)/sqrt(2.0)));
        E11b=0.5*(1.0-erff(R/sqrt(2.0)));
        Eyi=exp(medi+0.5*stdi*stdi);
        resp=(Si*Si*E2xy-Si*Exy*(Kc1+Kc2)+Kc1*Kc2*Ey)/(C1*C2);
        resp+=-mc1*(Si*Exy2b-Kc2*Ey2b)/C2-mc2*(Si*Exy1b-Kc1*Ey1b)/C1;
        resp+=-ms*(Si*Si*E2x-Si*Ex*(Kc1+Kc2)+Kc1*Kc2*E1);
        resp+=mc1*mc2*Eyi+mc1*ms*(Si*Ex2b-Kc2*E12b);
        resp+=mc2*ms*(Si*Ex1b-Kc1*E11b)/C1-ms*mc1*mc2;
        return(resp);
}


double m_cipirj(double medi, double medj, double stdi, double stdj,
                double corij, double Si, double Kci, double Kpi, double Ci,
                double Pi, double mc, double mp, double ms)
/*********************************************************************/
/* coskewness entre call e put sobre o ativo i e o ativo básico j        */
/* corij = coef. de correlacao entre os ativos i e j.                    */
/* Si = cotacao do ativo i em t=0;                                       */
/* Ci,Pi = premio da call e da put                                       */
/* Kci,Kpi = strike das calls                                            */
/*********************************************************************/
{
        double Exy, E2xy, Ex, E2x, Ey, E1, Exypb, Eypb, Expb, E1pb,
               Exycb, Eycb, Excb, E1cb, Eyi, Rc, Rp, R, resp;


        R=(log(Kpi/Si)-medi)/stdi;
        Exypb=0.5*exp(medi+medj+0.5*(stdi*stdi+stdj*stdj)
                    +stdi*stdj*corij)*
            (1.0+erff((R-stdi-corij*stdj)/sqrt(2.0)));
        Expb=0.5*exp(medi+0.5*stdi*stdi)*
             (1.0+erff((R-stdi)/sqrt(2.0)));
```

```
        Eypb=0.5*exp(medj+0.5*stdj*stdj)*
             (1.0+erff((R-stdj*corij)/sqrt(2.0)));
        E1pb=0.5*(1.0+erff(R/sqrt(2.0)));

        R=(log(Kci/Si)-medi)/stdi;
        Exycb=0.5*exp(medi+medj+0.5*(stdi*stdi+stdj*stdj)
                    +stdi*stdj*corij)*
            (1.0-erff((R-stdi-corij*stdj)/sqrt(2.0)));
        Excb=0.5*exp(medi+0.5*stdi*stdi)*
             (1.0-erff((R-stdi)/sqrt(2.0)));
        Eycb=0.5*exp(medj+0.5*stdj*stdj)*
             (1.0-erff((R-stdj*corij)/sqrt(2.0)));
        E1cb=0.5*(1.0-erff(R/sqrt(2.0)));
        Eyi=exp(medi+0.5*stdi*stdi);

        resp=-mc*(-Si*Exypb+Kpi*Eypb)/Pi-mp*(Si*Exycb-Kci*Eycb)/Ci;
        resp+=mc*mp*Eyi+mc*ms*(-Si*Expb+Kpi*E1pb)/Pi
              +mp*ms*(Si*Excb-Kci*E1cb)/Ci - mc*mp*ms;

        if (Kci < Kpi){
               Rc=(log(Kci/Si)-medi)/stdi;
               Rp=(log(Kpi/Si)-medi)/stdi;
               Exy=0.5*exp(medi+medj+0.5*(stdi*stdi+stdj*stdj)+
                           stdi*stdj*corij)*
                    (erff((Rp-stdi-corij*stdj)/sqrt(2.0))-
                     erff((Rc-stdi-corij*stdj)/sqrt(2.0)));
               E2xy=0.5*exp(2.0*medi+medj+2.0*stdi*stdi+0.5*stdj*stdj+
                           2.0*stdi*stdj*corij)*
                    (erff((Rp-2.0*stdi-corij*stdj)/sqrt(2.0))-
                     erff((Rc-2.0*stdi-corij*stdj)/sqrt(2.0)));
               Ex=0.5*exp(medi+0.5*stdi*stdi)*
                   (erff((Rp-stdi)/sqrt(2.0))-
                    erff((Rc-stdi)/sqrt(2.0)));
               Ey=0.5*exp(medj+0.5*stdj*stdj)*
                   (erff((Rp-stdj*corij)/sqrt(2.0))-
                    erff((Rc-stdj*corij)/sqrt(2.0)));
               E1=0.5*(erff(Rp/sqrt(2.0))-erff(Rc/sqrt(2.0)));
               E2x=0.5*exp(2.0*(medi+stdi*stdi))*
                   (erff((Rp-2.0*stdi)/sqrt(2.0))-
                    erff((Rc-2.0*stdi)/sqrt(2.0)));
               resp+=(-Si*Si*E2xy+Si*Exy*(Kci+Kpi)-Kci*Kpi*Ey)/(Ci*Pi);
               resp+=-ms*(-Si*Si*E2x+Si*Ex*(Kci+Kpi)-Kci*Kpi*E1)/(Ci*Pi);

        }
        return(resp);
}
```

```
double m_pipirj(double medi, double medj, double stdi, double stdj,
                double corij, double Si, double Kp1, double Kp2,
                double P1, double P2, double mp1, double mp2, double ms)
/**************************************************************************/
/* coskewness entre puts sobre o ativo i e o ativo básico j               */
/* corij = coef. de correlacao entre os ativos i e j.                     */
/* Si = cotacao do ativo i em t=0;                                        */
/* P1,P2 = premio das puts                                                */
/* Kp1,Kp2 = strike das puts                                              */
/**************************************************************************/
{
        double K, Exy, E2xy, Ex, E2x, Ey, E1, R, resp;

        K=dmin(Kp1,Kp2);
        R=(log(K/Si)-medi)/stdi;
        Exy=0.5*exp(medi+medj+0.5*(stdi*stdi+stdj*stdj)+
                                       stdi*stdj*corij)*
            (1.0+erff((R-stdi-corij*stdj)/sqrt(2.0)));
        E2xy=0.5*exp(2.0*medi+medj+2.0*stdi*stdi+0.5*stdj*stdj+
                     2.0*stdi*stdj*corij)*
             (1.0+erff((R-2.0*stdi-corij*stdj)/sqrt(2.0)));
        Ex=0.5*exp(medi+0.5*stdi*stdi)*
             (1.0+erff((R-stdi)/sqrt(2.0)));
        Ey=0.5*exp(medj+0.5*stdj*stdj)*
             (1.0+erff((R-stdj*corij)/sqrt(2.0)));
        E1=0.5*(1.0+erff(R/sqrt(2.0)));
        E2x=0.5*exp(2.0*(medi+stdi*stdi))*
             (1.0+erff((R-2.0*stdi)/sqrt(2.0)));
        R=(log(Kp2/Si)-medi)/stdi;
        Exy2b=0.5*exp(medi+medj+0.5*(stdi*stdi+stdj*stdj)+
                     stdi*stdj*corij)*
            (1.0+erff((R-stdi-corij*stdj)/sqrt(2.0)));
        Ex2b=0.5*exp(medi+0.5*stdi*stdi)*
           (1.0+erff((R-stdi)/sqrt(2.0)));
        Ey2b=0.5*exp(medj+0.5*stdj*stdj)*
           (1.0+erff((R-stdj*corij)/sqrt(2.0)));
        E12b=0.5*(1.0+erff(R/sqrt(2.0)));
        R=(log(Kp1/Si)-medi)/stdi;
        Exy1b=0.5*exp(medi+medj+0.5*(stdi*stdi+stdj*stdj)+
                     stdi*stdj*corij)*
            (1.0+erff((R-stdi-corij*stdj)/sqrt(2.0)));
        Ex1b=0.5*exp(medi+0.5*stdi*stdi)*
           (1.0+erff((R-stdi)/sqrt(2.0)));
        Ey1b=0.5*exp(medj+0.5*stdj*stdj)*
```

```
            (1.0+erff((R-stdj*corij)/sqrt(2.0)));
        E11b=0.5*(1.0+erff(R/sqrt(2.0)));
        Eyi=exp(medi+0.5*stdi*stdi);
        resp=(Si*Si*E2xy-Si*Exy*(Kp1+Kp2)+Kp1*Kp2*Ey)/(P1*P2);
        resp+=-mp1*(-Si*Exy2b+Kp2*Ey2b)/P2-mp2*(-Si*Exy1b+Kp1*Ey1b)/P1;
        resp+=-ms*(Si*Si*E2x-Si*Ex*(Kp1+Kp2)+Kp1*Kp2*E1);
        resp+=mp1*mp2*Eyi+mp1*ms*(-Si*Ex2b+Kp2*E12b);
        resp+=mp2*ms*(-Si*Ex1b+Kp1*E11b)/P1-ms*mp1*mp2;

}



double m_rirjrk(double medi, double medj, double medk,
                double stdi, double stdj, double stdk,
                double corij, double corik, double corjk)
/*************************************************************************/
/* coskewness entre os ativos básicos i,j,k;                             */
/* corij = coef. de correlacao entre os ativos i e j.                    */
/*************************************************************************/
{
        double Exyz, Exy, Exz, Eyz, Ex, Ey, Ez,resp;

        Exyz=exp(medi+medj+medk+
                 0.5*(stdi*stdi+stdj*stdj+stdk*stdk)
                 +stdi*stdj*corij+stdi*stdk*corik
                 +stdj*stdk*corjk);
        Exy=exp(medi+medj+
                0.5*(stdi*stdi+stdj*stdj)+stdi*stdj*corij);
        Exz=exp(medi+medk+
                0.5*(stdi*stdi+stdk*stdk)+stdi*stdk*corik);
        Eyz=exp(medj+medk+
                0.5*(stdj*stdj+stdk*stdk)+stdj*stdk*corjk);
        Ex=exp(medi+0.5*stdi*stdi);
        Ey=exp(medj+0.5*stdj*stdj);
        Ez=exp(medk+0.5*stdk*stdk);
        resp=Exyz-Ex*Eyz-Ey*Exz-Ez*Exy+2.0*Ex*Ey*Ez;
        return(resp);
}


double m_rirjck(double medi, double medj, double medk,
                double stdi, double stdj, double stdk,
                double corij, double corik, double corjk,
                double Sk, double Kck, double Ck,
                double msi, double msj, double mc)
```

```
/*************************************************************************/
/* medi,medj,medk=media do ativo i,j e k;                                */
/* stdi,stdj,stdk=desvio do ativo i,j e k;                               */
/* Ck,Kck=premio e strike da call ;                                      */
/* Sk=cotacao do ativo básico k em t=0.                                  */
/* corij=correlacao entre i e j;                                         */
/*************************************************************************/
{
        double Exyz,Exy,Exz,Eyz,Ez,Ex,Ey,E1,Exyi,Eyi,Exi,resp,
               R,a,A,B;

        R=(log(Kck/Sk)-medk)/stdk;
        a=1.0-corjk*corjk;
        A=corik*corjk-corij;
        B=corij*corjk-corik;
        Exyz=exp(medi+medj+medk+
                 0.5*(stdi*stdi+stdj*stdj+stdk*stdk)+
                 stdi*stdj*corij+stdi*stdk*corik+
                 stdj*stdk*corjk)/(2.0*sqrt(2.0));
        Exyz*=(1.0-erff((1.0/sqrt(2.0))*
              (R-stdi*corik-stdk-stdj*corjk)));
        Exy=exp(medi+medj+0.5*(stdi*stdi+stdj*stdj)+
                  stdi*stdj*corij)/(2.0*sqrt(2.0));
        Exy*=(1.0-erff((1.0/sqrt(2.0))*
             (R-stdi*corik-stdj*corjk)));
        Exz=exp(medi+medk+0.5*(stdi*stdi+stdk*stdk)+
                  stdi*stdk*corik)/(2.0*sqrt(2.0));
        Exz*=(1.0-erff((1.0/sqrt(2.0))*(R-stdi*corik-stdk)));
        Eyz=exp(medj+medk+0.5*(stdj*stdj+stdk*stdk)+
                 stdj*stdk*corjk)/(2.0*sqrt(2.0));
        Eyz*=(1.0-erff((1.0/sqrt(2.0))*(R-stdk-stdj*corjk)));
        Ex=exp(medi+0.5*(stdi*stdi))/(2.0*sqrt(2.0));
        Ex*=(1.0-erff((1.0/sqrt(2.0))*(R-corik*stdi)));
        Ey=exp(medj+0.5*stdj*stdj)/(2.0*sqrt(2.0));
        Ey*=(1.0-erff((1.0/sqrt(2.0))*(R-stdj*corjk)));
        Ez=exp(medk+0.5*stdk*stdk)/(2.0*sqrt(2.0));
        Ez*=(1.0-erff((1.0/sqrt(2.0))*(R-stdk)));
        E1=(1.0-erff(R/sqrt(2.0)))/(2.0*sqrt(2.0));
        Exyi=exp(medi+medj+0.5*(stdi*stdi+stdj*stdj)+corij*stdi*stdj);
        Eyi=exp(medj+0.5*stdj*stdj);
        Exi=exp(medi+0.5*stdi*stdi);
        resp=(Sk*Exyz-Kck*Exy)/Ck-msi*(Sk*Eyz-Kck*Ey)/Ck;
        resp+=-msj*(Sk*Exz-Kck*Ex)-mc*Exyi+msi*msj*(Sk*Ez-Kck*E1)/Ck;
        resp+=msi*mc*Eyi+msj*mc*Exi-msi*msj*mc;
        return(resp);
}
```

```
double m_rirjpk(double medi, double medj, double medk,
                double stdi, double stdj, double stdk,
                double corij, double corik, double corjk,
                double Sk, double Kpk, double Pk,
                double msi, double msj, double mp)
/************************************************************************/
/* medi,medj,medk=media do ativo i,j e k;                               */
/* stdi,stdj,stdk=desvio do ativo i,j e k;                              */
/* Pk,Kpk=premio e strike da put ;                                      */
/* Sk=cotacao do ativo básico k em t=0.                                 */
/* corij=correlacao entre i e j;                                        */
/************************************************************************/
{
        double Exyz,Exy,Exz,Eyz,Ez,Ex,Ey,E1,Exyi,Exi,Eyi,resp,
               R,a,A,B;


        R=(log(Kpk/Sk)-medk)/stdk;
        a=1.0-corjk*corjk;
        A=corik*corjk-corij;
        B=corij*corjk-corik;
        Exyz=exp(medi+medj+medk+
                 0.5*(stdi*stdi+stdj*stdj+stdk*stdk)+
                 stdi*stdj*corij+stdi*stdk*corik+
                 stdj*stdk*corjk)/(2.0*sqrt(2.0));
        Exyz*=(1.0+erff((1.0/sqrt(2.0))*(R-stdi*corik-stdk
              -stdj*corjk)));
        Exy=exp(medi+medj+0.5*(stdi*stdi+stdj*stdj)+
                  stdi*stdj*corij)/(2.0*sqrt(2.0));
        Exy*=(1.0+erff((1.0/sqrt(2.0))*(R-stdi*corik-stdj*corjk)));
        Exz=exp(medi+medk+0.5*(stdi*stdi+stdk*stdk)+
                  stdi*stdk*corik)/(2.0*sqrt(2.0));
        Exz*=(1.0+erff((1.0/sqrt(2.0))*(R-stdi*corik-stdk)));
        Eyz=exp(medj+medk+0.5*(stdj*stdj+stdk*stdk)+
                 stdj*stdk*corjk)/(2.0*sqrt(2.0));
        Eyz*=(1.0+erff((1.0/sqrt(2.0))*(R-stdk-stdj*corjk)));
        Ex=exp(medi+0.5*(stdi*stdi))/(2.0*sqrt(2.0));
        Ex*=(1.0+erff((1.0/sqrt(2.0))*(R-stdi*corik)));
        Ey=exp(medj+0.5*stdj*stdj)/(2.0*sqrt(2.0));
        Ey*=(1.0+erff((1.0/sqrt(2.0))*(R-stdj*corjk)));
        Ez=exp(medk+0.5*stdk*stdk)/(2.0*sqrt(2.0));
        Ez*=(1.0+erff((1.0/sqrt(2.0))*(R-stdk)));
        E1=(1.0+erff(R/sqrt(2.0)))/(2.0*sqrt(2.0));
        Exyi=exp(medi+medj+0.5*(stdi*stdi+stdj*stdj)+corij*stdi*stdj);
```

```
        Eyi=exp(medj+0.5*stdj*stdj);
        Exi=exp(medi+0.5*stdi*stdi);
        resp=(-Sk*Exyz+Kpk*Exy)/Pk-msi*(-Sk*Eyz+Kpk*Ey)/Pk;
        resp+=-msj*(-Sk*Exz+Kpk*Ex)-mp*Exyi+msi*msj*(-Sk*Ez+Kpk*E1)/Pk;
        resp+=msi*mp*Eyi+msj*mp*Exi-msi*msj*mp;
        return(resp);

}



/****************************************************************************/
/******     expressoes de assimetria que envolvem integrais duplas   ******/
/****************************************************************************/


double m_ricicj(double medi, double medj, double stdi, double stdj,
                double corij, double Si, double Sj, double Kci, double Kcj,
                double Ci, double Cj, double ms, double mci, double mcj)
/*****************************************************************************/
/* coskewness entre uma call sobre um ativo i, uma call sobre o ativo j    */
/* e o ativo básico i;                                                     */
/* corij = coef. de correlacao entre os ativos i e j.                      */
/* Si,Sj = cotacao do ativo i,j em t=0;                                    */
/* Ci, Cj = premio da call sobre i e j                                     */
/* Kci,Kcj = strikes das calls                                             */
/*****************************************************************************/
{
        double Exy, E2xy, Ex, E2x, Ey, Exyb, Exb, E2xa, Exa, Eyb, E1b,
               E1a, Exi, E1, Ri,Rj, resp, L;

        Ri=(log(Kci/Si)-medi)/stdi;
        Rj=(log(Kcj/Sj)-medj)/stdj;
        E2xy=(0.5/sqrt(2.0*pi))*
             exp(2.0*medi+medj+2.0*stdi*stdi
                 +0.5*stdj*stdj+2.0*stdi*stdj*corij);
        L=2.0*stdi+stdj*corij;
        E2xy*=I_skew(9,Ri,sqrt(2.0),1,-1.0,0.0,stdi,stdj,0.0,corij,0.0,
                    0.0,Ri,Rj,0.0,L);
        Exy=(0.5/sqrt(2.0*pi))*
            exp(medi+medj+0.5*(stdi*stdi+stdj*stdj)+
                              stdi*stdj*corij);
        L=stdi+stdj*corij;
        Exy*=I_skew(9,Ri,sqrt(2.0),1,-1.0,0.0,stdi,stdj,0.0,corij,0.0,
                    0.0,Ri,Rj,0.0,L);
        Ex=(0.5/sqrt(2.0*pi))*exp(medi+0.5*stdi*stdi);
```

```
        L=stdi;
        Ex*=I_skew(10,Ri,sqrt(2.0),1,-1.0,0.0,stdi,stdj,0.0,corij,0.0,
                    0.0,Ri,Rj,0.0,L);
        Ey=(0.5/sqrt(2.0*pi))*exp(medj+0.5*stdj*stdj);
        L=stdj*corij;
        Ey*=I_skew(9,Ri,sqrt(2.0),1,-1.0,0.0,stdi,stdj,0.0,corij,0.0,
                      0.0,Ri,Rj,0.0,L);
        E1=(0.5/sqrt(2.0*pi));
        L=0.0;
        E1*=I_skew(10,Ri,sqrt(2.0),1,-1.0,0.0,stdi,stdj,0.0,corij,0.0,
                      0.0,Ri,Rj,0.0,L);
        E2x=(0.5/sqrt(2.0*pi))*exp(2.0*(medi+stdi*stdi));
        L=2.0*stdi;
        E2x*=I_skew(10,Ri,sqrt(2.0),1,-1.0,0.0,stdi,stdj,0.0,corij,0.0,
                    0.0,Ri,Rj,0.0,L);
        Exyb=0.5*exp(medi+medj+0.5*(stdi*stdi+stdj*stdj)+stdi*stdj*corij)*
                 (1.0-erff((Rj-stdj-corij*stdi)/sqrt(2.0)));
        Exb=0.5*exp(medi+0.5*stdi*stdi)*
                 (1.0-erff((Rj-corij*stdi)/sqrt(2.0)));
        E2xa=0.5*exp(2.0*medi+2.0*stdi*stdi)*
                 (1.0-erff((Ri-2.0*stdi)/sqrt(2.0)));
        Exa=0.5*exp(medi+0.5*stdi*stdi)*
                 (1.0-erff((Ri-stdi)/sqrt(2.0)));
        Eyb=0.5*exp(medj+0.5*stdj*stdj)*
                 (1.0-erff((Rj-stdj)/sqrt(2.0)));
        E1b=0.5*(1.0-erff(Rj/sqrt(2.0)));
        E1a=0.5*(1.0-erff(Ri/sqrt(2.0)));
        Exi=exp(medi+0.5*stdi*stdi);
        resp=(Si*Sj*E2xy-Si*E2x*Kcj-Sj*Exy*Kci+Kci*Kcj*Ex)/(Ci*Cj);
        resp+=-ms*(Si*Sj*Exy-Si*Ex*Kcj-Sj*Ey*Kci+Kci*Kcj*E1)/(Ci*Cj);
        resp+=-mci*(Sj*Exyb-Kcj*Exb)/Cj-mcj*(Si*E2xa-Kci*Exa)/Ci;
        resp+=ms*mci*(Sj*Eyb-Kcj*E1b)/Cj+ms*mcj*(Si*Exa-Kci*E1a)/Ci;
        resp+=mci*mcj*Exi-ms*mci*mcj;
        return(resp);
}


double m_ricipj(double medi, double medj, double stdi, double stdj,
                double corij, double Si, double Sj, double Kci, double Kpj,
                double Ci, double Pj, double ms, double mc, double mp)
/**********************************************************************/
/* coskewness entre uma call sobre um ativo i, uma put sobre o ativo j     */
/* e o ativo básico i;                                                     */
/* corij = coef. de correlacao entre os ativos i e j.                      */
/* Si,Sj = cotacao do ativo i,j em t=0;                                    */
/* Ci = premio da call sobre i                                             */
```

```
/* Pj = premio da put sobre j                                          */
/* Kci,Kpj = strikes da call e da put                                  */
/**********************************************************************/
{
        double Exy, E2xy, Ex, E2x, Ey, E1, Exyb, Exb, E2xa, Exa, Eyb, E1b,
               E1a, Exi, Ri, Rj, resp, L;

        Ri=(log(Kci/Si)-medi)/stdi;
        Rj=(log(Kpj/Sj)-medj)/stdj;
        E2xy=(0.5/sqrt(2.0*pi))*
             exp(2.0*medi+medj+2.0*stdi*stdi+
                 0.5*stdj*stdj+2.0*stdi*stdj*corij);
        L=2.0*stdi+stdj*corij;
        E2xy*=I_skew(9,Ri,sqrt(2.0),1,1.0,0.0,stdi,stdj,0.0,corij,0.0,
                     0.0,Ri,Rj,0.0,L);
        Exy=(0.5/sqrt(2.0*pi))*
             exp(medi+medj+0.5*(stdi*stdi+stdj*stdj)+
                                  stdi*stdj*corij);
        L=stdi+stdj*corij;
        Exy*=I_skew(9,Ri,sqrt(2.0),1,1.0,0.0,stdi,stdj,0.0,corij,0.0,
                    0.0,Ri,Rj,0.0,L);
        Ex=(0.5/sqrt(2.0*pi))*exp(medi+0.5*stdi*stdi);
        L=stdi;
        Ex*=I_skew(10,Ri,sqrt(2.0),1,1.0,0.0,stdi,stdj,0.0,corij,0.0,
                   0.0,Ri,Rj,0.0,L);
        Ey=(0.5/sqrt(2.0*pi))*exp(medj+0.5*stdj*stdj);
        L=stdj*corij;
        Ey*=I_skew(9,Ri,sqrt(2.0),1,1.0,0.0,stdi,stdj,0.0,corij,0.0,
                   0.0,Ri,Rj,0.0,L);
        E1=(0.5/sqrt(2.0*pi));
        L=0.0;
        E1*=I_skew(10,Ri,sqrt(2.0),1,1.0,0.0,stdi,stdj,0.0,corij,0.0,
                   0.0,Ri,Rj,0.0,L);
        E2x=(0.5/sqrt(2.0*pi))*exp(2.0*(medi+stdi*stdi));
        L=2.0*stdi;
        E2x*=I_skew(10,Ri,sqrt(2.0),1,1.0,0.0,stdi,stdj,0.0,corij,0.0,
                    0.0,Ri,Rj,0.0,L);
        Exyb=0.5*exp(medi+medj+0.5*(stdi*stdi+stdj*stdj)+stdi*stdj*corij)*
                 (1.0+erff((Rj-stdj-corij*stdi)/sqrt(2.0)));
        Exb=0.5*exp(medi+0.5*stdi*stdi)*
                 (1.0+erff((Rj-corij*stdi)/sqrt(2.0)));
        E2xa=0.5*exp(2.0*medi+2.0*stdi*stdi)*
                 (1.0-erff((Ri-2.0*stdi)/sqrt(2.0)));
        Exa=0.5*exp(medi+0.5*stdi*stdi)*
                 (1.0-erff((Ri-stdi)/sqrt(2.0)));
        Eyb=0.5*exp(medj+0.5*stdj*stdj)*
```

```
                (1.0+erff((Rj-stdj)/sqrt(2.0)));
        E1b=0.5*(1.0+erff(Rj/sqrt(2.0)));
        E1a=0.5*(1.0-erff(Ri/sqrt(2.0)));
        Exi=exp(medi+0.5*stdi*stdi);
        resp=(-Si*Sj*E2xy+Si*E2x*Kcj+Sj*Exy*Kci-Kci*Kpj*Ex)/(Ci*Pj);
        resp+=-ms*(-Si*Sj*Exy+Si*Ex*Kpj+Sj*Ey*Kci-Kci*Kpj*E1)/(Ci*Pj);
        resp+=-mc*(-Sj*Exyb+Kpj*Exb)/Pj-mp*(Si*E2xa-Kci*Exa)/Ci;
        resp+=ms*mc*(-Sj*Eyb+Kpj*E1b)/Pj+ms*mp*(Si*Exa-Kci*E1a)/Ci;
        resp+=mc*mp*Exi-ms*mc*mp;
        return(resp);
}


double m_ripicj(double medi, double medj, double stdi, double stdj,
                double corij, double Si, double Sj, double Kpi,
                double Kcj, double Pi, double Cj,
                double ms, double mp, double mc)
/**********************************************************************/
/* coskewness entre uma call sobre um ativo j, uma put sobre o ativo i     */
/* e o ativo básico i;                                                     */
/* corij = coef. de correlacao entre os ativos i e j.                      */
/* Si,Sj = cotacao do ativo i,j em t=0;                                    */
/* Cj = premio da call sobre j                                             */
/* Pi = premio da put sobre i                                              */
/* Kpi,Kcj = strikes da call e da put                                      */
/**********************************************************************/
{
        double Exy, E2xy, Ex, E2x, Ey, E1, Ri, Rj, resp, L,
               Exyb, Exb, E2xa, Exa, E1a, Eyb, E1b, Exi;

        Ri=(log(Kpi/Si)-medi)/stdi;
        Rj=(log(Kcj/Sj)-medj)/stdj;
        E2xy=(0.5/sqrt(2.0*pi))*
             exp(2.0*medi+medj+2.0*stdi*stdi+0.5*stdj*stdj+
                 2.0*stdi*stdj*corij);
        L=2.0*stdi+stdj*corij;
        E2xy*=I_skew(9,Ri,sqrt(2.0),-1,-1.0,0.0,stdi,stdj,0.0,corij,0.0,
                     0.0,Ri,Rj,0.0,L);
        Exy=(0.5/sqrt(2.0*pi))*
             exp(medi+medj+0.5*(stdi*stdi+stdj*stdj)+
                                  stdi*stdj*corij);
        L=stdi+stdj*corij;
        Exy*=I_skew(9,Ri,sqrt(2.0),-1,-1.0,0.0,stdi,stdj,0.0,corij,0.0,
                    0.0,Ri,Rj,0.0,L);
        Ex=(0.5/sqrt(2.0*pi))*exp(medi+0.5*stdi*stdi);
        L=stdi;
```

```
        Ex*=I_skew(10,Ri,sqrt(2.0),-1,-1.0,0.0,stdi,stdj,0.0,corij,0.0,
                   0.0,Ri,Rj,0.0,L);
        Ey=(0.5/sqrt(2.0*pi))*exp(medj+0.5*stdj*stdj);
        L=stdj*corij;
        Ey*=I_skew(9,Ri,sqrt(2.0),-1,-1.0,0.0,stdi,stdj,0.0,corij,0.0,
                   0.0,Ri,Rj,0.0,L);
        E1=(0.5/sqrt(2.0*pi));
        L=0.0;
        E1*=I_skew(10,Ri,sqrt(2.0),-1,-1.0,0.0,stdi,stdj,0.0,corij,0.0,
                   0.0,Ri,Rj,0.0,L);
        E2x=(0.5/sqrt(2.0*pi))*exp(2.0*(medi+stdi*stdi));
        L=2.0*stdi;
        E2x*=I_skew(10,Ri,sqrt(2.0),-1,-1.0,0.0,stdi,stdj,0.0,corij,0.0,
                    0.0,Ri,Rj,0.0,L);
        Exyb=0.5*exp(medi+medj+0.5*(stdi*stdi+stdj*stdj)+stdi*stdj*corij)*
                  (1.0-erff((Rj-stdj-corij*stdi)/sqrt(2.0)));
        Exb=0.5*exp(medi+0.5*stdi*stdi)*
                  (1.0-erff((Rj-corij*stdi)/sqrt(2.0)));
        E2xa=0.5*exp(2.0*medi+2.0*stdi*stdi)*
                  (1.0+erff((Ri-2.0*stdi)/sqrt(2.0)));
        Exa=0.5*exp(medi+0.5*stdi*stdi)*
                  (1.0+erff((Ri-stdi)/sqrt(2.0)));
        Eyb=0.5*exp(medj+0.5*stdj*stdj)*
                  (1.0-erff((Rj-stdj)/sqrt(2.0)));
        E1b=0.5*(1.0-erff(Rj/sqrt(2.0)));
        E1a=0.5*(1.0+erff(Ri/sqrt(2.0)));
        Exi=exp(medi+0.5*stdi*stdi);
        resp=(-Si*Sj*E2xy+Si*E2x*Kcj+Sj*Exy*Kpi-Kpi*Kcj*Ex)/(Pi*Cj);
        resp+=-ms*(-Si*Sj*Exy+Si*Ex*Kcj+Sj*Ey*Kpi-Kpi*Kcj*E1)/(Pi*Cj);
        resp+=-mp*(Sj*Exyb-Kcj*Exb)/Cj-mc*(-Si*E2xa+Kpi*Exa)/Pi;
        resp+=ms*mp*(Sj*Eyb-Kcj*E1b)/Cj+ms*mc*(-Si*Exa+Kpi*E1a)/Pi;
        resp+=mc*mp*Exi-ms*mc*mp;
        return(resp);
}


double m_ripipj(double medi, double medj, double stdi, double stdj,
                double corij, double Si, double Sj, double Kpi,
                double Kpj, double Pi, double Pj,
                double ms, double mpi, double mpj)
/***************************************************************************/
/* coskewness entre puts sobre ativos i e j e o ativo básico i;           */
/* corij = coef. de correlacao entre os ativos i e j.                     */
/* Si,Sj = cotacao do ativo i,j em t=0;                                   */
/* Cj = premio da call sobre j                                            */
/* Pi = premio da put sobre i                                             */
```

```
/* Kpi,Kpj = strikes da call e da put                                   */
/************************************************************************/
{
        double Exy, E2xy, Ex, E2x, Ey, E1, Ri, Rj, resp, L,
               Exyb, Exb, E2xa, Exa, Eyb, E1b, Exa, E1a, Exi;

        Ri=(log(Kpi/Si)-medi)/stdi;
        Rj=(log(Kpj/Sj)-medj)/stdj;
        E2xy=(0.5/sqrt(2.0*pi))*
              exp(2.0*medi+medj+2.0*stdi*stdi+0.5*stdj*stdj+
                  2.0*stdi*stdj*corij);
        L=2.0*stdi+stdj*corij;
        E2xy*=I_skew(9,Ri,sqrt(2.0),-1,1.0,0.0,stdi,stdj,0.0,corij,0.0,
                     0.0,Ri,Rj,0.0,L);
        Exy=(0.5/sqrt(2.0*pi))*
             exp(medi+medj+0.5*(stdi*stdi+stdj*stdj)+
                                 stdi*stdj*corij);
        L=stdi+stdj*corij;
        Exy*=I_skew(9,Ri,sqrt(2.0),-1,1.0,0.0,stdi,stdj,0.0,corij,0.0,
                    0.0,Ri,Rj,0.0,L);
        Ex=(0.5/sqrt(2.0*pi))*exp(medi+0.5*stdi*stdi);
        L=stdi;
        Ex*=I_skew(10,Ri,sqrt(2.0),-1,1.0,0.0,stdi,stdj,0.0,corij,0.0,
                   0.0,Ri,Rj,0.0,L);
        Ey=(0.5/sqrt(2.0*pi))*exp(medj+0.5*stdj*stdj);
        L=stdj*corij;
        Ey*=I_skew(9,Ri,sqrt(2.0),-1,1.0,0.0,stdi,stdj,0.0,corij,0.0,
                   0.0,Ri,Rj,0.0,L);
        E1=(0.5/sqrt(2.0*pi));
        L=0.0;
        E1*=I_skew(10,Ri,sqrt(2.0),-1,1.0,0.0,stdi,stdj,0.0,corij,0.0,
                   0.0,Ri,Rj,0.0,L);
        E2x=(0.5/sqrt(2.0*pi))*exp(2.0*(medi+stdi*stdi));
        L=2.0*stdi;
        E2x*=I_skew(10,Ri,sqrt(2.0),-1,1.0,0.0,stdi,stdj,0.0,corij,0.0,
                    0.0,Ri,Rj,0.0,L);
        Exyb=0.5*exp(medi+medj+0.5*(stdi*stdi+stdj*stdj)+stdi*stdj*corij)*
                  (1.0+erff((Rj-stdj-corij*stdi)/sqrt(2.0)));
        Exb=0.5*exp(medi+0.5*stdi*stdi)*
                  (1.0+erff((Rj-corij*stdi)/sqrt(2.0)));
        E2xa=0.5*exp(2.0*medi+2.0*stdi*stdi)*
                  (1.0+erff((Ri-2.0*stdi)/sqrt(2.0)));
        Exa=0.5*exp(medi+0.5*stdi*stdi)*
                  (1.0+erff((Ri-stdi)/sqrt(2.0)));
        Eyb=0.5*exp(medj+0.5*stdj*stdj)*
                  (1.0+erff((Rj-stdj)/sqrt(2.0)));
```

```
        E1b=0.5*(1.0+erff(Rj/sqrt(2.0)));
        E1a=0.5*(1.0+erff(Ri/sqrt(2.0)));
        Exi=exp(medi+0.5*stdi*stdi);
        resp=(Si*Sj*E2xy-Si*E2x*Kpj-Sj*Exy*Kpi+Kpi*Kpj*Ex)/(Pi*Pj);
        resp+=-ms*(Si*Sj*Exy-Si*Ex*Kpj-Sj*Ey*Kpi+Kpi*Kpj*E1)/(Pi*Pj);
        resp+=-mpi*(-Sj*Exyb+Kpj*Exb)/Pj-mpj*(-Si*E2xa+Kpi*Exa)/Pi;
        resp+=ms*mpi*(-Sj*Eyb+Kpj*E1b)/Pj+ms*mpj*(-Si*Exa+Kpi*E1a)/Pi;
        resp+=mpi*mpj*Exi-ms*mpi*mpj;
        return(resp);
}


double m_cicicj(double medi, double medj, double stdi, double stdj,
                double corij, double Si, double Sj, double Kc1, double Kc2,
                double Kcj, double C1, double C2, double Cj,
                double mc1, double mc2, double mcj)
/*************************************************************************/
/* coskewness entre calls sobre ativos i e j ;                           */
/* corij = coef. de correlacao entre os ativos i e j.                    */
/* Si,Sj = cotacao do ativo i,j em t=0;                                  */
/* Cj,C1,C2 = premio da call sobre j e i.                                */
/* Kc1,Kc2,Kj = strikes das calls;                                       */
/*************************************************************************/
{
        double L, Exy, E2xy, Ex, E2x, Ey, E1, Ri, Rj, resp, K,
               Exy2b, Ex2b, Ey2b, E12b, Exy1b, Ex1b, Ey1b, E11b,
               E2xa, Exa, E1a, Eyb, E1b, Ex2, E12, Ex1, E11;

        K=dmax(Kc1,Kc2);
        Ri=(log(K/Si)-medi)/stdi;
        Rj=(log(Kcj/Sj)-medj)/stdj;
        E2xy=(0.5/sqrt(2.0*pi))*
              exp(2.0*medi+medj+2.0*stdi*stdi+0.5*stdj*stdj+
                  2.0*stdi*stdj*corij);
        L=2.0*stdi+stdj*corij;
        E2xy*=I_skew(9,Ri,sqrt(2.0),1,-1.0,0.0,stdi,stdj,0.0,corij,0.0,
                     0.0,Ri,Rj,0.0,L);
        Exy=(0.5/sqrt(2.0*pi))*
            exp(medi+medj+0.5*(stdi*stdi+stdj*stdj)+
                                 stdi*stdj*corij);
        L=stdi+stdj*corij;
        Exy*=I_skew(9,Ri,sqrt(2.0),1,-1.0,0.0,stdi,stdj,0.0,corij,0.0,
                    0.0,Ri,Rj,0.0,L);
        Ex=(0.5/sqrt(2.0*pi))*exp(medi+0.5*stdi*stdi);
        L=stdi;
        Ex*=I_skew(10,Ri,sqrt(2.0),1,-1.0,0.0,stdi,stdj,0.0,corij,0.0,
```

```
           0.0,Ri,Rj,0.0,L);
Ey=(0.5/sqrt(2.0*pi))*exp(medj+0.5*stdj*stdj);
L=stdj*corij;
Ey*=I_skew(9,Ri,sqrt(2.0),1,-1.0,0.0,stdi,stdj,0.0,corij,0.0,
           0.0,Ri,Rj,0.0,L);
E1=(0.5/sqrt(2.0*pi));
L=0.0;
E1*=I_skew(10,Ri,sqrt(2.0),1,-1.0,0.0,stdi,stdj,0.0,corij,0.0,
           0.0,Ri,Rj,0.0,L);
E2x=(0.5/sqrt(2.0*pi))*exp(2.0*(medi+stdi*stdi));
L=2.0*stdi;
E2x*=I_skew(10,Ri,sqrt(2.0),1,-1.0,0.0,stdi,stdj,0.0,corij,0.0,
            0.0,Ri,Rj,0.0,L);
E2xa=0.5*exp(2.0*medi+2.0*stdi*stdi)*(1.0-erff((Ri-stdi)/sqrt(2.0)));
Exa=0.5*exp(medi+0.5*stdi*stdi)*(1.0-erff((Ri-stdi)/sqrt(2.0)));
E1a=0.5*(1.0-erff(Ri/sqrt(2.0)));
Eyb=0.5*exp(medj+0.5*stdj*stdj)*(1.0-erff((Rj-stdj)/sqrt(2.0)));
E1b=0.5*(1.0-erff(Rj/sqrt(2.0)));

Ri=(log(Kc2/Si)-medi)/stdi;
Exy2b=(0.5/sqrt(2.0*pi))*
      exp(medi+medj+0.5*(stdi*stdi+stdj*stdj)+
L=stdi+stdj*corij;
Exy2b*=I_skew(9,Ri,sqrt(2.0),1,-1.0,0.0,stdi,stdj,0.0,corij,0.0,
            0.0,Ri,Rj,0.0,L);

Ex2b=(0.5/sqrt(2.0*pi))*exp(medi+0.5*stdi*stdi);
L=stdi;
Ex2b*=I_skew(10,Ri,sqrt(2.0),1,-1.0,0.0,stdi,stdj,0.0,corij,0.0,
           0.0,Ri,Rj,0.0,L);

Ey2b=(0.5/sqrt(2.0*pi))*exp(medj+0.5*stdj*stdj);
L=stdj*corij;
Ey2b*=I_skew(9,Ri,sqrt(2.0),1,-1.0,0.0,stdi,stdj,0.0,corij,0.0,
           0.0,Ri,Rj,0.0,L);

E12b=(0.5/sqrt(2.0*pi));
L=0.0;
E12b*=I_skew(10,Ri,sqrt(2.0),1,-1.0,0.0,stdi,stdj,0.0,corij,0.0,
           0.0,Ri,Rj,0.0,L);
Ex2=0.5*exp(medi+0.5*stdi*stdi)*(1.0-erff((Ri-stdi)/sqrt(2.0)));
E12=0.5*(1.0-erff(Ri/sqrt(2.0)));

Ri=(log(Kc1/Si)-medi)/stdi;
Exy1b=(0.5/sqrt(2.0*pi))*
      exp(medi+medj+0.5*(stdi*stdi+stdj*stdj)+
```

```
        L=stdi+stdj*corij;
        Exy1b*=I_skew(9,Ri,sqrt(2.0),1,-1.0,0.0,stdi,stdj,0.0,corij,0.0,
                   0.0,Ri,Rj,0.0,L);

        Ex1b=(0.5/sqrt(2.0*pi))*exp(medi+0.5*stdi*stdi);
        L=stdi;
        Ex1b*=I_skew(10,Ri,sqrt(2.0),1,-1.0,0.0,stdi,stdj,0.0,corij,0.0,
                  0.0,Ri,Rj,0.0,L);

        Ey1b=(0.5/sqrt(2.0*pi))*exp(medj+0.5*stdj*stdj);
        L=stdj*corij;
        Ey1b*=I_skew(9,Ri,sqrt(2.0),1,-1.0,0.0,stdi,stdj,0.0,corij,0.0,
                  0.0,Ri,Rj,0.0,L);

        E11b=(0.5/sqrt(2.0*pi));
        L=0.0;
        E11b*=I_skew(10,Ri,sqrt(2.0),1,-1.0,0.0,stdi,stdj,0.0,corij,0.0,
                  0.0,Ri,Rj,0.0,L);

        Ex1=0.5*exp(medi+0.5*stdi*stdi)*(1.0-erff((Ri-stdi)/sqrt(2.0)));
        E11=0.5*(1.0-erff(Ri/sqrt(2.0)));

        resp=(Si*Si*Sj*E2xy-Si*Si*E2x*Kcj-Si*Sj*(Kc1+Kc2)*Exy
              +Si*Kcj*(Kc1+Kc2)*Ex+Sj*Kc1*Kc2*Ey-Kc1*Kc2*Kcj*E1)/(C1*C2*Cj);
        resp+=-mc1*(Si*Sj*Exy2b-Si*Kcj*Ex2b-Kc2*Sj*Ey2b+Kc2*Kcj*E12b)/(C2*Cj);
        resp+=-mc2*(Si*Sj*Exy1b-Si*Kcj*Ex1b-Kc1*Sj*Ey1b+Kc1*Kcj*E11b)/(C1*Cj);
        resp+=-mcj*(Si*Si*E2xa-Si*(Kc1+Kc2)*Exa+Kc1*Kc2*E1a)/(C1*C2);
        resp+=mc1*mc2*(Sj*Eyb-Kcj*E1b)/Cb+mc1*mcj*(Si*Ex2-Kc2*E12)/C2;
        resp+=mc2*mcj*(Si*Ex1-Kc1*E11)/C1-mc1*mc2*mcj;
        return(resp);
}


double m_cicipj(double medi, double medj, double stdi, double stdj,
                double corij, double Si, double Sj, double Kc1, double Kc2,
                double Kpj, double C1, double C2, double Pj,
                double mc1, double mc2,  double mp)
/*************************************************************************/
/* coskewness entre calls sobre o ativo i e put sobre o ativo j ;        */
/* corij = coef. de correlacao entre os ativos i e j.                    */
/* Si,Sj = cotacao do ativo i,j em t=0;                                  */
/* C2,C1,Pj = premio das calls e da put.                                 */
/* Kc1,Kc2,Kpj = strikes das calls e da put;                             */
/*************************************************************************/
{
        double L, E2x, Exy, E2xy, Ex, Ey, E1, Ri, Rj, resp, K;
```

```
K=dmax(Kc1,Kc2);
Ri=(log(K/Si)-medi)/stdi;
Rj=(log(Kpj/Sj)-medj)/stdj;
E2xy=(0.5/sqrt(2.0*pi))*
     exp(2.0*medi+medj+2.0*stdi*stdi+0.5*stdj*stdj+
         2.0*stdi*stdj*corij);
L=2.0*stdi+stdj*corij;
E2xy*=I_skew(9,Ri,sqrt(2.0),1,1.0,0.0,stdi,stdj,0.0,corij,0.0,
             0.0,Ri,Rj,0.0,L);
Exy=(0.5/sqrt(2.0*pi))*
    exp(medi+medj+0.5*(stdi*stdi+stdj*stdj)+
                        stdi*stdj*corij);
L=stdi+stdj*corij;
Exy*=I_skew(9,Ri,sqrt(2.0),1,1.0,0.0,stdi,stdj,0.0,corij,0.0,
            0.0,Ri,Rj,0.0,L);
Ex=(0.5/sqrt(2.0*pi))*exp(medi+0.5*stdi*stdi);
L=stdi;
Ex*=I_skew(10,Ri,sqrt(2.0),1,1.0,0.0,stdi,stdj,0.0,corij,0.0,
           0.0,Ri,Rj,0.0,L);
Ey=(0.5/sqrt(2.0*pi))*exp(medj+0.5*stdj*stdj);
L=stdj*corij;
Ey*=I_skew(9,Ri,sqrt(2.0),1,1.0,0.0,stdi,stdj,0.0,corij,0.0,
           0.0,Ri,Rj,0.0,L);
E1=(0.5/sqrt(2.0*pi));
L=0.0;
E1*=I_skew(10,Ri,sqrt(2.0),1,1.0,0.0,stdi,stdj,0.0,corij,0.0,
           0.0,Ri,Rj,0.0,L);
E2x=(0.5/sqrt(2.0*pi))*exp(2.0*(medi+stdi*stdi));
L=2.0*stdi;
E2x*=I_skew(10,Ri,sqrt(2.0),1,1.0,0.0,stdi,stdj,0.0,corij,0.0,
            0.0,Ri,Rj,0.0,L);

E2xa=0.5*exp(2.0*medi+2.0*stdi*stdi)*(1.0-erff((Ri-stdi)/sqrt(2.0)));
Exa=0.5*exp(medi+0.5*stdi*stdi)*(1.0-erff((Ri-stdi)/sqrt(2.0)));
E1a=0.5*(1.0-erff(Ri/sqrt(2.0)));

Eyb=0.5*exp(medj+0.5*stdj*stdj)*(1.0-erff((Rj-stdj)/sqrt(2.0)));
E1b=0.5*(1.0+erff(Rj/sqrt(2.0)));

Ri=(log(Kc2/Si)-medi)/stdi;
Exy2b=(0.5/sqrt(2.0*pi))*
      exp(medi+medj+0.5*(stdi*stdi+stdj*stdj)+
L=stdi+stdj*corij;
Exy2b*=I_skew(9,Ri,sqrt(2.0),1,1.0,0.0,stdi,stdj,0.0,corij,0.0,
            0.0,Ri,Rj,0.0,L);
```

```
Ex2b=(0.5/sqrt(2.0*pi))*exp(medi+0.5*stdi*stdi);
L=stdi;
Ex2b*=I_skew(10,Ri,sqrt(2.0),1,1.0,0.0,stdi,stdj,0.0,corij,0.0,
          0.0,Ri,Rj,0.0,L);

Ey2b=(0.5/sqrt(2.0*pi))*exp(medj+0.5*stdj*stdj);
L=stdj*corij;
Ey2b*=I_skew(9,Ri,sqrt(2.0),1,1.0,0.0,stdi,stdj,0.0,corij,0.0,
          0.0,Ri,Rj,0.0,L);

E12b=(0.5/sqrt(2.0*pi));
L=0.0;
E12b*=I_skew(10,Ri,sqrt(2.0),1,1.0,0.0,stdi,stdj,0.0,corij,0.0,
          0.0,Ri,Rj,0.0,L);

Ex2=0.5*exp(medi+0.5*stdi*stdi)*(1.0-erff((Ri-stdi)/sqrt(2.0)));
E12=0.5*(1.0-erff(Ri/sqrt(2.0)));

Ri=(log(Kc1/Si)-medi)/stdi;
Exy1b=(0.5/sqrt(2.0*pi))*
      exp(medi+medj+0.5*(stdi*stdi+stdj*stdj)+
L=stdi+stdj*corij;
Exy1b*=I_skew(9,Ri,sqrt(2.0),1,1.0,0.0,stddi,stdj,0.0,corij,0.0,
           0.0,Ri,Rj,0.0,L);

Ex1b=(0.5/sqrt(2.0*pi))*exp(medi+0.5*stdi*stdi);
L=stdi;
Ex1b*=I_skew(10,Ri,sqrt(2.0),1,1.0,0.0,stdi,stdj,0.0,corij,0.0,
          0.0,Ri,Rj,0.0,L);

Ey1b=(0.5/sqrt(2.0*pi))*exp(medj+0.5*stdj*stdj);
L=stdj*corij;
Ey1b*=I_skew(9,Ri,sqrt(2.0),1,1.0,0.0,stdi,stdj,0.0,corij,0.0,
          0.0,Ri,Rj,0.0,L);

E11b=(0.5/sqrt(2.0*pi));
L=0.0;
E11b*=I_skew(10,Ri,sqrt(2.0),1,1.0,0.0,stdi,stdj,0.0,corij,0.0,
          0.0,Ri,Rj,0.0,L);

Ex1=0.5*exp(medi+0.5*stdi*stdi)*(1.0-erff((Ri-stdi)/sqrt(2.0)));
E11=0.5*(1.0-erff(Ri/sqrt(2.0)));

resp=(-Si*Si*Sj*E2xy+Si*Si*E2x*Kpj+Si*Sj*(Kc1+Kc2)*Exy
       -Si*Kpj*(Kc1+Kc2)*Ex-Sj*Kc1*Kc2*Ey+Kc1*Kc2*Kpj*E1)/(C1*C2*Pj);
```

```
        resp+=mc1*(Si*Sj*Exy2b-Si*Kpj*Ex2b-Kc2*Sj*Ey2b+Kc2*Kpj*E12b)/(C2*Pj);
        resp+=mc2*(Si*Sj*Exy1b-Si*Kpj*Ex1b-Kc1*Sj*Ey1b+Kc1*Kpj*E11b)/(C1*Pj);
        resp+=-mp*(Si*Si*E2xa-Si*(Kc1+Kc2)*Exa+Kc1*Kc2*E1a)/(C1*C2);
        resp+=mc1*mc2*(-Sj*Eyb+Kpj*E1b)/Pj+mc1*mp*(Si*Ex2-Kc2*E12)/C2;
        resp+=mc2*mp*(Si*Ex1-Kc1*E11)/C1-mc1*mc2*mp;

        return(resp);
}


double m_cipicj(double medi, double medj, double stdi, double stdj,
                double corij, double Si, double Sj, double Kci, double Kpi,
                double Kcj, double Ci, double Pi, double Cj,
                double mci, double mp, double mcj)
/*************************************************************************/
/* coskewness entre calls sobre os ativos i e j e put sobre o ativo i ;    */
/* corij = coef. de correlacao entre os ativos i e j.                      */
/* Si,Sj = cotacao do ativo i,j em t=0;                                    */
/* Ci,Cj,Pi = premio das calls e da put.                                   */
/* Kci,Kcj,Kpi = strikes das calls e da put;                               */
/*************************************************************************/
{
        double L,Exy, E2xy, Ex, E2x, Ey, E1, Rj, resp,Rc,Rp,
               Exypc, Expc,Eypc,E1pc,Exycc,Excc,Eycc,E1cc,
               Eyb,E1b,Exp,E1p,Exc,E1c;

        resp=0.0;
        Rc=(log(Kci/Si)-medi)/stdi;
        Rp=(log(Kpi/Si)-medi)/stdi;
        Rj=(log(Kcj/Sj)-medj)/stdj;
        if (Kci < Kpi){
              E2xy=(0.5/sqrt(2.0*pi))*
                    exp(2.0*medi+medj+2.0*stdi*stdi+0.5*stdj*stdj+
                        2.0*stdi*stdj*corij);
              L=2.0*stdi+stdj*corij;
              E2xy*=(I_skew(9,Rc,sqrt(2.0),1,-1.0,0.0,stdi,stdj,0.0,corij,0.0,
                     0.0,Rc,Rj,0.0,L)
                    -I_skew(9,Rp,sqrt(2.0),1,-1.0,0.0,stdi,stdj,0.0,corij,0.0,
                     0.0,Rp,Rj,0.0,L));
              Exy=(0.5/sqrt(2.0*pi))*
                   exp(medi+medj+0.5*(stdi*stdi+stdj*stdj)+
                                   stdi*stdj*corij);
              L=stdi+stdj*corij;
              Exy*=(I_skew(9,Rc,sqrt(2.0),1,-1.0,0.0,stdi,stdj,0.0,corij,0.0,
                    0.0,Rc,Rj,0.0,L)
                    -I_skew(9,Rp,sqrt(2.0),1,-1.0,0.0,stdi,stdj,0.0,corij,0.0,
```

```
                  0.0,Rp,Rj,0.0,L));
            Ex=(0.5/sqrt(2.0*pi))*exp(medi+0.5*stdi*stdi);
            L=stdi;
            Ex*=(I_skew(10,Rc,sqrt(2.0),1,-1.0,0.0,stdi,stdj,0.0,corij,0.0,
                        0.0,Rc,Rj,0.0,L)
                -I_skew(10,Rp,sqrt(2.0),1,-1.0,0.0,stdi,stdj,0.0,corij,0.0,
                        0.0,Rp,Rj,0.0,L));
            Ey=(0.5/sqrt(2.0*pi))*exp(medj+0.5*stdj*stdj);
            L=stdj*corij;
            Ey*=(I_skew(9,Rc,sqrt(2.0),1,-1.0,0.0,stdi,stdj,0.0,corij,0.0,
                        0.0,Rc,Rj,0.0,L)
                -I_skew(9,Rp,sqrt(2.0),1,-1.0,0.0,stdi,stdj,0.0,corij,0.0,
                        0.0,Rp,Rj,0.0,L));
            E1=(0.5/sqrt(2.0*pi));
            L=0.0;
            E1*=(I_skew(9,Rc,sqrt(2.0),1,-1.0,0.0,stdi,stdj,0.0,corij,0.0,
                        0.0,Rc,Rj,0.0,L)
                -I_skew(9,Rp,sqrt(2.0),1,-1.0,0.0,stdi,stdj,0.0,corij,0.0,
                        0.0,Rp,Rj,0.0,L));
            E2x=(0.5/sqrt(2.0*pi))*exp(2.0*(medi+stdi*stdi));
            L=2.0*stdi;
            E2x*=(I_skew(10,Rc,sqrt(2.0),1,-1.0,0.0,stdi,stdj,0.0,corij,0.0,
                         0.0,Rc,Rj,0.0,L)
                 -I_skew(10,Rp,sqrt(2.0),1,-1.0,0.0,stdi,stdj,0.0,corij,0.0,
                         0.0,Rp,Rj,0.0,L));
            E2xaa=0.5*exp(2.0*medi+2.0*stdi*stdi)*
                   (erff((Rc-2.0*stdi)/sqrt(2.0))-erf((Rp-2.0*stdi)/sqrt(2.0)));
            Exaa=0.5*exp(medi+0.5*stdi*stdi)*
                   (erff((Rc-stdi)/sqrt(2.0))-erf((Rp-stdi)/sqrt(2.0)));
            E1aa=0.5*(erff(Rc/sqrt(2.0))-erf(Rp/sqrt(2.0)));
            resp+=(-Si*Si*Sj*E2xy+Si*Si*E2x*Kcj+Si*Sj*(Kci+Kpi)*Exy
                   -Si*(Kci+Kpi)*Kcj*Ex-Kci*Kpi*Sj*Ey+Kci*Kpi*Kcj*E1)/(Ci*Cj*Pi);
            resp+=-mcj*(-Si*Si*E2xaa+Si*(Kci+Kpi)*Exaa-Kci*Kpi*E1aa)/(Pi*Ci);
      }

      Exypc=(0.5/sqrt(2.0*pi))*
            exp(medi+medj+0.5*(stdi*stdi+stdj*stdj)+
                                  stdi*stdj*corij);
      L=stdi+stdj*corij;
      Exypc*=I_skew(9,Rp,sqrt(2.0),-1,-1.0,0.0,stdi,stdj,0.0,corij,0.0,
                  0.0,Rc,Rj,0.0,L);
      Expc=(0.5/sqrt(2.0*pi))*exp(medi+0.5*stdi*stdi);
      L=stdi;
      Expc*=I_skew(10,Rp,sqrt(2.0),-1,-1.0,0.0,stdi,stdj,0.0,corij,0.0,
                 0.0,Rc,Rj,0.0,L);
      Eypc=(0.5/sqrt(2.0*pi))*exp(medj+0.5*stdj*stdj);
```

```
        L=stdj*corij;
        Eypc*=I_skew(9,Rp,sqrt(2.0),-1,-1.0,0.0,stdi,stdj,0.0,corij,0.0,
                   0.0,Rc,Rj,0.0,L);
        E1pc=(0.5/sqrt(2.0*pi));
        L=0.0;
        E1pc*=I_skew(10,Rp,sqrt(2.0),-1,-1.0,0.0,stdi,stdj,0.0,corij,0.0,
                   0.0,Rc,Rj,0.0,L);


        Exycc=(0.5/sqrt(2.0*pi))*
               exp(medi+medj+0.5*(stdi*stdi+stdj*stdj)+
                                   stdi*stdj*corij);
        L=stdi+stdj*corij;
        Exycc*=I_skew(9,Rc,sqrt(2.0),1,-1.0,0.0,stdi,stdj,0.0,corij,0.0,
                      0.0,Rc,Rj,0.0,L);
        Excc=(0.5/sqrt(2.0*pi))*exp(medi+0.5*stdi*stdi);
        L=stdi;
        Excc*=I_skew(10,Rc,sqrt(2.0),1,-1.0,0.0,stdi,stdj,0.0,corij,0.0,
                    0.0,Rc,Rj,0.0,L);
        Eycc=(0.5/sqrt(2.0*pi))*exp(medj+0.5*stdj*stdj);
        L=stdj*corij;
        Eycc*=I_skew(9,Rc,sqrt(2.0),1,-1.0,0.0,stdi,stdj,0.0,corij,0.0,
                      0.0,Rc,Rj,0.0,L);
        E1cc=(0.5/sqrt(2.0*pi));
        L=0.0;
        E1cc*=I_skew(10,Rc,sqrt(2.0),1,-1.0,0.0,stdi,stdj,0.0,corij,0.0,
                      0.0,Rc,Rj,0.0,L);

        Eyb=0.5*exp(medj+0.5*stdj*stdj)*(1.0-erff((Rj-stdj)/sqrt(2.0)));
        E1b=0.5*(1.0-erff(Rj/sqrt(2.0)));

        Exp=0.5*exp(medi+0.5*stdi*stdi)*(1.0+erff((Rp-stdi)/sqrt(2.0)));
        E1p=0.5*(1.0+erff(Rp/sqrt(2.0)));

        Exc=0.5*exp(medi+0.5*stdi*stdi)*(1.0-erff((Rc-stdi)/sqrt(2.0)));
        E1p=0.5*(1.0-erff(Rc/sqrt(2.0)));

        resp+=-mp*(Si*Sj*Exycc-Si*Kcj*Excc-Sj*kci*Eycc+Kci*Kcj*E1cc)/(Ci*Cj);
        resp+=mci*(Si*Sj*Exypc-Si*Kcj*Expc-Sj*kpi*Eypc+Kpi*Kcj*E1pc)/(Pi*Cj);
        resp+=mci*mp*(Sj*Eyb-Kcj*E1b)/Cj+mci*mcj*(-Si*Exp+Kpi*E1p)/Pi;
        resp+=mp*mcj*(Si*Exc-Kci*E1c)/Ci-mci*mcj*mp;

        return(resp);
}
```

```
double m_cipipj(double medi, double medj, double stdi, double stdj,
                double corij, double Si, double Sj, double Kci, double Kpi,
                double Kpj, double Ci, double Pi, double Pj,
                double mc, double mpi, double mpj)
/***********************************************************************/
/* coskewness entre puts sobre ativos i e j e call sobre o ativo i ;   */
/* corij = coef. de correlacao entre os ativos i e j.                  */
/* Si,Sj = cotacao do ativo i,j em t=0;                                */
/* Ci,Pi,Pj = premio da call e das puts.                               */
/* Kpi,Kpj,Kci = strikes das puts e da call;                           */
/***********************************************************************/
{
        double L,Exy, E2xy, Ex, E2x, Ey, E1, resp,Rp,Rc,Rj,
               Exypp, Expp, Eypp, E1pp, Exycp, Excp, Eycp, E1cp,
               E2xaa, Exaa, E1aa, Eyb, E1b, Exp, E1p, Exc, E1c;



        resp=0.0;
        Rc=(log(Kci/Si)-medi)/stdi;
        Rp=(log(Kpi/Si)-medi)/stdi;
        Rj=(log(Kpj/Sj)-medj)/stdj;
        if (Kci < Kpi){
              E2xy=(0.5/sqrt(2.0*pi))*
                    exp(2.0*medi+medj+2.0*stdi*stdi+0.5*stdj*stdj+
                        2.0*stdi*stdj*corij);
              L=2.0*stdi+stdj*corij;
              E2xy*=(I_skew(9,Rc,sqrt(2.0),1,1.0,0.0,stdi,stdj,0.0,corij,0.0,
                            0.0,Rc,Rj,0.0,L)
                    -I_skew(9,Rp,sqrt(2.0),1,1.0,0.0,stdi,stdj,0.0,corij,0.0,
                            0.0,Rp,Rj,0.0,L));
              Exy=(0.5/sqrt(2.0*pi))*
                   exp(medi+medj+0.5*(stdi*stdi+stdj*stdj)+
                                      stdi*stdj*corij);
              L=stdi+stdj*corij;
              Exy*=(I_skew(9,Rc,sqrt(2.0),1,1.0,0.0,stdi,stdj,0.0,corij,0.0,
                           0.0,Rc,Rj,0.0,L)
                   -I_skew(9,Rp,sqrt(2.0),1,1.0,0.0,stdi,stdj,0.0,corij,0.0,
                           0.0,Rp,Rj,0.0,L));
              Ex=(0.5/sqrt(2.0*pi))*exp(medi+0.5*stdi*stdi);
              L=stdi;
              Ex*=(I_skew(10,Rc,sqrt(2.0),1,1.0,0.0,stdi,stdj,0.0,corij,0.0,
                          0.0,Rc,Rj,0.0,L)
                  -I_skew(10,Rp,sqrt(2.0),1,1.0,0.0,stdi,stdj,0.0,corij,0.0,
                          0.0,Rp,Rj,0.0,L));
              Ey=(0.5/sqrt(2.0*pi))*exp(medj+0.5*stdj*stdj);
```

```
        L=stdj*corij;
        Ey*=(I_skew(9,Rc,sqrt(2.0),1,1.0,0.0,stdi,stdj,0.0,corij,0.0,
                    0.0,Rc,Rj,0.0,L)
            -I_skew(9,Rp,sqrt(2.0),1,1.0,0.0,stdi,stdj,0.0,corij,0.0,
                    0.0,Rp,Rj,0.0,L));
        E1=(0.5/sqrt(2.0*pi));
        L=0.0;
        E1*=(I_skew(10,Rc,sqrt(2.0),1,1.0,0.0,stdi,stdj,0.0,corij,0.0,
                    0.0,Rc,Rj,0.0,L)
            -I_skew(10,Rp,sqrt(2.0),1,1.0,0.0,stdi,stdj,0.0,corij,0.0,
                    0.0,Rp,Rj,0.0,L));
        E2x=(0.5/sqrt(2.0*pi))*exp(2.0*(medi+stdi*stdi));
        L=2.0*stdi;
        E2x*=(I_skew(10,Rc,sqrt(2.0),1,1.0,0.0,stdi,stdj,0.0,corij,0.0,
                     0.0,Rc,Rj,0.0,L)
             -I_skew(10,Rp,sqrt(2.0),1,1.0,0.0,stdi,stdj,0.0,corij,0.0,
                     0.0,Rp,Rj,0.0,L));
        E2xaa=0.5*exp(2.0*medi+2.0*stdi*stdi)*
               (erff((Rc-2.0*stdi)/sqrt(2.0))-erf((Rp-2.0*stdi)/sqrt(2.0)));
        Exaa=0.5*exp(medi+0.5*stdi*stdi)*
               (erff((Rc-stdi)/sqrt(2.0))-erf((Rp-stdi)/sqrt(2.0)));
        E1aa=0.5*(erff(Rc/sqrt(2.0))-erf(Rp/sqrt(2.0)));
        resp+=(Si*Si*Sj*E2xy-Si*Si*E2x*Kpj-Si*Sj*(Kci+Kpi)*Exy
               +Si*(Kci+Kpi)*Kpj*Ex+Kci*Kpi*Sj*Ey-Kci*Kpi*Kpj*E1)/(Ci*Pj*Pi);
        resp+=-mpj*(-Si*Si*E2xaa+Si*(Kci+Kpi)*Exaa-Kci*Kpi*E1aa)/(Pi*Ci);

 }


  Exycp=(0.5/sqrt(2.0*pi))*
         exp(medi+medj+0.5*(stdi*stdi+stdj*stdj)+
                             stdi*stdj*corij);
  L=stdi+stdj*corij;
  Exycp*=I_skew(9,Rc,sqrt(2.0),1,1.0,0.0,stdi,stdj,0.0,corij,0.0,
                0.0,Rc,Rj,0.0,L);
  Excp=(0.5/sqrt(2.0*pi))*exp(medi+0.5*stdi*stdi);
  L=stdi;
  Excp*=I_skew(10,Rc,sqrt(2.0),1,1.0,0.0,stdi,stdj,0.0,corij,0.0,
              0.0,Rc,Rj,0.0,L);
  Eycp=(0.5/sqrt(2.0*pi))*exp(medj+0.5*stdj*stdj);
  L=stdj*corij;
  Eycp*=I_skew(9,Rc,sqrt(2.0),1,1.0,0.0,stdi,stdj,0.0,corij,0.0,
                0.0,Rc,Rj,0.0,L);
  E1cp=(0.5/sqrt(2.0*pi));
  L=0.0;
  E1cp*=I_skew(10,Rc,sqrt(2.0),1,1.0,0.0,stdi,stdj,0.0,corij,0.0,
```

```
                  0.0,Rc,Rj,0.0,L);

        Exypp=(0.5/sqrt(2.0*pi))*
              exp(medi+medj+0.5*(stdi*stdi+stdj*stdj)+
                                    stdi*stdj*corij);
        L=stdi+stdj*corij;
        Exypp*=I_skew(9,Rp,sqrt(2.0),-1,1.0,0.0,stdi,stdj,0.0,corij,0.0,
                    0.0,Rc,Rj,0.0,L);
        Expp=(0.5/sqrt(2.0*pi))*exp(medi+0.5*stdi*stdi);
        L=stdi;
        Expp*=I_skew(10,Rp,sqrt(2.0),-1,1.0,0.0,stdi,stdj,0.0,corij,0.0,
                   0.0,Rc,Rj,0.0,L);
        Eypp=(0.5/sqrt(2.0*pi))*exp(medj+0.5*stdj*stdj);
        L=stdj*corij;
        Eypp*=I_skew(9,Rp,sqrt(2.0),-1,1.0,0.0,stdi,stdj,0.0,corij,0.0,
                   0.0,Rc,Rj,0.0,L);
        E1pp=(0.5/sqrt(2.0*pi));
        L=0.0;
        E1pp*=I_skew(10,Rp,sqrt(2.0),-1,1.0,0.0,stdi,stdj,0.0,corij,0.0,
                   0.0,Rc,Rj,0.0,L);



        Eyb=0.5*exp(medj+0.5*stdj*stdj)*(1.0+erff((Rj-stdj)/sqrt(2.0)));
        E1b=0.5*(1.0+erff(Rj/sqrt(2.0)));

        Exp=0.5*exp(medi+0.5*stdi*stdi)*(1.0+erff((Rp-stdi)/sqrt(2.0)));
        E1p=0.5*(1.0+erff(Rp/sqrt(2.0)));

        Exc=0.5*exp(medi+0.5*stdi*stdi)*(1.0-erff((Rc-stdi)/sqrt(2.0)));
        E1p=0.5*(1.0-erff(Rc/sqrt(2.0)));

        resp+=mpi*(Si*Sj*Exycp-Si*Kpj*Excp-Sj*kci*Eycp+Kpi*Kpj*E1cp)/(Ci*Pj);
        resp+=-mc*(Si*Sj*Exypp-Si*Kpj*Expp-Sj*kpi*Eypp+Kpi*Kpj*E1pp)/(Pi*Pj);
        resp+=mc*mpi*(-Sj*Eyb+Kpj*E1b)/Pj+mc*mpj*(-Si*Exp+Kpi*E1p)/Pi;
        resp+=mpi*mpj*(Si*Exc-Kci*E1c)/Ci-mc*mpi*mpj;

        return(resp);
}


double m_pipicj(double medi, double medj, double stdi, double stdj,
                double corij, double Si, double Sj, double Kp1, double Kp2,
                double Kcj, double P1, double P2, double Cj,
                double mp1,  double mp2, double mc)
/***************************************************************************/
```

```
/* coskewness entre uma call sobre um ativo j e  puts sobre o ativo i;     */
/* corij = coef. de correlacao entre os ativos i e j.                      */
/* Si,Sj = cotacao do ativo i,j em t=0;                                    */
/* Cj = premio da call sobre j                                             */
/* P1,P2 = premio das puts                                                 */
/* Kp1,Kp2,Kcj = strikes da put e da call                                  */
/**************************************************************************/
{
        double L, E2x, Exy, E2xy, Ex, Ey, E1, Ri, Rj, resp, K,
               Exy2b, Ex2b, Ey2b, E12b, Exy1b, Ex1b, Ey1b, E11b,
               E2xpp, Expp, E1pp, Eyb, E1b, Ex2, E12, Ex1, E11;

        K=dmin(Kp1,Kp2);
        Ri=(log(K/Si)-medi)/stdi;
        Rj=(log(Kcj/Sj)-medj)/stdj;
        E2xy=(0.5/sqrt(2.0*pi))*
              exp(2.0*medi+medj+2.0*stdi*stdi+0.5*stdj*stdj+
                  2.0*stdi*stdj*corij);
        L=2.0*stdi+stdj*corij;
        E2xy*=I_skew(9,Ri,sqrt(2.0),-1,-1.0,0.0,stdi,stdj,0.0,corij,0.0,
                     0.0,Ri,Rj,0.0,L);
        Exy=(0.5/sqrt(2.0*pi))*
             exp(medi+medj+0.5*(stdi*stdi+stdj*stdj)+
                                  stdi*stdj*corij);
        L=stdi+stdj*corij;
        Exy*=I_skew(9,Ri,sqrt(2.0),-1,-1.0,0.0,stdi,stdj,0.0,corij,0.0,
                    0.0,Ri,Rj,0.0,L);
        Ex=(0.5/sqrt(2.0*pi))*exp(medi+0.5*stdi*stdi);
        L=stdi;
        Ex*=I_skew(10,Ri,sqrt(2.0),-1,-1.0,0.0,stdi,stdj,0.0,corij,0.0,
                   0.0,Ri,Rj,0.0,L);
        Ey=(0.5/sqrt(2.0*pi))*exp(medj+0.5*stdj*stdj);
        L=stdj*corij;
        Ey*=I_skew(9,Ri,sqrt(2.0),-1,-1.0,0.0,stdi,stdj,0.0,corij,0.0,
                   0.0,Ri,Rj,0.0,L);
        E1=(0.5/sqrt(2.0*pi));
        L=0.0;
        E1*=I_skew(10,Ri,sqrt(2.0),-1,-1.0,0.0,stdi,stdj,0.0,corij,0.0,
                   0.0,Ri,Rj,0.0,L);
        E2x=(0.5/sqrt(2.0*pi))*exp(2.0*(medi+stdi*stdi));
        L=2.0*stdi;
        E2x*=I_skew(10,Ri,sqrt(2.0),-1,-1.0,0.0,stdi,stdj,0.0,corij,0.0,
                    0.0,Ri,Rj,0.0,L);
        E2xpp=0.5*exp(2.0*medi+2.0*stdi*stdi)*(1.0+erff((Ri-2.0*stdi)/sqrt(2.0)));
        Expp=0.5*exp(medi+0.5*stdi*stdi)*(1.0+erff((Ri-stdi)/sqrt(2.0)));
        Expp=0.5*(1.0+erff(Ri/sqrt(2.0)));
```

```
Ri=(log(Kp2/Si)-medi)/stdi;
Exy2b=(0.5/sqrt(2.0*pi))*
      exp(medi+medj+0.5*(stdi*stdi+stdj*stdj)+
                            stdi*stdj*corij);
L=stdi+stdj*corij;
Exy2b*=I_skew(9,Ri,sqrt(2.0),-1,-1.0,0.0,stdi,stdj,0.0,corij,0.0,
            0.0,Ri,Rj,0.0,L);
Ex2b=(0.5/sqrt(2.0*pi))*exp(medi+0.5*stdi*stdi);
L=stdi;
Ex2b*=I_skew(10,Ri,sqrt(2.0),-1,-1.0,0.0,stdi,stdj,0.0,corij,0.0,
           0.0,Ri,Rj,0.0,L);
Ey2b=(0.5/sqrt(2.0*pi))*exp(medj+0.5*stdj*stdj);
L=stdj*corij;
Ey2b*=I_skew(9,Ri,sqrt(2.0),-1,-1.0,0.0,stdi,stdj,0.0,corij,0.0,
           0.0,Ri,Rj,0.0,L);
E12b=(0.5/sqrt(2.0*pi));
L=0.0;
E12b*=I_skew(10,Ri,sqrt(2.0),-1,-1.0,0.0,stdi,stdj,0.0,corij,0.0,
           0.0,Ri,Rj,0.0,L);

Ex2=0.5*exp(medi+0.5*stdi*stdi)*(1.0-erff((Ri-stdj)/sqrt(2.0)));
E12=0.5*(1.0-erff(Ri/sqrt(2.0)));


Ri=(log(Kp1/Si)-medi)/stdi;
Exy1b=(0.5/sqrt(2.0*pi))*
      exp(medi+medj+0.5*(stdi*stdi+stdj*stdj)+
                            stdi*stdj*corij);
L=stdi+stdj*corij;
Exy1b*=I_skew(9,Ri,sqrt(2.0),-1,-1.0,0.0,stdi,stdj,0.0,corij,0.0,
            0.0,Ri,Rj,0.0,L);
Ex1b=(0.5/sqrt(2.0*pi))*exp(medi+0.5*stdi*stdi);
L=stdi;
Ex1b*=I_skew(10,Ri,sqrt(2.0),-1,-1.0,0.0,stdi,stdj,0.0,corij,0.0,
           0.0,Ri,Rj,0.0,L);
Ey1b=(0.5/sqrt(2.0*pi))*exp(medj+0.5*stdj*stdj);
L=stdj*corij;
Ey1b*=I_skew(9,Ri,sqrt(2.0),-1,-1.0,0.0,stdi,stdj,0.0,corij,0.0,
           0.0,Ri,Rj,0.0,L);
E11b=(0.5/sqrt(2.0*pi));
L=0.0;
E11b*=I_skew(10,Ri,sqrt(2.0),-1,-1.0,0.0,stdi,stdj,0.0,corij,0.0,
           0.0,Ri,Rj,0.0,L);

Eyb=0.5*exp(medj+0.5*stdj*stdj)*(1.0-erff((Rj-stdj)/sqrt(2.0)));
```

```
        E1b=0.5*(1.0-erff(Rj/sqrt(2.0)));

        Ex1=0.5*exp(medi+0.5*stdi*stdi)*(1.0-erff((Ri-stdj)/sqrt(2.0)));
        E11=0.5*(1.0-erff(Ri/sqrt(2.0)));


        resp=(Si*Si*Sj*E2xy-Si*Si*E2x*Kcj-Si*Sj*Exy*(Kp1+Kp2)+
              Si*Ex*Kcj*(Kp1+Kp2)+Kp1*Kp2*Sj*Ey-Kp1*Kp2*Kcj*E1)/(P1*P2*Cj);
        resp+=-mp1*(-Si*Sj*Exy2b+Si*Kcj*Ex2b+Sj*Ey2b*Kp2-Kp2*Kcj*E12b)/(P2*Cj);
        resp+=-mp2*(-Si*Sj*Exy1b+Si*Kcj*Ex1b+Sj*Ey1b*Kp1-Kp1*Kcj*E11b)/(P1*Cj);
        resp+=-mc*(Si*Si*E2xpp-Si*Expp*(Kp1+Kp2)+Kp1*Kp2*E1pp)/(P1*P2);
        resp+=mp1*mp2*(Sj*Eyb-Kcj*E1b)/Cj+mp1*mcj*(-Si*Ex2+Kp2*E12)/P2;
        resp+=mp2*mcj*(-Si*Ex1+Kp1*E11)/P1-mp1*mp2*mc;
        return(resp);
}


double m_pipipj(double medi, double medj, double stdi, double stdj,
                double corij, double Si, double Sj, double Kp1, double Kp2,
                double Kpj, double P1, double P2, double Pj,
                double mp1, double mp2, double mpj)
/****************************************************************************/
/* coskewness entre puts sobre ativos i e j ;                              */
/* corij = coef. de correlacao entre os ativos i e j.                      */
/* Si,Sj = cotacao do ativo i,j em t=0;                                    */
/* Pj,P1,P2 = premio das puts sobre os ativos j e i.                       */
/* Kp1,Kp2,Kpj = strikes das puts;                                         */
/****************************************************************************/
{
        double L, E2x, Exy, E2xy, Ex, Ey, E1, Ri, Rj, resp,K,
               Exy2b, Ex2b, Ey2b, E12b, Exy1b, Ex1b, Ey1b, E11b,
               E2xpp, Expp, E1pp, Eyb, E1b, Ex2, E12, Ex1, E11;

        K=dmin(Kp1,Kp2);
        Ri=(log(K/Si)-medi)/stdi;
        Rj=(log(Kpj/Sj)-medj)/stdj;
        E2xy=(0.5/sqrt(2.0*pi))*
             exp(2.0*medi+medj+2.0*stdi*stdi+0.5*stdj*stdj+
                 2.0*stdi*stdj*corij);
        L=2.0*stdi+stdj*corij;
        E2xy*=I_skew(9,Ri,sqrt(2.0),-1,1.0,0.0,stdi,stdj,0.0,corij,
                     0.0,0.0,Ri,Rj,0.0,L);
        Exy=(0.5/sqrt(2.0*pi))*
             exp(medi+medj+0.5*(stdi*stdi+stdj*stdj)+
                                 stdi*stdj*corij);
        L=stdi+stdj*corij;
```

```
Exy*=I_skew(9,Ri,sqrt(2.0),-1,1.0,0.0,stdi,stdj,0.0,corij,0.0,
            0.0,Ri,Rj,0.0,L);
Ex=(0.5/sqrt(2.0*pi))*exp(medi+0.5*stdi*stdi);
L=stdi;
Ex*=I_skew(10,Ri,sqrt(2.0),-1,1.0,0.0,stdi,stdj,0.0,corij,0.0,
           0.0,Ri,Rj,0.0,L);
Ey=(0.5/sqrt(2.0*pi))*exp(medj+0.5*stdj*stdj);
L=stdj*corij;
Ey*=I_skew(9,Ri,sqrt(2.0),-1,1.0,0.0,stdi,stdj,0.0,corij,0.0,
           0.0,Ri,Rj,0.0,L);
E1=(0.5/sqrt(2.0*pi));
L=0.0;
E1*=I_skew(10,Ri,sqrt(2.0),-1,1.0,0.0,stdi,stdj,0.0,corij,0.0,
           0.0,Ri,Rj,0.0,L);
E2x=(0.5/sqrt(2.0*pi))*exp(2.0*(medi+stdi*stdi));
L=2.0*stdi;
E2x*=I_skew(10,Ri,sqrt(2.0),-1,1.0,0.0,stdi,stdj,0.0,corij,0.0,
            0.0,Ri,Rj,0.0,L);
E2xpp=0.5*exp(2.0*medi+2.0*stdi*stdi)*(1.0+erff((Ri-2.0*stdi)/sqrt(2.0)));
Expp=0.5*exp(medi+0.5*stdi*stdi)*(1.0+erff((Ri-stdi)/sqrt(2.0)));
Expp=0.5*(1.0+erff(Ri/sqrt(2.0)));

Ri=(log(Kp2/Si)-medi)/stdi;
Exy2b=(0.5/sqrt(2.0*pi))*
      exp(medi+medj+0.5*(stdi*stdi+stdj*stdj)+
                            stdi*stdj*corij);
L=stdi+stdj*corij;
Exy2b*=I_skew(9,Ri,sqrt(2.0),-1,1.0,0.0,stdi,stdj,0.0,corij,0.0,
            0.0,Ri,Rj,0.0,L);
Ex2b=(0.5/sqrt(2.0*pi))*exp(medi+0.5*stdi*stdi);
L=stdi;
Ex2b*=I_skew(10,Ri,sqrt(2.0),-1,1.0,0.0,stdi,stdj,0.0,corij,0.0,
           0.0,Ri,Rj,0.0,L);
Ey2b=(0.5/sqrt(2.0*pi))*exp(medj+0.5*stdj*stdj);
L=stdj*corij;
Ey2b*=I_skew(9,Ri,sqrt(2.0),-1,1.0,0.0,stdi,stdj,0.0,corij,0.0,
           0.0,Ri,Rj,0.0,L);
E12b=(0.5/sqrt(2.0*pi));
L=0.0;
E12b*=I_skew(10,Ri,sqrt(2.0),-1,1.0,0.0,stdi,stdj,0.0,corij,0.0,
           0.0,Ri,Rj,0.0,L);

Ex2=0.5*exp(medi+0.5*stdi*stdi)*(1.0+erff((Ri-stdj)/sqrt(2.0)));
E12=0.5*(1.0+erff(Ri/sqrt(2.0)));
```

```
        Ri=(log(Kp1/Si)-medi)/stdi;
        Exy1b=(0.5/sqrt(2.0*pi))*
              exp(medi+medj+0.5*(stdi*stdi+stdj*stdj)+
                                   stdi*stdj*corij);
        L=stdi+stdj*corij;
        Exy1b*=I_skew(9,Ri,sqrt(2.0),-1,1.0,0.0,stdi,stdj,0.0,corij,0.0,
                    0.0,Ri,Rj,0.0,L);
        Ex1b=(0.5/sqrt(2.0*pi))*exp(medi+0.5*stdi*stdi);
        L=stdi;
        Ex1b*=I_skew(10,Ri,sqrt(2.0),-1,1.0,0.0,stdi,stdj,0.0,corij,0.0,
                   0.0,Ri,Rj,0.0,L);
        Ey1b=(0.5/sqrt(2.0*pi))*exp(medj+0.5*stdj*stdj);
        L=stdj*corij;
        Ey1b*=I_skew(9,Ri,sqrt(2.0),-1,1.0,0.0,stdi,stdj,0.0,corij,0.0,
                   0.0,Ri,Rj,0.0,L);
        E11b=(0.5/sqrt(2.0*pi));
        L=0.0;
        E11b*=I_skew(10,Ri,sqrt(2.0),-1,1.0,0.0,stdi,stdj,0.0,corij,0.0,
                   0.0,Ri,Rj,0.0,L);

        Eyb=0.5*exp(medj+0.5*stdj*stdj)*(1.0+erff((Rj-stdj)/sqrt(2.0)));
        E1b=0.5*(1.0+erff(Rj/sqrt(2.0)));

        Ex1=0.5*exp(medi+0.5*stdi*stdi)*(1.0-erff((Ri-stdj)/sqrt(2.0)));
        E11=0.5*(1.0-erff(Ri/sqrt(2.0)));


        resp=-(Si*Si*Sj*E2xy-Si*Si*E2x*Kpj-Si*Sj*Exy*(Kp1+Kp2)+
              Si*Ex*Kpj*(Kp1+Kp2)+Kp1*Kp2*Sj*Ey-Kp1*Kp2*Kpj*E1)/(P1*P2*Pj);
        resp+=mp1*(-Si*Sj*Exy2b+Si*Kpj*Ex2b+Sj*Ey2b*Kp2-Kp2*Kpj*E12b)/(P2*Pj);
        resp+=mp2*(-Si*Sj*Exy1b+Si*Kpj*Ex1b+Sj*Ey1b*Kp1-Kp1*Kpj*E11b)/(P1*Pj);
        resp+=-mpj*(Si*Si*E2xpp-Si*Expp*(Kp1+Kp2)+Kp1*Kp2*E1pp)/(P1*P2);
        resp+=mp1*mp2*(-Sj*Eyb+Kpj*E1b)/Pj+mp1*mpj*(-Si*Ex2+Kp2*E12)/P2;
        resp+=mp2*mpj*(-Si*Ex1+Kp1*E11)/P1-mp1*mp2*mpj;
        return(resp);

}


double m_ricjck(double medi, double medj, double medk,
                double stdi, double stdj, double stdk,
                double corij, double corik, double corjk,
                double Sj, double Sk, double Kcj, double Kck,
                double Cj, double Ck, double ms, double mcj,
                double mck)
/*************************************************************************/
```

```
/* coskewness entre ativo básico i e calls sobre ativos j e k;             */
/* medi,medj,medk=media do ativo i,j e k;                                  */
/* stdi,stdj,stdk=desvio do ativo i,j e k;                                 */
/* Cj,Ck,Kcj,Kck=cotacaos e strikes das calls ;                            */
/* Sj,Sk=cotacao dos ativos básicos j e k em t=0.                          */
/* corij=correlacao entre i e j;                                           */
/**************************************************************************/
{
        double Exyz,Exy,Exz,Eyz,Ez,Ex,Ey,E1,resp,
               L,A,B,Rj,Rk,a,Exzac,Exac,Exyab,Exab,Ezc,E1c,Eyb,E1b,Exi;

        Rj=(log(Kcj/Sj)-medj)/stdj;
        Rk=(log(Kck/Sk)-medk)/stdk;
        a=1.0-corjk*corjk;
        A=corik*corjk-corij;
        B=corij*corjk-corik;
        Exyz=exp(medi+medj+medk+
                 0.5*(stdi*stdi+stdj*stdj+stdk*stdk)+
                      stdi*stdj*corij+stdi*stdk*corik+
                      stdj*stdk*corjk)/(2.0*sqrt(2.0*pi));
        L=-stdi*(B+A*corjk)/a+stdj*corjk+stdk;
        Exyz*=I_skew(11,Rk,sqrt(2.0),1,-1.0,0.0,stdi,stdj,stdk,corij,corik,
                     corjk,0.0,Rj,Rk,L);
        Exy=exp(medi+medj+0.5*(stdi*stdi+stdj*stdj)+
                  stdi*stdj*corij)/(2.0*sqrt(2.0*pi));
        L=-stdi*(B+A*corjk)/a+stdj*corjk;
        Exy*=I_skew(11,Rk,sqrt(2.0),1,-1.0,0.0,stdi,stdj,stdk,corij,corik,
                    corjk,0.0,Rj,Rk,L);
        Exz=exp(medi+medk+0.5*(stdi*stdi+stdk*stdk)+
                  stdi*stdk*corik)/(2.0*sqrt(2.0*pi));
        L=-stdi*(B+A*corjk)/a+stdk;
        Exz*=I_skew(12,Rk,sqrt(2.0),1,-1.0,0.0,stdi,stdj,stdk,corij,corik,
                    corjk,0.0,Rj,Rk,L);
        Eyz=exp(medj+medk+0.5*(stdj*stdj+stdk*stdk)+
                 stdj*stdk*corjk)/(2.0*sqrt(2.0*pi));
        L=stdj*corjk+stdk;
        Eyz*=I_skew(13,Rk,sqrt(2.0),1,-1.0,0.0,stdi,stdj,stdk,corij,corik,
                    corjk,0.0,Rj,Rk,L);
        Ex=exp(medi+0.5*(stdi*stdi))/(2.0*sqrt(2.0*pi));
        L=-stdi*(B+A*corjk)/a;
        Ex*=I_skew(12,Rk,sqrt(2.0),1,-1.0,0.0,stdi,stdj,stdk,corij,corik,
                   corjk,0.0,Rj,Rk,L);
        Ey=exp(medj+0.5*stdj*stdj)/(2.0*sqrt(2.0*pi));
        L=stdj*corjk;
        Ey*=I_skew(13,Rk,sqrt(2.0),1,-1.0,0.0,stdi,stdj,stdk,corij,corik,
                   corjk,0.0,Rj,Rk,L);
```

```
        Ez=exp(medk+0.5*stdk*stdk)/(2.0*sqrt(2.0*pi));
        L=stdk;
        Ez*=I_skew(14,Rk,sqrt(2.0),1,-1.0,0.0,stdi,stdj,stdk,corij,corik,
                   corjk,0.0,Rj,Rk,L);
        L=0.0;
        E1=I_skew(14,Rk,sqrt(2.0),1,-1.0,0.0,stdi,stdj,stdk,corij,corik,
                  corjk,0.0,Rj,Rk,L)/(2.0*sqrt(2.0*pi));

        Exzac=exp(medi+medk+0.5*(stdi*stdi+stdk*stdk)+
                  stdi*stdk*corik)/(2.0*sqrt(2.0))*
                  (1.0-erff((Rk-corik*stdi-stdj)/sqrt(2.0)));
        Exac=exp(medi+0.5*stdi*stdi)/(2.0*sqrt(2.0))*
                  (1.0-erff((Rk-corik*stdi)/sqrt(2.0)));

        Exyab=exp(medi+medj+0.5*(stdi*stdi+stdj*stdj)+
                  stdi*stdj*corij)/(2.0*sqrt(2.0))*
                  (1.0-erff((Rj-corij*stdi-stdj)/sqrt(2.0)));
        Exab=exp(medi+0.5*stdi*stdi)/(2.0*sqrt(2.0))*
                  (1.0-erff((Rj-corij*stdi)/sqrt(2.0)));

        Ezc=exp(medk+0.5*stdk*stdk)/(2.0*sqrt(2.0))*
                  (1.0-erff((Rk-stdk)/sqrt(2.0)));
        E1c=(1.0-erff(Rk/sqrt(2.0)))/(2.0*sqrt(2.0));

        Eyb=exp(medj+0.5*stdj*stdj)/(2.0*sqrt(2.0))*
                  (1.0-erff((Rj-stdj)/sqrt(2.0)));
        E1b=(1.0-erff(Rj/sqrt(2.0)))/(2.0*sqrt(2.0));
        Exi=exp(medi+0.5*stdi*stdi);

        resp=(Sj*Sk*Exyz-Sj*Exy*Kck-Kcj*Sj*Exz+Kcj*Kck*Ex)/(Cj*Ck);
        resp+=-ms*(Sj*Sk*Eyz-Sj*Kck*Ey-Kcj*Sk*Ez+Kcj*Kck*E1)/(Cj*Ck);
        resp+=-mcj*(Sk*Exzac-Kck*Exac)/Ck;
        resp+=-mck*(Sj*Exyab-Kcj*Exab)/Cj;
        resp+=ms*mcj*(Sk*Ezc-Kck*E1c)/Ck;
        resp+=ms*mck*(Sj*Eyb-Kcj*E1b)/Cj;
        resp+=mcj*mck*Exi-ms*mcj*mck;
        return(resp);
}


double m_ricjpk(double medi, double medj, double medk,
                double stdi, double stdj, double stdk,
                double corij, double corik, double corjk,
                double Sj, double Sk, double Kcj, double Kpk,
                double Cj, double Pk, double ms, double mc,
                double mp)
```

```
/*********************************************************************/
/* coskewness entre ativo básico i, call sobre ativo j e put sobre ativo k;*/
/* medi,medj,medk=media do ativo i,j e k;                              */
/* stdi,stdj,stdk=desvio do ativo i,j e k;                             */
/* Cj,Pk,Kcj,Kpk=cotacaos e strikes das call e put ;                   */
/* Sj,Sk=cotacao dos ativos básicos i,j e k em t=0.                    */
/* corij=correlacao entre i e j;                                       */
/*********************************************************************/
{
        double Exyz,Exy,Exz,Eyz,Ez,Ex,Ey,E1,resp,
               Rj,Rk,a,L,A,B,Exzac,Exac,Exyab,Exab,Ezc,E1c,Eyb,E1b,Exi;

        Rj=(log(Kcj/Sj)-medj)/stdj;
        Rk=(log(Kpk/Sk)-medk)/stdk;
        a=1.0-corjk*corjk;
        A=corik*corjk-corij;
        B=corij*corjk-corik;
        Exyz=exp(medi+medj+medk+
             0.5*(stdi*stdi+stdj*stdj+stdk*stdk)+
                  stdi*stdj*corij+stdi*stdk*corik+
                  stdj*stdk*corjk)/(2.0*sqrt(2.0*pi));
        L=-stdi*(B+A*corjk)/a+stdj*corjk+stdk;
        Exyz*=I_skew(11,Rk,sqrt(2.0),-1,-1.0,0.0,stdi,stdj,stdk,corij,corik,
                     corjk,0.0,Rj,Rk,L);
        Exy=exp(medi+medj+0.5*(stdi*stdi+stdj*stdj)+
                stdi*stdj*corij)/(2.0*sqrt(2.0*pi));
        L=-stdi*(B+A*corjk)/a+stdj*corjk;
        Exy*=I_skew(11,Rk,sqrt(2.0),-1,-1.0,0.0,stdi,stdj,stdk,corij,corik,
                    corjk,0.0,Rj,Rk,L);
        Exz=exp(medi+medk+0.5*(stdi*stdi+stdk*stdk)+
                stdi*stdk*corik)/(2.0*sqrt(2.0*pi));
        L=-stdi*(B+A*corjk)/a+stdk;
        Exz*=I_skew(12,Rk,sqrt(2.0),-1,-1.0,0.0,stdi,stdj,stdk,corij,corik,
                    corjk,0.0,Rj,Rk,L);
        Eyz=exp(medj+medk+0.5*(stdj*stdj+stdk*stdk)+
                 stdj*stdk*corjk)/(2.0*sqrt(2.0*pi));
        L=stdj*corjk+stdk;
        Eyz*=I_skew(13,Rk,sqrt(2.0),-1,-1.0,0.0,stdi,stdj,stdk,corij,corik,
                    corjk,0.0,Rj,Rk,L);
        Ex=exp(medi+0.5*(stdi*stdi))/(2.0*sqrt(2.0*pi));
        L=-stdi*(B+A*corjk)/a;
        Ex*=I_skew(12,Rk,sqrt(2.0),-1,-1.0,0.0,stdi,stdj,stdk,corij,corik,
                   corjk,0.0,Rj,Rk,L);
        Ey=exp(medj+0.5*stdj*stdj)/(2.0*sqrt(2.0*pi));
        L=stdj*corjk;
        Ey*=I_skew(13,Rk,sqrt(2.0),-1,-1.0,0.0,stdi,stdj,stdk,corij,corik,
```

```
                    corjk,0.0,Rj,Rk,L);
        Ez=exp(medk+0.5*stdk*stdk)/(2.0*sqrt(2.0*pi));
        L=stdk;
        Ez*=I_skew(14,Rk,sqrt(2.0),-1,-1.0,0.0,stdi,stdj,stdk,corij,corik,
                    corjk,0.0,Rj,Rk,L);
        L=0.0;
        E1=I_skew(14,Rk,sqrt(2.0),-1,-1.0,0.0,stdi,stdj,stdk,corij,corik,
                   corjk,0.0,Rj,Rk,L)/(2.0*sqrt(2.0*pi));

        Exzac=exp(medi+medk+0.5*(stdi*stdi+stdk*stdk)+
                   stdi*stdk*corik)/(2.0*sqrt(2.0))*
                   (1.0+erff((Rk-corik*stdi-stdj)/sqrt(2.0)));
        Exac=exp(medi+0.5*stdi*stdi)/(2.0*sqrt(2.0))*
                   (1.0+erff((Rk-corik*stdi)/sqrt(2.0)));

        Exyab=exp(medi+medj+0.5*(stdi*stdi+stdj*stdj)+
                   stdi*stdj*corij)/(2.0*sqrt(2.0))*
                   (1.0-erff((Rj-corij*stdi-stdj)/sqrt(2.0)));
        Exab=exp(medi+0.5*stdi*stdi)/(2.0*sqrt(2.0))*
                   (1.0-erff((Rj-corij*stdi)/sqrt(2.0)));

        Ezc=exp(medk+0.5*stdk*stdk)/(2.0*sqrt(2.0))*
                   (1.0+erff((Rk-stdk)/sqrt(2.0)));
        E1c=(1.0+erff(Rk/sqrt(2.0)))/(2.0*sqrt(2.0));

        Eyb=exp(medj+0.5*stdj*stdj)/(2.0*sqrt(2.0))*
                   (1.0-erff((Rj-stdj)/sqrt(2.0)));
        E1b=(1.0-erff(Rj/sqrt(2.0)))/(2.0*sqrt(2.0));
        Exi=exp(medi+0.5*stdi*stdi);

        resp=-(Sj*Sk*Exyz-Sj*Exy*Kpk-Kcj*Sj*Exz+Kcj*Kpk*Ex)/(Cj*Pk);
        resp+=ms*(Sj*Sk*Eyz-Sj*Kpk*Ey-Kcj*Sk*Ez+Kcj*Kpk*E1)/(Cj*Pk);
        resp+=mc*(Sk*Exzac-Kpk*Exac)/Pk;
        resp+=-mp*(Sj*Exyab-Kcj*Exab)/Cj;
        resp+=ms*mc*(-Sk*Ezc+Kpk*E1c)/Pk;
        resp+=ms*mp*(Sj*Eyb-Kcj*E1b)/Cj;
        resp+=mc*mp*Exi-ms*mc*mp;
        return(resp);
}


double m_ripjpk(double medi, double medj, double medk,
                double stdi, double stdj, double stdk,
                double corij, double corik, double corjk,
                double Sj, double Sk, double Kpj, double Kpk,
                double Pj, double Pk, double ms, double mpj,
```

```
              double mpk)
/***************************************************************************/
/* coskewness entre ativo básico i, call sobre ativo j e put sobre ativo k;*/
/* medi,medj,medk=media do ativo i,j e k;                                  */
/* stdi,stdj,stdk=desvio do ativo i,j e k;                                 */
/* Pj,Pk,Kpj,Kpk=cotacaos e strikes das puts ;                             */
/* Sj,Sk=cotacao dos ativos básicos j e k em t=0.                          */
/* corij=correlacao entre i e j;                                           */
/***************************************************************************/
{
        double Exyz,Exy,Exz,Eyz,Ez,Ex,Ey,E1,resp,
               Rj,Rk,a,L,A,B,Exzac,Exac,Exyab,Exab,Ezc,E1c,Eyb,E1b,Exi;

        Rj=(log(Kpj/Sj)-medj)/stdj;
        Rk=(log(Kpk/Sk)-medk)/stdk;
        a=1.0-corjk*corjk;
        A=corik*corjk-corij;
        B=corij*corjk-corik;
        Exyz=exp(medi+medj+medk+
                 0.5*(stdi*stdi+stdj*stdj+stdk*stdk)+
                      stdi*stdj*corij+stdi*stdk*corik+
                      stdj*stdk*corjk)/(2.0*sqrt(2.0*pi));
        L=-stdi*(B+A*corjk)/a+stdj*corjk+stdk;
        Exyz*=I_skew(11,Rk,sqrt(2.0),-1,1.0,0.0,stdi,stdj,stdk,corij,corik,
                     corjk,0.0,Rj,Rk,L);
        Exy=exp(medi+medj+0.5*(stdi*stdi+stdj*stdj)+
                stdi*stdj*corij)/(2.0*sqrt(2.0*pi));
        L=-stdi*(B+A*corjk)/a+stdj*corjk;
        Exy*=I_skew(11,Rk,sqrt(2.0),-1,1.0,0.0,stdi,stdj,stdk,corij,corik,
                    corjk,0.0,Rj,Rk,L);
        Exz=exp(medi+medk+0.5*(stdi*stdi+stdk*stdk)+
                stdi*stdk*corik)/(2.0*sqrt(2.0*pi));
        L=-stdi*(B+A*corjk)/a+stdk;
        Exz*=I_skew(12,Rk,sqrt(2.0),-1,1.0,0.0,stdi,stdj,stdk,corij,corik,
                    corjk,0.0,Rj,Rk,L);
        Eyz=exp(medj+medk+0.5*(stdj*stdj+stdk*stdk)+
                 stdj*stdk*corjk)/(2.0*sqrt(2.0*pi));
        L=stdj*corjk+stdk;
        Eyz*=I_skew(13,Rk,sqrt(2.0),-1,1.0,0.0,stdi,stdj,stdk,corij,corik,
                    corjk,0.0,Rj,Rk,L);
        Ex=exp(medi+0.5*(stdi*stdi))/(2.0*sqrt(2.0*pi));
        L=-stdi*(B+A*corjk)/a;
        Ex*=I_skew(12,Rk,sqrt(2.0),-1,1.0,0.0,stdi,stdj,stdk,corij,corik,
                   corjk,0.0,Rj,Rk,L);
        Ey=exp(medj+0.5*stdj*stdj)/(2.0*sqrt(2.0*pi));
        L=stdj*corjk;
```

```
        Ey*=I_skew(13,Rk,sqrt(2.0),-1,1.0,0.0,stdi,stdj,stdk,corij,corik,
                   corjk,0.0,Rj,Rk,L);
        Ez=exp(medk+0.5*stdk*stdk)/(2.0*sqrt(2.0*pi));
        L=stdk;
        Ez*=I_skew(14,Rk,sqrt(2.0),-1,1.0,0.0,stdi,stdj,stdk,corij,corik,
                   corjk,0.0,Rj,Rk,L);
        L=0.0;
        E1=I_skew(14,Rk,sqrt(2.0),-1,1.0,0.0,stdi,stdj,stdk,corij,corik,
                  corjk,0.0,Rj,Rk,L)/(2.0*sqrt(2.0*pi));


        Exzac=exp(medi+medk+0.5*(stdi*stdi+stdk*stdk)+
                  stdi*stdk*corik)/(2.0*sqrt(2.0))*
                  (1.0+erff((Rk-corik*stdi-stdj)/sqrt(2.0)));
        Exac=exp(medi+0.5*stdi*stdi)/(2.0*sqrt(2.0))*
                  (1.0+erff((Rk-corik*stdi)/sqrt(2.0)));

        Exyab=exp(medi+medj+0.5*(stdi*stdi+stdj*stdj)+
                  stdi*stdj*corij)/(2.0*sqrt(2.0))*
                  (1.0+erff((Rj-corij*stdi-stdj)/sqrt(2.0)));
        Exab=exp(medi+0.5*stdi*stdi)/(2.0*sqrt(2.0))*
                  (1.0+erff((Rj-corij*stdi)/sqrt(2.0)));

        Ezc=exp(medk+0.5*stdk*stdk)/(2.0*sqrt(2.0))*
                  (1.0+erff((Rk-stdk)/sqrt(2.0)));
        E1c=(1.0+erff(Rk/sqrt(2.0)))/(2.0*sqrt(2.0));

        Eyb=exp(medj+0.5*stdj*stdj)/(2.0*sqrt(2.0))*
                  (1.0+erff((Rj-stdj)/sqrt(2.0)));
        E1b=(1.0+erff(Rj/sqrt(2.0)))/(2.0*sqrt(2.0));
        Exi=exp(medi+0.5*stdi*stdi);

        resp=(Sj*Sk*Exyz-Sj*Exy*Kpk-Kpj*Sj*Exz+Kpj*Kpk*Ex)/(Pj*Pk);
        resp+=-ms*(Sj*Sk*Eyz-Sj*Kpk*Ey-Kpj*Sk*Ez+Kpj*Kpk*E1)/(Pj*Pk);
        resp+=mpj*(Sk*Exzac-Kpk*Exac)/Pk;
        resp+=mpk*(Sj*Exyab-Kpj*Exab)/Pj;
        resp+=ms*mpj*(-Sk*Ezc+Kpk*E1c)/Pk;
        resp+=ms*mpk*(-Sj*Eyb+Kpj*E1b)/Pj;
        resp+=mpj*mpk*Exi-ms*mpj*mpk;
        return(resp);
}


/**************************************************************************/
```

```
/******    expressoes de assimetria que envolvem integrais triplas    *****/
/***********************************************************************/

double m_cicjck(double medi, double medj, double medk,
                double stdi, double stdj, double stdk,
                double corij, double corik, double corjk,
                double Si, double Sj, double Sk,
                double Kci, double Kcj, double Kck,
                double Ci, double Cj, double Ck, double *x, double *w,
                double mci, double mcj, double mck )
/***********************************************************************/
/* medi,medj,medk=media dos ativos i,j e k;                            */
/* stdi,stdj,stdk=desvio dos ativos i,j e k;                           */
/* Ci,Cj,Ck,Kci,Kcj,Kck=premios e strikes das calls ;                  */
/* Si,Sj,Sk=cotacao dos ativos básicos i,j e k em t=0.                 */
/* corij=correlacao entre i e j;                                       */
/* x=vetor que contém os nos do grid gaussiano;                        */
/* w=vetor de pesos do grid gaussiano;                                 */
/***********************************************************************/
{
        double Exyz,Exy,Exz,Eyz,Ez,Ex,Ey,E1,resp,
               Ri,Rj,Rk,c,Gb,G, L, T, B, C,
               Eyzbc,Eybc,Ezbc,E1bc,
               Exzac,Exac,Ezac,E1ac,
               Exyab,Exab,Eyab,E1ab,
               Ezc,E1c,Eyb,E1b,Exa,E1a;

        int l;

        Exyz=0.0;
        c=1.0-corij*corij;
        C=corij*corik-corjk;
        B=corij*corjk-corik;
        Ri=(log(Kci/Si)-medi)/stdi;
        Rj=(log(Kcj/Sj)-medj)/stdj;
        Rk=(log(Kck/Sk)-medk)/stdk;
        T=(stdk*B-stdi*c+corij*(-stdj*c+stdk*C))/c;
        L=(Ri+T)/sqrt(2.0);
        Gb=(1.0/sqrt(2.0*c))*(Rj-stdj*c+stdk*C-corij*sqrt(2.0)*L+corij*T);
        for (l=0;l<16;l++){     /* 16 e'o numero de pontos de gauss */
               G=Gb-(x[l]*sqrt(2.0)*corij)*(1.0/sqrt(2.0*c));
               Exyz+=w[l]*exp(-2.0*x[l]*L)*
                     I_skew(15,G,1.0,+1,-1.0,x[l],stdi,stdj,stdk,
                            corij,corik,corjk,Ri,Rj,Rk,0.0);
               /* I calcula a integral de G  ate' +(-)infinito da funcao
               func, que depende de phi (x[i] e'um ponto do grid de phi
```

```
        indiretamente e de psi. */
}
Exyz*=exp(medi+medj+medk
          +0.5*(stdi*stdi+stdj*stdj+stdk*stdk)
          +stdi*stdj*corij+stdi*stdk*corik
          +stdj*stdk*corjk-sqr(L))/(2.0*pi);
Exy=0.0;
T=-stdi+corij*-stdj;
L=(Ri+T)/sqrt(2.0);
Gb=(1.0/sqrt(2.0*c))*(Rj-stdj*c-corij*sqrt(2.0)*L+corij*T);
for (l=0;l<16;l++){     /* 16 e'o numero de pontos de gauss */
        G=Gb-(x[l]*sqrt(2.0)*corij)*(1.0/sqrt(2.0*c));
        Exy+=w[l]*exp(-2.0*x[l]*L)
             *I_skew(16,G,1.0,+1,-1.0,x[l],stdi,stdj,stdk,corij,
                     corik,corjk,Ri,Rj,Rk,0.0);
}
Exy*=exp(medi+medj+0.5*(stdi*stdi+stdj*stdj)
         +stdi*stdj*corij-sqr(L))/(2.0*pi);
Exz=0.0;
T=(stdk*B-stdi*c+corij*stdk*C)/c;
L=(Ri+T)/sqrt(2.0);
Gb=(1.0/sqrt(2.0*c))*(Rj+stdk*C-corij*sqrt(2.0)*L+corij*T);
for (l=0;l<16;l++){     /* 16 e'o numero de pontos de gauss */
        G=Gb-(x[l]*sqrt(2.0)*corij)*(1.0/sqrt(2.0*c));
        Exz+=w[l]*exp(-2.0*x[l]*L)
             *I_skew(17,G,1.0,+1,-1.0,x[l],stdi,stdj,stdk,corij,
                     corik,corjk,Ri,Rj,Rk,0.0);
}
Exz*=exp(medi+medk+0.5*(stdi*stdi+stdk*stdk)
         +stdi*stdk*corik-sqr(L))/(2.0*pi);
Eyz=0.0;
T=(stdk*B+corij*(-stdj*c+stdk*C))/c;
L=(Ri+T)/sqrt(2.0);
Gb=(1.0/sqrt(2.0*c))*(Rj-stdj*c+stdk*C-corij*sqrt(2.0)*L+corij*T);
for (l=0;l<16;l++){     /* 16 e'o numero de pontos de gauss */
        G=Gb-(x[l]*sqrt(2.0)*corij)*(1.0/sqrt(2.0*c));
        Eyz+=w[l]*exp(-2.0*x[l]*L)
             *I_skew(18,G,1.0,+1,-1.0,x[l],stdi,stdj,stdk,corij,
                     corik,corjk,Ri,Rj,Rk,0.0);
}
Eyz*=exp(medj+medk+0.5*(stdj*stdj+stdk*stdk)+
         stdj*stdk*corjk-sqr(L))/(2.0*pi);
Ex=0.0;
T=-stdi;
L=(Ri+T)/sqrt(2.0);
Gb=(1.0/sqrt(2.0*c))*(Rj-corij*sqrt(2.0)*L+corij*T);
```

```
for (l=0;l<16;l++){     /* 16 e'o numero de pontos de gauss */
        G=Gb-(x[l]*sqrt(2.0)*corij)*(1.0/sqrt(2.0*c));
        Ex+=w[l]*exp(-2.0*x[l]*L)
            *I_skew(19,G,1.0,+1,-1.0,x[l],stdi,stdj,stdk,corij,
                    corik,corjk,Ri,Rj,Rk,0.0);
}
Ex*=exp(medi+0.5*(stdi*stdi)-sqr(L))/(2.0*pi);
Ey=0.0;
T=corij*-stdj;
L=(Ri+T)/sqrt(2.0);
Gb=(1.0/sqrt(2.0*c))*(Rj-stdj*c-corij*sqrt(2.0)*L+corij*T);
for (l=0;l<16;l++){     /* 16 e'o numero de pontos de gauss */
        G=Gb-(x[l]*sqrt(2.0)*corij)*(1.0/sqrt(2.0*c));
        Ey+=w[l]*exp(-2.0*x[l]*L)
            *I_skew(20,G,1.0,+1,-1.0,x[l],stdi,stdj,stdk,corij,
                    corik,corjk,Ri,Rj,Rk,0.0);
}
Ey*=exp(medj+0.5*stdj*stdj-sqr(L))/(2.0*pi);
Ez=0.0;
T=(stdk*B+corij*stdk*C)/c;
L=(Ri+T)/sqrt(2.0);
Gb=(1.0/sqrt(2.0*c))*(Rj+stdk*C-corij*sqrt(2.0)*L+corij*T);
for (l=0;l<16;l++){     /* 16 e'o numero de pontos de gauss */
        G=Gb-(x[l]*sqrt(2.0)*corij)*(1.0/sqrt(2.0*c));
        Ez+=w[l]*exp(-2.0*x[l]*L)
            *I_skew(21,G,1.0,+1,-1.0,x[l],stdi,stdj,stdk,corij,
                    corik,corjk,Ri,Rj,Rk,0.0);
}
Ez*=exp(medk+0.5*stdk*stdk-sqr(L))/(2.0*pi);
E1=0.0;
T=0.0;
L=Ri/sqrt(2.0);
Gb=(1.0/sqrt(2.0*c))*(Rj-corij*sqrt(2.0)*L);
for (l=0;l<16;l++){     /* 16 e'o numero de pontos de gauss */
        G=Gb-(x[l]*sqrt(2.0)*corij)*(1.0/sqrt(2.0*c));
        E1+=w[l]*exp(-2.0*x[l]*L)
            *I_skew(22,G,1.0,+1,-1.0,x[l],stdi,stdj,stdk,corij,
                    corik,corjk,Ri,Rj,Rk,0.0);
}
E1*=exp(-sqr(L))/(2.0*pi);

Exyab=(0.5/sqrt(2.0*pi))*
       exp(medi+medj+0.5*(stdi*stdi+stdj*stdj)+
                           stdi*stdj*corij);
L=stdi+stdj*corij;
Exyab*=I_skew(9,Ri,sqrt(2.0),1,-1.0,0.0,stdi,stdj,0.0,corij,0.0,
```

```
              0.0,Ri,Rj,0.0,L);
Exab=(0.5/sqrt(2.0*pi))*exp(medi+0.5*stdi*stdi);
L=stdi;
Exab*=I_skew(10,Ri,sqrt(2.0),1,-1.0,0.0,stdi,stdj,0.0,corij,0.0,
            0.0,Ri,Rj,0.0,L);
Eyab=(0.5/sqrt(2.0*pi))*exp(medj+0.5*stdj*stdj);
L=stdj*corij;
Eyab*=I_skew(9,Ri,sqrt(2.0),1,-1.0,0.0,stdi,stdj,0.0,corij,0.0,
              0.0,Ri,Rj,0.0,L);
E1ab=(0.5/sqrt(2.0*pi));
L=0.0;
E1ab*=I_skew(10,Ri,sqrt(2.0),1,-1.0,0.0,stdi,stdj,0.0,corij,0.0,
              0.0,Ri,Rj,0.0,L);

Eyzbc=(0.5/sqrt(2.0*pi))*
       exp(medk+medj+0.5*(stdk*stdk+stdj*stdj)+
                          stdk*stdj*corjk);
L=stdk+stdj*corjk;
Eyzbc*=I_skew(9,Rk,sqrt(2.0),1,-1.0,0.0,stdi,stdj,0.0,0.0,0.0,
              corjk,Ri,Rj,0.0,L);
Ezbc=(0.5/sqrt(2.0*pi))*exp(medk+0.5*stdk*stdk);
L=stdk;
Ezbc*=I_skew(10,Rk,sqrt(2.0),1,-1.0,0.0,stdi,stdj,0.0,0.0,0.0,
            corjk,Ri,Rj,0.0,L);
Eybc=(0.5/sqrt(2.0*pi))*exp(medj+0.5*stdj*stdj);
L=stdj*corjk;
Eybc*=I_skew(9,Rk,sqrt(2.0),1,-1.0,0.0,stdi,stdj,0.0,0.0,0.0,
              corjk,Ri,Rj,0.0,L);
E1bc=(0.5/sqrt(2.0*pi));
L=0.0;
E1bc*=I_skew(10,Rk,sqrt(2.0),1,-1.0,0.0,stdi,stdj,0.0,0.0,0.0,
              corjk,Ri,Rj,0.0,L);

Exzac=(0.5/sqrt(2.0*pi))*
       exp(medk+medi+0.5*(stdk*stdk+stdi*stdi)+
                          stdk*stdi*corik);
L=stdk+stdi*corik;
Exzac*=I_skew(9,Rk,sqrt(2.0),1,-1.0,0.0,stdi,stdj,0.0,0.0,corik,
              0.0,Ri,Rj,0.0,L);
Ezac=(0.5/sqrt(2.0*pi))*exp(medk+0.5*stdk*stdk);
L=stdk;
Ezac*=I_skew(10,Rk,sqrt(2.0),1,-1.0,0.0,stdi,stdj,0.0,0.0,corik,
            0.0,Ri,Rj,0.0,L);
Exac=(0.5/sqrt(2.0*pi))*exp(medi+0.5*stdi*stdi);
L=stdi*corik;
Exac*=I_skew(9,Rk,sqrt(2.0),1,-1.0,0.0,stdi,stdj,0.0,0.0,corik,
```

```
                  0.0,Ri,Rj,0.0,L);
        E1ac=(0.5/sqrt(2.0*pi));
        L=0.0;
        E1ac*=I_skew(10,Rk,sqrt(2.0),1,-1.0,0.0,stdi,stdj,0.0,0.0,corik,
                  0.0,Ri,Rj,0.0,L);

        Ezc=0.5*exp(medk+0.5*stdk*stdk)*(1.0-erff((Rk-stdk)/sqrt(2.0)));
        Ezc=0.5*(1.0-erff(Rk/sqrt(2.0)));

        Eyb=0.5*exp(medj+0.5*stdj*stdj)*(1.0-erff((Rj-stdj)/sqrt(2.0)));
        E1b=0.5*(1.0-erff(Rj/sqrt(2.0)));

        Exa=0.5*exp(medi+0.5*stdi*stdi)*(1.0-erff((Ri-stdi)/sqrt(2.0)));
        E1a=0.5*(1.0-erff(Ri/sqrt(2.0)));

        resp=(Si*Sj*Sk*Exyz-Si*Sj*Exy*Kck-Si*Sk*Exz*Kcj+Si*Ex*Kcj*Kck
               +Sj*Ey*Kci*Kck+Sk*Ez*Kcj*Kci-Kci*Kcj*Kck*E1)/(Ci*Cj*Ck);
        resp+=-mci*(Sj*Sk*Eyzbc-Sj*Kck*Eybc-Kcj*Sk*Ezbc+Kcj*Kck*E1bc)/(Cj*Ck);
        resp+=-mcj*(Si*Sk*Exzac-Si*Kck*Exac-Kci*Sk*Ezac+Kci*Kck*E1ac)/(Ci*Ck);
        resp+=-mck*(Si*Sj*Exyab-Si*Kcj*Exab-Kci*Sj*Eyab+Kci*Kcj*E1ab)/(Ci*Cj);
        resp+=mci*mcj*(Sk*Ezc-Kck*E1c)/Ck+mci*mck*(Sj*Eyb-Kcj*E1b)/Cj;
        resp+=mcj*mck*(Si*Exa-Kci*E1a)/Ci-mci*mcj*mck;
        return(resp);
}


double m_cicjpk(double medi, double medj, double medk,
                double stdi, double stdj, double stdk,
                double corij, double corik, double corjk,
                double Si, double Sj, double Sk,
                double Kci, double Kcj, double Kpk,
                double Ci, double Cj, double Pk, double *x, double *w,
                double mci, double mcj, double mp)
/*************************************************************************/
/* medi,medj,medk=media do ativo i,j e k;                               */
/* stdi,stdj,stdk=desvio do ativo i,j e k;                              */
/* Ci,Cj,Kci,Kcj=premios e strikes das calls ;                          */
/* Pk,Kpk=cotacao e strike da put;                                      */
/* Si,Sj,Sk=cotacao dos ativos básicos i,j e k em t=0.                  */
/* corij=correlacao entre i e j;                                        */
/*************************************************************************/
{
        double Exyz,Exy,Exz,Eyz,Ez,Ex,Ey,E1,resp,
               Ri,Rj,Rk,c,G, Gb, C, B, L, T;
        int l;
```

```
Exyz=0.0;
c=1.0-corij*corij;
C=corij*corik-corjk;
B=corij*corjk-corik;
Ri=(log(Kci/Si)-medi)/stdi;
Rj=(log(Kcj/Sj)-medj)/stdj;
Rk=(log(Kpk/Sk)-medk)/stdk;
T=(stdk*B-stdi*c+corij*(-stdj*c+stdk*C))/c;
L=(Ri+T)/sqrt(2.0);
Gb=(1.0/sqrt(2.0*c))*(Rj-stdj*c+stdk*C-corij*sqrt(2.0)*L+corij*T);
for (l=0;l<16;l++){     /* 16 e'o numero de pontos de gauss */
        G=Gb-(x[l]*sqrt(2.0)*corij)*(1.0/sqrt(2.0*c));
        Exyz+=w[l]*exp(-2.0*x[l]*L)
              *I_skew(15,G,1.0,+1,1.0,x[l],stdi,stdj,stdk,
                     corij,corik,corjk,Ri,Rj,Rk,0.0);
        /* I calcula a integral de G  ate' +(-)infinito da funcao
        func, que depende de phi (x[i] e'um ponto do grid de phi
        indiretamente e de psi. */
}
Exyz*=exp(medi+medj+medk
          +0.5*(stdi*stdi+stdj*stdj+stdk*stdk)
          +stdi*stdj*corij+stdi*stdk*corik
          +stdj*stdk*corjk-sqr(L))/(2.0*pi);
Exy=0.0;
T=-stdi+corij*-stdj;
L=(Ri+T)/sqrt(2.0);
Gb=(1.0/sqrt(2.0*c))*(Rj-stdj*c-corij*sqrt(2.0)*L+corij*T);
for (l=0;l<16;l++){     /* 16 e'o numero de pontos de gauss */
        G=Gb-(x[l]*sqrt(2.0)*corij)*(1.0/sqrt(2.0*c));
        Exy+=w[l]*exp(-2.0*x[l]*L)
             *I_skew(16,G,1.0,+1,1.0,x[l],stdi,stdj,stdk,corij,
                    corik,corjk,Ri,Rj,Rk,0.0);
}
Exy*=exp(medi+medj+0.5*(stdi*stdi+stdj*stdj)
         +stdi*stdj*corij-sqr(L))/(2.0*pi);
Exz=0.0;
T=(stdk*B-stdi*c+corij*stdk*C)/c;
L=(Ri+T)/sqrt(2.0);
Gb=(1.0/sqrt(2.0*c))*(Rj+stdk*C-corij*sqrt(2.0)*L+corij*T);
for (l=0;l<16;l++){     /* 16 e'o numero de pontos de gauss */
        G=Gb-(x[l]*sqrt(2.0)*corij)*(1.0/sqrt(2.0*c));
        Exz+=w[l]*exp(-2.0*x[l]*L)
             *I_skew(17,G,1.0,+1,1.0,x[l],stdi,stdj,stdk,corij,
                    corik,corjk,Ri,Rj,Rk,0.0);
}
```

```
Exz*=exp(medi+medk+0.5*(stdi*stdi+stdk*stdk)
         +stdi*stdk*corik-sqr(L))/(2.0*pi);
Eyz=0.0;
T=(stdk*B+corij*(-stdj*c+stdk*C))/c;
L=(Ri+T)/sqrt(2.0);
Gb=(1.0/sqrt(2.0*c))*(Rj-stdj*c+stdk*C-corij*sqrt(2.0)*L+corij*T);
for (l=0;l<16;l++){     /* 16 e'o numero de pontos de gauss */
        G=Gb-(x[l]*sqrt(2.0)*corij)*(1.0/sqrt(2.0*c));
        Eyz+=w[l]*exp(-2.0*x[l]*L)
             *I_skew(18,G,1.0,+1,1.0,x[l],stdi,stdj,stdk,corij,
                     corik,corjk,Ri,Rj,Rk,0.0);
}
Eyz*=exp(medj+medk+0.5*(stdj*stdj+stdk*stdk)+
         stdj*stdk*corjk-sqr(L))/(2.0*pi);
Ex=0.0;
T=-stdi;
L=(Ri+T)/sqrt(2.0);
Gb=(1.0/sqrt(2.0*c))*(Rj-corij*sqrt(2.0)*L+corij*T);
for (l=0;l<16;l++){     /* 16 e'o numero de pontos de gauss */
        G=Gb-(x[l]*sqrt(2.0)*corij)*(1.0/sqrt(2.0*c));
        Ex+=w[l]*exp(-2.0*x[l]*L)
            *I_skew(19,G,1.0,+1,1.0,x[l],stdi,stdj,stdk,corij,
                    corik,corjk,Ri,Rj,Rk,0.0);
}
Ex*=exp(medi+0.5*(stdi*stdi)-sqr(L))/(2.0*pi);
Ey=0.0;
T=corij*-stdj;
L=(Ri+T)/sqrt(2.0);
Gb=(1.0/sqrt(2.0*c))*(Rj-stdj*c-corij*sqrt(2.0)*L+corij*T);
for (l=0;l<16;l++){     /* 16 e'o numero de pontos de gauss */
        G=Gb-(x[l]*sqrt(2.0)*corij)*(1.0/sqrt(2.0*c));
        Ey+=w[l]*exp(-2.0*x[l]*L)
            *I_skew(20,G,1.0,+1,1.0,x[l],stdi,stdj,stdk,corij,
                    corik,corjk,Ri,Rj,Rk,0.0);
}
Ey*=exp(medj+0.5*stdj*stdj-sqr(L))/(2.0*pi);
Ez=0.0;
T=(stdk*B+corij*stdk*C)/c;
L=(Ri+T)/sqrt(2.0);
Gb=(1.0/sqrt(2.0*c))*(Rj+stdk*C-corij*sqrt(2.0)*L+corij*T);
for (l=0;l<16;l++){     /* 16 e'o numero de pontos de gauss */
        G=Gb-(x[l]*sqrt(2.0)*corij)*(1.0/sqrt(2.0*c));
        Ez+=w[l]*exp(-2.0*x[l]*L)
            *I_skew(21,G,1.0,+1,1.0,x[l],stdi,stdj,stdk,corij,
                    corik,corjk,Ri,Rj,Rk,0.0);
}
```

```
Ez*=exp(medk+0.5*stdk*stdk-sqr(L))/(2.0*pi);
E1=0.0;
T=0.0;
L=Ri/sqrt(2.0);
Gb=(1.0/sqrt(2.0*c))*(Rj-corij*sqrt(2.0)*L);
for (l=0;l<16;l++){     /* 16 e'o numero de pontos de gauss */
        G=Gb-(x[l]*sqrt(2.0)*corij)*(1.0/sqrt(2.0*c));
        E1+=w[l]*exp(-2.0*x[l]*L)
            *I_skew(22,G,1.0,+1,1.0,x[l],stdi,stdj,stdk,corij,
                    corik,corjk,Ri,Rj,Rk,0.0);
}
E1*=exp(-sqr(L))/(2.0*pi);
Exyab=(0.5/sqrt(2.0*pi))*
       exp(medi+medj+0.5*(stdi*stdi+stdj*stdj)+
                           stdi*stdj*corij);
L=stdi+stdj*corij;
Exyab*=I_skew(9,Ri,sqrt(2.0),1,-1.0,0.0,stdi,stdj,0.0,corij,0.0,
              0.0,Ri,Rj,0.0,L);
Exab=(0.5/sqrt(2.0*pi))*exp(medi+0.5*stdi*stdi);
L=stdi;
Exab*=I_skew(10,Ri,sqrt(2.0),1,-1.0,0.0,stdi,stdj,0.0,corij,0.0,
            0.0,Ri,Rj,0.0,L);
Eyab=(0.5/sqrt(2.0*pi))*exp(medj+0.5*stdj*stdj);
L=stdj*corij;
Eyab*=I_skew(9,Ri,sqrt(2.0),1,-1.0,0.0,stdi,stdj,0.0,corij,0.0,
              0.0,Ri,Rj,0.0,L);
E1ab=(0.5/sqrt(2.0*pi));
L=0.0;
E1ab*=I_skew(10,Ri,sqrt(2.0),1,-1.0,0.0,stdi,stdj,0.0,corij,0.0,
              0.0,Ri,Rj,0.0,L);

Eyzbc=(0.5/sqrt(2.0*pi))*
       exp(medk+medj+0.5*(stdk*stdk+stdj*stdj)+
                           stdk*stdj*corjk);
L=stdk+stdj*corjk;
Eyzbc*=I_skew(9,Rk,sqrt(2.0),-1,-1.0,0.0,stdi,stdj,0.0,0.0,0.0,
              corjk,Ri,Rj,0.0,L);
Ezbc=(0.5/sqrt(2.0*pi))*exp(medk+0.5*stdk*stdk);
L=stdk;
Ezbc*=I_skew(10,Rk,sqrt(2.0),-1,-1.0,0.0,stdi,stdj,0.0,0.0,0.0,
            corjk,Ri,Rj,0.0,L);
Eybc=(0.5/sqrt(2.0*pi))*exp(medj+0.5*stdj*stdj);
L=stdj*corjk;
Eybc*=I_skew(9,Rk,sqrt(2.0),-1,-1.0,0.0,stdi,stdj,0.0,0.0,0.0,
              corjk,Ri,Rj,0.0,L);
E1bc=(0.5/sqrt(2.0*pi));
```

```
        L=0.0;
        E1bc*=I_skew(10,Rk,sqrt(2.0),-1,-1.0,0.0,stdi,stdj,0.0,0.0,0.0,
                     corjk,Ri,Rj,0.0,L);

        Exzac=(0.5/sqrt(2.0*pi))*
               exp(medk+medi+0.5*(stdk*stdk+stdi*stdi)+
                                   stdk*stdi*corik);
        L=stdk+stdi*corik;
        Exzac*=I_skew(9,Rk,sqrt(2.0),-1,-1.0,0.0,stdi,stdj,0.0,0.0,corik,
                     0.0,Ri,Rj,0.0,L);
        Ezac=(0.5/sqrt(2.0*pi))*exp(medk+0.5*stdk*stdk);
        L=stdk;
        Ezac*=I_skew(10,Rk,sqrt(2.0),-1,-1.0,0.0,stdi,stdj,0.0,0.0,corik,
                   0.0,Ri,Rj,0.0,L);
        Exac=(0.5/sqrt(2.0*pi))*exp(medi+0.5*stdi*stdi);
        L=stdi*corik;
        Exac*=I_skew(9,Rk,sqrt(2.0),-1,-1.0,0.0,stdi,stdj,0.0,0.0,corik,
                     0.0,Ri,Rj,0.0,L);
        E1ac=(0.5/sqrt(2.0*pi));
        L=0.0;
        E1ac*=I_skew(10,Rk,sqrt(2.0),-1,-1.0,0.0,stdi,stdj,0.0,0.0,corik,
                     0.0,Ri,Rj,0.0,L);

        Ezc=0.5*exp(medk+0.5*stdk*stdk)*(1.0+erff((Rk-stdk)/sqrt(2.0)));
        E1c=0.5*(1.0+erff(Rk/sqrt(2.0)));

        Eyb=0.5*exp(medj+0.5*stdj*stdj)*(1.0-erff((Rj-stdj)/sqrt(2.0)));
        E1b=0.5*(1.0-erff(Rj/sqrt(2.0)));

        Exa=0.5*exp(medi+0.5*stdi*stdi)*(1.0-erff((Ri-stdi)/sqrt(2.0)));
        E1a=0.5*(1.0-erff(Ri/sqrt(2.0)));

        resp=-(Si*Sj*Sk*Exyz-Si*Sj*Exy*Kpk-Si*Sk*Exz*Kcj+Si*Ex*Kcj*Kpk
               +Sj*Ey*Kci*Kpk+Sk*Ez*Kcj*Kci-Kci*Kcj*Kpk*E1)/(Ci*Cj*Pk);
        resp+=mci*(Sj*Sk*Eyzbc-Sj*Kpk*Eybc-Kcj*Sk*Ezbc+Kcj*Kpk*E1bc)/(Cj*Pk);
        resp+=mcj*(Si*Sk*Exzac-Si*Kpk*Exac-Kci*Sk*Ezac+Kci*Kpk*E1ac)/(Ci*Pk);
        resp+=-mpk*(Si*Sj*Exyab-Si*Kcj*Exab-Kci*Sj*Eyab+Kci*Kcj*E1ab)/(Ci*Cj);
        resp+=mci*mcj*(-Sk*Ezc+Kpk*E1c)/Pk+mci*mpk*(Sj*Eyb-Kcj*E1b)/Cj;
        resp+=mcj*mpk*(Si*Exa-Kci*E1a)/Ci-mci*mcj*mpk;
        return(resp);

}


double m_cipjpk(double medi, double medj, double medk,
```

```
            double stdi, double stdj, double stdk,
            double corij, double corik, double corjk,
            double Si, double Sj, double Sk,
            double Kci, double Kpj, double Kpk,
            double Ci, double Pj, double Pk, double *x, double *w,
            double mc, double mpj, double mpk)
/**********************************************************************/
/* medi,medj,medk=media do ativo i,j e k;                             */
/* stdi,stdj,stdk=desvio do ativo i,j e k;                            */
/* Ci,Kci=cotacao e strike da call ;                                  */
/* Pj,Pk,Kpj,Kpk=cotacao e strike das puts;                           */
/* Si,Sj,Sk=cotacao dos ativos básicos i,j e k em t=0.                */
/* corij=correlacao entre i e j;                                      */
/**********************************************************************/
{
        double Exyz,Exy,Exz,Eyz,Ez,Ex,Ey,E1,resp,
               Ri,Rj,Rk,c,G, Gb, L, T, B, C;
        int l;


        Exyz=0.0;
        c=1.0-corij*corij;
        C=corij*corik-corjk;
        B=corij*corjk-corik;
        Ri=(log(Kci/Si)-medi)/stdi;
        Rj=(log(Kpj/Sj)-medj)/stdj;
        Rk=(log(Kpk/Sk)-medk)/stdk;
        T=(stdk*B-stdi*c+corij*(-stdj*c+stdk*C))/c;
        L=(Ri+T)/sqrt(2.0);
        Gb=(1.0/sqrt(2.0*c))*(Rj-stdj*c+stdk*C-corij*sqrt(2.0)*L+corij*T);
        for (l=0;l<16;l++){     /* 16 e'o numero de pontos de gauss */
                G=Gb-(x[l]*sqrt(2.0)*corij)*(1.0/sqrt(2.0*c));
                Exyz+=w[l]*exp(-2.0*x[l]*L)
                      *I_skew(15,G,1.0,-1,1.0,x[l],stdi,stdj,stdk,
                              corij,corik,corjk,Ri,Rj,Rk,0.0);
                /* I calcula a integral de G  ate' +(-)infinito da funcao
                func, que depende de phi (x[i] e'um ponto do grid de phi
                indiretamente e de psi. */
        }
        Exyz*=exp(medi+medj+medk
                  +0.5*(stdi*stdi+stdj*stdj+stdk*stdk)
                  +stdi*stdj*corij+stdi*stdk*corik
                  +stdj*stdk*corjk-sqr(L))/(2.0*pi);
        Exy=0.0;
        T=-stdi+corij*-stdj;
        L=(Ri+T)/sqrt(2.0);
```

```
Gb=(1.0/sqrt(2.0*c))*(Rj-stdj*c-corij*sqrt(2.0)*L+corij*T);
for (l=0;l<16;l++){     /* 16 e'o numero de pontos de gauss */
        G=Gb-(x[l]*sqrt(2.0)*corij)*(1.0/sqrt(2.0*c));
        Exy+=w[l]*exp(-2.0*x[l]*L)
             *I_skew(16,G,1.0,-1,1.0,x[l],stdi,stdj,stdk,corij,
                     corik,corjk,Ri,Rj,Rk,0.0);
}
Exy*=exp(medi+medj+0.5*(stdi*stdi+stdj*stdj)
         +stdi*stdj*corij-sqr(L))/(2.0*pi);
Exz=0.0;
T=(stdk*B-stdi*c+corij*stdk*C)/c;
L=(Ri+T)/sqrt(2.0);
Gb=(1.0/sqrt(2.0*c))*(Rj+stdk*C-corij*sqrt(2.0)*L+corij*T);
for (l=0;l<16;l++){     /* 16 e'o numero de pontos de gauss */
        G=Gb-(x[l]*sqrt(2.0)*corij)*(1.0/sqrt(2.0*c));
        Exz+=w[l]*exp(-2.0*x[l]*L)
             *I_skew(17,G,1.0,-1,1.0,x[l],stdi,stdj,stdk,corij,
                     corik,corjk,Ri,Rj,Rk,0.0);
}
Exz*=exp(medi+medk+0.5*(stdi*stdi+stdk*stdk)
         +stdi*stdk*corik-sqr(L))/(2.0*pi);
Eyz=0.0;
T=(stdk*B+corij*(-stdj*c+stdk*C))/c;
L=(Ri+T)/sqrt(2.0);
Gb=(1.0/sqrt(2.0*c))*(Rj-stdj*c+stdk*C-corij*sqrt(2.0)*L+corij*T);
for (l=0;l<16;l++){     /* 16 e'o numero de pontos de gauss */
        G=Gb-(x[l]*sqrt(2.0)*corij)*(1.0/sqrt(2.0*c));
        Eyz+=w[l]*exp(-2.0*x[l]*L)
             *I_skew(18,G,1.0,-1,1.0,x[l],stdi,stdj,stdk,corij,
                     corik,corjk,Ri,Rj,Rk,0.0);
}
Eyz*=exp(medj+medk+0.5*(stdj*stdj+stdk*stdk)+
         stdj*stdk*corjk-sqr(L))/(2.0*pi);
Ex=0.0;
T=-stdi;
L=(Ri+T)/sqrt(2.0);
Gb=(1.0/sqrt(2.0*c))*(Rj-corij*sqrt(2.0)*L+corij*T);
for (l=0;l<16;l++){     /* 16 e'o numero de pontos de gauss */
        G=Gb-(x[l]*sqrt(2.0)*corij)*(1.0/sqrt(2.0*c));
        Ex+=w[l]*exp(-2.0*x[l]*L)
            *I_skew(19,G,1.0,-1,1.0,x[l],stdi,stdj,stdk,corij,
                    corik,corjk,Ri,Rj,Rk,0.0);
}
Ex*=exp(medi+0.5*(stdi*stdi)-sqr(L))/(2.0*pi);
Ey=0.0;
T=corij*-stdj;
```

```
L=(Ri+T)/sqrt(2.0);
Gb=(1.0/sqrt(2.0*c))*(Rj-stdj*c-corij*sqrt(2.0)*L+corij*T);
for (l=0;l<16;l++){     /* 16 e'o numero de pontos de gauss */
        G=Gb-(x[l]*sqrt(2.0)*corij)*(1.0/sqrt(2.0*c));
        Ey+=w[l]*exp(-2.0*x[l]*L)
            *I_skew(20,G,1.0,-1,1.0,x[l],stdi,stdj,stdk,corij,
                    corik,corjk,Ri,Rj,Rk,0.0);
}
Ey*=exp(medj+0.5*stdj*stdj-sqr(L))/(2.0*pi);
Ez=0.0;
T=(stdk*B+corij*stdk*C)/c;
L=(Ri+T)/sqrt(2.0);
Gb=(1.0/sqrt(2.0*c))*(Rj+stdk*C-corij*sqrt(2.0)*L+corij*T);
for (l=0;l<16;l++){     /* 16 e'o numero de pontos de gauss */
        G=Gb-(x[l]*sqrt(2.0)*corij)*(1.0/sqrt(2.0*c));
        Ez+=w[l]*exp(-2.0*x[l]*L)
            *I_skew(21,G,1.0,-1,1.0,x[l],stdi,stdj,stdk,corij,
                    corik,corjk,Ri,Rj,Rk,0.0);
}
Ez*=exp(medk+0.5*stdk*stdk-sqr(L))/(2.0*pi);
E1=0.0;
T=0.0;
L=Ri/sqrt(2.0);
Gb=(1.0/sqrt(2.0*c))*(Rj-corij*sqrt(2.0)*L);
for (l=0;l<16;l++){     /* 16 e'o numero de pontos de gauss */
        G=Gb-(x[l]*sqrt(2.0)*corij)*(1.0/sqrt(2.0*c));
        E1+=w[l]*exp(-2.0*x[l]*L)
            *I_skew(22,G,1.0,-1,1.0,x[l],stdi,stdj,stdk,corij,
                    corik,corjk,Ri,Rj,Rk,0.0);
}
E1*=exp(-sqr(L))/(2.0*pi);
Exyab=(0.5/sqrt(2.0*pi))*
       exp(medi+medj+0.5*(stdi*stdi+stdj*stdj)+
                          stdi*stdj*corij);
L=stdi+stdj*corij;
Exyab*=I_skew(9,Ri,sqrt(2.0),1,-1.0,0.0,stdi,stdj,0.0,corij,0.0,
              0.0,Ri,Rj,0.0,L);
Exab=(0.5/sqrt(2.0*pi))*exp(medi+0.5*stdi*stdi);
L=stdi;
Exab*=I_skew(10,Ri,sqrt(2.0),1,-1.0,0.0,stdi,stdj,0.0,corij,0.0,
            0.0,Ri,Rj,0.0,L);
Eyab=(0.5/sqrt(2.0*pi))*exp(medj+0.5*stdj*stdj);
L=stdj*corij;
Eyab*=I_skew(9,Ri,sqrt(2.0),1,-1.0,0.0,stdi,stdj,0.0,corij,0.0,
              0.0,Ri,Rj,0.0,L);
E1ab=(0.5/sqrt(2.0*pi));
```

```
L=0.0;
E1ab*=I_skew(10,Ri,sqrt(2.0),1,-1.0,0.0,stdi,stdj,0.0,corij,0.0,
              0.0,Ri,Rj,0.0,L);

Eyzbc=(0.5/sqrt(2.0*pi))*
       exp(medk+medj+0.5*(stdk*stdk+stdj*stdj)+
                           stdk*stdj*corjk);
L=stdk+stdj*corjk;
Eyzbc*=I_skew(9,Rk,sqrt(2.0),-1,-1.0,0.0,stdi,stdj,0.0,0.0,0.0,
              corjk,Ri,Rj,0.0,L);
Ezbc=(0.5/sqrt(2.0*pi))*exp(medk+0.5*stdk*stdk);
L=stdk;
Ezbc*=I_skew(10,Rk,sqrt(2.0),-1,-1.0,0.0,stdi,stdj,0.0,0.0,0.0,
            corjk,Ri,Rj,0.0,L);
Eybc=(0.5/sqrt(2.0*pi))*exp(medj+0.5*stdj*stdj);
L=stdj*corjk;
Eybc*=I_skew(9,Rk,sqrt(2.0),-1,-1.0,0.0,stdi,stdj,0.0,0.0,0.0,
              corjk,Ri,Rj,0.0,L);
E1bc=(0.5/sqrt(2.0*pi));
L=0.0;
E1bc*=I_skew(10,Rk,sqrt(2.0),-1,-1.0,0.0,stdi,stdj,0.0,0.0,0.0,
              corjk,Ri,Rj,0.0,L);

Exzac=(0.5/sqrt(2.0*pi))*
       exp(medk+medi+0.5*(stdk*stdk+stdi*stdi)+
                           stdk*stdi*corik);
L=stdk+stdi*corik;
Exzac*=I_skew(9,Rk,sqrt(2.0),-1,-1.0,0.0,stdi,stdj,0.0,0.0,corik,
              0.0,Ri,Rj,0.0,L);
Ezac=(0.5/sqrt(2.0*pi))*exp(medk+0.5*stdk*stdk);
L=stdk;
Ezac*=I_skew(10,Rk,sqrt(2.0),-1,-1.0,0.0,stdi,stdj,0.0,0.0,corik,
            0.0,Ri,Rj,0.0,L);
Exac=(0.5/sqrt(2.0*pi))*exp(medi+0.5*stdi*stdi);
L=stdi*corik;
Exac*=I_skew(9,Rk,sqrt(2.0),-1,-1.0,0.0,stdi,stdj,0.0,0.0,corik,
              0.0,Ri,Rj,0.0,L);
E1ac=(0.5/sqrt(2.0*pi));
L=0.0;
E1ac*=I_skew(10,Rk,sqrt(2.0),-1,-1.0,0.0,stdi,stdj,0.0,0.0,corik,
              0.0,Ri,Rj,0.0,L);

Ezc=0.5*exp(medk+0.5*stdk*stdk)*(1.0+erff((Rk-stdk)/sqrt(2.0)));
E1c=0.5*(1.0+erff(Rk/sqrt(2.0)));

Eyb=0.5*exp(medj+0.5*stdj*stdj)*(1.0+erff((Rj-stdj)/sqrt(2.0)));
```

```
        E1b=0.5*(1.0+erff(Rj/sqrt(2.0)));

        Exa=0.5*exp(medi+0.5*stdi*stdi)*(1.0-erff((Ri-stdi)/sqrt(2.0)));
        E1a=0.5*(1.0-erff(Ri/sqrt(2.0)));

        resp=(Si*Sj*Sk*Exyz-Si*Sj*Exy*Kpk-Si*Sk*Exz*Kpj+Si*Ex*Kpj*Kpk
               +Sj*Ey*Kci*Kpk+Sk*Ez*Kpj*Kci-Kci*Kpj*Kpk*E1)/(Ci*Pj*Pk);
        resp+=-mci*(Sj*Sk*Eyzbc-Sj*Kpk*Eybc-Kpj*Sk*Ezbc+Kpj*Kpk*E1bc)/(Pj*Pk);
        resp+=mpj*(Si*Sk*Exzac-Si*Kpk*Exac-Kci*Sk*Ezac+Kci*Kpk*E1ac)/(Ci*Pk);
        resp+=mpk*(Si*Sj*Exyab-Si*Kpj*Exab-Kci*Sj*Eyab+Kci*Kpj*E1ab)/(Ci*Pj);
        resp+=mci*mpj*(-Sk*Ezc+Kpk*E1c)/Pk+mci*mpk*(-Sj*Eyb+Kpj*E1b)/Pj;
        resp+=mpj*mpk*(Si*Exa-Kci*E1a)/Ci-mci*mpj*mpk;
        return(resp);

}



double m_pipjpk(double medi, double medj, double medk,
                double stdi, double stdj, double stdk,
                double corij, double corik, double corjk,
                double Si, double Sj, double Sk,
                double Kpi, double Kpj, double Kpk,
                double Pi, double Pj, double Pk, double *x, double *w,
                double mpi, double mpj, double mpk)
/*************************************************************************/
/* medi,medj,medk=media do ativo i,j e k;                               */
/* stdi,stdj,stdk=desvio do ativo i,j e k;                              */
/* Pi,Pj,Pk,Kpi,Kpj,Kpk=premios e strikes das puts ;                    */
/* Si,Sj,Sk=cotacao dos ativos básicos i,j e k em t=0.                  */
/* corij=correlacao entre i e j;                                        */
/*************************************************************************/
{
        double Exyz,Exy,Exz,Eyz,Ez,Ex,Ey,E1,resp,
               Ri,Rj,Rk,c,G, Gb, L, T, B, C;
        int l;

        Exyz=0.0;
        c=1.0-corij*corij;
        C=corij*corik-corjk;
        B=corij*corjk-corik;
        Ri=(log(Kpi/Si)-medi)/stdi;
        Rj=(log(Kpj/Sj)-medj)/stdj;
        Rk=(log(Kpk/Sk)-medk)/stdk;
        T=(stdk*B-stdi*c+corij*(-stdj*c+stdk*C))/c;
        L=(Ri+T)/sqrt(2.0);
```

```
Gb=(1.0/sqrt(2.0*c))*(Rj-stdj*c+stdk*C-corij*sqrt(2.0)*L+corij*T);
for (l=0;l<16;l++){     /* 16 e'o numero de pontos de gauss */
        G=Gb-(-x[l]*sqrt(2.0)*corij)*(1.0/sqrt(2.0*c));
        Exyz+=w[l]*exp(-2.0*-x[l]*L)
              *I_skew(15,G,1.0,-1,1.0,-x[l],stdi,stdj,stdk,
                     corij,corik,corjk,Ri,Rj,Rk,0.0);
        /* I calcula a integral de G  ate' +(-)infinito da funcao
        func, que depende de phi (x[i] e'um ponto do grid de phi
        indiretamente e de psi. */
}
Exyz*=exp(medi+medj+medk
          +0.5*(stdi*stdi+stdj*stdj+stdk*stdk)
          +stdi*stdj*corij+stdi*stdk*corik
          +stdj*stdk*corjk-sqr(L))/(2.0*pi);
Exy=0.0;
T=-stdi+corij*-stdj;
L=(Ri+T)/sqrt(2.0);
Gb=(1.0/sqrt(2.0*c))*(Rj-stdj*c-corij*sqrt(2.0)*L+corij*T);
for (l=0;l<16;l++){     /* 16 e'o numero de pontos de gauss */
        G=Gb-(-x[l]*sqrt(2.0)*corij)*(1.0/sqrt(2.0*c));
        Exy+=w[l]*exp(-2.0*-x[l]*L)
             *I_skew(16,G,1.0,-1,1.0,-x[l],stdi,stdj,stdk,corij,
                     corik,corjk,Ri,Rj,Rk,0.0);
}
Exy*=exp(medi+medj+0.5*(stdi*stdi+stdj*stdj)
         +stdi*stdj*corij-sqr(L))/(2.0*pi);
Exz=0.0;
T=(stdk*B-stdi*c+corij*stdk*C)/c;
L=(Ri+T)/sqrt(2.0);
Gb=(1.0/sqrt(2.0*c))*(Rj+stdk*C-corij*sqrt(2.0)*L+corij*T);
for (l=0;l<16;l++){     /* 16 e'o numero de pontos de gauss */
        G=Gb-(-x[l]*sqrt(2.0)*corij)*(1.0/sqrt(2.0*c));
        Exz+=w[l]*exp(-2.0*-x[l]*L)
             *I_skew(17,G,1.0,-1,1.0,-x[l],stdi,stdj,stdk,corij,
                     corik,corjk,Ri,Rj,Rk,0.0);
}
Exz*=exp(medi+medk+0.5*(stdi*stdi+stdk*stdk)
         +stdi*stdk*corik-sqr(L))/(2.0*pi);
Eyz=0.0;
T=(stdk*B+corij*(-stdj*c+stdk*C))/c;
L=(Ri+T)/sqrt(2.0);
Gb=(1.0/sqrt(2.0*c))*(Rj-stdj*c+stdk*C-corij*sqrt(2.0)*L+corij*T);
for (l=0;l<16;l++){     /* 16 e'o numero de pontos de gauss */
        G=Gb-(-x[l]*sqrt(2.0)*corij)*(1.0/sqrt(2.0*c));
        Eyz+=w[l]*exp(-2.0*-x[l]*L)
             *I_skew(18,G,1.0,-1,1.0,-x[l],stdi,stdj,stdk,corij,
```

```
                    corik,corjk,Ri,Rj,Rk,0.0);
}
Eyz*=exp(medj+medk+0.5*(stdj*stdj+stdk*stdk)+
         stdj*stdk*corjk-sqr(L))/(2.0*pi);
Ex=0.0;
T=-stdi;
L=(Ri+T)/sqrt(2.0);
Gb=(1.0/sqrt(2.0*c))*(Rj-corij*sqrt(2.0)*L+corij*T);
for (l=0;l<16;l++){     /* 16 e'o numero de pontos de gauss */
        G=Gb-(-x[l]*sqrt(2.0)*corij)*(1.0/sqrt(2.0*c));
        Ex+=w[l]*exp(-2.0*-x[l]*L)
            *I_skew(19,G,1.0,-1,1.0,-x[l],stdi,stdj,stdk,corij,
                    corik,corjk,Ri,Rj,Rk,0.0);
}
Ex*=exp(medi+0.5*(stdi*stdi)-sqr(L))/(2.0*pi);
Ey=0.0;
T=corij*-stdj;
L=(Ri+T)/sqrt(2.0);
Gb=(1.0/sqrt(2.0*c))*(Rj-stdj*c-corij*sqrt(2.0)*L+corij*T);
for (l=0;l<16;l++){     /* 16 e'o numero de pontos de gauss */
        G=Gb-(-x[l]*sqrt(2.0)*corij)*(1.0/sqrt(2.0*c));
        Ey+=w[l]*exp(-2.0*-x[l]*L)
            *I_skew(20,G,1.0,-1,1.0,-x[l],stdi,stdj,stdk,corij,
                    corik,corjk,Ri,Rj,Rk,0.0);
}
Ey*=exp(medj+0.5*stdj*stdj-sqr(L))/(2.0*pi);
Ez=0.0;
T=(stdk*B+corij*stdk*C)/c;
L=(Ri+T)/sqrt(2.0);
Gb=(1.0/sqrt(2.0*c))*(Rj+stdk*C-corij*sqrt(2.0)*L+corij*T);
for (l=0;l<16;l++){     /* 16 e'o numero de pontos de gauss */
        G=Gb-(-x[l]*sqrt(2.0)*corij)*(1.0/sqrt(2.0*c));
        Ez+=w[l]*exp(-2.0*-x[l]*L)
            *I_skew(21,G,1.0,-1,1.0,-x[l],stdi,stdj,stdk,corij,
                    corik,corjk,Ri,Rj,Rk,0.0);
}
Ez*=exp(medk+0.5*stdk*stdk-sqr(L))/(2.0*pi);
E1=0.0;
T=0.0;
L=Ri/sqrt(2.0);
Gb=(1.0/sqrt(2.0*c))*(Rj-corij*sqrt(2.0)*L);
for (l=0;l<16;l++){     /* 16 e'o numero de pontos de gauss */
        G=Gb-(-x[l]*sqrt(2.0)*corij)*(1.0/sqrt(2.0*c));
        E1+=w[l]*exp(-2.0*-x[l]*L)
            *I_skew(22,G,1.0,-1,1.0,-x[l],stdi,stdj,stdk,corij,
                    corik,corjk,Ri,Rj,Rk,0.0);
```

```
}
E1*=exp(-sqr(L))/(2.0*pi);
Exyab=(0.5/sqrt(2.0*pi))*
       exp(medi+medj+0.5*(stdi*stdi+stdj*stdj)+
                           stdi*stdj*corij);
L=stdi+stdj*corij;
Exyab*=I_skew(9,Ri,sqrt(2.0),-1,-1.0,0.0,stdi,stdj,0.0,corij,0.0,
              0.0,Ri,Rj,0.0,L);
Exab=(0.5/sqrt(2.0*pi))*exp(medi+0.5*stdi*stdi);
L=stdi;
Exab*=I_skew(10,Ri,sqrt(2.0),-1,-1.0,0.0,stdi,stdj,0.0,corij,0.0,
            0.0,Ri,Rj,0.0,L);
Eyab=(0.5/sqrt(2.0*pi))*exp(medj+0.5*stdj*stdj);
L=stdj*corij;
Eyab*=I_skew(9,Ri,sqrt(2.0),-1,-1.0,0.0,stdi,stdj,0.0,corij,0.0,
              0.0,Ri,Rj,0.0,L);
E1ab=(0.5/sqrt(2.0*pi));
L=0.0;
E1ab*=I_skew(10,Ri,sqrt(2.0),-1,-1.0,0.0,stdi,stdj,0.0,corij,0.0,
              0.0,Ri,Rj,0.0,L);

Eyzbc=(0.5/sqrt(2.0*pi))*
       exp(medk+medj+0.5*(stdk*stdk+stdj*stdj)+
                           stdk*stdj*corjk);
L=stdk+stdj*corjk;
Eyzbc*=I_skew(9,Rk,sqrt(2.0),-1,-1.0,0.0,stdi,stdj,0.0,0.0,0.0,
              corjk,Ri,Rj,0.0,L);
Ezbc=(0.5/sqrt(2.0*pi))*exp(medk+0.5*stdk*stdk);
L=stdk;
Ezbc*=I_skew(10,Rk,sqrt(2.0),-1,-1.0,0.0,stdi,stdj,0.0,0.0,0.0,
            corjk,Ri,Rj,0.0,L);
Eybc=(0.5/sqrt(2.0*pi))*exp(medj+0.5*stdj*stdj);
L=stdj*corjk;
Eybc*=I_skew(9,Rk,sqrt(2.0),-1,-1.0,0.0,stdi,stdj,0.0,0.0,0.0,
              corjk,Ri,Rj,0.0,L);
E1bc=(0.5/sqrt(2.0*pi));
L=0.0;
E1bc*=I_skew(10,Rk,sqrt(2.0),-1,-1.0,0.0,stdi,stdj,0.0,0.0,0.0,
              corjk,Ri,Rj,0.0,L);

Exzac=(0.5/sqrt(2.0*pi))*
       exp(medk+medi+0.5*(stdk*stdk+stdi*stdi)+
                           stdk*stdi*corik);
L=stdk+stdi*corik;
Exzac*=I_skew(9,Rk,sqrt(2.0),-1,-1.0,0.0,stdi,stdj,0.0,0.0,corik,
              0.0,Ri,Rj,0.0,L);
```

```
        Ezac=(0.5/sqrt(2.0*pi))*exp(medk+0.5*stdk*stdk);
        L=stdk;
        Ezac*=I_skew(10,Rk,sqrt(2.0),-1,-1.0,0.0,stdi,stdj,0.0,0.0,corik,
                    0.0,Ri,Rj,0.0,L);
        Exac=(0.5/sqrt(2.0*pi))*exp(medi+0.5*stdi*stdi);
        L=stdi*corik;
        Exac*=I_skew(9,Rk,sqrt(2.0),-1,-1.0,0.0,stdi,stdj,0.0,0.0,corik,
                      0.0,Ri,Rj,0.0,L);
        E1ac=(0.5/sqrt(2.0*pi));
        L=0.0;
        E1ac*=I_skew(10,Rk,sqrt(2.0),-1,-1.0,0.0,stdi,stdj,0.0,0.0,corik,
                      0.0,Ri,Rj,0.0,L);

        Ezc=0.5*exp(medk+0.5*stdk*stdk)*(1.0+erff((Rk-stdk)/sqrt(2.0)));
        E1c=0.5*(1.0+erff(Rk/sqrt(2.0)));

        Eyb=0.5*exp(medj+0.5*stdj*stdj)*(1.0+erff((Rj-stdj)/sqrt(2.0)));
        E1b=0.5*(1.0+erff(Rj/sqrt(2.0)));

        Exa=0.5*exp(medi+0.5*stdi*stdi)*(1.0-erff((Ri-stdi)/sqrt(2.0)));
        E1a=0.5*(1.0-erff(Ri/sqrt(2.0)));

        resp=-(Si*Sj*Sk*Exyz-Si*Sj*Exy*Kpk-Si*Sk*Exz*Kpj+Si*Ex*Kpj*Kpk
               +Sj*Ey*Kpi*Kpk+Sk*Ez*Kpj*Kci-Kpi*Kpj*Kpk*E1)/(Pi*Pj*Pk);
        resp+=mpi*(Sj*Sk*Eyzbc-Sj*Kpk*Eybc-Kpj*Sk*Ezbc+Kpj*Kpk*E1bc)/(Pj*Pk);
        resp+=mpj*(Si*Sk*Exzac-Si*Kpk*Exac-Kpi*Sk*Ezac+Kci*Kpk*E1ac)/(Pi*Pk);
        resp+=mpk*(Si*Sj*Exyab-Si*Kpj*Exab-Kpi*Sj*Eyab+Kci*Kpj*E1ab)/(Pi*Pj);
        resp+=mpi*mpj*(-Sk*Ezc+Kpk*E1c)/Pk+mpi*mpk*(-Sj*Eyb+Kpj*E1b)/Pj;
        resp+=mpj*mpk*(-Si*Exa+Kpi*E1a)/Pi-mpi*mpj*mpk;
        return(resp);

}


double sss(int i, int j, int k, double *med, double *std, double **cor,
           int *atv)
/***********************************************************************/
/* rotina de chamada das funcoes de coskewness somente entre ativos     */
/* básicos;                                                             */
/* aqui sera' chamada a funcao que calcula o coskewness entre o i-ésimo, */
/* j-esimo                                                              */
/* e k-esimo ativo básico;                                              */
/* dependendo se i,j e k sao iguais ou nao, sera' chamada uma funcao    */
/* diferente;                                                           */
/* no vetor atv, atv[i] contém o respectivo ativo básico referente ao   */
/* i-ésimo elemento do portfolio.                                       */
```

```
/************************************************************************/
{
        double resp=1.0E+308;

        if ((atv[i] == atv[j]) && (atv[i] == atv[k])){
                resp=m_ririri(med[atv[i]],std[atv[i]]);
        }
        if ((atv[i] == atv[j]) && (atv[j] != atv[k])){
                resp=m_ririrj(med[atv[i]],med[atv[k]],std[atv[i]],std[atv[k]],
                              cor[atv[i]][atv[k]]);
        }
        if ((atv[i] != atv[j]) && (atv[j] == atv[k])){
                resp=m_ririrj(med[atv[j]],med[atv[i]],std[atv[j]],std[atv[i]],
                              cor[atv[i]][atv[j]]);
        }
        if ((atv[i] == atv[k]) && (atv[j] != atv[k])){
                resp=m_ririrj(med[atv[i]],med[atv[j]],std[atv[i]],std[atv[j]],
                              cor[atv[i]][atv[j]]);
        }
        if ((atv[i] != atv[j]) && (atv[i] != atv[k]) && (atv[j] != atv[k])){
                resp=m_rirjrk(med[atv[i]],med[atv[j]],med[atv[k]],std[atv[i]],
                              std[atv[j]],std[atv[k]],cor[atv[i]][atv[j]],
                              cor[atv[i]][atv[k]],cor[atv[j]][atv[k]]);
        }
        return(resp);
}


double ssc(int i, int j, int k, double *med, double *std, double **cor,
           double *S, double *K, double *P, int *atv, double *esp)
/************************************************************************/
/* rotina de chamada das funcoes de coskewness entre ativos básicos e call;*/
/* dependendo se i,j e k sao iguais ou nao, sera' chamada uma funcao      */
/* diferente;                                                             */
/*  no vetor atv, atv[i] contém o respectivo ativo básico referente ao    */
/* i-ésimo elemento do portfolio.                                         */
/************************************************************************/
{
        double resp=1.0E+308;

        if ((atv[i] == atv[j]) && (atv[i] == atv[k])){
                resp=m_ririci(med[atv[i]],std[atv[i]],S[k],K[k],P[k],esp[i],
                              esp[k]);
        }
        if ((atv[i] == atv[j]) && (atv[j] != atv[k])){
                resp=m_riricj(med[atv[i]],med[atv[k]],std[atv[i]],std[atv[k]],
```

```
                        cor[atv[i]][atv[k]],S[k],K[k],P[k],esp[i],esp[k]);
        }
        if ((atv[i] != atv[j]) && (atv[j] == atv[k])){
                resp=m_ricirj(med[atv[j]],med[atv[i]],std[atv[j]],std[atv[i]],
                        cor[atv[i]][atv[j]],S[k],K[k],P[k],esp[i],esp[k]);
        }
        if ((atv[i] == atv[k]) && (atv[j] != atv[k])){
                resp=m_ricirj(med[atv[i]],med[atv[j]],std[atv[i]],std[atv[j]],
                              cor[atv[i]][atv[j]],S[k],K[k],P[k],esp[i],esp[j],
                              esp[k]);
        }
        if ((atv[i] != atv[j]) && (atv[i] != atv[k]) && (atv[j] != atv[k])){
                resp=m_rirjck(med[atv[i]],med[atv[j]],med[atv[k]],std[atv[i]],
                              std[atv[j]],std[atv[k]],cor[atv[i]][atv[j]],
                              cor[atv[i]][atv[k]],cor[atv[j]][atv[k]],
                              S[k],K[k],P[k],esp[i],esp[j],esp[k]);
        }
        return(resp);
}


double ssp(int i, int j, int k, double *med, double *std, double **cor,
           double *S, double *K, double *P, int *atv, double *esp)
/**************************************************************************/
/* rotina de chamada das funcoes de coskewness entre ativos básicos e put; */
/* dependendo se i,j e k sao iguais ou nao, sera' chamada uma funcao       */
/* diferente.                                                              */
/**************************************************************************/
{
        double resp=1.0E+308;

        if ((atv[i] == atv[j]) && (atv[i] == atv[k])){
                resp=m_riripi(med[atv[i]],std[atv[i]],S[k],K[k],P[k],esp[i],
                              esp[k]);
        }
        if ((atv[i] == atv[j]) && (atv[j] != atv[k])){
                resp=m_riripj(med[atv[i]],med[atv[k]],std[atv[i]],std[atv[k]],
                              cor[atv[i]][atv[k]],S[k],K[k],P[k],esp[i],esp[k]);
        }
        if ((atv[i] != atv[j]) && (atv[j] == atv[k])){
                resp=m_ripirj(med[atv[j]],med[atv[i]],std[atv[j]],std[atv[i]],
                              cor[atv[i]][atv[j]],S[k],K[k],P[k],esp[i],
                              esp[j],esp[k]);
        }
        if ((atv[i] == atv[k]) && (atv[j] != atv[k])){
                resp=m_ripirj(med[atv[i]],med[atv[j]],std[atv[i]],std[atv[j]],
```

```
                            cor[atv[i]][atv[j]],S[k],K[k],P[k],esp[i],
                            esp[j],esp[k]);
        }
        if ((atv[i] != atv[j]) && (atv[i] != atv[k]) && (atv[j] != atv[k])){
                resp=m_rirjpk(med[atv[i]],med[atv[j]],med[atv[k]],std[atv[i]],
                            std[atv[j]],std[atv[k]],cor[atv[i]][atv[j]],
                            cor[atv[i]][atv[k]],cor[atv[j]][atv[k]],
                            S[k],K[k],P[k],esp[i],esp[j],esp[k]);
        }
        return(resp);
}


double scc(int i, int j, int k, double *med, double *std, double **cor,
           double *S, double *K, double *P, int *atv, double *esp)
/**************************************************************************/
/* rotina de chamada das funcoes de coskewness entre ativos básicos e call;*/
/* dependendo se i,j e k sao iguais ou nao, sera' chamada uma funcao       */
/* diferente.                                                              */
/**************************************************************************/
{
        double resp=1.0E+308;

        if ((atv[i] == atv[j]) && (atv[i] == atv[k])){
                resp=m_ricici(med[atv[i]],std[atv[i]],S[k],K[j],K[k],P[j],P[k],
                            esp[i],esp[j],esp[k]);
        }
        if ((atv[i] == atv[j]) && (atv[j] != atv[k])){
                resp=m_ricicj(med[atv[i]],med[atv[k]],std[atv[i]],std[atv[k]],
                            cor[atv[i]][atv[k]],S[j],S[k],K[j],K[k],P[j],P[k],
                            esp[i],esp[j],esp[k]);
        }
        if ((atv[i] != atv[j]) && (atv[j] == atv[k])){ //j=k i
                resp=m_cicirj(med[atv[j]],med[atv[i]],std[atv[j]],std[atv[i]],
                            cor[atv[i]][atv[j]],S[j],K[j],K[k],P[j],P[k],
                            esp[i],esp[j],esp[k]);
        }
        if ((atv[i] == atv[k]) && (atv[j] != atv[k])){
                resp=m_ricicj(med[atv[i]],med[atv[j]],std[atv[i]],std[atv[j]],
                            cor[atv[i]][atv[j]],S[k],S[j],K[k],K[j],P[k],P[j],
                            esp[i],esp[j],esp[k]);
        }
        if ((atv[i] != atv[j]) && (atv[i] != atv[k]) && (atv[j] != atv[k])){
                resp=m_ricjck(med[atv[i]],med[atv[j]],med[atv[k]],std[atv[i]],
                            std[atv[j]],std[atv[k]],cor[atv[i]][atv[j]],
                            cor[atv[i]][atv[k]], cor[atv[j]][atv[k]],
```

```
                              S[j],S[k],K[j],K[k],P[j],P[k],esp[i],esp[j],esp[k]);
        }
        return(resp);
}


double scp(int i, int j, int k, double *med, double *std, double **cor,
           double *S, double *K, double *P, int *atv, double *esp)
/***********************************************************************/
/* rotina de chamada das funcoes de coskewness entre 1 ativo básico,       */
/* 1 call e 1 put;  dependendo se i,j e k sao iguais ou nao, sera' chamada */
/* uma funcao diferente; no vetor atv, atv[i] contém o respectivo ativo    */
/* básico referente ao i-ésimo elemento do portfolio.                      */
/***********************************************************************/
{
        double resp=1.0E+308;

        if ((atv[i] == atv[j]) && (atv[i] == atv[k])){
                resp=m_ricipi(med[atv[i]],std[atv[i]],S[k],K[j],K[k],P[j],P[k],
                              esp[i],esp[j],esp[k]);
        }
        if ((atv[i] == atv[j]) && (atv[j] != atv[k])){
                resp=m_ricipj(med[atv[i]],med[atv[k]],std[atv[i]],std[atv[k]],
                              cor[atv[i]][atv[k]],S[j],S[k],K[j],K[k],P[j],P[k],
                              esp[i],esp[j],esp[k]);
        }
        if ((atv[i] != atv[j]) && (atv[j] == atv[k])){
                resp=m_cipirj(med[atv[j]],med[atv[i]],std[atv[j]],std[atv[i]],
                              cor[atv[i]][atv[j]],S[j],K[j],K[k],P[j],P[k],
                              esp[i],esp[j],esp[k]);
        }
        if ((atv[i] == atv[k]) && (atv[j] != atv[k])){
                resp=m_ripicj(med[atv[i]],med[atv[j]],std[atv[i]],std[atv[j]],
                              cor[atv[i]][atv[j]],S[k],S[j],K[k],K[j],P[k],P[j],
                              esp[i],esp[j],esp[k]);
        }
        if ((atv[i] != atv[j]) && (atv[i] != atv[k]) && (atv[j] != atv[k])){
                resp=m_ricjpk(med[atv[i]],med[atv[j]],med[atv[k]],std[atv[i]],
                              std[atv[j]],std[atv[k]],cor[atv[i]][atv[j]],
                              cor[atv[i]][atv[k]],cor[atv[j]][atv[k]],
                              S[j],S[k],K[j],K[k],P[j],P[k],esp[i],esp[j],
                              esp[k]);
        }
        return(resp);
}
```

```
double spp(int i, int j, int k, double *med, double *std, double **cor,
           double *S, double *K, double *P, int *atv, double *esp)
/************************************************************************/
/* rotina de chamada das funcoes de coskewness entre 1 ativo básico e 2 puts*/
/************************************************************************/
{
        double resp=1.0E+308;

        if ((atv[i] == atv[j]) && (atv[i] == atv[k])){
                resp=m_ripipi(med[atv[i]],std[atv[i]],S[k],K[j],K[k],P[j],P[k],
                              esp[i],esp[j],esp[k]);
        }
        if ((atv[i] == atv[j]) && (atv[j] != atv[k])){
                resp=m_ripipj(med[atv[i]],med[atv[k]],std[atv[i]],std[atv[k]],
                              cor[atv[i]][atv[k]],S[j],S[k],K[j],K[k],P[j],P[k],
                              esp[i],esp[j],esp[k]);
        }
        if ((atv[i] != atv[j]) && (atv[j] == atv[k])){
                resp=m_pipirj(med[atv[j]],med[atv[i]],std[atv[j]],std[atv[i]],
                              cor[atv[i]][atv[j]],S[j],K[j],K[k],P[j],P[k],esp[i],
                              esp[j],esp[k]);
        }
        if ((atv[i] == atv[k]) && (atv[j] != atv[k])){
                resp=m_ripipj(med[atv[i]],med[atv[j]],std[atv[i]],std[atv[j]],
                              cor[atv[i]][atv[j]],S[k],S[j],K[k],K[j],P[k],P[j],
                              esp[i],esp[j],esp[k]);
        }
        if ((atv[i] != atv[j]) && (atv[i] != atv[k]) && (atv[j] != atv[k])){
                resp=m_ripjpk(med[atv[i]],med[atv[j]],med[atv[k]],std[atv[i]],
                              std[atv[j]],std[atv[k]],cor[atv[i]][atv[j]],
                              cor[atv[i]][atv[k]],cor[atv[j]][atv[k]],
                              S[j],S[k],K[j],K[k],P[j],P[k],esp[i],esp[j],
                              esp[k]);
        }
        return(resp);
}


double ccc(int i, int j, int k, double *med, double *std, double **cor,
           double *S, double *K, double *P, int *atv, double *x, double *w,
           double *esp)
/************************************************************************/
/* rotina de chamada das funcoes de coskewness somente entre calls       */
/************************************************************************/
{
```

```
        double resp=1.0E+308;

        if ((atv[i] == atv[j]) && (atv[i] == atv[k])){
                resp=m_cicici(med[atv[i]],std[atv[i]],S[k],K[i],K[j],K[k],
                              P[i],P[j],P[k],esp[i],esp[j],esp[k]);
        }
        if ((atv[i] == atv[j]) && (atv[j] != atv[k])){
                resp=m_cicicj(med[atv[i]],med[atv[k]],std[atv[i]],std[atv[k]],
                              cor[atv[i]][atv[k]],S[j],S[k],K[i],K[j],K[k],
                              P[i],P[j],P[k],esp[i],esp[j],esp[k]);
        }
        if ((atv[i] != atv[j]) && (atv[j] == atv[k])){
                resp=m_cicicj(med[atv[j]],med[atv[i]],std[atv[j]],std[atv[i]],
                              cor[atv[i]][atv[j]],S[j],S[i],K[j],K[k],K[i],
                              P[j],P[k],P[i],esp[i],esp[j],esp[k]);
        }
        if ((atv[i] == atv[k]) && (atv[j] != atv[k])){
                resp=m_cicicj(med[atv[i]],med[atv[j]],std[atv[i]],std[atv[j]],
                              cor[atv[i]][atv[j]],S[k],S[j],K[i],K[k],K[j],
                              P[i],P[k],P[j],esp[i],esp[j],esp[k]);
        }
        if ((atv[i] != atv[j]) && (atv[i] != atv[k]) && (atv[j] != atv[k])){
                resp=m_cicjck(med[atv[i]],med[atv[j]],med[atv[k]],std[atv[i]],
                              std[atv[j]],std[atv[k]],cor[atv[i]][atv[j]],
                              cor[atv[i]][atv[k]],cor[atv[j]][atv[k]],
                              S[i],S[j],S[k],K[i],K[j],K[k],P[i],P[j],P[k],x,w,
                              esp[i],esp[j],esp[k]);
        }
        return(resp);
}


double ccp(int i, int j, int k, double *med, double *std, double **cor,
           double *S, double *K, double *P, int *atv, double *x, double *w,
           double *esp )
/**************************************************************************/
/* rotina de chamada das funcoes de coskewness entre 2 calls e 1 put.     */
/**************************************************************************/
{
        double resp=1.0E+308;

        if ((atv[i] == atv[j]) && (atv[i] == atv[k])){
                resp=m_cicipi(med[atv[i]],std[atv[i]],S[i],K[i],K[j],K[k],
                              P[i],P[j],P[k],esp[i],esp[j],esp[k]);
        }
        if ((atv[i] == atv[j]) && (atv[j] != atv[k])){
```

```
                resp=m_cicipj(med[atv[i]],med[atv[k]],std[atv[i]],std[atv[k]],
                              cor[atv[i]][atv[k]],S[j],S[k],K[i],K[j],K[k],P[i],
                              P[j],P[k],esp[i],esp[j],esp[k]);
        }
        if ((atv[i] != atv[j]) && (atv[j] == atv[k])){
                resp=m_cipicj(med[atv[j]],med[atv[i]],std[atv[j]],std[atv[i]],
                              cor[atv[i]][atv[j]],S[j],S[i],K[j],K[k],K[i],
                              P[j],P[k],P[i],esp[i],esp[j],esp[k]);
        }
        if ((atv[i] == atv[k]) && (atv[j] != atv[k])){
                resp=m_cipicj(med[atv[i]],med[atv[j]],std[atv[i]],std[atv[j]],
                              cor[atv[i]][atv[j]],S[k],S[j],K[i],K[k],K[j],P[i],
                              P[k],P[j],esp[i],esp[j],esp[k]);
        }
        if ((atv[i] != atv[j]) && (atv[i] != atv[k]) && (atv[j] != atv[k])){
                resp=m_cicjpk(med[atv[i]],med[atv[j]],med[atv[k]],std[atv[i]],
                              std[atv[j]],std[atv[k]],cor[atv[i]][atv[j]],
                              cor[atv[i]][atv[k]],cor[atv[j]][atv[k]],
                              S[i],S[j],S[k],K[i],K[j],K[k],P[i],P[j],P[k],x,w,
                              esp[i],esp[j],esp[k]);
        }
        return(resp);
}


double cpp(int i, int j, int k, double *med, double *std, double **cor,
           double *S, double *K, double *P, int *atv, double *x, double *w,
           double *esp)
/**************************************************************************/
/* rotina de chamada das funcoes de coskewness entre 1 call e 2 puts.     */
/**************************************************************************/
{
        double resp=1.0E+308;

        if ((atv[i] == atv[j]) && (atv[i] == atv[k])){
                resp=m_cipipi(med[atv[i]],std[atv[i]],S[i],K[i],K[j],K[k],
                              P[i],P[j],P[k],esp[i],esp[j],esp[k]);
        }
        if ((atv[i] == atv[j]) && (atv[j] != atv[k])){
                resp=m_cipipj(med[atv[i]],med[atv[k]],std[atv[i]],std[atv[k]],
                              cor[atv[i]][atv[k]],S[j],S[k],K[i],K[j],K[k],
                              P[i],P[j],P[k],esp[i],esp[j],esp[k]);
        }
        if ((atv[i] != atv[j]) && (atv[j] == atv[k])){
                resp=m_pipicj(med[atv[j]],med[atv[i]],std[atv[j]],std[atv[i]],
                             cor[atv[i]][atv[j]],S[j],S[i],K[j],K[k],K[i],
```

```
                    P[j],P[k],P[i],esp[i],esp[j],esp[k]);
        }
        if ((atv[i] == atv[k]) && (atv[j] != atv[k])){
                resp=m_cipipj(med[atv[i]],med[atv[j]],std[atv[i]],std[atv[j]],
                            cor[atv[i]][atv[j]],S[k],S[j],K[i],K[k],K[j],P[i],
                            P[k],P[j],esp[i],esp[j],esp[k]);
        }
        if ((atv[i] != atv[j]) && (atv[i] != atv[k]) && (atv[j] != atv[k])){
                resp=m_cipjpk(med[atv[i]],med[atv[j]],med[atv[k]],std[atv[i]],
                              std[atv[j]],std[atv[k]],cor[atv[i]][atv[j]],
                              cor[atv[i]][atv[k]],cor[atv[j]][atv[k]],
                              S[i],S[j],S[k],K[i],K[j],K[k],P[i],P[j],P[k],x,w,
                              esp[i],esp[j],esp[k]);
        }
        return(resp);
}


double ppp(int i, int j, int k, double *med, double *std, double **cor,
           double *S, double *K, double *P, int *atv, double *x, double *w,
           double *esp)
/*****************************************************************************/
/* rotina de chamada das funcoes de coskewness somente entre puts.          */
/*****************************************************************************/
{
        double resp=1.0E+308;

        if ((atv[i] == atv[j]) && (atv[i] == atv[k])){
                resp=m_pipipi(med[atv[i]],std[atv[i]],S[i],K[i],K[j],K[k],
                              P[i],P[j],P[k],esp[i],esp[j],esp[k]);
        }
        if ((atv[i] == atv[j]) && (atv[j] != atv[k])){
                resp=m_pipipj(med[atv[i]],med[atv[k]],std[atv[i]],std[atv[k]],
                              cor[atv[i]][atv[k]],S[i],S[k],K[i],K[j],K[k],
                              P[i],P[j],P[k],esp[i],esp[j],esp[k]);
        }
        if ((atv[i] != atv[j]) && (atv[j] == atv[k])){
                resp=m_pipipj(med[atv[j]],med[atv[i]],std[atv[j]],std[atv[i]],
                              cor[atv[i]][atv[j]],S[j],S[i],K[j],K[k],K[i],P[j],
                              P[k],P[i],esp[i],esp[j],esp[k]);
        }
        if ((atv[i] == atv[k]) && (atv[j] != atv[k])){
                resp=m_pipipj(med[atv[i]],med[atv[j]],std[atv[i]],std[atv[j]],
                              cor[atv[i]][atv[j]],S[k],S[j],K[i],K[k],K[j],P[i],
                              P[k],P[j],esp[i],esp[j],esp[k]);
        }
```

```
        if ((atv[i] != atv[j]) && (atv[i] != atv[k]) && (atv[j] != atv[k])){
                resp=m_pipjpk(med[atv[i]],med[atv[j]],med[atv[k]],std[atv[i]],
                              std[atv[j]],std[atv[k]],cor[atv[i]][atv[j]],
                              cor[atv[i]][atv[k]],cor[atv[j]][atv[k]],S[i],S[j],
                              S[k],K[i],K[j],K[k],P[i],P[j],P[k],x,w,esp[i],
                              esp[j],esp[k]);
        }
        return(resp);
}


double f_skew(char opcao, double sgn, double x, double y, double stdi,
              double stdj, double stdk, double corij, double corik,
              double corjk, double Ri, double Rj, double Rk, double a)
/****************************************************************************/
/* contém o corpo das funcoes integrando das integrais envolvidas nas      */
/* expressoes de assimetria entre derivativos.                              */
/* x, y = sao abcissas na integracao gaussiana e de romberg,respectivamente*/
/* sgn = sinal da funcao erff abaixo.                                       */
/* Ri,Rk = constantes.                                                      */
/****************************************************************************/
{
        double resp, c, C, R, B, T, L;

        c=1.0-corij*corij;
        C=corij*corik-corjk;
        B=corij*corjk-corik;
        R=1.0-corij*corij-corik*corik-corjk*corjk+2.0*corij*corik*corjk;
        switch(opcao)
        {

                case 9: /* relativo a E2xy,Exy,Ey*/
                        resp=Rj-y*corij-stdj*(1.0-corij*corij);
                        resp/=sqrt(2.0*(1.0-corij*corij));
                        resp=(1.0+sgn*erff(resp))*exp(-sqr(y-a)/2.0);
                        return(resp);

                case 10: /* relativo a Ex,E2x,E1*/
                        resp=Rj-y*corij;
                        resp/=sqrt(2.0*(1.0-corij*corij));
                        resp=(1.0+sgn*erff(resp))*exp(-sqr(y-a)/2.0);
                        return(resp);

                case 11: /* relativo a Exyz*/
                        resp=Rj-y*corjk+stdi*(corik*corjk-corij)
                              -stdj*(1.0-corjk*corjk);
```

```
        resp/=sqrt(2.0*(1.0-corjk*corjk));
        resp=(1.0+sgn*erff(resp))*exp(-sqr(y-a)/2.0);
        return(resp);

case 12: /* relativo a Exy,Exz,Ex*/
        resp=Rj-y*corjk+stdi*(corik*corjk-corij);
        resp/=sqrt(2.0*(1.0-corjk*corjk));
        resp=(1.0+sgn*erff(resp))*exp(-sqr(y-a)/2.0);
        return(resp);

case 13: /* relativo a Eyz,Ey*/
        resp=Rj-y*corjk-stdj*(1.0-corjk*corjk);
        resp/=sqrt(2.0*(1.0-corjk*corjk));
        resp=(1.0+sgn*erff(resp))*exp(-sqr(y-a)/2.0);
        return(resp);

case 14: /* relativo a Ez,E1*/
        resp=Rj-y*corjk;
        resp/=sqrt(2.0*(1.0-corjk*corjk));
        resp=(1.0+sgn*erff(resp))*exp(-sqr(y-a)/2.0);
        return(resp);

case 15: /* relativo a Exyz*/
        T=(stdk*B-stdi*c+corij*(-stdj*c+stdk*C))/c;
        L=(Ri+T)/sqrt(2.0);
        resp=Rk+(x*(B*sqrt(2.0)+C*corij*sqrt(2.0))+B*L*sqrt(2.0)
             -B*T+C*sqrt(2.0*c)*y+C*c*stdj-C*C*stdk
             +C*corij*sqrt(2.0)*L-C*corij*T)/c-stdk*R;
        resp*=sqrt(c/(2.0*R));
        resp=(1.0+sgn*erff(resp))*exp(-sqr(y-a));
        return(resp);

case 16: /* relativo a Exy*/
        T=(-stdi*c+corij*(-stdj*c))/c;
        L=(Ri+T)/sqrt(2.0);
        resp=Rk+(x*(B*sqrt(2.0)+C*corij*sqrt(2.0))+B*L*sqrt(2.0)
             -B*T+C*sqrt(2.0*c)*y+C*c*stdj+C*corij*sqrt(2.0)*L
             -C*corij*T)/c;
        resp*=sqrt(c/(2.0*R));
        resp=(1.0+sgn*erff(resp))*exp(-sqr(y-a));
        return(resp);


case 17: /* relativo a Exz*/
        T=(stdk*B-stdi*c+corij*(stdk*C))/c;
        L=(Ri+T)/sqrt(2.0);
```

```
        resp=Rk+(x*(B*sqrt(2.0)+C*corij*sqrt(2.0))+B*L*sqrt(2.0)
             -B*T+C*sqrt(2.0*c)*y-C*C*stdk+C*corij*sqrt(2.0)*L
             -C*corij*T)/c-stdk*R;
        resp*=sqrt(c/(2.0*R));
        resp=(1.0+sgn*erff(resp))*exp(-sqr(y-a));
        return(resp);

case 18: /* relativo a Eyz*/
        T=(stdk*B+corij*(-stdj*c+stdk*C))/c;
        L=(Ri+T)/sqrt(2.0);
        resp=Rk+(x*(B*sqrt(2.0)+C*corij*sqrt(2.0))+B*L*sqrt(2.0)
             -B*T+C*sqrt(2.0*c)*y+C*c*stdj-C*C*stdk
             +C*corij*sqrt(2.0)*L-C*corij*T)/c-stdk*R;
        resp*=sqrt(c/(2.0*R));
        resp=(1.0+sgn*erff(resp))*exp(-sqr(y-a));
        return(resp);

case 19: /* relativo a Ex*/
        T=-stdi;
        L=(Ri+T)/sqrt(2.0);
        resp=Rk+(x*(B*sqrt(2.0)+C*corij*sqrt(2.0))+B*L*sqrt(2.0)
             -B*T+C*sqrt(2.0*c)*y+C*corij*sqrt(2.0)*L
             -C*corij*T)/c;
        resp*=sqrt(c/(2.0*R));
        resp=(1.0+sgn*erff(resp))*exp(-sqr(y-a));
        return(resp);


case 20: /* relativo a Ey*/
        T=corij*-stdj;
        L=(Ri+T)/sqrt(2.0);
        resp=Rk+(x*(B*sqrt(2.0)+C*corij*sqrt(2.0))+B*L*sqrt(2.0)
             -B*T+C*sqrt(2.0*c)*y+C*c*stdj+C*corij*sqrt(2.0)*L
             -C*corij*T)/c;
        resp*=sqrt(c/(2.0*R));
        resp=(1.0+sgn*erff(resp))*exp(-sqr(y-a));
        return(resp); //-2.0*x*Ri

case 21: /* relativo a Ez*/
        T=(stdk*B+corij*stdk*C)/c;
        L=(Ri+T)/sqrt(2.0);
        resp=Rk+(x*(B*sqrt(2.0)+C*corij*sqrt(2.0))+B*L*sqrt(2.0)
             -B*T+C*sqrt(2.0*c)*y-C*C*stdk+C*corij*sqrt(2.0)*L
             -C*corij*T)/c-stdk*R;
        resp*=sqrt(c/(2.0*R));
        resp=(1.0+sgn*erff(resp))*exp(-sqr(y-a));
```

```
                        return(resp);

                case 22: /* relativo a E1*/
                        T=0.0;
                        L=Ri/sqrt(2.0);
                        resp=Rk+(x*(B*sqrt(2.0)+C*corij*sqrt(2.0))+B*L*sqrt(2.0)
                             -B*T+C*sqrt(2.0*c)*y+C*corij*sqrt(2.0)*L
                             -C*corij*T)/c;
                        resp*=sqrt(c/(2.0*R));
                        resp=(1.0+sgn*erff(resp))*exp(-sqr(y-a));
                        return(resp);

               case 41: // teste
                        resp=(1-erff(y+x+2))*exp(-y*y-y);
                        //resp=(1-erff(3*y-x-2))*exp(-y*y-y);
                        return(resp);
        }
        printf(" \n deu erro !! ");
        return(1.0E308);
}


/*****************************************************************************/
/**         rotinas de integração para o cálculo das assimetrias           **/
/*****************************************************************************/


double romb_skew(char opcao, double a, double b, double sgn, double phi,
                 double stdi, double stdj, double stdk, double corij,
                 double corik, double corjk, double Ri, double Rj,
                 double Rk, double L)
/*****************************************************************************/
/* método de romberg para integracao em intervalos fechados (a,b)          */
/* L = termo da funcao que leva a exponencial : exp[(x-L)^2/2]             */
/*****************************************************************************/
{
        double *T=NULL, *F=NULL, *X=NULL;
        double h,resp;
        int i,j,n,fim,l;


        X=aloc_vet(1);
        T=aloc_vet(1);
        F=aloc_vet(1);
        n=1;
        fim=0;
```

```
        T[0]=-1;
        while ((!fim) && (n<9)){
                for (l=2,j=1;j<n;j++) l<<=1;   /* l=2^n */
                T=realoca(T,n+1);
                F=realoca(F,l+1);
                X=realoca(X,l+1);
                T[n]=0.0;
                h=(b-a)/l;
                if (n>1){
                        for (j=l/2;j>=1;j--){
                                X[2*j]=X[j];
                                F[2*j]=F[j];
                        }
                        for (j=1;j<=l/2;j++){
                                i=2*j-1;
                                X[i]=X[i-1]+h;
                                F[i]=f_skew(opcao,sgn,phi,X[i],stdi,stdj,
                                            stdk,corij,corik,corjk,Ri,Rj,
                                            Rk,L);
                        }
                        for (i=0;i<l;i++) T[n]+=F[i]+F[i+1];
                        T[n]*=0.5*h;
                        if ((fabs(T[n]-T[n-1]) < 0.00001) | n==8) fim=1;
                        else n++;
                }
                else{
                        for (i=0;i<=l;i++){
                                X[i]=a+i*h;
                                F[i]=f_skew(opcao,sgn,phi,X[i],stdi,stdj,
                                            stdk,corij,corik,corjk,Ri,Rj,
                                            Rk,L);
                        }
                        for (i=0;i<l;i++) T[n]+=F[i]+F[i+1];
                        T[n]*=0.5*h;
                        n++;
                }
        }
        resp=extrap(T,n);
        free(T);
        free(X);
        free(F);
        return(resp);
}
```

```
double trap_skew(char opcao, double a, double delta, double lim, double sgn,
                 double phi, double stdi, double stdj, double stdk,
                 double corij, double corik, double corjk, double Ri,
                 double Rj, double Rk, double L)
/**************************************************************************/
/* procedimento que aplica o método do trapezio + extrapol. de Richardson  */
/* para calcular uma integral.                                            */
/* phi = no'da integracao gaussiana;                                      */
/* a = limite finito da integral ( a integral e'de "a" ate' +infinito ou de*/
/* -infinito ate' "a".                                                    */
/* opcao = indice da funcao integrando na rotina f_skew(...);             */
/* L=termo da exponencial f(x)*exp[((x-L)/delta)^2];                      */
/* lim=+1 se estiver integrando de "a" ate' +infinito  e -1 se de          */
/* -infinito  ate' "a";                                                   */
/* sgn=+1 se na funcao integrando temos (1+erf(..)) e e' -1 se for         */
/* (1-erf(..));                                                           */
/* Ri,Rj,Rk = constantes.                                                 */
/* delta : vem da expressao f(x) = exp{-[(x-a)/delta]^2}                  */
/**************************************************************************/

{

        double *F=NULL, *X=NULL, *T=NULL;
        double quad, erro;
        int nf,k, n;
        long m, Nf,i, j,l,N;


        X=aloc_vet(3);
        F=aloc_vet(3);
        T=aloc_vet(3);
        N=3;
        n=1;
        erro=1.0;
        while (( (erro > 0.0001) | (erro < -0.0001)) | (n <= 8))
        {
                if (n>1)
                {
                        for (m=2,j=1;j<n;j++) m<<=1;
                        Nf=(m-1)*2+1;
                        X=realoca(X,Nf);
                        F=realoca(F,Nf);
                        T=realoca(T,n);
                        for (i=N-1;i>0;i--)
                        {
                                        X[2*i]=X[i];
```

```
                                F[2*i]=F[i];
                }
                m=0;
                j=0;
                for (k=0;k<n-1;k++)
                {
                        for (l=1,i=1;i<=n-k;i++) l<<=1;
                        m+=l;
                        for (i=j;i< m/2;i++)
                        {
                            X[2*i+1]=X[2*i]+lim*delta/l;
                            F[2*i+1]=f_skew(opcao,sgn,phi,X[2*i+1],
                                            stdi,stdj,stdk,corij,
                                            corik,corjk,Ri,Rj,Rk,L);
                        }
                        j=i;
                }
                for (i=Nf-2;i<Nf;i++)
                {
                        X[i]=X[i-1]+lim*delta/2;
                        F[i]=f_skew(opcao,sgn,phi,X[i],stdi,stdj,stdk,
                                    corij,corik,corjk,Ri,Rj,Rk,L);
                }
                N=Nf;
        }
        else for (i=0;i<=2;i++)
                  {
                         X[i]=a+lim*i*delta/2;
                         F[i]=f_skew(opcao,sgn,phi,X[i],stdi,stdj,
                                     stdk,corij,corik,corjk,Ri,Rj,
                                     Rk,L);
             }
        T[n]=trapezio(F,n,delta);
        if (n > 1) erro = T[n]-T[n-1];
        n++;
}
n--;
nf=n;
for (k=1;k<nf;k++)
{
        for (m=4,j=1;j<k;j++) m<<=2;
        for (n=1;n<=(nf-k);n++)
        {
                quad=(m*T[n+1]-T[n])/(m-1);
                T[n]=quad;
        }
```

```
        }
        free(X);
        free(F);
        free(T);
        return(quad);
}


double I_skew(char opcao, double G, double delta, double lim,
              double sgn, double phi, double stdi, double stdj, double stdk,
              double corij, double corik, double corjk, double Ri, double Rj,
              double Rk, double L)
/*************************************************************************/
/* esta funcao chama as rotinas de integracao numerica apropriadas para o  */
/* calculo da integral desejada;                                           */
/* opcao = indice da funcao integrando na rotina f_skew(...);              */
/* G=extremo finito da integral;                                           */
/* L=termo da exponencial f(x)*exp[((x-L)/delta)^2];                       */
/* lim=+1 se estiver integrando de G ate' +infinito  e -1 se de -infinity  */
/*     ate' G;                                                             */
/* sgn=+1 se na funcao integrando temos (1+erf(..)) e e' -1 se for         */
/* (1-erf(..));                                                            */
/* phi = no' da integracao gaussiana.                                      */
/* Ri,Rj,Rk = constantes.                                                  */
/* delta : vem da expressao f(x) = exp{-[(x-L)/delta]^2}                   */
/*************************************************************************/
{
        double I;

        if (fabs(G-L) < 0.5*delta){
                I=trap_skew(opcao,G,delta,lim,sgn,phi,stdi,stdj,stdk,corij,
                            corik,corjk,Ri,Rj,Rk,L);
                return(I);
        }
        else{
                if ((G-L)*lim > 0.0){
                        I=trap_skew(opcao,G,delta,lim,sgn,phi,stdi,stdj,stdk,
                                    corij,corik,corjk,Ri,Rj,Rk,L);
                        return(I);
                }
                else{
                        I=trap_skew(opcao,L,delta,lim,sgn,phi,stdi,stdj,stdk,
                                    corij,corik,corjk,Ri,Rj,Rk,L);
                        if (G > L)
                           I+=romb_skew(opcao,L,G,sgn,phi,stdi,stdj,
                                        stdk,corij,corik,corjk,Ri,Rj,Rk,L);
```

```
                              else
                                 I+=romb_skew(opcao,G,L,sgn,phi,stdi,stdj,
                                         stdk,corij,corik,corjk,Ri,Rj,Rk,L);
                              return(I);
                    }
          }
}
```

Apêndice B

Implementações em Matlab e Gams

B.1 Algoritmo para o cálculo de nós e pesos da Quadratura Gaussiana

```
/*************************************************/
/********            COEFGAUS.M         **********/
/*************************************************/

n=16;
integ(1)=0.5*sqrt(pi);
integ(2)=0.5;
for i=1:2*n-1
   integ(i+2)=i*0.5*integ(i);
end;
b=[];
A=[];
x=[];
y=[];
a=[];
for i=1:n
        for j=1:n A(i,j)=integ(i+j-1); end;
        b(i)=-integ(i+n);
end;
b=b';
[L,U]=lu(A);
invL=inv(L);
invU=inv(U);
y=invL*b;
x=invU*y;
r=A*x-b;
```

```
eaux=invL*r;
e=invU*eaux;
x1=x-e;
r=A*x1-b;
while norm(r) > 0.00000001
x=x1;
eaux=invL*r;
e=invU*eaux;
x1=x-e;
r=A*x1-b;
end
c=x1;
c(n+1)=1;
aux=c;
for i=1:n+1
  c(i)=aux(n+1-i+1);
end;
x=roots(c);
x=sort(x);
for i=1:n
        a(i)=0;
        t=x(i);
        y=x;
        y(i)=[];
        [P]=produto(t,y);
        ind=-1;
        for j=n:-1:1
            ind=ind*(-1);
            [l]=Lij(n,i,j-1,x);
            a(i)=a(i)+ind*l*integ(j);
        end;
        a(i)=a(i)/P;
end;
a=a'


/*****    Lij.m    *****/

function[r]=Lij(n,i,j,x)
if (n>3)
        y=x;
        y(i)=[];
        if (j<n-2 & j>0)
            [r]=perm(n-1,n-j-1,y);
        end;
        if (j==n-1)
```

```
            r=1;
        end;
        if (j==n-2)
                um=ones(1,n-1);
                r=y'*um';
        end;
        if (j==0)
                r=1;
                for k=1:n-1
                   r=r*y(k);
                end;
        end;
end;
if (n==2)
        i;j;
        y=x;
        y(i)=[];
        if (j==0)
          r=y;
        end;
        if (j==1)
          r=1;
        end;
end;
if (n==3)
        y=x;
        y(i)=[];
        if (j==2)
             r=1;
        end;
        if (j==0)
           r=y(1)*y(2);
        end;
        if (j==1)
          r=y(1)+y(2);
        end;
end;
```

```
/*****    produto.m   *****/

function[P]=produto(t,y)
d=-y+t;
P=prod(d);
```

B.2 Arquivo em GAMS - Seleção de Portfólios usando Assimetria

```
TITLE  OTIMIZACAO DE PORTFOLIOS USANDO MOMENTOS DE TERCEIRA ORDEM

SET I componentes formadores do portfolio
    /BB4, PET4, TEL4, VAL43, OTC16, OTC17, OTP19, OTP24/

    ITERA conjunto para dirigir iteracoes de alpha / ITA-1*ITA-20 /

    ITERE conjunto para dirigir iteracoes de eta / ITE-1*ITE-1 /;


ALIAS (I,J);

ALIAS (I,K);

SCALAR

ETA parametro de ponderacao da media /1.5/
ALPHA parametro de ponderacao do skew /0.0/;

PARAMETER

VALUEA(ITERA) usado para atualizar alpha
VALUEE(ITERE) usado para atualizar eta

ESP(I) taxa de retorno simples esperada para os componentes do portfolio

/BB4    -0.1030799667
 PET4    0.1766551319
 TEL4    0.1228039581
 VAL4    0.0153334211
 OTC16   0.3859981177
 OTC17   0.7067222457
 OTP19  -0.8211298158
 OTP24  -0.5926258435/;


TABLE COV(I,J) array de variancia e covariancia dos componentes do portfolio

          BB4             PET4         TEL4           VAL4
OTC16       OTC17          OTP19          OTP24
BB4   1.0000000000   0.1823564539   0.1578521444   0.0888047461   0.1568451573
0.1547705883   -0.1073096454   -0.1383966747
PET4  0.1823564539   1.0000000000   0.5147879950   0.6293648575  0.5118030420
0.5054410381   -0.3461757989   -0.4485565937
TEL4  0.1578521444   0.5147879950   1.0000000000   0.5835617107  0.9963686397
0.9869885907   -0.6444407927   -0.8502748970
VAL4  0.0888047461   0.6293648575   0.5835617107   1.0000000000  0.5804297367
```

```
0.5735589167   -0.3892180077   -0.5061052322
OTC16  0.1568451573   0.5118030420   0.9963686397   0.5804297367 1.0000000000
0.9953744913   -0.5969581835   -0.8280633925
OTC17  0.1547705883   0.5054410381   0.9869885907   0.5735589167  0.9953744913
1.0000000000   -0.5247821186   -0.7875704685
OTP19  -0.1073096454   -0.3461757989   -0.6444407927   -0.3892180077
-0.5969581835   -0.5247821186   1.0000000000   0.8654714267
OTP24  -0.1383966747   -0.4485565937   -0.8502748970   -0.5061052322
-0.8280633925   -0.7875704685   0.8654714267   1.0000000000;



TABLE M(I,J,K) array de skewness e coskewness

                 BB4             PET4          VAL4           WHM3
OTC16            OTC17            OTP19          OTP24
BB4.BB4      0.0505385252    0.0030752561    0.0023308007    0.0012892656
0.0132785857    0.0207538627    -0.3311915830    -0.3386035243
BB4.PET4      0.0030752561    0.0007138585    0.0003400919    0.0002965267
0.0022198392    0.0040654380    0.0013325650    0.0008032973
BB4.TEL4      0.0023308007    0.0003400919    0.0004391309    0.0002133898
0.0029733847    0.0056692507    0.0024797822    0.0018084445
BB4.VAL4      0.0012892656    0.0002965267    0.0002133898    0.0002450382
0.0015037447    0.0029612989    0.0015394342    0.0011557139
BB4.OTC16    0.0132785857    0.0022198392    0.0029733847    0.0015037447
0.0188913176    0.0347165842    0.0114986329    0.0081968353
BB4.OTC17    0.0207538627    0.0040654380    0.0056692507    0.0029612989
0.0347165842    0.0609494097    0.0134423026    0.0087850325
BB4.OTP19    -0.3311915830    0.0013325650    0.0024797822    0.0015394342
0.0114986329    0.0134423026    -0.0157566302    -0.0138419035
BB4.OTP24    -0.3386035243    0.0008032973    0.0018084445    0.0011557139
0.0081968353    0.0087850325    -0.0138419035    -0.0127156809
PET4.BB4    0.0030752561    0.0007138585    0.0003400919    0.0002965267
0.0022198392    0.0040654380    0.0013325650    0.0008032973
PET4.PET4    0.0007138585    0.0024051754    0.0008989431    0.0011675090
0.0055552581    0.0096056800    -0.4871690301    -0.5089445838
PET4.TEL4    0.0003400919    0.0008989431    0.0007573136    0.0006039625
0.0050792918    0.0096074894    0.0040030421    0.0028953468
PET4.VAL4    0.0002965267    0.0011675090    0.0006039625    0.0010345408
0.0038907043    0.0070424573    0.0020847373    0.0012027831
PET4.OTC16    0.0022198392    0.0055552581    0.0050792918    0.0038907043
0.0321063469    0.0586652091    0.0185536755    0.0130896705
PET4.OTC17    0.0040654380    0.0096056800    0.0096074894    0.0070424573
0.0586652091    0.1026386252    0.0216346313    0.0139114253
PET4.OTP19    0.0013325650    -0.4871690301    0.0040030421    0.0020847373
0.0185536755    0.0216346313    -0.0255747717    -0.0224587425
PET4.OTP24    0.0008032973    -0.5089445838    0.0028953468    0.0012027831
0.0130896705    0.0139114253    -0.0224587425    -0.0206404453
TEL4.BB4    0.0023308007    0.0003400919    0.0004391309    0.0002133898
0.0029733847    0.0056692507    0.0024797822    0.0018084445
TEL4.PET4    0.0003400919    0.0008989431    0.0007573136    0.0006039625
```

```
0.0050792918    0.0096074894    0.0040030421    0.0028953468
TEL4.TEL4   0.0004391309    0.0007573136    0.0015265161    0.0007793724
0.0099321075    0.0182933718    0.0063348778    0.0044177621
TEL4.VAL4   0.0002133898    0.0006039625    0.0007793724    0.0008174455
0.0051925756    0.0097663174    0.0039242142    0.0028199591
TEL4.OTC16   0.0029733847    0.0050792918    0.0099321075    0.0051925756
0.0617328400    0.1106294470    0.0293001546    0.0197347840
TEL4.OTC17   0.0056692507    0.0096074894    0.0182933718    0.0097663174
0.1106294470    0.1912471487    0.0337895302    0.0201606941
TEL4.OTP19   0.0024797822    0.0040030421    0.0063348778    0.0039242142
0.0293001546    0.0337895302    -0.0414146218    -0.0363098170
TEL4.OTP24   0.0018084445    0.0028953468    0.0044177621    0.0028199591
0.0197347840    0.0201606941    -0.0363098170    -0.0334200513
VAL4.BB4   0.0012892656    0.0002965267    0.0002133898    0.0002450382
0.0015037447    0.0029612989    0.0015394342    0.0011557139
VAL4.PET4   0.0002965267    0.0011675090    0.0006039625    0.0010345408
0.0038907043    0.0070424573    0.0020847373    0.0012027831
VAL4.TEL4   0.0002133898    0.0006039625    0.0007793724    0.0008174455
0.0051925756    0.0097663174    0.0039242142    0.0028199591
VAL4.VAL4   0.0002450382    0.0010345408    0.0008174455    0.0016572883
0.0050854663    0.0088699060    -0.3547914660    -0.3784112515
VAL4.OTC16   0.0015037447    0.0038907043    0.0051925756    0.0050854663
0.0327037764    0.0595133146    0.0181822160    0.0127229373
VAL4.OTC17   0.0029612989    0.0070424573    0.0097663174    0.0088699060
0.0595133146    0.1038630354    0.0211585210    0.0134302029
VAL4.OTP19   0.0015394342    0.0020847373    0.0039242142    -0.3547914660
0.0181822160    0.0211585210    -0.0251774405    -0.0221034374
VAL4.OTP24   0.0011557139    0.0012027831    0.0028199591    -0.3784112515
0.0127229373    0.0134302029    -0.0221034374    -0.0203203229
OTC16.BB4   0.0132785857    0.0022198392    0.0029733847    0.0015037447
0.0188913176    0.0347165842    0.0114986329    0.0081968353
OTC16.PET4   0.0022198392    0.0055552581    0.0050792918    0.0038907043
0.0321063469    0.0586652091    0.0185536755    0.0130896705
OTC16.TEL4   0.0029733847    0.0050792918    0.0099321075    0.0051925756
0.0617328400    0.1106294470    0.0293001546    0.0197347840
OTC16.VAL4   0.0015037447    0.0038907043    0.0051925756    0.0050854663
0.0327037764    0.0595133146    0.0181822160    0.0127229373
OTC16.OTC16   0.0188913176    0.0321063469    0.0617328400    0.0327037764
0.3735381203    0.6575132436    0.1394270743    0.0907463435
OTC16.OTC17   0.0347165842    0.0586652091    0.1106294470    0.0595133146
0.6575132436    1.1235934363    0.1587891402    0.0889325628
OTC16.OTP19   0.0114986329    0.0185536755    0.0293001546    0.0181822160
0.1394270743    0.1587891402    -0.1964613828    -0.1742178881
OTC16.OTP24   0.0081968353    0.0130896705    0.0197347840    0.0127229373
0.0907463435    0.0889325628    -0.1742178881    -0.1644250366
OTC17.BB4   0.0207538627    0.0040654380    0.0056692507    0.0029612989
0.0347165842    0.0609494097    0.0134423026    0.0087850325
OTC17.PET4   0.0040654380    0.0096056800    0.0096074894    0.0070424573
0.0586652091    0.1026386252    0.0216346313    0.0139114253
OTC17.TEL4   0.0056692507    0.0096074894    0.0182933718    0.0097663174
0.1106294470    0.1912471487    0.0337895302    0.0201606941
OTC17.VAL4   0.0029612989    0.0070424573    0.0097663174    0.0088699060
```

```
0.0595133146    0.1038630354    0.0211585210    0.0134302029
OTC17.OTC16   0.0347165842    0.0586652091    0.1106294470    0.0595133146
0.6575132436    1.1235934363    0.1587891402    0.0889325628
OTC17.OTC17   0.0609494097    0.1026386252    0.1912471487    0.1038630354
1.1235934363    1.8815840333    0.1735543668    0.0738254268
OTC17.OTP19   0.0134423026   0.0216346313    0.0337895302    0.0211585210
0.1587891402    0.1735543668    -0.2489452686    -0.2183983394
OTC17.OTP24   0.0087850325   0.0139114253    0.0201606941    0.0134302029
0.0889325628    0.0738254268    -0.2183983394    -0.2098481702
OTP19.BB4   -0.3311915830    0.0013325650    0.0024797822    0.0015394342
0.0114986329    0.0134423026    -0.0157566302    -0.0138419035
OTP19.PET4   0.0013325650    -0.4871690301    0.0040030421    0.0020847373
0.0185536755    0.0216346313    -0.0255747717    -0.0224587425
OTP19.TEL4   0.0024797822    0.0040030421    0.0063348778    0.0039242142
0.0293001546    0.0337895302    -0.0414146218    -0.0363098170
OTP19.VAL4   0.0015394342    0.0020847373    0.0039242142    -0.3547914660
0.0181822160    0.0211585210    -0.0251774405    -0.0221034374
OTP19.OTC16  0.0114986329   0.0185536755    0.0293001546    0.0181822160
0.1394270743    0.1587891402    -0.1964613828    -0.1742178881
OTP19.OTC17  0.0134423026   0.0216346313    0.0337895302    0.0211585210
0.1587891402    0.1735543668    -0.2489452686    -0.2183983394
OTP19.OTP19  -0.0157566302  -0.0255747717  -0.0414146218  -0.0251774405
-0.1964613828    -0.2489452686    0.2134013931    0.1947074869
OTP19.OTP24  -0.0138419035  -0.0224587425  -0.0363098170  -0.0221034374
-0.1742178881    -0.2183983394    0.1947074869    0.1748466011
OTP24.BB4   -0.3386035243   0.0008032973    0.0018084445    0.0011557139
0.0081968353    0.0087850325    -0.0138419035    -0.0127156809
OTP24.PET4   0.0008032973  -0.5089445838    0.0028953468    0.0012027831
0.0130896705    0.0139114253    -0.0224587425    -0.0206404453
OTP24.TEL4   0.0018084445   0.0028953468    0.0044177621    0.0028199591
0.0197347840    0.0201606941    -0.0363098170    -0.0334200513
OTP24.VAL4   0.0011557139    0.0012027831    0.0028199591    -0.3784112515
0.0127229373    0.0134302029    -0.0221034374   -0.0203203229
OTP24.OTC16  0.0081968353  0.0130896705    0.0197347840    0.0127229373
0.0907463435    0.0889325628    -0.1742178881    -0.1644250366
OTP24.OTC17  0.0087850325  0.0139114253    0.0201606941    0.0134302029
0.0889325628    0.0738254268    -0.2183983394    -0.2098481702
OTP24.OTP19  -0.0138419035  -0.0224587425  -0.0363098170  -0.0221034374
-0.1742178881    -0.2183983394    0.1947074869    0.1748466011
OTP24.OTP24  -0.0127156809  -0.0206404453 -0.0334200513  -0.0203203229
-0.1644250366  -0.2098481702    0.1748466011    0.1633187616;


VARIABLES

X(I) fracao do total do investimento no ativo I
U    utilidade esperada do portfolio
C    covariancia do portfolio
E    retorno esperado do portfolio
S    skewness do portfolio;
```

```
POSITIVE VARIABLE X;

VALUEA("ITA-1")=ALPHA;

VALUEE("ITE-1")=ETA;

EQUATIONS

UTIL funcao utilidade esperada de terceira ordem
FSUM fracoes devem somar 1.0
COVARP covariancia do portfolio
EXRETP retorno esperado do portfolio
SKEWP  terceiro momento do portfolio;



UTIL.. U =E= ETA*SUM(I,X(I)*ESP(I))-0.5*SUM(I,X(I)*SUM(J,X(J)*COV(I,J)))
             +ALPHA*SUM(I,X(I)*SUM(J,X(J)*SUM(K,X(K)*M(I,J,K))));
FSUM.. SUM(I,X(I)) =E= 1.0;
COVARP.. C =E= SUM(I,X(I)*SUM(J,X(J)*COV(I,J)));
EXRETP.. E =E= SUM(I,X(I)*ESP(I));
SKEWP..  S =E= SUM(I,X(I)*SUM(J,X(J)*SUM(K,X(K)*M(I,J,K))));

MODEL PORTFOLIO /ALL/;

FILE MED, COVAR, COSKEW, RELAT;
PUT RELAT; RELAT.ND=6;
PUT MED; MED.ND=6; PUT ´E=[´/;
PUT COVAR; COVAR.ND=6; PUT ´C=[´/;
PUT COSKEW; COSKEW.ND=6; PUT ´S=[´/;

LOOP(ITERE,

     VALUEE(ITERE+1) = VALUEE(ITERE)+0.1;

     ETA=VALUEE(ITERE);

     LOOP(ITERA,

                VALUEA(ITERA+1) = VALUEA(ITERA)+0.01;

                ALPHA=VALUEA(ITERA);

                SOLVE PORTFOLIO USING NLP MAXIMIZING U;

                DISPLAY  ETA, ALPHA, U.L, E.L, C.L, S.L, X.L;

                PUT RELAT; PUT ´ALPHA = ´; PUT ALPHA/;

                PUT ´ETA = ´; PUT ETA/;

                PUT ´FUNCAO OBJETIVO = ´; PUT U.L//;
```

```
                PUT ´Composition of portfolio:´/;

                LOOP(I,PUT I.TL, X.L(I) /);

                PUT ´-------------------------------------------------´//;

                PUT MED; PUT E.L/;

                PUT COVAR; PUT C.L/;

                PUT COSKEW; PUT S.L/;
    );

);
PUT MED; PUT ´]´;
PUT COVAR; PUT ´]´;
PUT COSKEW; PUT ´]´;
```

Referências Bibliográficas

[1] Julio Michael Stern e Celma de Oliveira Ribeiro. *Métodos de Otimização em Finanças.*

[2] G.J.Alexander and J.C.Francis. *Portfolio Analysis.* Prentice-Hall, Englewood Cliffs, New Jersey, 1986.

[3] K.J.Cohen J.A.Pogue. *An Empirical Evaluation of Alternative Portfolio Selection Models.* Journal of Business, 1967.

[4] J.Hull. *Introduction to Futures and Options Markets.* Prentice-Hall,1991.

[5] J.E.Ingersoll. *Theory of Financial Decision Making.* Studies in Financial Economics, Rowman and Littlefield, Savage, Maryland, 1987. cd

[6] H.M.Markowitz. *Mean-Variance Analisys in Portfolio Choice and Capital Markets.* Basil Blackwell, Cambridge, Massachusetts, 1987.

[7] W.F.Sharpe. *Portfolio Theory and Capital Markets.* McGraw-Hill, New York, 1970.

[8] H.R.Schwarz. *Numerical Analisys.* John Wiley & Sons, 1989.

[9] T. Kariya. *Quantitative Methods for Portfolio Analysis.* Kluwer, 1993.

[10] H.R.Stoll and R.E.Whaley. *Futures and Options* Soult-Western, 1993.

[11] K.J.Arrow. *Aspects of the Theory of Risky-Bearing. Lectures* Helsinki, 1964.

[12] Tsiang. *The Rationale of the Mean-Standard Deviation Analysis, Skewness Preference, and the Demand for Money.* The American Economic Review 62, 1972.

[13] Stoer & Bulirsch. *Introduction to Numerical Analysis.* Springer-Verlag, 1980.

[14] Engeln-Mullges & Frank Uhlig. *Numerical Algorithms with C.* Springer-Verlag, 1996.

[15] J.C.Singleton & J. Wingender. *Skewness Persistence in Common Stock Returns.* Journal of Financial and Quantitative Analysis, vol. 21, n.3, September/1986.

[16] P. Chunhachinda, K. Dandapani, S. Harnid, A. J. Prakash. *Portfolio Selection and Skewness: evidence from international stock markets* . Journal of Banking & Finance 21(1997), pp.143-147.

[17] A. Kraus& R. H. Litzenberger. *Skewness Preference and the Valuation of Risk Assets.* The Journal of Finance, vol 31, n.4, September/1976.

[18] F. D. Arditti & H. Levy. *Portfolio Efficiency Analysis in Three Moments: the Multiperiod Case.* The Journal of Finance, vol.30, n.3, June/1975.

[19] H. Levy & H. M. Markowitz. *Approximating Expected Utility by a Function of Mean and Variance.* The American Economic Review, vol.69, n.3, June/1979.

[20] M. Kijima& M. Ohnishi. *Mean-Risk analysis of risk aversion and wealth effects on optimal portfolios with multiple investments opportunities.* Annals of Operation Research, 45(1993), pp.147-163.

[21] A. Halston. *A First Course in Numerical Analysis.* McGraw-Hill, 1965.

[22] H. M. Markowitz. *Portfolio Selection - Efficient Diversification of Investments.* Blackwell, 1995.