

"A FEA e a USP respeitam os direitos autorais deste trabalho. Nós acreditamos que a melhor proteção contra o uso ilegítimo deste texto é a publicação online. Além de preservar o conteúdo motiva-nos oferecer à sociedade o conhecimento produzido no âmbito da universidade pública e dar publicidade ao esforço do pesquisador. Entretanto, caso não seja do interesse do autor manter o documento online, pedimos compreensão em relação à iniciativa e o contato pelo e-mail [bibfea@usp.br](mailto:bibfea@usp.br) para que possamos tomar as providências cabíveis (remoção da tese ou dissertação da BDTD)."

**UNIVERSIDADE DE SÃO PAULO**  
**FACULDADE DE ECONOMIA, ADMINISTRAÇÃO E CONTABILIDADE**  
**DEPARTAMENTO DE ECONOMIA**  
**PROGRAMA DE DOUTOURADO EM TEORIA ECONÔMICA**

**UM NOVO ALGORITMO PARA APLICAÇÕES DAS SEQUÊNCIAS DE SOBOL À  
PRECIFICAÇÃO DE DERIVATIVOS FINANCEIROS**

**Thierry Barbe**  
**Orientador: Prof. Dr. Marcos Eugênio da Silva**

**SÃO PAULO**  
**2005**

Prof. Dr. Adolpho José Melfi  
Reitor da Universidade de São Paulo

Profa. Dra. Maria Teresa Leme Fleury  
Diretora da Faculdade de Economia, Administração e Contabilidade

Prof. Dr. Ricardo Abramovay  
Chefe do Departamento de Economia

Profa. Dra. Fabiana Rocha  
Coordenadora do Programa de Pós-Graduação em Economia

**THIERRY BARBE**

**UM NOVO ALGORITMO PARA APLICAÇÕES DAS SEQUÊNCIAS DE SOBOL À  
PRECIFICAÇÃO DE DERIVATIVOS FINANCEIROS**

Tese apresentada ao Departamento de Economia da Faculdade  
de Economia, Administração e Contabilidade da Universidade  
de São Paulo como requisito para obtenção do título de  
Doutor em Economia.

**Orientador: Prof. Dr. Marcos Eugênio da Silva**

**SÃO PAULO**

**2005**

Tese defendida e aprovada, em 29.09.2005, no Programa de Pós-Graduação em Economia, pela seguinte comissão julgadora:

Prof. Dr. Marcos Eugênio da Silva  
Prof. Dr. José Carlos de Souza Santos  
Prof. Dr. José de Oliveira Siqueira  
Prof. Dr. Oswaldo Luiz do Valle Costa  
Prof. Dr. Rogério Rosenfeld

Barbe, Thierry.

Um novo algoritmo para aplicação das seqüências de Sobol à precificação de derivativos financeiros / Thierry Barbe – São Paulo 2005.

68p.

Tese(Doutorado) - Universidade de São Paulo, 2005

Bibliografia.

1. Derivativos I. Universidade de São Paulo, Faculdade de Economia, Administração e Contabilidade II. Título.

CDD - 332.645

**À Ana Paula**

**Sempre presente nos momentos mais difíceis.**

Agradeço a minha mulher Ana Paula pela compreensão demonstrada em relação aos sacrifícios necessários à elaboração de uma tese e pelo apoio incondicional, desde sempre, às minhas veleidades acadêmicas.

Agradeço o professor e orientador Dr. Marcos Eugênio da Silva pela confiança depositada em mim e pelos conselhos sempre justos e úteis que contribuíram para a edificação desta tese.

## RESUMO

A simulação Monte Carlo adquiriu enorme popularidade durante a segunda metade do século XX tanto pela facilidade com a qual o método é implementado como pela sua eficácia. No entanto o seu uso depende de um grande esforço computacional que muitas vezes limita o seu alcance. Por isso muito esforço tem sido despendido na tentativa de encontrar maneiras de reduzir este ônus. Técnicas de redução de variância são um exemplo desta linha de pesquisa. Um dos métodos que também visa uma convergência mais rápida consiste em escolher antecipadamente os números “aleatórios” com os quais a simulação será efetuada. Quando estes são oriundos de seqüências de baixa discrepância esta técnica recebe o nome de simulação quase Monte Carlo. Este método mostrou-se mais preciso que a simulação Monte Carlo tradicional para uma ampla gama de problemas. Todavia, o seu uso restringiu-se a simulações desenvolvidas em espaços de pequena dimensão pois, como os demais métodos deterministas, ele sofre da “maldição da dimensionalidade”. Criou-se então o desafio de estender o uso da simulação quase Monte Carlo a problemas de dimensão elevada. Várias soluções tem sido propostas mas nenhuma delas tem conseguido lidar com problemas para as quais a dimensão alcança níveis altíssimos. Além disso demonstrou-se que os poucos resultados satisfatórios em dimensões médias e altas resultavam de propriedades específicas das simulações que as tornavam unicamente sensíveis a um subespaço de dimensão muito menor. Definiu-se então as noções de dimensão efetiva e dimensão nominal de uma simulação Monte Carlo. O nosso trabalho apresenta um método que permite estender o uso das seqüências de Sobol para simulações de dimensão muito maior do que o geralmente admitido como seguro pelos usuários. Isso passa pela descoberta de uma propriedade de uniformidade implícita nos números direcionais e pela implementação de um algoritmo que gera números direcionais eficientes. Em seguida, a construção é avaliada na resolução de integrais de teste e na precificação de instrumentos que pertencem a três famílias de derivativos financeiros. Pela primeira vez na literatura sobre quase Monte Carlo em finanças a análise leva em consideração a dicotomia entre dimensão nominal e dimensão efetiva, o que permite a realização de testes desprovidos de vieses. Conseguimos excelentes resultados para simulações Monte Carlo de dimensão efetiva até 2000.



## ABSTRACT

During the latter half of the twentieth century Monte Carlo gained enormous popularity, both for its efficiency and ease of implementation. However, its use requires great computational cost, a factor that has limited its expansion. This is why a great deal of effort has been spent in an effort to discover ways to reduce this burden. Variance reduction techniques are an example of this line of research. Another technique that seeks faster convergence consists in picking beforehand the “random” numbers used in the simulation. When these numbers are chosen from low discrepancy sequences the method is then called quasi Monte Carlo and has proved more efficient than traditional Monte Carlo for solving a vast array of problems. Nonetheless, its use has been confined to simulations in low dimensional spaces, for, like other deterministic methods, it suffers from the curse of dimensionality. Researchers are now challenged to extend the use of quasi Monte Carlo to problems of very high dimension. Various solutions have been suggested but none has managed to deal with problems of very high dimensions. Furthermore, it has been shown that in various cases where satisfactory results were reached for medium to high dimensional problems the simulation was in fact sensitive only to a subset of much lower dimension. This generated the distinction between nominal and effective dimension of the Monte Carlo simulation. Our work presents a method that unleashes the use of Sobol sequences to dimensions much higher than those for which it has insofar proved operational. This goes through the discovery of a uniformity property implicit in the directional numbers and the implementation of an algorithm that generates efficient directional numbers. The construction is then tested in different settings which range from the estimation of test integrals to the pricing of instruments that belong to three different families of financial derivatives. For the first time in the quasi Monte Carlo literature in finance the analysis takes into consideration the dichotomy between nominal and effective dimension thus reducing much of the bias present in previous tests. Excellent results were obtained for simulations with effective dimension ranging up to 2000.

# Sumário

<b>1</b>	<b>Introdução</b>	<b>1</b>
1.1	Objetivo do trabalho . . . . .	1
1.2	Estrutura da tese . . . . .	2
<b>2</b>	<b>Simulação Quase Monte Carlo</b>	<b>5</b>
2.1	O método de Monte Carlo . . . . .	5
2.1.1	Simulação Monte Carlo . . . . .	5
2.1.2	Aplicação em finanças . . . . .	7
2.1.3	Simulação quase Monte Carlo . . . . .	9
2.2	Seqüências de baixa discrepância . . . . .	9
2.2.1	Seqüências uniformes módulo 1 . . . . .	9
2.2.2	A discrepância de uma seqüência . . . . .	10
2.2.3	Algumas propriedades . . . . .	11
2.2.4	Redes (t,m,d) e seqüências (t,d) . . . . .	11
2.3	As construções clássicas . . . . .	12
2.3.1	Irracionais . . . . .	12
2.3.2	Seqüências de Van der Corput . . . . .	13
2.3.3	As seqüências de Hammersley e Halton . . . . .	13
2.3.4	Seqüências de Faure . . . . .	14
2.3.5	Seqüências de Sobol . . . . .	15
2.3.6	Construções recentes . . . . .	15
2.4	As seqüências de baixa discrepância na prática . . . . .	16
2.4.1	A difusão da simulação quase Monte Carlo . . . . .	16
2.4.2	Resultados paradoxais . . . . .	16

2.4.3	Soluções propostas . . . . .	18
<b>3</b>	<b>Construção da seqüência de Sobol</b>	<b>21</b>
3.1	A construção clássica . . . . .	21
3.1.1	Conceitos preliminares e notação . . . . .	21
3.1.2	Geração das seqüências de Sobol . . . . .	22
3.1.3	Exemplo . . . . .	23
3.2	Uniformidade das seqüências de Sobol . . . . .	26
3.2.1	As seqüências de Sobol são seqüências (t,d) na base 2 . . . . .	26
3.2.2	As propriedades A e B . . . . .	27
3.2.3	Seqüências anômalas . . . . .	28
3.3	Um novo algoritmo para a escolha dos direcionais . . . . .	29
3.3.1	Números direcionais eficientes . . . . .	29
3.3.2	Exemplo . . . . .	31
3.3.3	Um procedimento para geração dos números direcionais arbitrários . . . . .	35
<b>4</b>	<b>A dimensão efetiva</b>	<b>39</b>
4.1	Definição . . . . .	40
4.1.1	A decomposição ANOVA . . . . .	40
4.1.2	A dimensão efetiva . . . . .	41
4.2	O cálculo da dimensão efetiva . . . . .	42
4.3	Integrais de teste . . . . .	43
4.4	Comparação com o método de Joe & Kuo . . . . .	44
4.4.1	O método de Joe & Kuo . . . . .	45
4.4.2	Os resultados . . . . .	45
<b>5</b>	<b>Aplicações em finanças</b>	<b>49</b>
5.1	Opções com barreira . . . . .	50
5.1.1	O produto . . . . .	50
5.1.2	Resultados . . . . .	51
5.2	Opção basket asiática . . . . .	53
5.2.1	O produto . . . . .	53
5.2.2	Resultados . . . . .	55

5.3	Opção asiática . . . . .	57
5.3.1	O produto . . . . .	57
5.3.2	A dimensão efetiva . . . . .	58
5.3.3	Resultados . . . . .	60
6	Conclusão	63

# Capítulo 1

## Introdução

### 1.1 Objetivo do trabalho

- Ao longo dos últimos anos a simulação quase Monte Carlo conquistou uma ampla audiência. A maioria dos livros que aborda, de uma forma ou de outra, a simulação Monte Carlo, possui uma seção voltada a este método. Exemplos são as obras de Hull [2000], Jäckel [2002], Glasserman [2004] e Jorion [2001]. Na prática sente-se que a adoção do método por parte dos usuários de finanças é tímida. Vários fatores contribuem para este quadro. Acreditamos que um dos principais seja a ausência de clareza quanto à eficácia do método. Considerando-se a magnitude das somas envolvidas, a reticência em adotar metodologias em fase de sedimentação é compreensível.

O recente surgimento da noção de dimensão efetiva tem propiciado um grande passo no sentido de sistematizar os resultados apresentados na literatura. As limitações das seqüências de baixa discrepância estão sendo mapeadas com mais rigor. Tudo leva a crer que o método padece mesmo da “maldição da dimensionalidade”<sup>1</sup>. Este reconhecimento gerou um desafio: como permitir o uso de simulação quase Monte Carlo em dimensões muito grandes porém passíveis de acontecerem na prática.

Seguindo esta linha de pesquisa o nosso trabalho consistiu em estender o uso da seqüência de Sobol a dimensões efetivas altas (acima de 1000). Para isso mostramos que certas configurações de números direcionais permitem alcançar critérios predeterminados de uniformidade. Um novo

---

<sup>1</sup>O termo maldição da dimensionalidade é decorrente da rápida divergência de métodos numéricos de integração conforme a dimensão do espaço no qual se integra aumenta.

algoritmo que gera números que obedecem aos critérios especificados implica em seqüências de Sobol desprovidas dos vieses documentados na literatura. Implementamos o algoritmo e submetemos a nova construção tanto a integrais de teste como também à aplicação que mais nos interessam, a precificação de derivativos financeiros.

## 1.2 Estrutura da tese

A introdução tem como propósito apresentar o arcabouço no qual nos inserimos ao longo deste trabalho e o roteiro utilizado para sua exposição.

No capítulo 2 oferecemos uma pequena introdução à simulação quase Monte Carlo. Após breve referência à simulação Monte Carlo apresentamos os critérios que definem as seqüências utilizadas, as construções clássicas e as propriedades das estimativas geradas através deste método. O entendimento das grandes linhas da teoria e uma familiaridade com as seqüências testadas na literatura é crucial para compreender o nosso trabalho. Aproveitamos a mesma seção para efetuar um levantamento dos resultados obtidos com simulação quase Monte Carlo no âmbito financeiro.

O capítulo 3 é inteiramente dedicado às seqüências de Sobol. Começamos com a construção clássica dos números Sobol. Os problemas apresentados pelo método inicialmente proposto são ilustrados graficamente. Certos critérios podem melhorar o desempenho das simulações. Estes são descritos e comentados. Finalmente, sugerimos uma modificação na construção das seqüências que permite estender a utilização dos números de Sobol a problemas de dimensão muito grande. Esta é a principal contribuição do nosso trabalho para a literatura. Abordamos também a questão da implementação do método, tópico crucial, por se tratar de um procedimento numérico.

O capítulo 4 é o primeiro a conter resultados empíricos. Inicialmente o conceito de dimensão efetiva é exposto, de modo a permitir uma avaliação rigorosa dos avanços gerados pela metodologia de escolha dos números de Sobol. O primeiro teste ao qual submetemos a nova construção consiste em estimar o valor de uma integral de dimensão que pode atingir 1000. Os resultados obtidos são comparados com os de Joe e Kuo [2003], que representam o registro de melhor desempenho do método de Sobol que se conhece para problemas desta dimensão.

O capítulo 5 inclui aplicações do método em finanças. A nossa busca por um método que permite diminuir o custo operacional da simulação Monte Carlo parte de um interesse na

precificação de opções “path-dependent- cujo pagamento depende da trajetória do ativo subjacente. Pode-se considerar que atingimos o nosso objetivo se as simulações quase Monte Carlo efetuadas com o uso do algoritmo proposto atingem um resultado satisfatório com um número significativamente menor de trajetórias que o método tradicional. Para avaliarmos o desempenho da simulação quase Monte Carlo escolhemos três tipos de instrumentos: opções asiática, opções com barreira e opções basket asiáticas. Trabalhamos sempre com exercícios que exigem um bom desempenho em altas dimensões. A análise comporta também, no primeiro teste, o cálculo da dimensão efetiva, de modo a permitir uma avaliação rigorosa dos resultados alcançados.

Concluimos com uma rápida recapitulação dos elementos que formam a espinha dorsal desta pesquisa e oferecemos algumas sugestões para futuros trabalhos.





## Capítulo 2

# Simulação Quase Monte Carlo

Antes de escrevermos algo sobre simulação quase Monte Carlo é conveniente mencionarmos a simulação Monte Carlo e a maneira pela qual esta é utilizada em finanças. Em seguida tratamos vários aspectos da simulação quase Monte Carlo. Isso inclui a noção geral de seqüência de baixa discrepância, as diferentes construções existentes e os resultados obtidos em aplicações financeiras.

### 2.1 O método de Monte Carlo

O problema de precificar um derivativo através de simulação Monte Carlo requer que ele seja formulado em termos de uma integral, ou seja, através do valor esperado de um processo estocástico adequado. Mostramos primeiro como a simulação Monte Carlo se aplica à resolução de uma integral e apresentamos um exemplo de aplicação em finanças.

#### 2.1.1 Simulação Monte Carlo

Nas suas grandes linhas, a simulação Monte Carlo é um método numérico destinado a resolver problemas matemáticos através de simulação de variáveis aleatórias. O nome Monte Carlo é oriundo do cassino do principado, já que um dos mecanismos mais simples para gerar números aleatórios é a roleta. A lei dos grandes números e o teorema central do limite conferem ao método as propriedades que o tornam tão atraente, ou seja, a ausência de viés na estimação e uma distribuição normal para o erro, facilitando assim o uso da estimativa. A resolução de integrais é o terreno mais fértil para aplicação do método probabilista. Utilizamos este exemplo

para listar as principais propriedades da simulação Monte Carlo. O leitor interessado numa descrição mais rigorosa dos tópicos a seguir pode consultar os livros de Niederreiter [1992] e Sobol [1993].

Seja  $f$  uma função integrável no espaço  $[0, 1]^d$ . Representamos a integral como,

$$I(f) = \int_{[0,1]^d} f(x) dx$$

Supõe-se que são gerados  $n$  pontos  $u_1, u_2, \dots$  distribuídos de maneira uniforme em  $[0, 1]^d$ . Se avaliarmos  $f$  em cada um desses pontos e calcularmos a média dos resultados obtidos então obtemos uma estimativa do valor de  $I(f)$ :

$$\hat{I}_n(f) = \frac{1}{n} \sum_{i=1}^n f(u_i)$$

A lei forte dos grandes números garante a convergência em probabilidade de  $\hat{I}_n(f)$ ,

$$\hat{I}_n(f) \longrightarrow I(f)$$

Se  $f \in L^2$  podemos definir,

$$\sigma^2(f) = \int_{[0,1]^d} (f(x) - I(f))^2 dx$$

O erro,  $\hat{I}_n(f) - I(f)$ , é portanto distribuído de acordo com uma normal de média 0 e desvio padrão  $\sigma(f)\sqrt{n}^{-1}$ . É possível estimar  $\sigma(f)$  através de,

$$s(f) = \sqrt{\frac{1}{n} \sum_{i=1}^n (f(u_i) - \hat{I}_n(f))^2}$$

Destes resultados decorre a ordem de grandeza do erro obtido através da simulação Monte Carlo, de valor  $\sqrt{N}^{-1}$ . Portanto para diminuir o erro de uma casa decimal é necessário multiplicar por 100 o número de pontos utilizados na simulação.

Conforme relatado por Niederreiter [1992], o método de Monte Carlo é inferior aos métodos numéricos tradicionais, por exemplo a regra trapezoidal, para integração em espaços de dimensão inferior ou igual a 3. Porém o maior atrativo da simulação Monte Carlo é que a ordem de grandeza do seu erro é independente do tamanho ou da dimensão do espaço no qual se integra. Em contraste, a regra trapezoidal possui erro de ordem  $O(n^{-2/d})$ . Essa dependência da ordem do erro em relação à dimensão é uma característica dos métodos deterministas, inclusive da simulação quase Monte Carlo, objeto de nossa pesquisa.

### 2.1.2 Aplicação em finanças

Um dos pilares da teoria financeira pode ser resumido da seguinte maneira: sob certas condições, dentre as quais a neutralidade em relação ao risco, o preço de um ativo é uma esperança dos seus possíveis valores em diferentes estados da natureza. Uma aplicação deste princípio no espaço euclidiano para ativos cujo retorno segue um movimento browniano nos insere no mundo de Black & Scholes. Vejamos, neste contexto, como calcular o preço de uma opção de compra europeia sobre uma ação.

Imaginemos que estamos no instante  $t = 0$  e que a opção vence no momento  $T$ . O seu valor então será de,

$$C_T = (S_T - K)^+$$

onde  $S_t$  é o valor da ação no tempo  $t$  e  $K$  é o preço de exercício. Se denotarmos por  $r$  a taxa de juros livre de risco, então o valor esperado de  $C_T$ , a preços correntes, corresponde a,

$$C_0 = \exp^{-rT} E[(S_T - K)^+] \quad (2.1)$$

Para calcularmos esta esperança necessitamos da distribuição de  $S_T$ . Uma das hipóteses do modelo de Black & Scholes é que  $S_t$ , num mundo neutro ao risco, obedece à seguinte equação diferencial estocástica,

$$\frac{dS_t}{S_t} = rdt + \sigma dW_t \quad (2.2)$$

onde  $\sigma$  é um dos parâmetros do modelo e  $W_t$  um movimento browniano de variância  $t$ . A solução desta equação diferencial corresponde a,

$$S_T = S_0 \exp\left(\left(r - \frac{1}{2}\sigma^2\right)T + \sigma W_T\right) \quad (2.3)$$

O movimento browniano  $W_T$  tem distribuição normal com média 0 e variância  $T$ . Ele pode ser decomposto em  $z\sqrt{T}$ , onde  $z \sim N(0, 1)$ , o que permite reescrever a equação anterior sob a forma,

$$S_T = S_0 \exp\left(\left(r - \frac{1}{2}\sigma^2\right)T + \sigma\sqrt{T}z\right) \quad (2.4)$$

Vemos que o logaritmo do preço possui distribuição normal, logo  $S_T$  possui distribuição lognormal. Portanto a equação 2.1 corresponde à integral da fórmula do pagamento final em relação à distribuição lognormal.

Para estimarmos o preço da opção no instante 0,  $C_0$ , é suficiente gerar uma grande quantidade de preços  $S_T$  de modo a simular a sua distribuição e conseqüentemente a distribuição dos

pagamentos da opção em  $T$ . A média desses valores a preços correntes corresponde à estimativa pela simulação Monte Carlo do prêmio teórico dado pelo modelo de Black & Scholes. Em suma, calcular o valor atual de uma opção "plain vanilla" via simulação Monte Carlo requer o cumprimento do seguinte roteiro,

1. Decidir qual será o "tamanho" da simulação,  $n$ .
2. Gerar  $n$  realizações de variáveis aleatórias distribuídas de acordo com a normal padrão,  $z_1, z_2, \dots, z_n$ .
3. Inserir essas realizações na equação 2.4 de modo a simular  $n$  preços  $S_{T_1}, S_{T_2}, \dots, S_{T_n}$ .
4. Para cada um dos preços gerados na etapa anterior calcular o valor da opção  $C_{T_1}, C_{T_2}, \dots, C_{T_n}$ .
5. Finalmente obtemos o preço da opção no instante 0, ao calcular a média dos  $C_{T_i}$ ,  $1 \leq i \leq n$ , o que equivale a estimar  $E[(S_T - K)^+]$ , e descontá-la pelo fator  $\exp^{-rT}$ .

Este exemplo permite compreender o enorme potencial da simulação Monte Carlo no âmbito da precificação de derivativos. Uma extensão imediata consiste em quebrar a dinâmica de  $S_t$  em vários passos no tempo, o que permite obter uma trajetória de preços para a ação do instante 0 ao instante  $T$ . A facilidade com a qual o método se adapta à precificação de derivativos que dependem de trajetórias de uma ou mais variáveis aleatórias é a razão da popularidade do método Monte Carlo nos mercados financeiros. Opções asiáticas e opções digitais são aplicações já tradicionais do método. Todavia, novos terrenos são constantemente desbravados. É o caso das opções americanas, para as quais a simulação Monte Carlo vem ganhando espaço em detrimento dos métodos que fazem uso de árvores.

Mas a simulação Monte Carlo apresenta também alguns inconvenientes. Um deles é que a obtenção de estimativas razoáveis requer um tamanho,  $n$ , muito grande, e portanto um tempo de computação não desprezível. Em mercados mundialmente interligados, em que os preços dos ativos financeiros variam na ordem de segundos, qualquer perda de tempo pode resultar em oportunidades de negócios desperdiçadas. Esse é um dos principais motivos pela fertilidade das pesquisas que visam a descobrir métodos que permitem a diminuição do tamanho da simulação sem comprometer as estimativas. Dentre os terrenos explorados, a simulação quase Monte Carlo ocupa um lugar de destaque pelos seus resultados encorajadores.

### 2.1.3 Simulação quase Monte Carlo

A simulação quase Monte Carlo pode ser descrita de modo informal como uma simulação Monte Carlo na qual os números aleatórios são substituídos por números escolhidos cuidadosamente e, portanto, já conhecidos. Ao contrário da simulação Monte Carlo a simulação quase Monte Carlo não tenta replicar nenhuma aleatoriedade. Pelo contrário, faz-se uso de seqüências distribuídas da maneira mais uniforme possível. Em suma, trata-se de construir uma seqüência de números com propriedades muito específicas que, no exemplo anterior, substituem as realizações de  $z$  na etapa 2 do procedimento listado.

A vantagem do método em relação à simulação Monte Carlo reside na magnitude do erro máximo, de ordem  $O(1/n^{1-\epsilon})$ . Sob certas condições o  $\epsilon$  pode ser desprezível, o que caracteriza uma melhora nítida em relação à simulação Monte Carlo que apelidamos neste trabalho de método tradicional. Todavia, veremos que o mesmo  $\epsilon$  é também função crescente da dimensão do espaço no qual se calcula a integral. Este é o principal gargalo do método. Até agora não foi possível fornecer técnicas que tenham um desempenho satisfatório em dimensões altas com exceção da proposta promissora de Joe e Kuo [2003]. O objeto deste trabalho consiste justamente em tentar aumentar o conjunto de situações para as quais o método quase Monte Carlo pode ser utilizado.

## 2.2 Seqüências de baixa discrepância

Nas próximas páginas apresentamos as principais definições e resultados necessários a um entendimento da literatura sobre simulação quase Monte Carlo em finanças. O nosso principal objetivo consiste em familiarizar o leitor com alguns conceitos básicos, de modo a podermos utilizar a linguagem já sedimentada na literatura.

### 2.2.1 Seqüências uniformes módulo 1

A teoria que trata de seqüências distribuídas de maneira uniforme é recente, originando-se de um artigo de Herman Weyl [1916]. Nosso principal interesse reside nas seqüências uniformes módulo 1.

Seja  $[y]$  a parte inteira de  $y$ . Se fizermos  $x = y - [y]$  temos  $x \in [0, 1)$ . Diz-se então que a seqüência  $\{x_n\}$  constituída da parte fracionária da seqüência  $\{y_n\}$  é uniformemente distribuída

módulo 1 se para cada par  $a$  e  $b$  de números reais com  $0 \leq a < b < 1$  e para  $N \in \mathbb{Z}^+$ , temos

$$\lim_{N \rightarrow \infty} \frac{\#(\{x_n^N\} \cap [a, b))}{N} = b - a$$

onde  $\#$  é utilizado para a cardinalidade de um conjunto e  $\{x_n^N\}$  representa o conjunto formado pelos termos da sequência  $\{x_n\}$  de índice inferior ou igual a  $N$ .

### 2.2.2 A discrepância de uma sequência

Podemos pensar intuitivamente que algumas seqüências são distribuídas de maneira mais ou menos uniforme que outras. Para conseguirmos efetuar uma classificação dentre as seqüências é necessário algum critério de dispersão. A medida mais popular na literatura é a discrepância. Nas palavras de Niederreiter [1992] “a discrepância pode ser vista como uma medida quantitativa de desvio em relação à distribuição uniforme”. Apresentamos abaixo as suas formas mais conhecidas. Seja  $P$  um conjunto de  $N$  pontos no cubo unitário  $[0, 1)^d$ . Seja  $\beta$  uma família de subconjuntos de  $[0, 1)^d$  mensuráveis no sentido de Lebesgue. De modo geral a discrepância de  $P$  é obtida através de:

$$D_N(\beta; P) = \sup_{B \in \beta} \left| \frac{\#(B \cap P)}{N} - \lambda_d(B) \right|$$

onde  $\lambda(B)$  é a medida de Lebesgue do conjunto  $B$ . A escolha de  $\beta$  determina o tipo de discrepância.

Apresentamos as duas definições mais utilizadas de discrepância.

- A discrepância estrela  $D_N^* = D_N(\beta^*; P)$  onde  $\beta^*$  é a família de todos os subintervalos de  $[0, 1)^d$  do tipo  $\prod_{i=1}^d [0, u_i)$ .
- A discrepância extrema  $D_N(P) = D_N(\beta; P)$  onde  $\beta$  é a família de todos os subintervalos do tipo  $\prod_{i=1}^d [u_i, v_i)$ .

Seja  $D_N(w)$  a discrepância da seqüência  $\{w_n\}$ . As duas definições de discrepância permitem caracterizar seqüências distribuídas de maneira uniforme no sentido definido anteriormente. De acordo com Niederreiter & Kuipers [1974] as três propriedades que seguem são equivalentes:

1.  $\{w_n\}$  é distribuída de maneira uniforme módulo 1.
2.  $\lim_{N \rightarrow \infty} D_N(w) = 0$
3.  $\lim_{N \rightarrow \infty} D_N^*(w) = 0$

### 2.2.3 Algumas propriedades

Pelo último resultado apresentado, as seqüências uniforme módulo 1 são também chamadas de seqüências de baixa discrepância. Utilizaremos as duas denominações ao longo deste trabalho. A estas soma-se mais uma, seqüência quase aleatória, oriunda de simulação quase Monte Carlo. De fato, como mencionamos na introdução, esta pode ser descrita como uma simulação Monte Carlo na qual utiliza-se seqüências de baixa discrepância no lugar das variáveis aleatórias. Vejamos como a simulação quase Monte Carlo se comporta na resolução da mesma integral utilizada na introdução para apresentar o método de Monte Carlo. A metodologia de cálculo é idêntica e portanto nos concentramos unicamente no erro de aproximação da integral.

A discrepância nos fornece um resultado muito poderoso, o limite de erro de integração. No caso unidimensional temos a desigualdade de Koksma. Se  $f$  é de variação limitada  $V(f)$  no intervalo  $[0, 1]$ , então para qualquer conjunto de pontos  $P \in [0, 1]$  temos:

$$\left| \frac{1}{N} \sum_{n=1}^N f(x_n) - \int_0^1 f(u) du \right| \leq V(f) D_N^*(P)$$

No caso n-dimensional temos a desigualdade de Koksma-Hlawka; se  $f$  é de variação limitada  $V(f)$  no sentido de Hardy e Krause no intervalo  $[0, 1]^d$ , então para qualquer conjunto de pontos  $P \in [0, 1]^d$  temos:

$$\left| \frac{1}{N} \sum_{n=1}^N f(x_n) - \int_{[0,1]^d} f(u) du \right| \leq V(f) D_N^*(P)$$

A discrepância não é só útil na análise de seqüências uniformemente distribuídas e na obtenção de limites de erro de integração. Ela também é usada como medida de calibração de seqüências pseudo-aleatórias. Os resultados acima são um incentivo forte à busca de construções de baixa discrepância. Antes de prosseguirmos, insistimos no fato que as expressões acima constituem um limite para o erro. No caso da simulação Monte Carlo as ordens de grandeza para este são de tipo probabilista, ou seja podem eventualmente ser excedidas. Por isso, apesar destas fórmulas fornecerem alguma sensibilidade sobre possíveis comparações entre os dois métodos não se pode efetuar uma justaposição direta das duas ordens.

### 2.2.4 Redes (t,m,d) e seqüências (t,d)

Introduziremos agora a noção de rede (t,m,d) e de seqüência (t,d). Um paralelepípedo na base  $b$ ,  $p_b$ , também conhecido como intervalo elementar na base  $b$ , ( $b \geq 2$ ), é um subconjunto do

cubo unitário  $[0, 1)^d$  da forma,

$$p_b = \prod_{j=1}^d \left[ \frac{a_j}{b^{k_j}}, \frac{a_j + 1}{b^{k_j}} \right) \quad (2.5)$$

onde  $k_j \in \{0, 1, \dots\}$  e  $a_j \in \{0, 1, \dots, b^{k_j} - 1\}$ . O volume de um intervalo elementar é dado por,

$$V(p_b) = \frac{1}{b^{k_1 + \dots + k_d}}$$

Sejam os inteiros  $t$  e  $m$  tais que  $0 \leq t \leq m$ . Uma rede  $(t, m, d)$  na base  $b$  é um conjunto de  $b^m$  pontos no cubo unitário tal que cada intervalo elementar de volume  $b^{t-m}$  possui exatamente  $b^t$  pontos. Vamos escolher a base 2 - a base das seqüências de Sobol - para ilustrar este conceito. Se escolhermos  $m = 3$  e  $t = 1$ , no plano unitário  $[0, 1)^2$ , uma rede  $(1, 3, 2)$  comportará exatamente  $2^1$  pontos em cada intervalo elementar de volume  $2^{1-3} = 0,25$ . Separando o plano unitário em quatro quadrantes, cada um deles conterá exatamente 2 pontos da seqüência.

Uma seqüência é  $(t, d)$  no cubo unitário se para todo  $m > t$  cada segmento do tipo  $\{x_i : j b^m < i \leq (j+1) b^m\}$ ,  $j = 0, 1, \dots$  é uma rede  $(t, m, d)$  na base  $b$ . Chamaremos os segmentos na base  $b = 2$  de segmentos binários. Pelas definições acima é obviamente preferível obter-se o menor  $t$  possível. Ao mesmo tempo é desejável também que a base seja pequena pois as propriedades são então válidas para conjuntos menores de números ( $b^m$  pontos). Isso explica em parte o sucesso das seqüências de Sobol na prática.

## 2.3 As construções clássicas

Apresentamos agora as principais seqüências utilizadas em simulação quase Monte Carlo. Dentre elas, as mais usadas são sem dúvida as de Halton, Sobol e Faure.

### 2.3.1 Irracionais

As construções que fazem uso de irracionais são as mais simples de serem geradas. Por exemplo a seqüência  $\{n\theta\}$  com  $\theta$  irracional é uniforme módulo um. Podemos generalizar este último resultado. Seja o polinômio:

$$p(x) = \alpha_m x^m + \alpha_{m-1} x^{m-1} + \dots + \alpha_0$$

Então a seqüência  $\{p(n)\}$  é uniforme módulo 1 se pelo menos um dos  $\alpha_i$  é irracional.



Na prática essas seqüências não são muito utilizadas. Um exemplo de uso em finanças é dado por Birge, [1996], um dos primeiros a aplicar o método de quase Monte Carlo em finanças.

### 2.3.2 Seqüências de Van der Corput

A seqüência de Van der Corput é o fundamento para várias outras seqüências e por isso a apresentamos.

O procedimento a seguir para geração dos números começa com a escolha de uma base,  $b$ , estritamente maior que 1. Todo número inteiro positivo  $k$  possui uma representação nesta base,

$$k = \sum_{j=0}^{\infty} a_j(k) b^j \quad (2.6)$$

onde os coeficientes  $a_j$  pertencem a  $\{0, 1, \dots, b-1\}$ . A função radical inversa efetua um mapeamento dos inteiros no intervalo  $[0, 1)$  usando os coeficientes para representar frações na base  $b$ ,

$$\phi_b(k) = \sum_{j=0}^{\infty} \frac{a_j(k)}{b^{j+1}} \quad (2.7)$$

A seqüência de Van der Corput é dada por  $\{\phi_b(n)\}$ . A ordem da discrepância estrela dos  $n$  primeiros pontos de Van der Corput é  $O(\log n/n)$ .

### 2.3.3 As seqüências de Hammersley e Halton

Uma simples extensão das seqüências de Van der Corput nos leva às construções de Hammersley e Halton. Mantendo a mesma notação utilizada para as seqüências de Van der Corput um termo de ordem  $n$  de uma seqüência de Halton de dimensão  $d$  é dado por,

$$x_k = (\phi_{b_1}(k), \phi_{b_2}(k), \dots, \phi_{b_d}(k)) \quad (2.8)$$

Requer-se que as bases sejam coprimos. Bases menores são preferíveis. É comum utilizar os  $d$  primeiros primos. A seqüência de Hammersley é obtida substituindo os termos da primeira dimensão por  $\{k/n\}$ , o que implica em escolher de antemão o tamanho da simulação. Ela é dada por,

$$x_k = (k/n, \phi_{b_1}(k), \dots, \phi_{b_{d-1}}(k)) \quad (2.9)$$

Obtém-se melhor uniformidade desta maneira.

A discrepância estrela dos primeiros  $n$  pontos de Halton de dimensão  $d$  com bases  $b_1, \dots, b_d$  é dada por,

$$D^*(x_0, \dots, x_{n-1}) \leq C_d(b_1, \dots, b_d) \frac{(\log n)^d}{n} + O\left(\frac{(\log n)^{d-1}}{n}\right) \quad (2.10)$$

No caso da seqüência de Hammersley temos,

$$D^*(x_0, \dots, x_{n-1}) \leq C_{d-1}(b_1, \dots, b_{d-1}) \frac{(\log n)^{d-1}}{n} + O\left(\frac{(\log n)^{d-2}}{n}\right) \quad (2.11)$$

Ao se tomar os primeiros  $d$  primos minimiza-se o termo  $C_d$ . Porém a escolha de coprimos gera também seqüências de baixa discrepância. Como para todas elas, a discrepância aumenta com a dimensão. Isso ocorre neste caso porque  $C_d$  aumenta exponencialmente com a dimensão.

### 2.3.4 Seqüências de Faure

Os problemas associados a dimensões altas para as seqüências de Halton e Hammersley são em parte causadas pela necessidade de trabalhar-se com bases muito grandes. Faure [1982] sugeriu um método que utiliza a mesma base para todas as dimensões. As coordenadas da seqüência de Faure são permutações de uma única seqüência de Van der Corput.

O procedimento começa com a escolha da base. Esta necessita ser maior que a dimensão da seqüência. Como o ideal é mantê-la menor possível a escolha consiste geralmente no menor primo superior à dimensão da seqüência. Todo inteiro é dado por,

$$k = \sum_{l=0}^{\infty} a_l(k) b^l \quad (2.12)$$

A  $i$ -ésima coordenada do  $k$ -ésimo termo de uma seqüência de Faure é dada por,

$$x_k^i = \sum_{j=1}^{\infty} \frac{y_j^{(i)}(k)}{b^j} \quad (2.13)$$

onde,

$$y_j^{(i)}(k) = \sum_{l=0}^{\infty} \binom{l}{j-1} (i-1)^{l-j+1} a_l(k) \mod b \quad (2.14)$$

Existem maneiras muito econômicas para implementar a construção de Faure. Como a seqüência não será usada não abordamos o assunto neste trabalho. Porém, uma boa referência para implementação é o artigo de Galanti & Jung [1997].

Faure [1982] mostra que a discrepância de uma seqüência de dimensão  $d$  é,

$$D^*(x_1, \dots, x_d) = F_d \frac{(\log n)^d}{n} + O\left(\frac{(\log n)^{d-1}}{n}\right) \quad (2.15)$$

Neste caso  $F_d$  tende a zero quando  $d$  tende a infinito. Isso por se já denota uma nítida melhora em relação às seqüências de Halton e Hammersley.

### 2.3.5 Seqüências de Sobol

Os números de Sobol [1967] são também permutações de seqüências de Van der Corput. A diferença em relação aos números de Faure é que a base das seqüências de Van der Corput é sempre 2. Isso facilita enormemente o aspecto computacional da construção. Deixamos a construção da seqüência de Sobol para o próximo capítulo pois é importante tê-la em mente para entender o algoritmo proposto nesta tese. Dentre as seqüências apresentadas ela é a mais utilizada. Os números de Sobol formam uma seqüência  $(t,d)$  na base 2.

### 2.3.6 Construções recentes

A pesquisa atual merece menção pois afinal os geradores apresentados até agora tem 30 a 40 anos. Construções recentes não teriam demonstrado resultados superiores ? Um campo de pesquisa que aparenta ser muito promissor, pelo menos no que tange às propriedades assintóticas das seqüências quase aleatórias é a geometria algébrica. Nesta linha, duas construções devidas a Niederreiter e Xing [1995, 1996] apresentam propriedades superiores às seqüências de Sobol. Os autores mostram que para suas seqüências  $(t, d)$ , o crescimento do parâmetro  $t$  conforme cresce a dimensão é muito mais lento do que para as seqüências de Sobol. Por exemplo, para  $d = 20$  o menor  $t$  que uma seqüência de Sobol pode atingir é 71 enquanto que a última construção de Niederreiter e Xing [1996] é uma seqüência  $(21, 20)$  na base 2.

Infelizmente as aplicações destes novos procedimentos não geram resultados animadores. Jäckel [2002] relata que para conseguir estimativas de preços de opções superiores às obtidas pelo método de Sobol são necessários tamanhos que superam o admissível na prática. Atingir as propriedades assintóticas das novas construções não seria factível dentro de um ambiente normal de simulação. Estes resultados confirmam as observações de Barbe [2001] que mostra através de simulações para cálculo de Value at Risk, e precificações de opções asiáticas e binárias, que as seqüências de Sobol são superiores à seqüência de Niederreiter e Xing [1996].

Não incluímos aqui uma discussão sobre métodos considerados híbridos como por exemplo o embaralhamento de seqüências quase aleatórias ou misturas de números quase aleatórios com números pseudo-aleatórios, pelos motivos que seguem. Em primeiro lugar os resultados obtidos

através destas metodologias não são superiores aos obtidos com ferramentas tradicionais, e, em segunda instância, porque não podemos considerá-las como construções primitivas. Na verdade todos estes métodos dependem de outras construções, geralmente os números de Faure ou de Sobol.

## 2.4 As seqüências de baixa discrepância na prática

### 2.4.1 A difusão da simulação quase Monte Carlo

Conforme salientado na introdução, a adoção da simulação quase Monte Carlo tem sido muito tímida por parte dos praticantes do mercado financeiro. Nos livros de larga audiência como Hull [2000] o método é mencionado mas não há nenhuma indicação sobre implementação ou sobre situações nas quais é preferível ou não utilizá-lo. Isso é claramente atribuível aos problemas que a simulação quase Monte Carlo apresenta em dimensões altas (de forma subjetiva  $d = 50$  pode ser considerado um patamar a partir do qual as seqüências de baixa discrepância podem apresentar sérios problemas). O fracasso do método em tratar sistematicamente tais casos gera uma sensação de insegurança que tem esfriado a reação dos praticantes a seu respeito.

Pelo que vimos até agora não é de todo surpreendente que muita pesquisa tenha sido consagrada a eliminar esta restrição. Várias soluções têm sido adiantadas mas nenhuma delas parece ter sido adotada por uma ampla gama de usuários. Isto pode ser atribuído ao estado embrionário no qual o método de quase Monte Carlo se encontra. Porém, acreditamos que outros dois fatores contribuem para este quadro. Um deles é a abordagem caso a caso que tem sido adotada por vários autores, onde mostra-se que uma variante de um método determinista funciona para um instrumento financeiro específico, reduzindo assim o seu escopo de aplicações. O outro fator tem a ver com a implementação. Em muitos casos a complexidade do método proposto reduz o seu apelo. O ideal seria portanto encontrar uma técnica que não seja sensível à dimensão do problema, de uso geral, e fácil de ser implementada. Ao buscarmos uma alternativa às demais técnicas propostas, tentamos sempre manter esses princípios como linha diretriz.

### 2.4.2 Resultados paradoxais

O uso dos métodos de baixa discrepância em finanças é relativamente recente. As grandes contribuições pioneiras de Paskov e Traub [1995], Galanti e Jung [1997], Joy, Boyle e Tan

[1996] foram publicadas na segunda metade dos anos noventa. Todavia, em outras disciplinas, notadamente em física, os pesquisadores possuem há mais tempo boas noções sobre as vantagens (e desvantagens) da simulação quase Monte Carlo. Exemplos são os trabalhos de Lécot [1991] e Morokoff e Caflisch [1993]. É interessante notar que os autores concluem que,

“error reduction for quasi-Monte Carlo is limited as the spatial dimension increases”

Isto é aparente em Caflisch [1994, 1995]. Ao contrário, os resultados obtidos em finanças eram muito mais encorajadores. De fato, Paskov e Traub [1995] precificam uma CMO de dimensão 360 e mostram que a simulação quase Monte Carlo possui desempenho superior ao método tradicional mesmo com o uso de técnicas de redução de variância. A sequência quase aleatória de Sobol é gerada pelo programa proprietário FINDER. Ao mesmo tempo, em Galanti e Jung [1997] as sequências quase aleatórias deixam de ser eficazes para dimensões médias mas apresentam um bom desempenho quando a dimensão passa de média para alta. Às vezes a sequência de Faure apresenta melhor desempenho, outras vezes a sequência de Sobol leva a melhor. Joy, Boyle e Tan [1996] não se aventuram em territórios nos quais a dimensão é alta. O problema de maior dimensão é a precificação de uma opção asiática de 52 passos no tempo. Eles relatam que a sequência de Faure apresentou um desempenho superior aos demais métodos. Uma pesquisa posterior de Boyle [1997] aumenta a dimensão até 100. As sequências de Faure e Sobol são testadas. Para estas dimensões as sequências de Faure deixaram de ser operacionais. No entanto as sequências de Sobol geram estimativas muito próximas do valor teórico. Mesmo assim, a aproximação diminui de qualidade conforme a dimensão cresce.

Esses resultados um tanto conflitantes podem ser explicados pelo conceito de dimensão efetiva do problema, conforme demonstrado por Caflisch, Morokoff e Owen [1997]. Nesta pesquisa tenta-se explicar a razão dos resultados surpreendentes alcançados em finanças. Eles concluem que a dimensão que importa para, por exemplo, a CMO apresentada por Paskov e Traub [1995] é muito inferior à 360, dimensão nominal do problema. Eles enfatizam que,

“quasi-Monte Carlo methods provide significant improvements in accuracy and computational speed for problems of small to moderate dimension”.

Enfim, esta aparente confusão não ajuda a popularizar o método, mesmo para problemas de baixa dimensão, para os quais o desempenho superior das sequências de baixa discrepância é comprovado. Baseado na literatura, o agente interessado em utilizar o método de quase Monte Carlo pergunta-se então,

- por que certas seqüências possuem melhor performance ?
- por que certas seqüências deixam de ser operacionais para um conjunto intermediário de dimensões mas voltam a apresentar bons resultados para dimensões altas ?
- por que motivo pode-se utilizar uma simulação quase Monte Carlo para uma CMO de dimensão 360 e não para uma opção asiática de dimensão 100 ?

. A timidez na adoção da simulação quase Monte Carlo é compreensível. Concluimos que a dimensão nominal do problema deve ceder o lugar ao conceito de dimensão efetiva para que se possa julgar com mais precisão a eficiência das construções propostas. Feito isso é seguro dizer que as construções utilizadas em finanças até agora não apresentam desempenho seguro o suficiente em dimensões efetivas altas (acima de 50) para que possam ser adotadas no meio financeiro.

Um outro ponto que pode indispor um praticante a utilizar o método de quase Monte Carlo é a ausência de intervalos de confiança para a estimativa. Um linha de pesquisa segue os passos de Cranley and Paterson [1976] e tenta obter regiões de confiança para as estimativas obtidas via quase Monte Carlo. Vejamos agora o que tem sido proposto para estender o método para altas dimensões efetivas.

### 2.4.3 Soluções propostas

Pesquisadores tentam estender o escopo da simulação quase Monte Carlo de diversas maneiras. Como já mencionado podemos classificar as abordagens sob dois grandes conjuntos. Uma abordagem tenta gerar seqüências com propriedades mais atraentes, sejam elas novas ou simples transformações de seqüências já existentes. A outra tenta modificar o problema a ser resolvido para que possa ser tratado com os meios existentes.

O procedimento mais nobre é exemplificado pelos trabalhos de Niederreiter e Xing [1995, 1996]. Conforme já salientado os primeiros resultados aparentam ainda apontar para a superioridade dos números de Sobol. Outro procedimento consiste em embaralhar seqüências deterministas. Numa tentativa de determinar a variância das simulações quase Monte Carlo, Owen [1994, 1997] mostra que seu método de embaralhamento mantém as propriedades de baixa discrepância das seqüências deterministas e ajuda a tornar o método mais preciso em altas dimensões. Uma adaptação do seu método - de modo a simplificar a sua implementação -

foi testada por Tan e Boyle [2000] no âmbito financeiro. Eles precificam uma opção asiática de dimensão nominal 50, 250 e 365. As seqüências embaralhadas mostram desempenho melhor quando comparadas com o método tradicional e com o método de quase Monte Carlo puro.

Por outro lado, pode-se transformar o problema tratado via simulação Monte Carlo para diminuir sua dimensão efetiva. De fato, quando o problema consiste em precificar opções cuja trajetória precisa ser simulada, é possível reduzir a dimensão desta através de uma discretização cuidadosamente escolhida. Uma exposição muito clara deste procedimento pode ser encontrada em Moskowitz e Caffisch [1996]. Os autores utilizam uma ponte browniana para avaliar integrais de linha de Feynman Kac. Os resultados são muito promissores mas nos testes relatados a dimensão nominal não passa de 32.

A primeira abordagem nos parece muito mais interessante pois busca maior generalidade através de um procedimento que pode ser aplicado a todos os problemas. O grupo de simulações para as quais é possível a aplicação de uma técnica de redução de dimensão não é muito amplo. Um exemplo deste conjunto é o cálculo de Value at Risk. Este problema, por acaso, foi testada por Papageorgiou e Paskov [1999] porém unicamente com 34 fatores de risco.





## Capítulo 3

# Construção da seqüência de Sobol

### 3.1 A construção clássica

#### 3.1.1 Conceitos preliminares e notação

Chamamos de cubo unitário o espaço  $[0, 1]^d$ . Uma seqüência  $\{S_n\}$  no cubo unitário é uma função  $S : \mathbb{N} \rightarrow [0, 1]^d$ . A cada inteiro positivo  $n$  é associado um ponto  $S_n = (s_{n_1}, s_{n_2}, \dots, s_{n_d})$ . Cada uma das  $d$  coordenadas dos pontos da seqüência forma também uma seqüência notada  $\{s_{n_j}\}$ ,  $1 \leq j \leq d$ .  $s_{n_j}$  representa a  $j$ -ésima coordenada do termo  $S_n$ .

Faremos uso da representação binária de inteiros não negativos e de frações binárias no cubo unitário. No caso de um  $n \in \mathbb{Z}^+$  sua representação binária será dada por uma seqüência de bits,  $\{b_{n_i}\}$  na qual o  $n$ -ésimo termo representa o coeficiente associado à potência  $2^{i-1}$ . Assim o número 13, é dado pela seqüência 1, 0, 1, 1, 0, 0, ... pois,

$$13 = 1 \times 2^0 + 0 \times 2^1 + 1 \times 2^2 + 1 \times 2^3 + 0 \times 2^4 + \dots$$

Vale notar que isso não corresponde à representação habitual. Atuamos aqui como se houvesse um espelho à direita da notação habitual, 1101. Naturalmente,  $b_{n_i} = 0$  para todo  $i > [\log_2 n] + 1$  onde  $[x]$  denota a parte inteira de  $x$ .

Uma fração binária  $x$  no cubo unitário é um número do tipo,

$$x = \frac{n}{2^k} \tag{3.1}$$

onde  $n$  é um inteiro positivo ímpar inferior a  $2^k$ . A sua representação binária é também dada por uma seqüência de bits  $\{b_{x_i}\}$  que representa os coeficientes das potências negativas de 2 à

direita do ponto decimal. Por exemplo,  $x = 9/2^{-4}$  é representado por  $1, 0, 0, 0, 1, 0, 0, \dots$  pois,

$$\frac{9}{16} = 1 \times 2^{-1} + 0 \times 2^{-2} + 0 \times 2^{-3} + 1 \times 2^{-4} + 0 \times 2^{-5} + \dots$$

Neste caso  $b_{x_i} = 0$  para todo  $i > k + 1$ . Vemos portanto que  $b_{x_i} = b_{n_i}$  com  $n = x2^k$ , o que explica a escolha da notação pouco usual para a representação binária dos naturais.

O símbolo  $\oplus$  representa a operação “bit by bit exclusive or” de  $\{0, 1\} \times \{0, 1\} \rightarrow \{0, 1\}$  de acordo com as regras,  $0 \oplus 0 = 0$ ,  $1 \oplus 1 = 0$ ,  $0 \oplus 1 = 1$  e  $1 \oplus 0 = 1$ .  $n \oplus m$  é chamado de XOR de  $n$  e  $m$ . Ao utilizarmos a notação  $n \oplus m$  para  $n$  e  $m$  inteiros positivos ou frações binárias estaremos efetuando o XOR dos termos de mesma ordem das seqüências que caracterizam a representação binária de  $n$  e  $m$ . Por exemplo  $5 \oplus 7 = 2$  pois efetuamos o XOR dos termos de mesma ordem das seqüências  $1, 0, 1, 0, \dots$  e  $1, 1, 1, 0, \dots$  obtendo assim  $0, 1, 0, 0, \dots$  o que equivale ao inteiro 2.

O corpo  $\mathbb{Z}_2$  corresponde ao conjunto  $\{0, 1\}$ , munido da adição módulo 2 (as regras são idênticas ao XOR de dois bits) e da multiplicação habitual. Estamos interessados nos polinômios irredutíveis de  $\mathbb{Z}_2$ . Seja  $p(x)$  um polinômio. Este é irredutível (ou primo) se não existirem polinômios  $g(x)$  e  $h(x)$  de grau estritamente positivo tais que  $g(x)h(x) = p(x)$ . Se  $p(x)$  for de grau  $g$  ele será apresentado sob a forma,

$$p(x) = x^g + a_1 x^{g-1} + a_2 x^{g-2} + \dots + a_{g-1} x + 1 \quad (3.2)$$

Por exemplo,  $x^3 + x^2 + 1$  é um dos polinômios irredutíveis de grau 3. Neste caso,  $a_1 = 1$ ,  $a_2 = 0$ .

### 3.1.2 Geração das seqüências de Sobol

Mantendo-se a mesma notação, a regra de formação de uma seqüência de Sobol  $\{S_n\}$  no cubo unitário de dimensão  $d$ ,  $[0, 1]^d$ , onde  $S_n = (s_{n_1}, \dots, s_{n_d})$  é dada por,

$$s_{n_j} = b_{n_1} v_{j_1} \oplus b_{n_2} v_{j_2} \oplus b_{n_3} v_{j_3} \dots \quad (3.3)$$

com  $1 \leq j \leq d$ . Introduzimos aqui uma seqüência  $\{v_{j_i}\}$  de frações binárias de denominador  $2^{-i}$  que chamaremos de números direcionais. Cada coordenada  $j$  de  $\{S_n\}$  depende de uma seqüência diferente de números direcionais. A construção de uma seqüência de Sobol requer

portanto a geração prévia das seqüências de números direcionais, o que veremos a seguir

Para cada  $j$ , escolhe-se um polinômio irreduzível no  $\mathbb{Z}_2$ . Imaginemos que para gerar a coordenada  $j$  tenha-se escolhido um polinômio de grau  $q$ . A função destes polinômios consiste em fornecer os coeficientes da seguinte fórmula recursiva,

$$v_{j_i} = a_{j_1} v_{j_{i-1}} \oplus a_{j_2} v_{j_{i-2}} \oplus \dots \oplus a_{j_{q-1}} v_{j_{i-q+1}} \oplus v_{j_{i-q}} \oplus \frac{v_{j_{i-q}}}{2^q} \quad (3.4)$$

Como vimos anteriormente todo número fracionário binário é dado pela equação (3.1) e portanto podemos utilizar uma forma mais intuitiva para a recursão acima,

$$m_{j_i} = 2a_{j_1} m_{j_{i-1}} \oplus 2^2 a_{j_2} m_{j_{i-2}} \oplus \dots \oplus 2^{q-1} a_{j_{q-1}} m_{j_{i-q+1}} \oplus 2^q m_{j_{i-q}} \oplus m_{j_{i-q}} \quad (3.5)$$

Como podemos observar é necessário escolher os  $q$  primeiros números direcionais para que seja possível aplicar a fórmula recursiva pela primeira vez. De acordo com o artigo de Sobol [1967], essa escolha deve simplesmente seguir duas restrições:  $m_{j_i}$  deve ser ímpar e estritamente inferior a  $2^i$ . Vamos agora ver o funcionamento dos procedimentos descritos acima através de um exemplo. Usamos o termo seqüências de Sobol durante a exposição acima porque, como foi mostrado, cada escolha de polinômios e de números direcionais arbitrários gera uma seqüência diferente. Vê-se claramente que os números direcionais arbitrários podem influenciar a distribuição dos pontos de Sobol.

### 3.1.3 Exemplo

Vamos construir uma seqüência de Sobol,  $\{S_n\}$  no  $[0, 1]^2$ . Para isso temos que escolher dois polinômios irreduzíveis no  $\mathbb{Z}_2$ . Por exemplo,

$$p_1(x) = x^3 + x + 1 \quad (3.6)$$

$$p_2(x) = x^3 + x^2 + 1 \quad (3.7)$$

Ambos os polinômios são de grau 3. Para  $j = 1$ , temos  $a_{1_1} = 0$ ,  $a_{1_2} = 1$  e para  $j = 2$ , temos  $a_{2_1} = 1$ ,  $a_{2_2} = 0$ . É comum representar os coeficientes como números binários ou números equivalentes na base 10. Neste caso temos os polinômios 1 e 2 na base dez, 01 e 10 respectivamente na base 2. Obtemos portanto, as seguintes relações recursivas,

$$m_{1_i} = 4m_{1_{i-2}} \oplus 8m_{1_{i-3}} \oplus m_{1_{i-3}} \quad (3.8)$$

$$m_{2_i} = 2m_{1_{i-1}} \oplus 8m_{1_{i-3}} \oplus m_{1_{i-3}} \quad (3.9)$$

Temos que escolher os três primeiros termos para cada uma das seqüências de números direcionais. Vamos estabelecer que  $m_{1_1} = 1$ ,  $m_{1_2} = 3$ ,  $m_{1_3} = 7$  e  $m_{2_1} = 1$ ,  $m_{2_2} = 3$ ,  $m_{2_3} = 5$ . Calculemos agora os próximos 2 valores para ambas as relações.

$$\begin{aligned} m_{1_4} &= 12 \oplus 8 \oplus 1 \\ &= 0011 \oplus 0001 \oplus 1000 \\ &= 1010 \\ &= 5 \end{aligned}$$

$$\begin{aligned} m_{1_5} &= 28 \oplus 24 \oplus 3 \\ &= 00111 \oplus 00011 \oplus 11000 \\ &= 11100 \\ &= 7 \end{aligned}$$

$$\begin{aligned} m_{2_4} &= 10 \oplus 8 \oplus 1 \\ &= 0101 \oplus 0001 \oplus 1000 \\ &= 1100 \\ &= 3 \end{aligned}$$

$$\begin{aligned} m_{2_5} &= 6 \oplus 24 \oplus 3 \\ &= 01100 \oplus 00011 \oplus 11000 \\ &= 10111 \\ &= 29 \end{aligned}$$

Somos agora capazes de gerar os 32 primeiros números de  $\{S_i\}$ . Vamos calcular dois exemplos, o termo de ordem 5 e o termo de ordem 11. Aplicamos a regra descrita na equação (18).

$$\begin{aligned}
 s_{5_1} &= b_{5_1}v_{1_1} \oplus b_{5_2}v_{1_2} \oplus b_{5_3}v_{1_3} \\
 &= 1 \times 0,100 \oplus 0 \times 0,110 \oplus 1 \times 0,111 \\
 &= 0,011 \\
 &= 0,375
 \end{aligned}$$

$$\begin{aligned}
 s_{5_2} &= b_{5_1}v_{2_1} \oplus b_{5_2}v_{2_2} \oplus b_{5_3}v_{2_3} \\
 &= 1 \times 0,100 \oplus 0 \times 0,110 \oplus 1 \times 0,101 \\
 &= 0,001 \\
 &= 0,125
 \end{aligned}$$

Calculamos agora as coordenadas do termo de ordem 11,

$$\begin{aligned}
 s_{11_1} &= b_{11_1}v_{1_1} \oplus b_{11_2}v_{1_2} \oplus b_{11_3}v_{1_3} \oplus b_{11_4}v_{1_4} \\
 &= 1 \times 0,1000 \oplus 1 \times 0,1100 \oplus 0 \times 0,1110 \oplus 1 \times 0,0101 \\
 &= 0,0001 \\
 &= 0,0625
 \end{aligned}$$

$$\begin{aligned}
 s_{11_2} &= b_{11_1}v_{2_1} \oplus b_{11_2}v_{2_2} \oplus b_{11_3}v_{2_3} \oplus b_{11_4}v_{2_4} \\
 &= 1 \times 0,1000 \oplus 1 \times 0,1100 \oplus 0 \times 0,1010 \oplus 1 \times 0,0011 \\
 &= 0,0111 \\
 &= 0,4375
 \end{aligned}$$

Obtivemos  $S_5 = (0,375; 0,125)$  e  $S_{11} = (0,0625; 0,4375)$ . Para que se possa ter uma boa idéia da aparência de uma seqüência de Sobol mostramos na tabela 3.1 os primeiros termos da seqüência gerada pela nossa escolha de polinômios e números direcionais arbitrários.

n	1,3,7	1,3,5
0	0	0
1	0.5	0.5
2	0.75	0.75
3	0.25	0.25
4	0.875	0.625
5	0.375	0.125
6	0.125	0.375
7	0.625	0.875
8	0.3125	0.1875
9	0.8125	0.6875
10	0.5625	0.9375
11	0.0625	0.4375
12	0.6875	0.5625
13	0.1875	0.0625
14	0.4375	0.3125
15	0.9375	0.8125

Tabela 3.1: Os primeiros 15 números de Sobol

Como pode ser observado, as coordenadas dos 4 primeiros termos da sequência são idênticas. Isso acontece porque os 2 primeiros números direcionais são os únicos que intervêm na formação de  $S_0$  a  $S_3$ . Ao mesmo tempo nota-se que os números da segunda coordenada são permutações da primeira coordenada para  $n$  abrangendo intervalos do tipo  $[2^k, 2^{k+1})$ . Veremos agora a propriedade principal das sequências de Sobol, a maneira uniforme pela qual os pontos preenchem o cubo unitário.

## 3.2 Uniformidade das sequências de Sobol

### 3.2.1 As sequências de Sobol são sequências (t,d) na base 2

Como já mencionamos acima a sequência de Sobol é uma rede (t,d) na base 2. Porém, na prática, somente simulações de dimensão muito baixa podem tirar proveito desta propriedade.

Para se ter uma idéia das dificuldades em aproveitar-se da uniformidade dos pontos conforme a dimensão vai aumentando, Sobol [1967] demonstra que para  $d = 20$  não se pode ter um  $t$  inferior ou igual a 71. Como os parâmetros  $m$  das redes  $(t, m, d)$  implícitas na sequência  $(t, d)$  necessitam ser superiores ou iguais a  $t$  então só se pode atingir o melhor refinamento possível (um ponto por paralelepípedo) a partir de segmentos binários de tamanho  $2^{71}$ . De acordo com Sobol [1967] este piso para  $t$  evolui exponencialmente com a dimensão, pois é dado por,

$$t_{\min} = \sum_{j=1}^d \deg(p_j) - d \quad (3.10)$$

onde  $\deg(p_j)$  corresponde ao grau do polinômio utilizado para a coordenada  $j$ . Como foi demonstrado em vários exercícios aplicados, a performance da simulação quase Monte Carlo deteriora-se muito rapidamente conforme a dimensão aumenta.

### 3.2.2 As propriedades A e B

Num trabalho posterior, Sobol [1976] mostra que alguns números direcionais são preferíveis a outros. Como critério de avaliação ele sugere as propriedades A e B.

Uma sequência de Sobol possui a propriedade A se, após dividirmos o cubo unitário  $[0, 1]^d$  em paralelepípedos na base 2 do tipo

$$R_2 = \prod_{j=1}^d \left[ t \frac{1}{2}, t \frac{1}{2} + \frac{1}{2} \right)$$

com  $t \in \{0, 1\}$ , e seccionarmos a sequência em segmentos binários de tamanho  $2^d$ , então cada paralelepípedo contém exatamente um ponto do segmento. Trata-se de um caso particular da propriedade apresentada na seção anterior onde é necessário garantir a distribuição adequada dos pontos para um conjunto extenso de partições do espaço em paralelepípedos. Agora, toma-se uma única partição. Sobol mostra que a propriedade A é atendida se,

$$\left\| \begin{pmatrix} v_{1,1}^1 & v_{2,1}^1 & \dots & v_{j,1}^1 \\ \vdots & \vdots & \ddots & \vdots \\ v_{1,1}^j & v_{2,1}^j & \dots & v_{j,1}^j \end{pmatrix} \right\| = 1 \pmod{2} \quad (3.11)$$

onde  $v_{l,k}^j$  representa o  $k$ -ésimo bit do  $l$ -ésimo número direcional utilizado para construir a  $j$ -ésima coordenada de um número de Sobol.

Sobol mostra também que se,

$$\left\| \begin{array}{cccc} v_{1,1}^1 & v_{2,1}^1 & \dots & v_{2d,1}^1 \\ v_{1,2}^1 & v_{2,2}^1 & \dots & v_{2d,2}^1 \\ \vdots & \vdots & \ddots & \vdots \\ v_{1,1}^j & v_{2,1}^j & \dots & v_{2d,1}^j \\ v_{1,2}^j & v_{2,2}^j & \dots & v_{2d,2}^j \end{array} \right\| = 1 \pmod{2} \quad (3.12)$$

então os segmentos de comprimento  $2^{2d}$  possuirão um ponto em cada paralelepípedo do tipo,

$$p_2 = \prod_{j=1}^d \left[ t \frac{1}{4}, t \frac{1}{4} + \frac{1}{4} \right)$$

com  $t \in \{0, 1, 2, 3\}$ . Neste caso os números direcionais atendem à propriedade B.

### 3.2.3 Sequências anômalas

É comum o uso de gráficos para mostrar que as seqüências de baixa discrepância preenchem o cubo unitário de forma mais uniforme que os números pseudo-aleatórios. Mostramos na parte de cima da figura 3.1 um caso clássico, uma comparação entre as projeções sobre as coordenadas 2 e 3 para números pseudo-aleatórios gerados com o Matlab e números de Sobol.

Porém este padrão de preenchimento para projeções arbitrárias tende a mostrar configurações anômalas conforme as dimensões tornam-se mais altas. Para ilustrar este caso construímos uma seqüência de dimensão 100 de acordo com o procedimento delineado acima. Os números direcionais são escolhidos de maneira aleatória respeitando-se é claro as restrições impostas pelo modelo. Após a geração dos números calculamos a matriz de correlação associada à seqüência. Conseguimos identificar as projeções anômalas através das correlações mais altas. Tipicamente, quando se utiliza o algoritmo convencional, encontra-se correlações na ordem de 75%. Alguns exemplos de projeções indesejáveis são apresentadas na parte de baixo da figura 3.1.

Observamos que as seqüências de melhor desempenho eram sempre aquelas desprovidas de projeções esdrúxulas. Uma tentativa de gerar pontos que não apresentam estes defeitos, utilizando-se como proxy a ausência de correlação, gerou excelentes resultados apresentados em Silva [2002]. O desempenho impressionante das seqüências obtidas empiricamente motivou



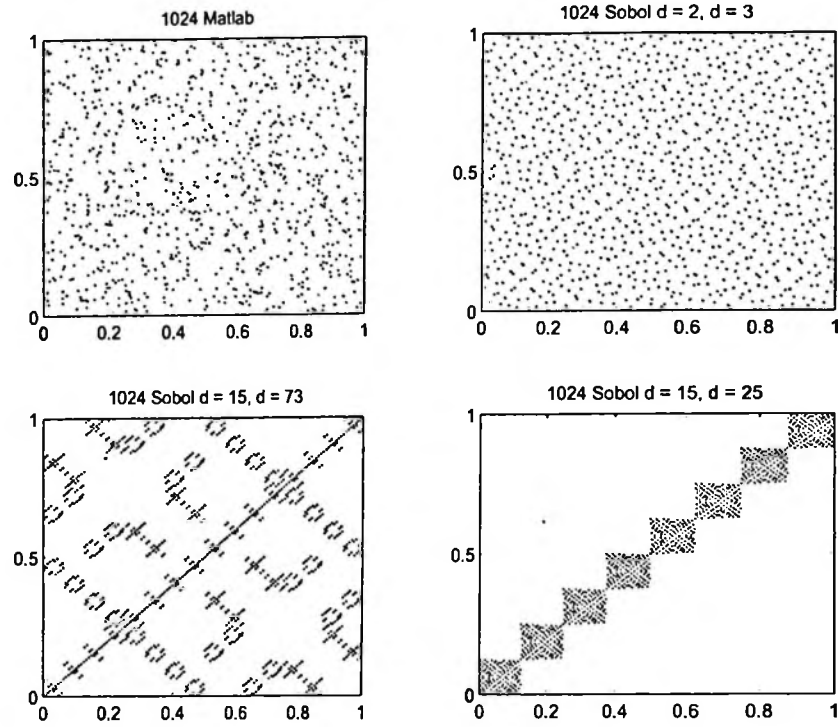


Figura 3.1: Sobol: projeções para construção clássica

a busca por regras claras e de fácil implementação. Apresentamos agora um método que produz seqüências desprovidas dos vieses que acabamos de salientar.

### 3.3 Um novo algoritmo para a escolha dos direcionais

#### 3.3.1 Números direcionais eficientes

O resultado que derivamos a seguir é o ponto central de nossa pesquisa. A sua validade nos permite em seguida desenvolver um algoritmo que gera seqüências de Sobol sem os vieses relatados acima. Segue inicialmente a notação utilizada no resto da seção.

Focamos em pares de coordenadas de  $\{S_n\}$ . Para facilitar a nossa notação trabalharemos unicamente com as coordenadas de índice 1 e 2 da seqüência. Estamos interessados na maneira pela qual as duas coordenadas geram pontos uniformemente espalhados no quadrado unitário  $[0, 1]^2$ . Para isto particionamos o quadrado unitário em  $2^{2t}$ ,  $t \geq 1$ , subquadrados dados pelos produtos de intervalos elementares do tipo,

$$I_{u,w} = \left[ u \frac{1}{2^t}, (u+1) \frac{1}{2^t} \right) \times \left[ w \frac{1}{2^t}, (w+1) \frac{1}{2^t} \right) \quad (3.13)$$

com  $u$  e  $w$  assumindo os valores de 0 a  $2^t - 1$ .

Cada coordenada depende de um conjunto próprio de números direcionais,  $\{v_n^1\}$  e  $\{v_n^2\}$ . Lembramos que  $v_{n,j}^m$  representa o  $j$ -ésimo bit do  $n$ -ésimo número direcional utilizado para construção da  $m$ -ésima coordenada. Introduzimos também a matriz  $V_{tk}$  que têm como elementos os  $t$  primeiros bits dos  $k+1$  primeiros números direcionais utilizados na construção das duas primeiras coordenadas. Se cada coluna representar um direcional, variando de 1 a  $k+1$ , e se cada linha representar um bit, variando de 1 a  $t$  para as duas coordenadas, obtemos<sup>1</sup> então a seguinte representação:

$$V_{tk} = \begin{vmatrix} v_{1,1}^1 & v_{2,1}^1 & \dots & v_{k,1}^1 & v_{k+1,1}^1 \\ v_{1,1}^2 & v_{2,1}^2 & \dots & v_{k,1}^2 & v_{k+1,1}^2 \\ v_{1,2}^1 & v_{2,2}^1 & \dots & v_{k,2}^1 & v_{k+1,2}^1 \\ v_{1,2}^2 & v_{2,2}^2 & \dots & v_{k,2}^2 & v_{k+1,2}^2 \\ \vdots & \vdots & & \vdots & \vdots \\ v_{1,t}^1 & v_{2,t}^1 & \dots & v_{k,t}^1 & v_{k+1,t}^1 \\ v_{1,t}^2 & v_{2,t}^2 & \dots & v_{k,t}^2 & v_{k+1,t}^2 \end{vmatrix} \quad (3.14)$$

Vemos que a matriz  $V_{tk}$  é de dimensão  $2t \times k+1$ . Podemos agora enunciar o seguinte resultado.

**Proposição 3.3.1.** *Sejam  $t$  e  $k$  dois inteiros positivos tais que  $2t \leq k$ . Se a matriz  $V_{tk}$  tiver posto  $2t$  e se  $n$  percorrer o segmento binário  $[2^k, 2^{k+1})$ , então cada intervalo elementar do tipo  $I_{u,w}$  possuirá exatamente  $2^{k-2t}$  pontos.*

**Demonstração 3.3.1.** *Um ponto  $(s_n^1, s_n^2)$ , com  $2^k \leq n < 2^{k+1}$  pertence a  $I_{u,w}$  quando os  $t$  primeiros bits de  $s_n^1$  coincidem com a representação binária de  $u$  ao mesmo tempo em que os  $t$  primeiros bits de  $s_n^2$  coincidem com a representação binária de  $w$ . A equação 3.3 mostra que os  $t$  primeiros bits de  $s_n^1$  são dados pela relação,*

$$\begin{aligned} v_{1,1}^1 b_{n,1} \oplus v_{2,1}^1 b_{n,2} \oplus \dots \oplus v_{k,1}^1 b_{n,k} \oplus v_{k+1,1}^1 b_{n,k+1} &= s_{n,1}^1 \\ v_{1,2}^1 b_{n,1} \oplus v_{2,2}^1 b_{n,2} \oplus \dots \oplus v_{k,2}^1 b_{n,k} \oplus v_{k+1,2}^1 b_{n,k+1} &= s_{n,2}^1 \\ &\vdots = \vdots \\ v_{1,t}^1 b_{n,1} \oplus v_{2,t}^1 b_{n,2} \oplus \dots \oplus v_{k,t}^1 b_{n,k} \oplus v_{k+1,t}^1 b_{n,k+1} &= s_{n,t}^1 \end{aligned}$$

<sup>1</sup>Gostaríamos de chamar a atenção do leitor para o fato que a notação encontra-se invertida em relação à habitual onde um elemento  $a_{ij}$  de uma matriz pertence à  $i$ -ésima linha e  $j$ -ésima coluna.

A mesma relação vale para os  $t$  primeiros bits de  $s_n^2$ ,

$$\begin{aligned} v_{1,1}^2 b_{n,1} \oplus v_{2,1}^2 b_{n,2} \oplus \dots \oplus v_{k,1}^2 b_{n,k} \oplus v_{k+1,1}^2 b_{n,k+1} &= s_{n,1}^2 \\ v_{1,2}^2 b_{n,1} \oplus v_{2,2}^2 b_{n,2} \oplus \dots \oplus v_{k,2}^2 b_{n,k} \oplus v_{k+1,2}^2 b_{n,k+1} &= s_{n,2}^2 \\ &\vdots \\ v_{1,t}^2 b_{n,1} \oplus v_{2,t}^2 b_{n,2} \oplus \dots \oplus v_{k,t}^2 b_{n,k} \oplus v_{k+1,t}^2 b_{n,k+1} &= s_{n,t}^2 \end{aligned}$$

Se “empilharmos” estes dois sistemas de equações reconhecemos a matriz  $V_{tk}$  como a matriz formada pelos coeficientes de cada linha. Como cada subquadrado de  $[0, 1]^2$  é caracterizado por um elemento do conjunto  $\{0, 1\}^{2t}$  - correspondente às possíveis combinações de  $u$  e  $w$  - para que cada um deles contenha pelo menos um ponto é suficiente que a matriz  $V_{tk}$  tenha posto  $2t$ . Reordenamos as  $k$  primeiras colunas de  $V_{tk}$  de tal modo que as  $2t$  primeiras colunas gerem  $\{0, 1\}^{2t}$ . A decomposição binária de  $n$  fornece os coeficientes das combinações lineares das colunas de  $V_{tk}$ . Separemos estes coeficientes nos primeiros  $2t$  e nos  $k + 1 - 2t$  restantes.  $n$  percorre o segmento binário  $[2^k, 2^{k+1})$ , o que implica que a  $k + 1$ -ésima coluna de  $V_{tk}$  é composta por 1. Tomamos a classe de equivalência entre os pontos dada pelos subquadrados aos quais eles pertencem. Cada combinação linear das primeiras  $2t$  colunas equivale a um subquadrado do plano unitário. Temos  $2^{k-2t}$  combinações lineares das  $k - 2t$  próximas colunas. Isso quer dizer que cada subquadrado gerado pelas  $2t$  primeiras colunas é mapeado  $2^{k-2t}$  vezes em outro subquadrado (nem sempre diferentes). Como este mapeamento é injetor, cada subquadrado recebe efetivamente  $2^{k-2t}$  pontos da sequência.

### 3.3.2 Exemplo

A melhor maneira de entender a propriedade enunciada acima é através de um exemplo. Vamos mostrar um caso em que as condições necessárias são atendidas e um caso na qual elas falham. Imagine que queremos gerar uma sequência bidimensional. Para facilitar a nossa tarefa escolhemos um polinômio de grau 10 de modo a podermos escolher os números direcionais. O critério de refinamento da sequência de Sobol é  $t = 2$ . Isso quer dizer que, se dividirmos o plano unitário em oito subquadrados dados pelos intervalos elementares,

$$I_{u,w} = \left[ u \frac{1}{4}, (u+1) \frac{1}{4} \right) \times \left[ w \frac{1}{4}, (w+1) \frac{1}{4} \right) \quad (3.15)$$

com  $u$  e  $w$  assumindo os valores de 0 a 3 então cada um deles deve conter o mesmo número de pontos. Vejamos primeiro a correspondência entre cada subquadrado e os inteiros de 0 a  $2^{2t}$ .

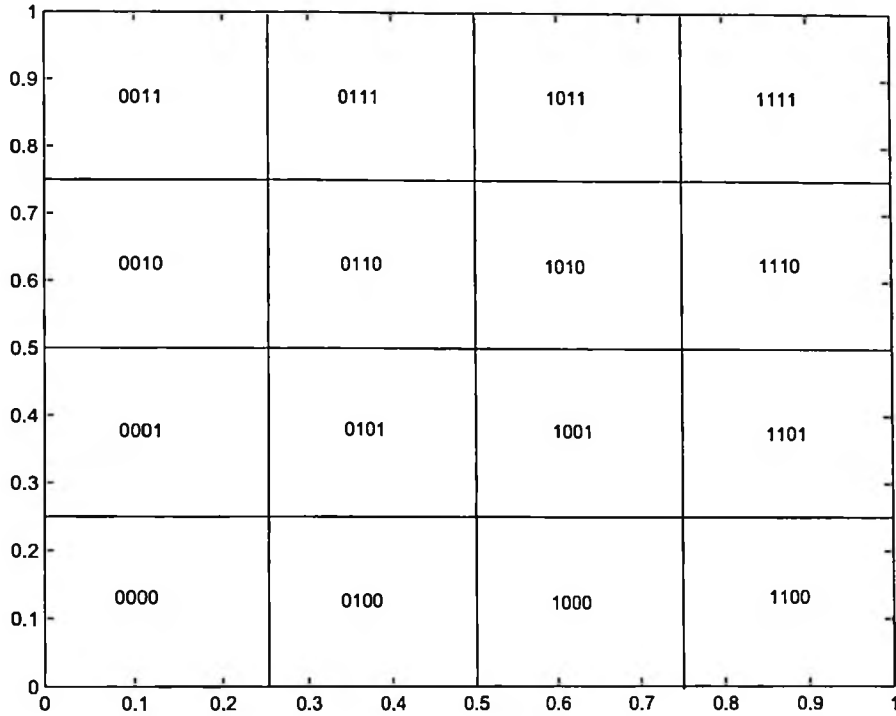


Figura 3.2: Mapeamento dos subquadrados quando  $t = 2$

Para isto basta tomar a concatenação da representação binária de  $u$  e  $w$ . Por exemplo quando  $u$  é igual a 1 e  $w$  é igual a 2 temos,

$$I_{1,2} = \left[ \frac{1}{4}, \frac{1}{2} \right) \times \left[ \frac{1}{2}, \frac{3}{4} \right)$$

equivalente ao inteiro 6, dado por 01 e 10, 0110. A figura 3.2 mostra a idéia de forma mais clara.

Vamos agora escolher  $k = 6$  ou seja a sequência a ser estudada é de tamanho 64. Neste caso gostaríamos que cada subquadrado tivesse  $2^2$  pontos. Para facilitar a nossa tarefa escolhemos dois polinômios de grau 5, representados na base dez pelos números 13 e 14. Inicialmente usamos direcionais gerados cuidadosamente para que as hipóteses da proposição citada acima sejam válidas. No caso, para o polinômio correspondente ao inteiro 13 os cinco primeiros direcionais são 1, 1, 1, 9 e 9 enquanto que para o polinômio representado por 14 temos 1, 1, 1, 3, 11. O sexto direcional é gerado pelas fórmulas recursivas. Obtemos respectivamente 27 e 51.

A matriz  $V_{26}$  é dada por,

$$V_{26} = \begin{bmatrix} 1 & 0 & 0 & 1 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 & 1 & 0 & 1 \\ 1 & 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 & 1 & 1 & 1 \end{bmatrix} \quad (3.16)$$

A matriz  $V_{26}$  é de posto  $4 = 2t$  o que garante o espalhamento adequado dos pontos. Isso pode ser comprovado ao observar-se a figura 3.3 onde mostramos os pontos de coordenadas geradas pelas duas seqüências de ordem 64 a 127, com pano de fundo a divisão do plano unitário dada por  $t = 2$ .

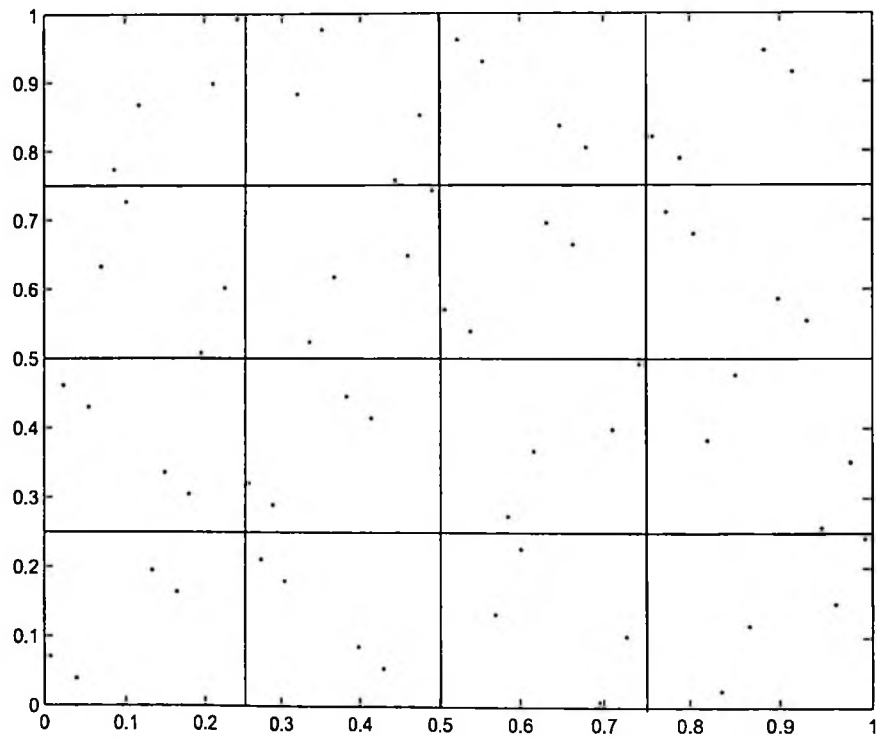


Figura 3.3: Mapeamento dos subquadrados quando  $t = 2$  - Critério de uniformidade atendido

Vamos agora modificar a matriz  $V_{26}$  através da escolha dos direcionais arbitrários. Queremos que a terceira coluna da matriz seja idêntica à primeira coluna. Para isso o primeiro bit do terceiro direcional necessita ser 1 enquanto que o segundo precisa ser nulo. Uma maneira de atingir este objetivo é de escolher o  $5 = 101$  como terceiro direcional para ambas as dimensões. Conseguimos replicar a primeira coluna. Mesmo assim a matriz possui ainda posto 4. Para forçarmos mais uma dependência linear gostaríamos também que a quarta coluna de  $V_{26}$  seja também idêntica à primeira. A escolha de  $11 = 1011$  como quarto número direcional para a

segunda coordenada assegura este resultado. Obtemos agora a matriz,

$$V_{26} = \begin{vmatrix} 1 & 0 & 1 & 1 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 & 1 & 0 & 1 \\ 1 & 0 & 1 & 1 & 0 & 1 & 1 \\ 0 & 1 & 0 & 0 & 1 & 1 & 1 \end{vmatrix} \quad (3.17)$$

de posto 3. A figura 3.4 apresenta os pontos dados pelas duas seqüências utilizadas como coordenadas mostra as conseqüências de forma clara. Existem subquadrados que não foram preenchidos. Em suma, como a matriz não consegue gerar o espaço  $\{0, 1\}^4$ , certas combinações de bits não são geradas. Neste caso os números 1,2,4,7,8,11, 13 e 14 não são fruto de nenhuma das combinação lineares oriundas da representação binária dos inteiros de 64 a 127.

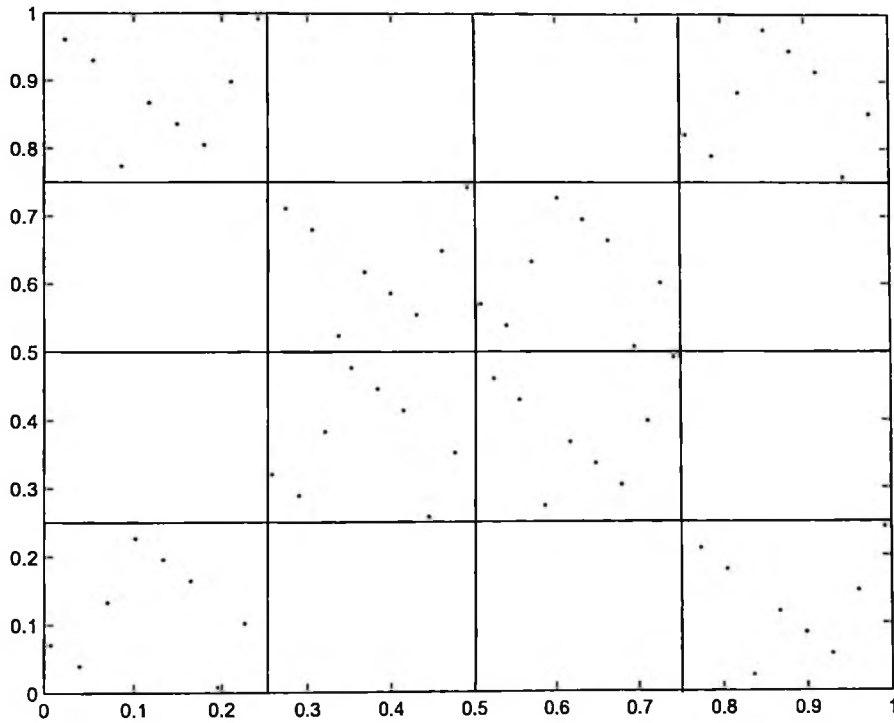


Figura 3.4: Mapeamento dos subquadrados quando  $t = 2$  - Critério de uniformidade não atendido

De modo geral, se utilizarmos correlações baixas como proxy para a ausência de projeções distorcidas vemos que a construção proposta atinge os seus objetivos. Para  $t = 1$  as maiores correlações em valor absoluto encontram-se na vizinhança de 18.50%. Consideramos este nível ainda alto. De fato podemos perceber padrões não desejáveis nas projeções. Um exemplo pode ser visto na figura 3.5 para o caso  $t = 1$ . O nosso requisito de distribuição dos pontos

é atendido porém dentro dos subquadrados a distribuição é longe de ser uniforme. Quando  $t = 2$  as correlações mais altas em valor absoluto caem para 4.50%. Ainda persistem projeções distorcidas porém os vieses são menos óbvios à vista, como pode-se ver na parte superior da figura 3.5 para  $t = 2$ . A parte inferior da figura 3.5 apresenta os piores casos possíveis para  $t = 3$  (esquerda) e  $t = 4$  (direita). Para os critérios  $t = 3$  e  $t = 4$  as correlações são muito próximas de zero. O próximo tópico apresenta o algoritmo proposto para a geração de seqüências bem comportadas.

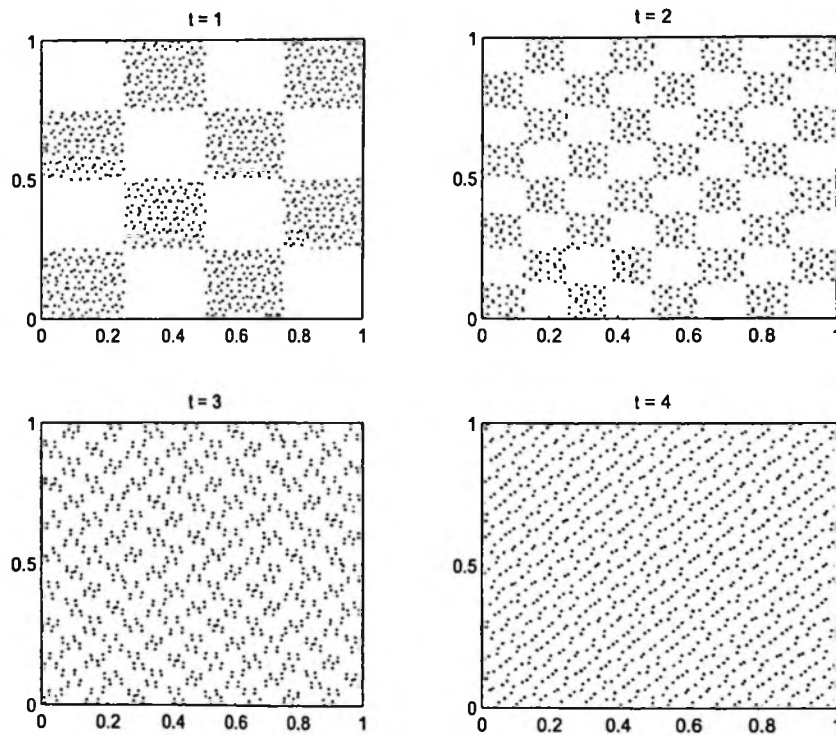


Figura 3.5: Piores cenários para o método sugerido

### 3.3.3 Um procedimento para geração dos números direcionais arbitrários

O exemplo dado acima é restrito a uma seqüência bi-dimensional. Temos todavia interesse em trabalhar com seqüências de maior dimensão. Neste caso o nosso critério de distribuição especifica que quaisquer que sejam as dimensões  $i$  e  $j$  escolhidas a projeção nestas duas dimensões possui um espalhamento que respeita a distribuição dos pontos nos subquadrados. Portanto as propriedades requeridas para a matriz  $V_{tk}$  precisam ser atendidas para todos os pares de dimensão da seqüência gerada. Na maioria dos casos a geração da seqüência depende

de mais números direcionais do que os que podem ser escolhidos arbitrariamente. A fórmula recursiva dada pelos polinômios entra em jogo. Este fator, aliado ao fato que deve se levar em consideração as dependências entre polinômios torna extremamente complexa a formulação de uma regra determinista para escolha dos números direcionais. Escolhemos não trilhar este caminho. Ao invés disto utilizamos o procedimento que descrevemos a seguir.

1. Determinar  $n$ , o número de pontos de Sobol que serão usados na simulação. Isso implica numa dimensão máxima e no melhor refinamento possível atingível através da escolha dos números direcionais. Os melhores resultados serão obviamente obtidos quando  $n$  for uma potência de 2.
2. Gerar os números direcionais para a primeira coordenada.
3. Para cada coordenada subsequente,  $i$ , gerar aleatoriamente os números direcionais arbitrários e calcular o posto da matriz  $V_{j,i}$  com  $1 \leq j < i$ . Se um deles for inferior a  $2t$  então gerar novamente os números direcionais. Se esses números passarem no teste para todas as dimensões anteriores então pode-se proceder para a próxima dimensão.

Explicamos cada passo em maiores detalhes.

Vimos anteriormente que uma das condições necessárias para que o esquema de preenchimento seja atendido é que  $2^k \leq n < 2^{k+1}$ , para  $2t \leq k$ . Isso implica que os subquadrados do quadrado unitário receberão sua cota de pontos unicamente quando o número de termos atingir  $2^k$ . Se o usuário deseja realizar uma simulação de tamanho previamente conhecido e diferente de uma potência de 2, é então preferível escolher a potência mais próxima. Nos testes a seguir algumas das simulações são efetuadas neste tipo de situação. Os resultados são favoráveis à construção apresentada. Naturalmente o melhor desempenho é obtido com potências de 2. Não consideramos restritiva esta característica da construção das sequências de Sobol. Na ampla maioria dos casos em que a simulação Monte Carlo é empregada no contexto financeiro a escolha do número de trajetórias é totalmente livre.

A escolha dos números direcionais para a primeira coordenada é totalmente livre. Não existe nenhuma restrição já que não há possibilidade de se obter uma projeção bi-dimensional.

A terceira parte do procedimento é um algoritmo que determina a escolha dos números direcionais. Os números direcionais devem, a princípio, obedecer simplesmente a dois critérios:

1. Cada número direcional deve ser ímpar



2. O  $i$ -ésimo número direcional deve ser estritamente inferior a  $2^i$ .

Dadas essas duas regras, a liberdade de escolha é grande. Para se ter uma idéia, para um polinômio de grau  $n$  já é possível escolher  $\prod_{i=0}^{n-1} 2^i$  direcionais. Para  $n = 10$  obtemos mais de 35 trilhões de escolhas possíveis. Um mapeamento sistemático de todas as combinações possíveis encontra-se tecnologicamente fora do nosso alcance. Utilizamos o gerador de números aleatórios RAN2 que pode ser encontrado em Press et al. [1992] para gerar números direcionais que atendem as restrições enunciadas. Uma vez os números escolhidos é necessário, caso  $k$  exceda o grau do polinômio, gerar os demais números direcionais requeridos para formação da matriz  $V_{ik}$  através das fórmulas recursivas dadas pelos polinômios irredutíveis. Uma vez obtidos os números direcionais é necessário testar se os números de Sobol que serão obtidos através destes possuem a propriedade desejada. Para isso cada projeção bi-dimensional deve possuir o espalhamento adequado.

É portanto necessário sempre testar os novos números direcionais escolhidos aleatoriamente com os demais números determinados até agora. O teste consiste em formar a matriz  $V_{ik}$  com os números direcionais escolhidos aleatoriamente e os números direcionais da dimensão na qual efetua-se a projeção. O posto desta matriz tem que ser igual a  $2^t$ . Por exemplo, quando geramos os números direcionais da segunda dimensão o teste é efetuado unicamente com os números escolhidos para a primeira dimensão. Para a terceira dimensão, testamos inicialmente os números escolhidos em conjunção com a primeira dimensão (equivalente a uma projeção da seqüência nas dimensões 1 e 3). Se verificarmos que o posto da matriz formada pelos bits dos números direcionais é de valor  $2^t$  então testamos os novos números direcionais com os da segunda dimensão. Caso a matriz não tenha posto igual ao número de linhas então novos números direcionais são escolhidos aleatoriamente. Como pode-se inferir o procedimento requer um critério de término caso os testes falhem repetidamente. A implementação mais simples utilizada neste trabalho estipula um número máximo de tentativas de 25 mil para a geração dos números direcionais da próxima coordenada. Uma vez atingido este ponto o algoritmo cessa de gerar números direcionais. A maneira escolhida para testar o posto da matriz consiste em testar as combinações lineares não-nulas possíveis de suas linhas. Caso uma delas resulte no vetor nulo então abortamos o teste e gera-se novamente um conjunto de números direcionais arbitrários. Como estamos trabalhando no espaço vetorial  $\{0, 1\}^{2^t}$  sobre  $\{0, 1\}$  o aspecto computacional é grandemente facilitado. Vejamos agora o conceito de dimensão efetiva, o que nos permite em

seguida julgar o desempenho da construção acima exposta em contextos desprovidos de vieses.

## Capítulo 4

### A dimensão efetiva

Tradicionalmente, a noção de dimensão de uma simulação Monte Carlo corresponde à dimensão do espaço no qual o problema é definido. Até agora utilizamos o exemplo de integrais no  $[0, 1]^d$  o que implica que cada integral é função de  $d$  variáveis. As simulações que visam a estimar o valor dessas integrais possuem portanto dimensão  $d$ . Porém, é intuitivo pensar que a imagem da função é mais afetada por certas variáveis do que por outras. Em suma, é muito provável que possamos isolar um subconjunto de  $[0, 1]^d$ , digamos por exemplo  $U = [0, 1]^i \times [0, 1]^j \times [0, 1]^k$  com  $i, j, k \in \{0, 1, \dots, d\}$  que contribui para a maior parte das variações na função. Conseqüentemente a precisão da simulação Monte Carlo depende fundamentalmente da qualidade dos números pseudo-aleatórios utilizados para o conjunto  $U$ . Logo, a identificação deste subconjunto de variáveis permitiria melhorar em muito os resultados da simulação. Ao mesmo tempo, se as variáveis que não pertencem a  $U$  não possuem influência significativa sobre a função pode-se falar de dimensão efetiva, a dimensão de  $U$ , em contraposição à dimensão nominal,  $d$ . O interesse por essa dicotomia é recente.

A teoria sobre dimensão efetiva encontra-se ainda em estado embrionário. As primeiras tentativas de isolar-se a dimensão efetiva da dimensão nominal são devidas a Caflisch e Morokoff [1997]. A introdução do conceito de dimensão efetiva é motivada pela tentativa de explicar o sucesso de métodos quase Monte Carlo em finanças. Como vimos anteriormente os resultados apresentados batiam de frente com a teoria que alegava que seqüências de baixa discrepância são inoperantes em altas dimensões. Como esperado, o conceito permitiu ao mesmo tempo explicar os paradoxos apresentados anteriormente e submeter as novas construções propostas a testes desprovidos de vieses. Neste capítulo apresentamos inicialmente as diversas definições

de dimensão efetiva. Em seguida mostra-se um algoritmo que permite o seu cálculo. Finalmente, como primeiro teste para o algoritmo de geração de números direcionais e aplicação do conceito de dimensão efetiva comparamos a construção proposta com a construção de melhor desempenho na atualidade, sugerida por Joe e Kuo [2003].

## 4.1 Definição

### 4.1.1 A decomposição ANOVA

Começamos com uma introdução na qual apresentamos as principais definições necessárias ao entendimento da dimensão efetiva. Além do trabalho já citado outras pesquisas de Caflisch, Morokoff e Owen [1997], Owen [2001] e Wang e Fang [2003] têm contribuído muito para a difusão do conceito. Wang e Fang [2003], baseados em trabalhos do Sobol [2001] e [1993] são os primeiros a delinear um algoritmo para o cálculo da dimensão efetiva.

A noção de dimensão efetiva é uma aplicação da análise de variância - ANOVA. Nas suas grandes linhas a análise de variância identifica os conjuntos de variáveis que mais contribuem para a variação da função. Esta noção de estatística multivariada foi estendida também à análise de algoritmos computacionais de grande complexidade para identificação das variáveis que mais influenciam o código.

Seja  $f$  uma função definida sobre  $X = [0, 1]^d$  tal que  $\int f(x)^2 dx < \infty$ . Define-se a média de  $f$ , denotada  $I$  como,

$$I(f) = \int_{[0,1]^d} f(x) dx$$

e a variância de  $f$ ,  $\sigma^2$  como,

$$\sigma^2(f) = \int_{[0,1]^d} (f(x) - I(f))^2 dx$$

Seja  $D = \{1, 2, \dots, d\}$  e  $u \subseteq D$ . A cardinalidade de  $u$  é denotada  $\#(u)$ . O complemento de  $u$  em relação a  $X$  é dado por  $-u$ . Um ponto qualquer de  $[0, 1]^d$  é dado por  $x = (x_1, \dots, x_d)$ . Utilizaremos  $x^u$  para expressar um vetor de  $\#(u)$  componentes  $x^j$  com  $j \in u$ . Seja  $[0, 1]^u$  o conjunto de todos os vetores  $x^u$ . A integral da função  $f$  sobre  $[0, 1]^u$  depende unicamente das coordenadas de índice em  $u$ . Ao integrarmos  $f$  as coordenadas de índice em  $-u$  são tomadas como constantes. A decomposição ANOVA permite expressar toda função de  $L^2$  como,

$$f(X) = \sum_{u \subseteq D} f_u(X) \quad (4.1)$$

onde os termos da soma, geralmente chamados de termos ANOVA, são definidos por,

$$f_u(X) = \int_{X^{-u}} f(X) dX^{-u} - \sum_{v \subseteq u} f_v(X) \quad (4.2)$$

onde por convenção  $f_\emptyset(X) = I(f)$  para todo  $X$ . A obtenção dos termos  $f_u$  é um processo recursivo. Os termos da soma possuem propriedades atraentes,

1. Para  $j \in u$ ,  $\int_0^1 f_u(x) dx_j = 0$ .

2. A decomposição ANOVA é ortogonal, ou seja,

$$\int_X f_u(x) f_v(x) dx = 0$$

quando  $u \neq v$ .

3. A variância de  $f$  corresponde a,

$$\sigma^2(f) = \sum_{u \subseteq D} \sigma_u^2(f)$$

onde,

$$\sigma_u^2(f) = \int_X [f_u(x)]^2 dx$$

para  $|u| > 0$ , com  $\sigma_\emptyset^2(f) = 0$ .

Essas definições e propriedades nos permitem agora definir as diferentes versões de dimensão efetiva.

#### 4.1.2 A dimensão efetiva

Existem vários conceitos de dimensão efetiva. Listamos aqui as duas principais definições.

Seja  $u$  um subconjunto de  $D$ . Pelo que vimos acima a variância que podemos atribuir a  $u$  é definida por,

$$T_u = \sum_{v \subseteq u} \sigma_v^2(f) \quad (4.3)$$

O efeito total de  $u$  na variância é definido por,

$$T_u^{tot} = \sum_{v \cap u \neq \emptyset} \sigma_v^2(f) = \sigma^2(f) - T_{D-u} \quad (4.4)$$

O efeito total de  $u$  na variância leva em consideração não apenas o efeito direto de  $u$ ,  $T_u$  mas também os efeitos devidos à interação dos membros de  $u$  com as demais variáveis. O tamanho

relativo de  $T_u$  ou  $T_u^{tot}$  em relação a  $\sigma^2(f)$  é uma indicativa da importância relativa de  $X^u$ . Quando  $T_u$  é grande em relação a  $\sigma^2(f)$  sabemos que a influência direta sobre a função é grande. Isso acontece também quando  $T_u^{tot}$  é importante mesmo que  $T_u$  seja pequeno. Quando ambos os números são pequenos sabemos que  $u$  não influencia muito a simulação e dependendo da magnitude das contribuições podemos escolher valores predeterminados para  $X^u$ . Podemos agora enunciar as duas principais medidas de dimensão efetiva.

A dimensão efetiva de  $f$  de acordo com o conceito de sobreposição é o menor  $d_s$  tal que,

$$\sum_{0 < \#(u) < d_s} \sigma_u^2(f) \geq p\sigma^2(f) \quad (4.5)$$

A dimensão efetiva de  $f$  de acordo com o conceito da truncagem é o menor  $d_t$  tal que,

$$T_{(1, \dots, d_t)} \geq p\sigma^2(f). \quad (4.6)$$

Como pode-se inferir a partir das definições de dimensão efetiva o seu cálculo não é trivial. São poucos os casos em que existe uma solução analítica fácil de ser obtida. Na maioria das vezes o próprio fato de termos interesse na dimensão efetiva implica que o cálculo de  $I(f)$  requer um método numérico. É difícil imaginar que os termos ANOVA possam ser obtidos muito mais facilmente. Vejamos agora como utilizar estes conceitos na prática.

## 4.2 O cálculo da dimensão efetiva

Para conseguirmos determinar a dimensão efetiva, tanto no sentido da sobreposição como no sentido da truncagem, é necessário calcular a variância,  $T_u$  correspondente a qualquer subconjunto de  $D$ . Vejamos como conseguir isto.

Para todo  $u \subset D$ , sejam  $x = (x_u, x_{-u})$  e  $y = (y_u, y_{-u})$ . Sobol [2001] deriva a seguinte relação para  $T_u$ ,

$$T_u = \int f(x)f(x_u, y_{-u})dx dy_{-u} - [I(f)]^2 \quad (4.7)$$

É necessário calcular-se os termos  $T_u$ ,  $I(f)$  e  $\sigma^2(f)$ . Isso requer a estimação das seguintes integrais,

$$\int f(x)dx \quad (4.8)$$

$$\int f(x)^2 dx \quad (4.9)$$

$$\int f(x_u, x_{-u})f(x_u, y_{-u})dxdy_{-u} \quad (4.10)$$

Conforme salientado acima é provável que seja necessário estimar estes valores. Adotamos neste trabalho a prática de utilizar o método de Monte Carlo. Em suma obtemos estimativas das integrais acima através de,

$$\hat{I}(f) = \frac{1}{n} \sum_{i=1}^n f(x_i) \quad (4.11)$$

$$\hat{\sigma}^2(f) = \frac{1}{n} \sum_{i=1}^n f^2(x_i) - [\hat{I}(f)]^2 \quad (4.12)$$

$$\hat{T}_u = \frac{1}{n} \sum_{i=1}^n f((x_i)_u, (x_i)_{-u})f((x_i)_u, (y_i)_{-u}) - [\hat{I}(f)]^2 \quad (4.13)$$

A dimensão efetiva no sentido da truncagem pode ser determinada pelo cálculo de  $T_u$  com  $u$  variando em  $\{1, \dots, l\}$ , com  $l = 1, 2, \dots, d$  até o momento em que um determinado percentil da variância é atingido. Já, para a dimensão efetiva no sentido da sobreposição, o cálculo é muito mais complexo no sentido computacional pois é necessário calcular  $T_u$  para subconjuntos de cardinalidade variando de 1 ao índice suficiente para satisfazer a desigualdade que define a dimensão. Para os problemas que nos interessam, de dimensões nominais que podem passar de mil, o cálculo da dimensão efetiva no sentido da sobreposição é inviável. O próximo parágrafo mostra a primeira aplicação do algoritmo proposto.

### 4.3 Integrais de teste

A busca de métodos numéricos eficientes para a estimação de integrais tem gerado uma ampla gama de funções ditas de teste. Estas funções são concebidas de modo a gerar os problemas de integração mais difíceis possíveis. Uma descrição de algumas classes de função de teste pode ser encontrada em Owen [2001]. Uma família muito popular de funções é dada por,

$$f(x) = \prod_{k=1}^d \frac{|4x_k - 2| + a_k}{1 + a_k} \quad (4.14)$$

Os parâmetros  $a_k$  permitem uma sintonia da função de modo a gerar maior ou menor interação entre as variáveis e controlar a importância das dimensões mais altas. Um grande atrativo desta classe de funções é que os termos necessários ao cômputo das dimensões efetivas de acordo com as definições acima são obtidos de maneira analítica. Temos respectivamente,

$$I(f) = 1 \quad (4.15)$$

$$\sigma^2(f) = \prod_{k=1}^d \left[ 1 + \frac{1}{3(1+a_k)^2} \right] - 1 \quad (4.16)$$

e,

$$T_u = \prod_{k \in u} \left[ 1 + \frac{1}{3(1+a_k)^2} \right] - 1 \quad (4.17)$$

Os tipos mais comuns de parametrização correspondem a  $a_k = k^z$ . Conforme  $z$  tende a zero a dimensão efetiva da integral torna-se cada vez maior. A próxima tabela mostra algumas dimensões efetivas no sentido da truncagem para  $z \in \{0, 1/3, 3/4, 1, 2\}$ .

$d$	$z = 0$	$z = 1/3$	$z = 3/4$	$z = 1$	$z = 2$
50	50	50	48	39	5
100	100	100	95	64	5
200	200	199	186	93	5
300	300	299	276	110	5
400	400	398	364	121	5
500	500	498	451	129	5
600	600	598	538	135	5
700	700	698	623	140	5
800	800	797	708	143	5
900	900	897	792	146	5
1000	1000	876	742	148	5

Tabela 4.1: Dimensões efetivas (truncagem) para vários parâmetros  $a_k$  em 4.14

Vemos nela que a dimensão efetiva para potências pequenas é muito próxima da dimensão nominal representada por  $d$ . Conforme  $z$  aumenta a dimensão efetiva diminui de maneira exponencial, dada uma dimensão nominal.

## 4.4 Comparação com o método de Joe & Kuo

Consideramos este primeiro teste ao qual submetemos o algoritmo de escolha dos direcionais o mais árduo de todos pois neste caso a dimensão efetiva não depende de simulação e pode ser definida com muita precisão. Não conhecemos na literatura nenhuma simulação quase Monte Carlo que apresente resultados tão bons quanto os relatados no trabalho de Joe e Kuo [2003].



Portanto, o objetivo desta seção consiste em efetuar uma comparação com o que pode ser considerado a construção mais eficiente na atualidade.

#### 4.4.1 O método de Joe & Kuo

Joe e Kuo [2003] desenvolvem um algoritmo de escolha de direcionais que atende à propriedade A de Sobol. Em suma a idéia consiste em gerar aleatoriamente os números direcionais arbitrários até que a proposição 3.3.1 seja satisfeita. Lembramos que o uso destes números garante que se o cubo  $[0, 1]^d$  for dividido pelos hiperplanos  $x_j = 1/2$  para  $j = 1, \dots, d$  em  $2^d$  cubos de volume idêntico então cada subsequência de  $2^d$  números não apresentará mais de um ponto em cada subcubo. Conforme os próprios autores relatam,

Property A is not that useful to have for large  $d$  because of the computational time required to approximate an integral using  $2^d$  points.

Efetivamente, para as dimensões testadas tanto aqui como no trabalho deles necessitaria-se de simulações de até  $2^{1000}$  para aproveitar plenamente a propriedade A. Isso é simplesmente inviável em situações reais. Todavia, para simulações de tamanho aceitáveis na prática, a propriedade A parece gerar ótimos resultados. No final, estes não deixam de gerar um pequeno desconforto pois se a propriedade almejada não entra em jogo e mesmo assim as estimativas das integrais são muito próximas do esperado deve haver incidência de outras propriedades que não foram mapeadas. Esta crítica é um dos fatores que nos motivou a buscar métodos que gerassem propriedades precisas para pequenas simulações.

#### 4.4.2 Os resultados

A função utilizada nos testes de Joe e Kuo [2003] é dada por,

$$f(x) = \prod_{k=1}^d \frac{|4x_k - 2| + k^{\frac{1}{3}}}{1 + k^{\frac{1}{3}}} \quad (4.18)$$

Isso corresponde a  $z = 1/3$ . Neste caso, como pode-se notar na tabela 4.1, a dimensão efetiva fica muito próxima da dimensão nominal. A escolha deste expoente é baseada no trabalho de Wang e Fang [2003]. Conforme salientado acima o resultado da integral de 4.18 pode ser obtido analiticamente.  $I(f) = 1$ . Os autores comparam as seqüências que atendem à propriedade A

com resultados obtidos com o uso de seqüências de Faure e lattice rules. Os números obtidos com as seqüências de Faure são bastante voláteis e possuem desempenho muito inferior ao obtido com a propriedade A. O método das lattice rules é mais preciso. Todavia, para as dimensões mais altas gera resultados inferiores (no sentido em que o valor absoluto do erro é maior) aos obtidos através do uso de números direcionais que satisfazem a propriedade A. Diante dessa hierarquia de performance decidimos não apresentar os demais métodos e nos restringirmos unicamente à comparação das simulações quase Monte Carlo com seqüências de Sobol.

A tabela 4.2 mostra os resultados obtidos. Não há menção no trabalho de Joe e Kuo [2003] da maneira pela qual seus resultados foram gerados. É possível que os autores tenham gerado um conjunto grande de números direcionais e escolhido em seguida um subconjunto para os testes. É também provável que tenham gerado uma única vez uma estimativa para cada integral e cada tamanho de simulação. Diante da incerteza sobre a metodologia usada escolhemos efetuar duas simulações para cada teste. Os números da tabela representam a melhor das duas estimativas. Naturalmente, ao passo em que efetuávamos os testes, encontramos conjuntos de números direcionais de performance muito superior à média apresentada. Porém, parece mais justo simular uma situação em que o usuário não possui uma experiência significativa na geração de seqüências de Sobol e não deseja utilizar direcionais sugeridos por outros usuários.

O critério utilizado para a escolha dos números direcionais é único. Não utilizamos o método misto que permite otimizar a ausência de correlação ao permitir graus de refinamento,  $t$ , variáveis. Deste modo podemos sentir com mais clareza o desempenho de cada critério. Os testes efetuados com problemas de finanças fazem uso do método misto. O número entre parênteses ao lado da estimativa da integral na tabela 4.2 informa o critério ou  $t$  utilizado para aquela estimativa. Como indicado anteriormente não é possível utilizar todos os critérios em todas as situações. Veremos isso com mais clareza ao precificar os ativos financeiros onde adotaremos o critério misto. Será então possível ver os limites de cada critério de acordo com a dimensão e o tamanho da simulação. Por enquanto notamos que só foi possível utilizar o critério  $t = 4$  até a dimensão 1000 com simulações de tamanho  $n = 128021$ . Também, para simulações de tamanho  $n = 1009$  não é possível gerar pontos que atendem ao primeiro critério para simulações de dimensão superior a 512. De um total de 83 comparações temos quase 50% efetuadas com o critério  $t = 4$ , aproximadamente 25% efetuadas com o critério  $t = 3$  e o resto com os critérios  $t = 1$  e  $t = 2$ .

Como critério de erro optamos pelo valor absoluto da diferença entre o valor estimado e o valor correto,  $I(f) = 1$ . Antes de compararmos os números com os de Joe e Kuo [2003] frisamos que do ponto de vista do exercício de estimação o método mostrou-se inteiramente adequado. A partir de simulações de  $n = 16001$ , número que pode ser considerado até otimista quando a dificuldade da integral é levada em consideração, as estimativas encontram-se muito próximas de 1. Portanto a comparação com os resultados de Joe e Kuo [2003] não é espúria. Em relação a estes o algoritmo proposto possui erro menor em 71.08% das simulações. Pode-se notar alguns padrões. Se, por força de hábito, separarmos a tabela em quatro zonas, vemos que a comparação é desfavorável ao algoritmo proposto unicamente na zona em que as dimensões são altas e o tamanho da simulação é pequeno. Neste caso obtemos resultados superiores unicamente em 33% das simulações. Porém isso não nos preocupa pois trata-se de uma região na qual nenhum dos métodos seria utilizado na prática. Isso não corresponderia a uma tática segura por parte do usuário. Além disso, não somos capazes de gerar pontos com critérios superiores a dois. Em situações normais e dimensões até 500 o nosso método apresenta resultados mais precisos em 75% das simulações. O aspecto mais encorajador é o desempenho na zona onde o número de dimensões é muito alto e as simulações de tamanho realista. Nesta região obtemos estimativas mais precisas em 90% dos casos. O método parece apontar para uma melhora sensível para dimensões muito altas.

Terminamos com uma pequena observação sobre o tamanho das simulações. Estes não correspondem aos tamanhos ótimos para a metodologia proposta. Efetivamente, o fato de não usarmos potências de 2 impossibilita o uso preciso de um ciclo de preenchimento. Algumas impurezas - falta de pontos em alguns subquadrados - são introduzidas no processo. No entanto optamos por manter os tamanhos definidos por Joe e Kuo [2003]<sup>1</sup> para mostrar que pode-se fugir de tamanhos predeterminados sem gerar grandes distorções.

Vejamos agora como o algoritmo se comporta com problemas na esfera financeira e de dimensão efetiva ainda maiores.

---

<sup>1</sup> Os tamanhos, todos primos, foram escolhidos para permitir a comparação com as lattice rules

d = 50	N = 1009	N = 1997	N = 4001	N = 8009	N = 16001	N = 32003	N = 64007	N = 128021
JK	0.9743	0.9544	0.9838	0.9982	0.9995	0.9976	0.9969	0.9975
BS	0.9697 (3)	1.0088 (4)	1.0017 (4)	0.999 (4)	1.0028 (4)	1.004 (4)	0.9994 (4)	0.9975 (4)
d = 100	1009	1997	4001	8009	16001	32003	64007	128021
JK	1.0157	0.9288	0.9649	0.9866	0.9985	0.9817	0.9955	0.9995
BS	0.9858 (2)	0.9577 (3)	1.05 (4)	1.0079 (4)	0.9914 (4)	0.9963 (4)	0.9995 (4)	1.0042 (4)
d = 200	1009	1997	4001	8009	16001	32003	64007	128021
JK	1.1109	0.8877	0.9070	1.0312	1.0216	0.9772	0.9744	0.9925
BS	0.9188 (2)	0.959 (2)	0.92 (3)	1.0118 (4)	0.9892 (4)	0.9776 (4)	0.9949 (4)	1.0056 (4)
d = 300	1009	1997	4001	8009	16001	32003	64007	128021
JK	0.9770	0.8715	0.9305	0.9546	1.0060	0.9712	0.9770	0.9843
BS	1.0004 (1)	0.9185 (2)	0.9773 (3)	0.9636 (3)	0.9824 (4)	0.9836 (4)	0.9933 (4)	1.0105 (4)
d = 400	1009	1997	4001	8009	16001	32003	64007	128021
JK	1.1780	0.8908	0.9710	0.9527	1.0203	0.9754	0.9779	0.9725
BS	0.9191 (1)	0.9278 (1)	0.952 (2)	1.0045 (3)	0.992 (3)	0.9793 (4)	0.9768 (4)	0.997 (4)
d = 500	1009	1997	4001	8009	16001	32003	64007	128021
JK	1.0567	0.8548	0.9700	0.9704	0.9768	0.9634	0.9745	0.9670
BS	0.8854 (1)	1.6145 (1)	1.0149 (2)	0.9563 (3)	1.0028 (3)	1.0151 (4)	1.0124 (4)	1.0198 (4)
d = 600	1009	1997	4001	8009	16001	32003	64007	128021
JK	0.8735	0.7674	1.0130	0.9874	0.9677	0.9416	0.9580	0.9564
BS	-	0.8382 (1)	1.0158 (1)	0.9768 (3)	1.0077 (3)	0.982 (4)	1.0191 (4)	1.0223 (4)
d = 700	1009	1997	4001	8009	16001	32003	64007	128021
JK	0.8752	0.8241	1.0067	1.0078	0.9987	0.9236	0.9517	0.9475
BS	-	0.8259 (1)	0.8927 (1)	0.9721 (2)	0.9808 (3)	1.0335 (3)	1.0165 (4)	1.009 (4)
d = 800	1009	1997	4001	8009	16001	32003	64007	128021
JK	0.9139	0.8822	1.0926	1.0560	1.0210	0.9454	0.9330	0.9359
BS	-	0.8548 (1)	1.0103 (1)	0.93 (2)	0.9827 (3)	0.9399 (3)	0.9585 (4)	1.0016 (4)
d = 900	1009	1997	4001	8009	16001	32003	64007	128021
JK	0.7779	1.0225	1.0593	1.0991	1.0470	0.9115	0.9688	0.9615
BS	-	0.9257 (1)	1.0875 (1)	1.0692 (2)	0.9612 (3)	1.0104 (3)	0.9825 (4)	1.0052 (4)
d = 1000	1009	1997	4001	8009	16001	32003	64007	128021
JK	0.8348	1.0443	1.0710	1.0862	1.0483	0.9188	0.9735	0.9683
BS	-	1.267 (1)	1.0235 (1)	1.1619 (2)	0.9827 (3)	1.0178 (3)	0.9974 (3)	0.9736 (4)

Tabela 4.2: Comparação do método proposto (BS) com os resultados de Joe e Kuo (JK)

## Capítulo 5

### Aplicações em finanças

A busca por um método mais eficiente para a resolução de problemas geralmente tratados via simulação Monte Carlo é motivada pelo grande ônus em termos de tempo e máquina normalmente requerido pelo método. O incentivo tornou-se maior ainda nos últimos anos com o rápido crescimento do número de produtos financeiros que dependem de simulação, método que, hoje, faz parte integrante da rotina diária de vários operadores e integrantes de áreas de risco. Para testarmos a nossa construção neste âmbito escolhemos três produtos clássicos, uma opção com barreira, uma opção basket asiática e uma opção asiática. Essas opções dependem da trajetória do ativo subjacente ao longo do tempo e são facilmente precificadas através de simulação Monte Carlo (ver a introdução). Em todos os casos efetuar-se-á uma comparação entre o método tradicional e o método de Sobol com direcionais escolhidos através do algoritmo proposto neste trabalho. O método tradicional foi implementado com o gerador de números pseudo-aleatórios RAN2 apresentado em Press et al. [1992]. Este gerador foi utilizado em várias pesquisas, por exemplo Galanti e Jung [1997], Boyle e Tan [2000], e é considerado satisfatório na literatura. O algoritmo de Moro, usado para inversão da distribuição normal padrão merece também menção. Ele é também usado na literatura e considerado satisfatório, Galanti e Jung [1997].

Em termos de performance não há diferença entre os métodos. Parte-se do princípio que a escolha dos números direcionais é efetuada anteriormente. A comparação entre os métodos é portanto imediata, basta calcular a razão dos números de trajetória para o mesmo critério de convergência. Por exemplo, para um erro de 1% como critério de convergência, se o método 1 não ultrapassar este patamar a partir de  $n = 200$  enquanto que o método 2 ainda o viola até  $n = 600$  então o método 1 é três vezes mais rápido que o método 2. Apresentamos primeiro

o caso da opção com barreira e em seguida a opção basket asiática. Devida à complexidade e ao tempo requerido para as simulações que estimam a dimensão efetiva trabalha-se unicamente com a dimensão nominal. De fato, para se calcular, por exemplo, a dimensão efetiva de um problema de dimensão nominal 1000 são necessárias 1000 simulações Monte Carlo de um número adequado de trajetórias. Optando por simulações de 10 mil pontos atingimos a cifra de 10 milhões de trajetórias. Como é necessário efetuar-se o cálculo para dimensões nominais variando de 250 a 2000 vê-se que o custo computacional é muito grande. Apresentamos a dimensão efetiva para as opções asiáticas, resultados gerados em cinco estações trabalhando em paralelo durante 3 dias.

## 5.1 Opções com barreira

### 5.1.1 O produto

Opções com barreira são de estrutura simples. O pagamento final depende do ativo subjacente ter ultrapassado ou não uma barreira predeterminada. Existem várias consequências possíveis relacionadas a este evento. Por exemplo, uma opção “down and in” só passa a valer se o ativo subjacente passar uma barreira situada em nível inferior ao valor atual do ativo. Uma opção “down and out”, ao contrário, cessa de valer se o preço do ativo subjacente for menor, em algum momento da vida do derivativo, do que a barreira. O mesmo raciocínio aplica-se às opções “up and in” e “up and out”. Normalmente as partes envolvidas na transação escolhem o preço de fechamento diário, ou alguma média dos últimos preços transacionados, do ativo subjacente como preço de referência. No caso de uma opção de compra “down and out”, se em algum momento este preço for inferior à barreira estipulada a opção cessa de existir. Se isso não ocorrer durante o tempo de vida da opção então o seu valor no final é dado pelo máximo entre a diferença entre o valor a vista da ação e o strike e 0.

Escolhemos aqui opções de compra com barreira do tipo “down and out”. Os parâmetros necessários para a simulação são  $S = 100$ ,  $K = 120$ ,  $B = 95$ ,  $r = 0,2$  e  $\sigma = 0,2$ , onde  $S$  é o preço à vista do ativo subjacente no momento em que precifica-se a opção,  $K$  é o preço de exercício,  $B$  é o valor da barreira,  $r$  é a taxa de juros livre de risco e  $\sigma$  a volatilidade do ativo subjacente, estas últimas expressas em termos anuais. Escolhemos observar o preço do ativo ao final de cada dia útil, tornando cada passo no tempo equivalente a  $1/252$  unidades de tempo.

O tempo até o vencimento é portanto equivalente à dimensão da simulação - equivalente ao número de preços observados durante cada trajetória - vezes  $1/252$ . As dimensões vão de 250 a 2000 em intervalos de 250. Temos portanto 8 opções. Elas diferem entre si pelo prazo até o vencimento.

### 5.1.2 Resultados

O critério escolhido para a geração dos números direcionais é o que chamamos de misto. Começamos a gerar direcionais que atendem ao critério  $t = 4$ . O número máximo de tentativas para cada nova dimensão é fixado em 25 mil. Se atingirmos este limite passa-se para o critério imediatamente inferior  $t = 3, 2, 1$ , até conseguirmos novos direcionais. Desta forma estamos sempre trabalhando com o melhor grau de refinamento possível das distribuições dos pontos nos hiperquadrados. Por exemplo, para simulações de tamanho 4096 conseguimos utilizar o critério  $t = 4$  até a dimensão 115,  $t = 3$  até a dimensão 373,  $t = 2$  até a dimensão 964 e como era fácil prever o critério  $t = 1$  até a dimensão 2048. A tabela 5.1 mostra a evolução dos critérios utilizados ao passo em que o tamanho da simulação aumenta.

t	4096	8192	16384	32768	65536	131072
3	115	228	445	892	1783	-
2	373	721	1422	-	-	-
1	964	1939	-	-	-	-

Tabela 5.1: Graus de refinamento para opção com barreira até  $d = 2000$

Lê-se a tabela da seguinte maneira. Começamos sempre com  $t = 4$ . Quando não conseguimos mais gerar números direcionais que atendem às condições necessárias para  $t = 4$  tentamos gerar direcionais que satisfazem o critério  $t = 3$ . Isto corresponde à primeira linha da tabela. Vê-se por exemplo, que para simulações de tamanho 16384 geramos os direcionais até a dimensão 892 de acordo com o critério  $t = 4$ . Quando não há entrada, caso de simulações de tamanho 131072, não foi necessário mudar de critério. Todos os direcionais foram gerados de acordo com o critério mais fino implementado. Lembramos que a escolha do refinamento é decisão do usuário. Podemos também observar que a fronteira até a qual pode-se utilizar um critério dobra aproximadamente conforme o tamanho da simulação aumenta para a próxima potência de 2. Não apresentamos esta tabela para os demais experimentos pois a escolha dos

direcionais é independente do problema em questão.

Na tabela 5.2 mostramos os erros obtidos com os métodos tradicionais e quase Monte Carlo. Para podermos auferir a precisão dos métodos é necessária alguma medida de erro. Optamos

d = 250	$N = 2^{12}$	$N = 2^{13}$	$N = 2^{14}$	$N = 2^{15}$	$N = 2^{16}$	$N = 2^{17}$
MC	3.63	1.63	0.74	0.64	0.03	0.47
SQMC	1.53	0.09	0.97	0.59	0.11	0.30
d = 500	4096	8192	16384	32768	65536	131072
MC	0.17	2.09	1.39	0.67	0.21	0.30
SQMC	0.39	0.51	0.61	0.49	0.14	0.38
d = 750	4096	8192	16384	32768	65536	131072
MC	0.50	1.25	3.06	0.39	0.03	0.46
SQMC	0.06	0.70	0.30	0.63	0.23	0.29
d = 1000	4096	8192	16384	32768	65536	131072
MC	2.03	0.49	0.47	0.08	0.50	0.76
SQMC	0.46	0.98	0.45	0.49	0.01	0.18
d = 1250	4096	8192	16384	32768	65536	131072
MC	0.65	1.39	1.25	0.80	0.41	0.75
SQMC	0.20	0.83	0.24	0.52	0.04	0.09
d = 1500	4096	8192	16384	32768	65536	131072
MC	1.25	2.08	0.41	0.79	0.82	0.34
SQMC	0.05	0.71	0.68	0.61	0.15	0.34
d = 1750	4096	8192	16384	32768	65536	131072
MC	3.28	3.01	0.64	1.79	0.94	0.06
SQMC	0.17	0.51	0.56	0.88	0.01	0.62
d = 2000	4096	8192	16384	32768	65536	131072
MC	1.82	0.65	0.53	0.71	0.18	0.32
SQMC	0.09	0.05	0.35	0.66	0.18	0.43

Tabela 5.2: Opção com barreira: resultados

aqui pelo valor absoluto do desvio percentual em relação ao preço justo aproximado. Este preço justo é obtido através de simulações Monte Carlo de tamanho  $n = 1$  milhão de trajetórias. Se notarmos  $p$  este valor e  $\hat{p}$  o preço obtido no exercício de comparação então o erro  $e$  é dado por,

$$e = \left| \frac{\hat{p} - p}{p} \right| \quad (5.1)$$

Em primeira instância é interessante notar que apenas uma das simulações quase Monte Carlo



apresenta erro superior a 1%. Trata-se da precificação da opção de dimensão 250 com 4096 trajetórias. As demais encontram-se abaixo de 1%. Em contraste o método tradicional apresenta erro superior a 1% em 30% dos casos. Neste grupo de valores encontramos 4 valores acima de 3%. É surpreendente notar que para dimensões acima de mil o erro obtido com 4096 pontos é muito baixo, sendo de 0,20%, 0,05%, 0,17% e 0,09% para as dimensões respectivas de 1250, 1500, 1750 e 2000. O método tradicional apresenta erros de 0,65%, 1,25%, 3,28% e 1,82%. A diferença é gritante. Ao todo, o método quase Monte Carlo mostrou-se mais preciso em 77,08% das vezes, o que constitui um resultado encorajador quando as dimensões são levadas em consideração. Ambos os métodos convergem para o resultado justo embora não de forma monotônica. É freqüente o erro aumentar com o tamanho da simulação. Calculamos o número de vezes em que um resultado piora com o aumento do número de trajetórias. Por exemplo se a simulação de tamanho 4096 apresentou melhor desempenho que as simulações de tamanho 8192 e 32768 então entendemos que a simulação apresentou resultados inferiores duas vezes mesmo que a simulação de 32768 pontos gere um erro inferior à de 8192 pontos. De acordo com este critério obtemos 33 piores para o método tradicional e 52 para o método quase Monte Carlo. Parece que a convergência é mais suave para a simulação Monte Carlo. Uma explicação possível é que no caso do método tradicional partimos de erros muito maiores para dimensões pequenas. A simulação quase Monte Carlo tende a “barrigar” um pouco no sentido em que os resultados obtidos para simulações de tamanhos intermediários, na sua maioria, são superiores aos obtidos com simulações de tamanho pequeno ou muito grande. Mesmo assim a magnitude desses erros não invalida o uso do método. Mesmo sem noção da dimensão nominal das simulações efetuadas neste experimento achamos os resultados alentadores pois testes com problemas semelhantes em Barbe [2001] mostram uma deterioração rápida dos resultados para simulações efetuadas com a construção clássica de Sobol e as seqüências de Niederreiter-Xing. Simulações de dimensão inferior a 100 já divergiam significativamente do resultado esperado.

## 5.2 Opção basket asiática

### 5.2.1 O produto

O segundo experimento em finanças faz uso de uma opção basket asiática. Trata-se de uma opção cujo pagamento depende da média de preços de vários ativos agrupados numa cesta

(basket). Mostramos a seguir o seu funcionamento de forma mais detalhada.

No instante inicial monta-se uma carteira de  $k$  ativos,  $s_i$ ,  $1 \leq i \leq k$ , em quantidades  $q_i$ ,  $1 \leq i \leq k$ . O valor da carteira  $\Pi_0$  no instante inicial  $t = 0$  é,

$$\Pi_0 = \sum_{i=1}^k s_i^0 q_i$$

Supõe-se agora que os preços são observados simultaneamente para cada ativo em  $d$  momentos diferentes. São estes preços que vão compor a média da cesta. Em cada momento em que os  $k$  preços são observados obtemos um novo valor para a carteira. No momento em que a opção vence temos uma coleção de  $d$  preços para a carteira da qual efetuamos a média  $\mu_\Pi$ ,

$$\mu_\Pi = \frac{1}{d} \sum_{j=1}^d \sum_{i=1}^k s_i^j q_i$$

O valor  $c$  de uma opção de compra no seu vencimento é dado por,

$$c = \max(\mu_\Pi - K, 0) \quad (5.2)$$

onde  $K$  é o preço de exercício da opção.

Ao contrário do caso anterior e do que segue, temos aqui uma dependência entre os ativos que deve ser levada em consideração. Não podemos simplesmente gerar  $k$  trajetórias independentes e calcular o valor fictício da carteira em cada passo. É necessário que as trajetórias possuam a estrutura de correlação presente entre os ativos. Para isto os choques brownianos devem conservar este padrão. A decomposição de Choleski da matriz de correlação entre os retornos dos ativos permite atingir este objetivo. Se  $\Lambda$  é a matriz de correlação, a sua decomposição de Choleski gera uma matriz triangular inferior  $C$  tal que  $\Lambda = CC'$ . Se premultiplicarmos um vetor de números aleatórios  $Z$  distribuídos de acordo com a normal padrão por  $C$  então obtemos um vetor de números aleatórios que obedece à estrutura de correlação entre os ativos. Como pode ser inferido, a dimensão do problema é agora ampliada pelo número de ativos. Seguindo a notação utilizada acima para descrever o produto a simulação Monte Carlo é de dimensão  $k \times d$ .

Para facilitar a montagem do experimento escolhemos dez ativos listados na Bovespa. Deste modo não é necessário chutar uma matriz de covariância. Na tabela 5.3 encontram-se os preços e as quantidades de cada ação que compõem a carteira.

O valor inicial desta carteira é de R\$99.998,74. O teste é conduzido com opções de compra de preço de exercício  $K = 105.000,00$ . O número de observações varia de opção para opção.

Ativo	Preço	Quantidade
$s_1^0$	36.10	277.00
$s_2^0$	118.78	84.00
$s_3^0$	71.52	140.00
$s_4^0$	48.30	207.00
$s_5^0$	47.80	209.00
$s_6^0$	29.48	339.00
$s_7^0$	2.68	3,738.00
$s_8^0$	94.51	106.00
$s_9^0$	32.05	312.00
$s_{10}^0$	83.26	120.00

Tabela 5.3: Parâmetros para as opções basket asiáticas

Esta variável nos permite modificar a dimensão da simulação. Nos nossos testes utilizamos dimensões que variam de 250 a 2000 em intervalos de 250. Isso quer dizer que observamos o preço da carteira 25 vezes para a opção de menor dimensionalidade e 200 vezes para a opção de maior dimensionalidade. Como no caso anterior, escolhemos observar os preços no final de cada dia útil o que torna o tempo entre cada observação,  $dt$ , equivalente a  $1/252$ .

### 5.2.2 Resultados

Não existe fórmula analítica para a precificação deste tipo de instrumento. É necessário determinar novamente um preço “justo”. Para isto recorremos à simulação Monte Carlo. Efetuamos uma simulação Monte Carlo de 1 milhão de trajetórias para cada opção. Os preços obtidos são comparados em seguida com as estimativas oferecidas pelo método tradicional e pelo método quase Monte Carlo apresentado neste trabalho. Utilizamos novamente o mesmo critério de erro que no caso das opções com barreira. Reproduzimos na tabela 5.4 os erros obtidos com os dois métodos.

Em termos gerais o método tradicional apresentou uma estimativa de pior qualidade em exatos 2/3 dos casos. De todos os experimentos - incluindo o próximo - este foi o menos desfavorável à simulação Monte Carlo. Os erros obtidos através dos dois métodos são claramente inaceitáveis para simulações de tamanho 4096. Estranhamente para este tamanho de simulação

d	Método	$N = 2^{12}$	$N = 2^{13}$	$N = 2^{14}$	$N = 2^{15}$	$N = 2^{16}$	$N = 2^{17}$
250	SMC	-5.32	-0.43	-0.76	0.65	-1.36	-2.03
	SQMC	-3.06	-0.28	0.57	0.71	0.07	-0.03
500	SMC	2.21	-0.70	0.40	-1.44	-0.24	0.20
	SQMC	-2.50	-0.75	-0.04	0.78	-0.21	-0.17
750	SMC	5.30	-0.91	-0.88	0.11	0.65	-0.24
	SQMC	-1.80	-1.19	-0.24	0.47	-0.53	-0.38
1000	SMC	-4.21	0.78	1.85	-0.31	-0.10	-0.10
	SQMC	-0.95	-0.62	0.07	0.55	-0.22	-0.05
1250	SMC	1.26	-3.58	-1.45	-1.43	1.30	-0.36
	SQMC	-0.99	-0.60	-0.24	0.10	-0.44	-0.25
1500	SMC	0.86	1.88	0.35	0.65	0.27	-0.64
	SQMC	-1.08	-0.68	-0.48	-0.12	-0.55	-0.35
1750	SMC	-0.34	0.82	-0.50	-0.88	0.12	0.03
	SQMC	-0.45	-0.31	-0.25	0.11	-0.20	-0.08
2000	SMC	0.16	-1.76	0.18	-0.20	-1.36	0.22
	SQMC	-0.48	-0.33	-0.35	-0.09	-0.29	-0.22

Tabela 5.4: Opções basket asiáticas: resultados

evidencia-se uma melhora em ambos os casos conforme a dimensão do problema aumenta. Este resultado não é esperado para a simulação quase Monte Carlo. Esperava-se que, no melhor dos casos, as ordens de grandeza dos erros permanecesse constante. É o que de fato parece acontecer para as simulações de tamanho superior. Com exceção de um resultado, todas as estimativas geradas pela simulação quase Monte Carlo que apresentam erro acima de 1% são obtidas com  $N = 4096$ . A exceção é um erro de 1,19% para a opção de dimensão 750 com  $N = 8192$ . Em contraste, a simulação Monte Carlo gera erros maiores que 1% para todos os tamanhos de simulação, chegando a apresentar um erro de 2,03% para a opção de dimensão 250 quando  $N = 131072$ . Em ambos os casos os erros como um todo diminuem com o tamanho da simulação embora não de forma monotônica. Como todos os erros possuem média nula calculamos o desvio padrão amostral de ambos os métodos. Estes são de 0,69% e 1,63% respectivamente para a simulação quase Monte Carlo e o método tradicional. Naturalmente este número tem que ser manuseado com cuidado pois mistura várias opções e vários tamanhos de simulação. Porém, trata-se de outro indício de que a sequência de Sobol parece se comportar de forma menos errática.

No que tange ao objetivo principal do nosso trabalho, a extensão do uso de seqüências de Sobol a dimensões muito superiores às apresentadas na literatura até este momento, constatamos que novamente o método não piorou conforme aumenta a dimensão do problema. Vemos, por exemplo, que para simulações de dimensão 1750 e 2000 obtemos erros sempre inferiores a 0,50%. As médias dos erros obtidos para todos os tamanhos de simulação são respectivamente de 0,20% e 0,29%. Os resultados são muito encorajadores pois qualquer tentativa de gerar estimativas com números escolhidos arbitrariamente gera erros descomunais. Por exemplo, uma tentativa de precificar uma opção que depende de 225 observações (de dimensão 2250), com números direcionais obtidos até a dimensão 2000 de acordo com o nosso algoritmo acrescidos de números escolhidos arbitrariamente para os 250 polinômios seguintes, apresenta erros na vizinhança de 50,00%! Por si só o bom comportamento da simulação quase Monte Carlo para as dimensões muito grandes já representa um belo avanço. É claro que existe a possibilidade da dimensão efetiva ser muito inferior à dimensão nominal, porém a discrepância entre resultados obtidos entre simulações quase Monte Carlo que usam ou não o algoritmo de escolha dos direcionais parece apontar para outra direção. De qualquer modo implementamos no próximo exemplo uma simulação para determinar a dimensão efetiva do problema sob consideração.

## 5.3 Opção asiática

### 5.3.1 O produto

Um dos exercícios clássicos para testes de métodos que visam uma melhora em relação ao método tradicional é a opção asiática. Trata-se de um instrumento que depende de uma média de preços observados. Quando a média é geométrica existe uma solução analítica para o preço da opção. Quando trata-se de outro tipo de média requer-se então métodos numéricos. Para o caso da média aritmética o procedimento mais comum para obtenção do preço é a simulação Monte Carlo. Este será o instrumento fictício que iremos precificar. Escolhemos de forma totalmente livre os parâmetros de uma opção de compra européia. São estes,  $S = 100$ ,  $r = 20\%$ ,  $\sigma = 20\%$ ,  $X = 100$ , onde  $S$  representa o preço do ativo subjacente no momento em que a opção é precificada,  $r$  a taxa de juros livre de risco continuamente composta até o vencimento da opção,  $\sigma$  a volatilidade do subjacente e  $X$  o preço de exercício da opção. O tempo até o vencimento é variável. Mantemos fixo o intervalo entre dois preços em  $dt = 1/252$ , como se estivéssemos

observando o preço do ativo subjacente ao final de cada dia. O tempo até o vencimento da opção é dado pelo número de preços que compõem a média vezes o intervalo de tempo. Naturalmente a dimensão da simulação é dada pelo número de preços que compõem a média. O payoff  $\Pi$  desta opção corresponde a,

$$\Pi = \left( \frac{1}{n} \sum_{t=1}^T S_t - X, 0 \right)^+ \quad (5.3)$$

A precificação requer portanto que sejam simuladas um grande número de trajetórias. Para cada uma delas calcula-se o payoff da opção, gera-se a média do preço destas opções e traz-se esta média a valor presente. Decidimos testar opções de dimensão nominal variando de 250 a 2500. O preço esperado é obtido através de uma simulação Monte Carlo de tamanho 6 milhões. Antes de prosseguirmos com os resultados vejamos primeiro como se comporta a dimensão efetiva para este exercício.

### 5.3.2 A dimensão efetiva

Para o cálculo da dimensão efetiva utilizamos o procedimento descrito na seção anterior baseado no resultado de Sobol [2001]. Para isso precisamos calcular, seguindo a notação já estabelecida, várias vezes o termo  $T_u$ . A integral que fornece o preço da opção  $c$  é da forma,

$$c = \int_{[0,1]^d} \exp^{-rT} \left( 0, \frac{1}{d} \sum_{j=1}^d S \exp \left[ \left( r - \frac{\sigma^2}{2} \right) t_j + \sigma \sqrt{t_j} \sum_{i=1}^j \phi^{-1}(x_i) \right] - K \right) dx \quad (5.4)$$

onde  $\phi^{-1}$  é a inversa da distribuição normal padrão. Escolhemos sempre o mesmo conjunto inicial  $u = \{1, \dots, l\}$  com  $l = 50$ . Incrementamos  $l$  de um em um até atingir a dimensão nominal. Cada  $T_u$  é obtido através de uma simulação Monte Carlo de tamanho  $n = 10.000$  o que gera uma variância amostral não desprezível. A variância do integrando é obtida através de uma simulação Monte Carlo de tamanho 5 milhões. Pode-se observar na figura 5.1 a trajetória de  $T_u$  para os problemas de dimensão 750, 1250, 1750 e 2250. É razoável avançar que existe um padrão para o exercício em questão. A contribuição de  $T_u$  aumenta muito rapidamente até determinado patamar e parece se estabilizar num nível próximo à variância da função para uma dimensão em torno de 2/3 ou 3/4 da dimensão nominal. A variância da função corresponde à reta mais espessa (e vermelha) em cada um dos gráficos.

Escolhemos o parâmetro mais utilizado na literatura que trata de dimensão efetiva,  $p = 99\%$ . Diante da variância amostral é difícil dizer de fato quando o termo  $T_u$  ultrapassa 99% da variância. Para isso estipulamos a seguinte regra. Efetua-se uma média móvel de tamanho

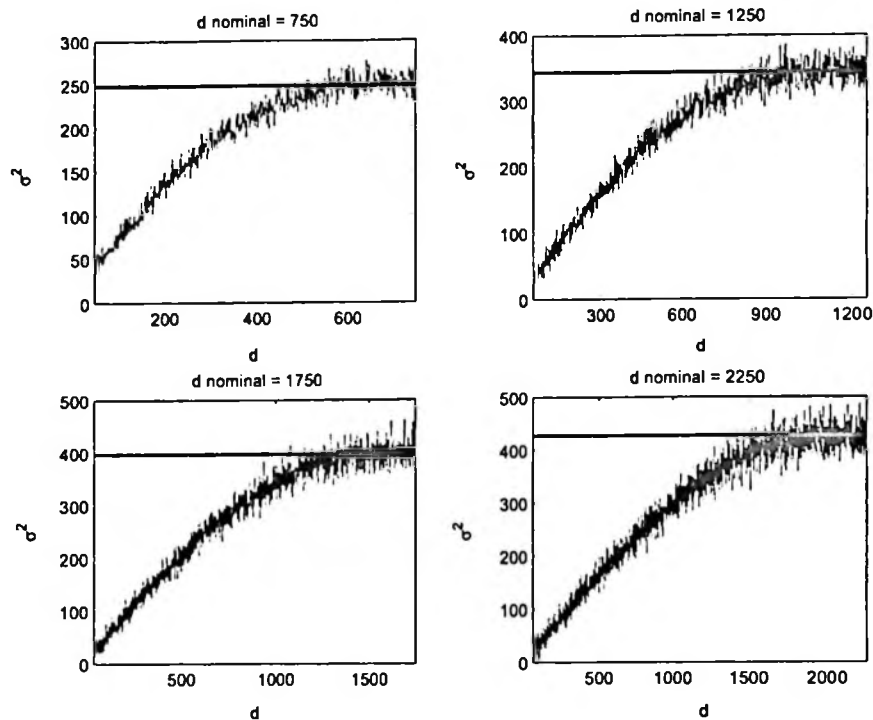


Figura 5.1: Opção asiática: dimensão efetiva  $d$

10. Quando esta média ultrapassa pela primeira vez o patamar crítico então toma-se o maior membro do primeiro conjunto da série (no sentido do  $u$  contido nos demais conjuntos) como dimensão efetiva. Obtemos a tabela 5.3.

Nela vê-se a dimensão nominal,  $DN$ , a variância do integrando  $\sigma^2(f)$ , o nível crítico correspondente a 99% da variância, a dimensão efetiva  $DE$  e a razão da dimensão efetiva sobre a dimensão nominal. De acordo com a nossa regra a dimensão efetiva parece corresponder a um pouco mais do que 75% da dimensão nominal. Não parece haver nenhum padrão de convergência para um determinado piso. Ao contrário, tudo indica que este percentual pode ser válido para qualquer dimensão. A maior dimensão efetiva que será testada corresponde a pouco menos que 2000. Em si isto já representa um grande avanço. Lembramos que na literatura a opção asiática de maior dimensão nominal é de 250, em Boyle e Tan [2000]. Vejamos agora como a simulação quase Monte Carlo se comporta em frente ao método tradicional para o qual utiliza-se o gerador RAN2 de Press et al. [1992], adotado em grande parte dos trabalhos em finanças que tratam de simulação quase Monte Carlo.

DN	$\sigma^2(f)$	$p \times \sigma^2(f)$	DE	DE / DN
250	88.93	88.04	218	87.20%
500	178.25	176.47	382	76.40%
750	250.96	248.45	576	76.80%
1000	305.58	302.52	762	76.20%
1250	347.92	344.44	947	75.76%
1500	379.11	375.31	1107	73.80%
1750	401.06	397.05	1361	77.77%
2000	414.41	410.27	1535	76.75%
2250	431.59	427.27	1788	79.47%
2500	434.79	430.44	1952	78.08%

Tabela 5.5: Dimensão efetiva para opção asiática

### 5.3.3 Resultados

Não existe uma fórmula analítica para o preço de uma opção de compra asiática que depende da média aritmética. É portanto necessário estipular um “preço justo” para a opção. Para isto utilizamos o método tradicional. Para cada dimensão efetuamos seis simulações Monte Carlo de tamanho 1 milhão. Trata-se portanto do equivalente a uma simulação Monte Carlo de tamanho seis milhões. Ao efetuar cada uma das seis simulações calculamos os preços que seriam obtidos com simulações de tamanho  $2^j$  com  $j = 12, 13, \dots, 17$ . Formamos assim uma matriz  $6 \times 6$ , seis simulações e seis tamanhos. Para que possamos comparar o método proposto com uma simulação Monte Carlo diferente, escolhemos os preços que constam na diagonal desta matriz.

Mostramos agora a comparação entre os dois métodos. Apresentamos agora os resultados em formato gráfico na figura 5.2. É possível assim obter-se uma idéia da convergência dos métodos.

Para facilitar a leitura omitimos o tamanho da simulação. Salvo nos casos das dimensões nominais 2250 e 2500 começamos com simulações de tamanho  $2^{12}$ . Cada instância de marcador corresponde a um tamanho diferente, equivalente a duas vezes o último. Para as simulações de dimensão nominal 2250 e 2500 o menor tamanho é de  $2^{13}$ . Mostramos a dimensão efetiva acima de cada pequeno gráfico. A linha preta mais espessa corresponde ao “preço justo” mencionado acima. Os resultados obtidos com a metodologia proposta são representados pela linha azul



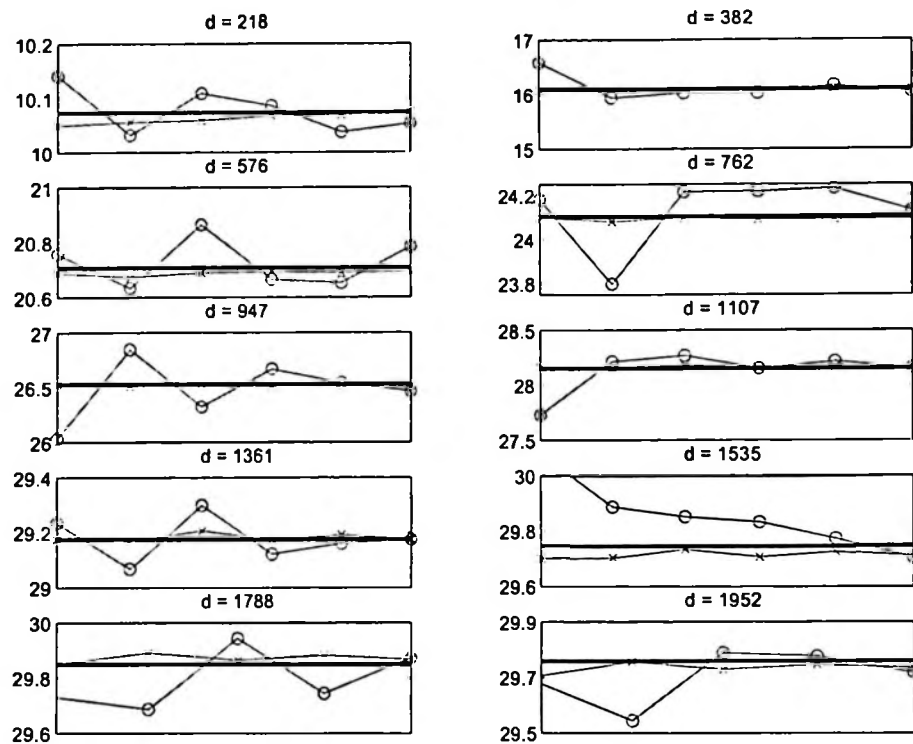


Figura 5.2: Opção asiática: comparação entre o método de Monte Carlo - círculo - e quase Monte Carlo - cruz - com algoritmo proposto ( $d$  = dimensão efetiva)

com marcador em forma de cruz. Os resultados oriundos da simulação Monte Carlo são representados em verde com marcadores em forma de círculo.

Não há dúvidas quanto à superioridade das seqüências de Sobol tanto no que tange à precisão como em relação à variação da estimativa. Para todas as dimensões efetivas o método adere muito mais ao preço esperado. Para dimensões efetivas comparáveis com o teste da seção anterior a diferença é avassaladora pois quase Monte Carlo mostra uma convergência muito rápida e precisa. Em compensação o método tradicional oscila bastante. Quando a dimensão efetiva aumenta e passa dos níveis testados anteriormente o método quase Monte Carlo continua plenamente operacional. Ele mostra-se ainda superior ao método de Monte Carlo porém a diferença de performance parece se atenuar um pouco. Em todas as simulações o método tradicional só conseguiu erro de aproximação menor em uma única vez, para simulação de tamanho  $2^{15}$  de dimensão nominal 2500. Em suma, mostramos agora que além de ser possível estender o uso das seqüências de Sobol para dimensões muito altas os resultados obtidos mostram-se nitidamente superiores aos obtidos pelo método tradicional.



# Capítulo 6

## Conclusão

Após apresentarmos as bases da simulação quase Monte Carlo e suas principais características expusemos uma metodologia para construção de seqüências de Sobol que permite o tratamento de problemas de dimensão muito alta. O método é baseado numa configuração específica dos bits dos números direcionais que gera projeções bem comportadas nos hiperplanos. A implementação do método é obtida através da geração aleatória de números direcionais e verificação da aderência destes aos critérios necessários para atender as propriedades almeçadas. Com a ferramenta em mãos submetemos o novo algoritmo a testes que visavam medir o seu desempenho em situações reais.

Para que as comparações fossem desprovidas de viés incorporamos à nossa análise, quando possível, o conceito de dimensão efetiva, que oferece um tipo de denominador comum para os diversos testes possíveis. A exposição sistemática da dimensão efetiva de um produto financeiro ao testar um método numérico é novidade na literatura. Os exercícios anteriores utilizavam o conceito de dimensão nominal, o que acabava gerando resultados paradoxais. Um algoritmo para cálculo da dimensão efetiva de opções asiáticas foi desenvolvido.

Enfim, a eficácia do método proposto foi averiguada em quatro aplicações. A primeira consiste numa integral de teste já utilizada em outras pesquisas. O novo método mostrou-se superior aos melhores resultados obtidos até agora, expostos em Joe e Kuo [2003]. A aplicação a problemas de finanças tinha um duplo objetivo: mostrar que é possível utilizar a simulação quase Monte Carlo em altas dimensões e testar o seu desempenho em relação à simulação Monte Carlo. Ambos foram atingidos. Em primeiro lugar demonstramos que as situações típicas - o de opções cujo pagamento final depende da trajetória do ativo subjacente - encontradas em finanças

são um terreno fértil para as seqüências de Sobol, mesmo em dimensões efetivas muito grandes. Os nossos testes se limitaram subjetivamente à marca de 2000 mas não se deteriora quando lidamos com problemas de dimensão superiores. Em segundo lugar mostramos que, de modo geral, a construção proposta consegue lidar com dimensões altas de modo mais eficiente que o método Monte Carlo. Essa superioridade é sistemática e não parece ser função do produto tratado.

Enquanto perseguíamos o nosso objetivo surgiram novas direções e eventuais melhoras que merecem consideração. Em primeiro lugar um mapeamento rigoroso da dimensão efetiva de cada produto financeiro seria de grande auxílio aos usuários de simulação quase Monte Carlo. A escolha do método e do tamanho da simulação dependem desta informação. Tê-la em mente pode resultar em grandes ganhos de eficiência. A determinação da dimensão no sentido da sobreposição para aplicações financeiras é um campo totalmente virgem. As exigências computacionais necessárias ao cálculo da dimensão no sentido da sobreposição não permitem uma investigação sistemática de tal. Diante da tecnologia atual são necessários resultados analíticos que permitam a simplificação do cálculo da dimensão. Existe ainda a possibilidade da dimensão no sentido da sobreposição dos problemas utilizados ser baixa, o que teria como consequência diminuir o alcance dos resultados sobre simulação quase Monte Carlo.

No que tange ao algoritmo proposto seria interessante descobrir se existe uma maneira de escolher os direcionais sem que seja necessário gerá-los de forma aleatória. Um estudo sistemático dos números que atendem aos critérios estipulados pode desencobrir padrões que ajudariam em muito no que diz respeito à implementação do método. Não investigamos também os limites de uso de cada critério em relação à dimensão, para cada tamanho de simulação. Isso pode representar um ganho substancial de tempo caso o usuário opte por não utilizar números direcionais previamente gerados. Uma investigação dos ganhos marginais obtidos passando-se de um critério de refinamento ao critério imediatamente superior pode também ser uma informação valiosa para o usuário. Finalmente, testes com critérios mais fortes de refinamento  $t = 5, 6, \dots$  devem ressaltar mais ainda a eficácia do método proposto em relação à simulação Monte Carlo. Concluimos com a esperança que os resultados obtidos ao cabo deste trabalho incitem alguns participantes do mercado financeiro a pensar seriamente na alternativa quase Monte Carlo.

## Referências

- [1996] P. A. Ackworth. M. Broadie and P. Glasserman. A comparison of some Monte Carlo and quasi-Monte Carlo techniques for option pricing. *Monte Carlo and quasi-Monte Carlo Methods. Lecture Notes in Statistics* 127 (Springer-Verlag, 1996).
- [2001] T. Barbe. Sequências quase aleatórias: aplicações em finanças. *Dissertação de mestrado FEA - USP*. (2001).
- [1996] R. Birge. Quasi-Monte Carlo approaches to option pricing. *University of Michigan Working Paper*. (1996).
- [1997] P.P. Boyle. M. Broadie and P. Glasserman. Monte Carlo methods for security pricing. *Journal of Economic Dynamics and Control* 21 (1997).
- [1997] R. Caflisch. W. Morokoff and A. Owen. Valuation of mortgage-backed securities using brownian bridges to reduce effective dimension, in: B. Dupire ed, *Monte Carlo Simulation in Finance* (Risk Publications, London, 1999) 301–314.
- [2000] J. Cheng and M. J. Druzdzel. Computational investigation of low-discrepancy sequences in simulation algorithms for Bayesian networks. *Proceedings of the 16th Annual Conference on Uncertainty in Artificial Intelligence* (Morgan-Kaufmann 2000).
- [1976] R. Cranley and T.N.L. Patterson. Randomizations of number theoretic methods for multiple integration. *Siam Journal on Numerical Analysis* 13 (1976) 904–914.
- [2002] K. Dowd. *Measuring Market Risk* (Wiley, 2002).
- [1982] H. Faure. Discrepances de suites associées à un système de numération (en dimension  $s$ ), *Acta Arithmetica* 41 (1982) 337–351.

- [1997] S. Galanti and A. Jung. Low-discrepancy sequences: Monte Carlo simulation of option prices. *The Journal of Derivatives* (Fall 1997) 63–83.
- [2004] P. Glasserman. *Monte Carlo Methods in Financial Engineering* (Springer, 2004).
- [1983] G.H. Golub and C.F. Van Loan. *Matrix Computations* (John Hopkins University Press, Baltimore, 1983).
- [2000] J. Hull. *Options, Futures and Other Derivative Securities* (Prentice-Hall, New York, 2000).
- [2002] P. Jäckel. *Monte Carlo Methods in Finance* (Wiley, New York, 2002).
- [2003] S. Joe and F. Kuo. Remark on Algorithm 659: Implementing Sobol’s Quasirandom Sequence Generator. *ACM Transactions on Mathematical Software* 29 (March 2003) 49–57.
- [2001] P. Jorion. *Value at Risk, the New Benchmark for Managing Financial Risk* (McGraw-Hill, New York, 2001).
- [1996] C. Joy. P.P. Boyle and K.S. Tan. Quasi-Monte Carlo methods in numerical finance. *Management Science* 42 (1996) 926–938.
- [1990] A.G.Z. Kemna and A.C.F. Vorst. A pricing method for options based on average asset values. *Journal of Banking and Finance* 14 (1990) 113–129.
- [1991] C. Lécot. A Quasi-Monte Carlo method for the Boltzman equation. *Mathematics of Computation* 56 (1991) 621–644.
- [1993] W. Morokoff and R. Caflisch. A Quasi-Monte Carlo approach to particle simulation of the heat equation. *Siam Journal on Numerical Analysis* 30 (1991) 1558–1573.
- [1994] W. Morokoff and R. Caflisch. Quasi-random sequences and their discrepancies. *Siam J. Sci. Stat. Computing* 15 (1994) 1251–1279.
- [1995] W. Morokoff and R. Caflisch. Quasi-Monte Carlo integration. *J. Comp. Phys.* 122 (1995) 218–230.
- [1996] B. Moskowitz and R. Caflisch. Smoothness and dimension reduction in quasi-Monte Carlo methods. *Mathl. Comput. Modelling* 23 (1996) 37–54.

- [1992] H. Niederreiter. *Random number generation and quasi-Monte Carlo methods*. (SIAM 1992).
- [1974] H. Niederreiter and L. Kuipers. *Uniform distribution of sequences*. (John Wiley & Sons 1974).
- [1995] H. Niederreiter and C.P. Xing. Low-discrepancy sequences obtained from algebraic function fields over finite fields. *Acta Arith.* , **72** (1995) 281–298.
- [1996] H. Niederreiter and C.P. Xing. Low-discrepancy sequences and global function fields with many rational places. *Finite Fields Applications* , **2** (1996) 241–273.
- [1994] A. Owen. Lattice sampling revisited: Monte Carlo variance of means over randomized orthogonal arrays. *The Annals of Statistics* **22** (1994) 930–945.
- [1997] A. Owen. Monte Carlo variance of scrambled net quadrature. *Siam Journal on Numerical Analysis* **34** (1994) 1884–1910.
- [2001] A. Owen. The dimension distribution and quadrature test functions. Technical Report, Stanford University, 2001.
- [1999] A. Papageorgiou and S. Paskov. Deterministic simulation for risk management. *The Journal of Portfolio Management* (May 1999) 122–127.
- [1995] S.H. Paskov and J.F. Traub. Faster valuation of financial derivatives. *The Journal of Portfolio Management* (Fall 1995).
- [1992] W. Press. S. Teukolsky. T.V. William and P.F. Brian. *Numerical Recipes in C, 2nd edition* , Cambridge University Press, Cambridge (1992).
- [2002] M.E. Silva. Simulação de quase Monte Carlo em finanças: quebrando a maldição da dimensionalidade. *Resenha da BM&F*, **151**, (Sep-Oct 2002).
- [2005] M.E.Silva e T. Barbe. Quasi-Monte Carlo in finance: extending for problems of high effective dimension (a ser publicado). *Economia Aplicada*.
- [1967] I.M.Sobol. On the distribution of points in a cube and the approximate evaluation of integrals. *USSR computational mathematics and mathematical physics*. **7** 86–112 (1967).

- [1976] I.M. Sobol. Uniformly distributed sequences with additional uniform properties. *USSR Computational Mathematics and Mathematical Physics* **16** (1976)
- [1993] I. M. Sobol. Sensitivity estimates for nonlinear mathematical models. *Math. Modeling Comput. Exp.* **1** (1993) 407–414.
- [1993] I. M. Sobol. *A primer for the Monte Carlo method* (CRC Press, 1994).
- [2001] I. M. Sobol. Global sensitivity indices for nonlinear mathematical models and their Monte Carlo estimates. *Math. Modeling Comput. Exp.* **55** (2001) 271–280.
- [2000] K.S. Tan and P.P. Boyle. Applications of randomized low discrepancy sequences o the valuation of complex securities. *Journal of Economic Dynamics and Control* **24** (2000) 1747–1782.
- [2003] X. Wang and K.T. Fang. The effective dimension and quasi Monte Carlo integration. *Journal of Complexity* **19** (2003) 101–124.
- [1916] H. Weyl. Über die gleichverteilung von zahlen mod eins. *Math. Ann.* **77** (1916) 313–352.