

2. Técnicas de Integração

2.1. Integração por diferença finita

Em muitos casos de interesse não sabemos como obter analiticamente a integral de uma função. Duas soluções possíveis são: procurar uma aproximação analítica ou recorrer a alguma técnica de integração numérica. Técnicas analíticas, quando acompanhadas de algum teorema que lhes deem suporte, são sempre preferíveis. Nos casos em que mesmo alguma aproximação analítica não é possível somos obrigados a recorrer às técnicas numéricas. Mesmo neste caso não existe um método geral e eficiente. Por exemplo, se a função varia suavemente ela pode ser integrada por métodos relativamente simples. Por outro lado, funções que apresentam grandes variações, como máximos e mínimos acentuados, são um desafio para serem avaliadas numericamente. Para o propósito deste curso será suficiente discutirmos apenas os casos mais simples. A integração, compreendida como o cálculo da área sob uma curva, pode ser feita dividindo-se a área total em polígonos para os quais calculamos as áreas individuais dos polígonos e ao final do cálculo os somamos para obter a área total desejada [Carnahan]. Este é o bem conhecido método dos trapézios exemplificado na figura 1. Este método simples é eficiente para a estimativa de integrais de funções lisas em uma ou duas dimensões, contudo torna-se ineficiente para funções menos bem comportadas ou em dimensões mais altas.

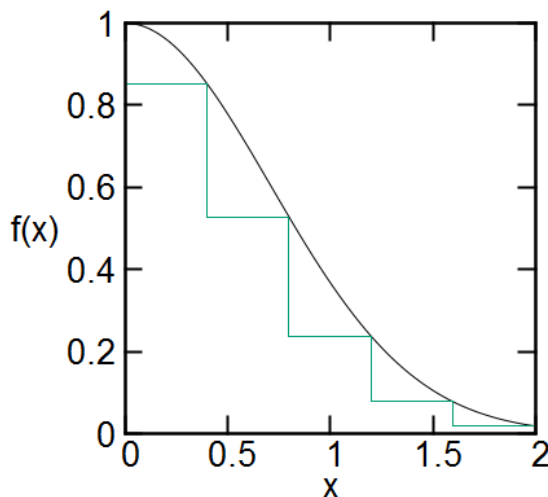


Figura 1.1a Esquema para o cálculo da integral de $f(x)$ usando a soma de retângulos.

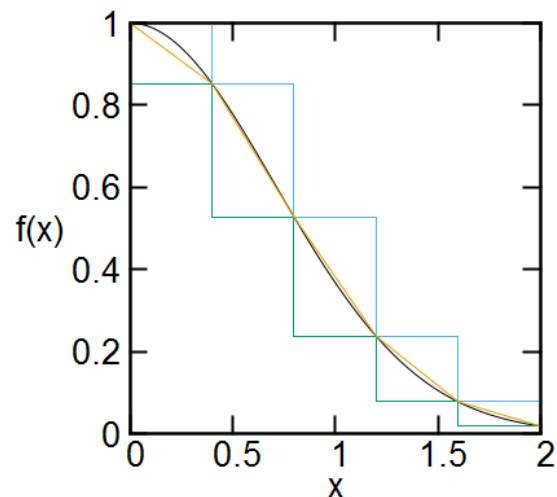


Figura 1.1b Esquema para o cálculo da integral usando os valores médios de dois retângulos

Nas figuras 1 a e b acima mostramos como é o esquema para o cálculo da integral de $f(x)$ usando retângulos. Na primeira figura selecionamos N pontos sobre a curva. Neste caso $N = 6$. Os pontos foram tomados igualmente espaçados, mas isto não é necessário. Os vértices de cada retângulo então nos pontos: $(x_0, f(x_0))$,

$(x_0 + \Delta x, f(x_0 + \Delta x)), (x_0 + 2\Delta x, f(x_0 + 2\Delta x)), \dots$ e assim por diante. Isto é mostrado como uma curva azul na figura. Uma aproximação para a área sob a curva é a soma das áreas dos retângulos. O cálculo é tanto melhor quanto menor for Δx . Uma estimativa melhor consiste em tomar a média entre os retângulos "superiores" e "inferiores" como mostrado na figura 1.b. Observe que isto é o mesmo que aproximar a curva por segmentos de reta que passam pelos pontos $(x_0 + (n-1)\Delta x, f(x_0 + (n-1)\Delta x)), (x_0 + n\Delta x, f(x_0 + n\Delta x)) \dots$, e calcular a área abaixo de cada segmento de reta. Esta aproximação pode ser melhorada ainda mais se tomarmos curvas mais complexas para aproximar segmentos de $f(x)$. Por exemplo, uma parábola $ax^2 + bx + c$. Neste caso precisamos de três pontos dentro do segmento para obter a, b e c .

Um algoritmo simples para escrever o código poderia ser:

Algoritmo 2.1

```

1 - Defina os limites de integração de  $f(x)$ ,  $(x_0, x_N)$ 
2 - Defina o tamanho da partição  $\Delta x = \frac{x_N - x_0}{N}$ 
3 - Faça  $Integral = 0$ 
4 - Loop
5 -  $Integral = Integral + \Delta x f(x_0 + n\Delta x)$ 
6 - Fim do Loop

```

Exercícios - 2.1

1. Usando os esquemas descritos acima, calcule a integral da função $f(x) = \sqrt{1-x^2}$ no intervalo $[0,1]$ para $f(x) > 0$. O resultado deve ser $\pi/4$.
2. Repita o problema acima usando o esquema da figura 1.b.
3. Faça um programa e execute, usando uma aproximação de parábola para calcular a área de um quarto do disco de raio 1.
4. Compare os resultados obtidos em 1, 2 e 3. Qual deles dá uma melhor aproximação para um mesmo número de partições? Compare o tempo gasto usando cada um dos métodos.

Uma pergunta importante agora é: Qual o erro que cometemos ao usar as técnicas de integração descritas acima. Para responder esta pergunta vamos começar escrevendo a definição:

$$\int_{x_0}^{x_N} f(x) dx = \lim_{N \rightarrow \infty} \sum_{n=0}^{N-1} \Delta x f(x_n) \quad \text{Eq. 2.1}$$

Observe que $f(x)$ é uma função lisa ela pode ser escrita como sua série de Taylor em torno do ponto x_n , isto é:

$$f(x + \Delta x) = f(x_n) + \Delta x f'(x_n) + \Delta x^2 f''(x_n) + \dots \quad \text{Eq. 2.2}$$

Ou

$$f(x - \Delta x) = f(x_n) - \Delta x f'(x_n) + \Delta x^2 f''(x_n) + \dots \quad \text{Eq. 2.3}$$

Se somarmos estas duas equações teremos

$$f(x + \Delta x) + f(x - \Delta x) = f(x_n) = \frac{1}{2} [f(x + \Delta x) + f(x - \Delta x)] \quad \text{Eq. 2.4}$$

Substituindo na expressão na equação (1)

$$\int_{x_0}^{x_N} f(x) dx = \lim_{N \rightarrow \infty} \sum_{n=0}^{N-1} \Delta x [f(x + \Delta x) + f(x - \Delta x)] \quad \text{Eq. 2.5}$$

Tomando N muito grande e “esquecendo” o limite, temos uma aproximação para a integral. Observe que o próximo termo que teríamos de incluir no cálculo da integral seria de ordem Δx^2 . Assim erro cometido é de ordem Δx^2 .

Exercício - 2.2

Usando os cálculos feitos nos Exercícios - 1, observe este comportamento em seus cálculos.

2.2. Integração-Extensão para dimensões mais altas

Imediatamente pensamos em como estender esta técnica para dimensões mais altas. Isto pode ser feito de maneira direta para calcular o hipervolume debaixo de uma superfície. Temos que lembrar que ao invés de um segmento de reta ligando pontos, teremos agora hiperplanos ligando os segmentos de hipercurvas. As figuras 1.2a e 1.2b mostram o esquema em 2 dimensões.

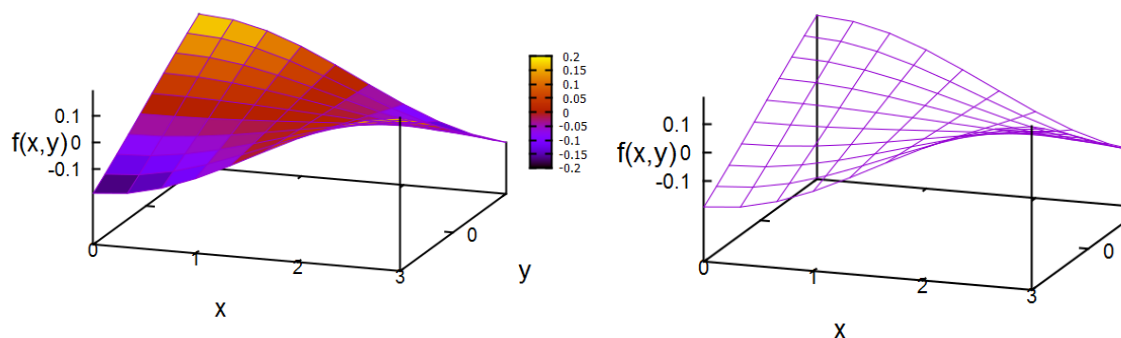


Figura 1.2a A figura mostra uma superfície sob a qual desejamos calcular o volume. O procedimento é similar àquele feito para o cálculo da área sob a curva.

Figura 1.2b O procedimento é aproximar cada secção da superfície por um plano definido pelos seus quatro vértices. Calcula-se o volume sob cada plano. A soma de todos esses pequenos volumes dá uma estimativa do volume sob a superfície.

Exercícios - 2.3

1. Calcule o volume sob a curva $e^{0.1(x^2+y^2)}$ usando o esquema de partições.
2. Mostre que o erro cometido é de segunda ordem em $\Delta x, \Delta y$.

O método de integração discutido acima e suas variantes é bastante conveniente e eficiente em um grande número de casos. No entanto, deve-se tomar muito cuidado ao empregá-lo de forma indiscriminada. Existem situações patológicas quando a integranda tem um polo ou um ponto de mínimo (ou máximo) muito agudo (comum em física de partículas), que resistem bravamente ao método. Para estes casos foram desenvolvidos métodos específicos muito eficientes que não discutiremos aqui [Poles]. Em outras situações em que é necessário fazer integrais em dimensões mais altas o método também se torna ineficiente. As integrações usando partições convergem proporcionalmente ao número de partições, tornando inviável seu uso em integrações em mais que 3 dimensões devido ao longo tempo computacional. Nestas situações o método de Monte Carlo, que discutiremos a seguir, é muito mais eficiente, como veremos a seguir.

2.3. Integração por Monte Carlo

O que é o método de Monte Carlo [MonteCarlo]?

O que chamamos de Monte Carlo não é um método específico, pronto para uso em qualquer circunstância. De fato, este é um nome comum para muitas técnicas baseadas em amostragem e física estatística, para investigar problemas tão variados e diversos quanto em ciência dos materiais, biofísica, física estatística, economia e muitos outros. Voltaremos a este assunto nos próximos tópicos, agora, vamos usá-lo como uma técnica simples e poderosa para o cálculo de integrais.

A primeira pergunta que vem à mente é: Se os métodos *analíticos* de integração, como discutidos anteriormente funcionam tão bem, por que desenvolver outro baseado em amostragem estatística?

Considere os métodos discutidos acima. Como vimos o erro cometido usando retângulos é proporcional a $\Delta x \sim 1/N$, onde N é o número de partições. Se usarmos trapézios o erro será proporcional a Δx^2 . Métodos mais sofisticados podem levar a erros ainda menores, porém a complexidade aumenta geometricamente. Se a função a ser integrada é em d dimensões, teremos que fazer $N^{\frac{1}{d}}$ partições em cada direção de integração, o que nos deixa com um erro proporcional a $1/N^{\frac{k}{d}}$, onde k é a precisão do método (1 ou 2 no caso presente). Em um método estocástico, como é Monte Carlo, o teorema central do limite nos garante que o erro é proporcional a $\frac{1}{\sqrt{N}}$, independente de d ! Comparando estas duas fórmulas vemos que o método de Monte

Carlo começa a ser competitivo em $N \sim 4$. Em geral, uma simulação de um sistema físico trata com sistemas onde $d > 10^6$!

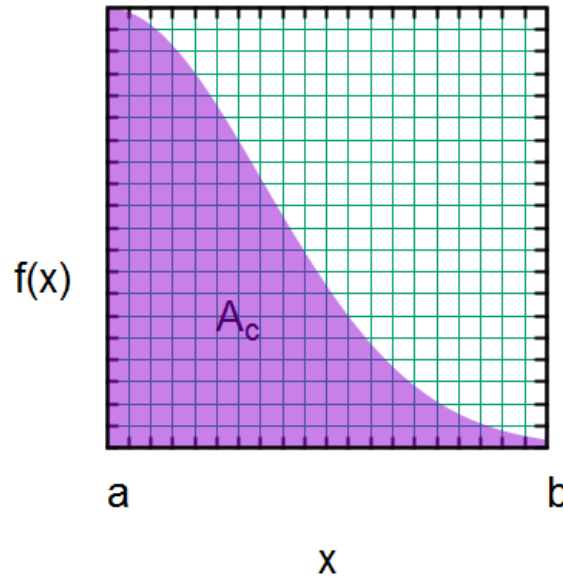


Figura 1. Erro! Nenhum texto com o estilo especificado foi encontrado no documento. **3** A área que procuramos calcular é definida abaixo da curva $f(x)$, A_c .

Para discutir o método considere o cálculo da área (integral) sob a curva da figura Figura 1.3.

$$A_c = \int_a^b f(x) dx \quad \text{Eq. 2.6}$$

A área que procuramos calcular é A_c e a área total do quadrado que envolve a figura é A . Se dividirmos o sistema em pequenos quadrados, como mostrado, aproximações para A_c e A são dadas pelo número de quadrados N_c e N respectivamente, vezes a área de cada quadrado. Isto é

$$\frac{A_c}{A} \doteq \frac{N_c}{N} \quad \text{Eq. 2.7}$$

Dividindo membro a membro obtemos

$$A_c \doteq \frac{N_c}{N} A, \quad \text{Eq. 2.8}$$

Onde A é a área, conhecida, do quadrado que contém a curva. Nosso problema agora é estimar N e N_c . Para fazer isto vamos usar um método estocástico. A ideia é gerar aleatoriamente um conjunto de pontos $\{(x_1, y_1), \dots, (x_n, y_n), \dots, (x_N, y_N)\}$ dentro do intervalo $\{a \leq x \leq b\}$, $\{f(a) \leq y \leq f(b)\}$, distribuídos uniformemente no intervalo de integração. Contando o número de pontos N e N_c , e ao final usando a equação Eq. 2.8 teremos uma estimativa de A_c . Linguagens como Python, C++ ou FORTRAN têm um gerador de números pseudoaleatórios intrínseco, que gera sequências não correlacionadas no intervalo $(0,1)$. Chamamos de pseudoaleatórios pois estas sequências têm um

período (A sequência de números se repete). Um gerador é tão melhor quanto menos correlacionada e maior é a sequência que fornece. Bons geradores podem ter sequências da ordem de 10^{100} antes de se repetirem. Os disponíveis naquelas linguagens têm uma sequência bem menor, mas isto não será um problema neste estágio do curso. Tendo em mãos nosso gerador temos que adaptá-lo aos intervalos definidos no problema. Isto é feito de forma simples. Suponha que o aleatório gerado seja r_x . Basta fazer um deslocamento do zero: $r'_x = a + r_x$, este novo número estará no intervalo $(a, a + 1)$, agora fazemos uma reescala: $X = a + (b - a)r_x$ de modo que quando $r_x = 0, X = a$ e $r_x = 1, X = b$. Com expediente similar obtemos $Y = f(a) + (f(b) - f(a))r_y$.

O que devemos fazer é gerar N duplas (r_x, r_y) , contar o número que cai abaixo da curva e ao final usar a equação Eq. 2.8 para estimar a integral.

Um algoritmo simples seria:

Algoritmo 2.1

```

Defina N
Faça N_c = 0
Loop N
   $r_x = \text{Random}(\text{seed})$ 
   $X = a + (b - a)r_x$ 
   $r_y = \text{Random}(\text{seed})$ 
   $Y = f(a) + (f(b) - f(a))r_y$ 
  Se  $(X, Y)$  esta abaixo da curva faça N_c = N_c + 1
Fim do Loop
Area = N_c/N

```

Muitos geradores de números (pseudo) aleatórios em FORTRAN retornam um vetor de tamanho N quando chamados. Por exemplo `CALL Random_Number(r)` gera um vetor com um número de elementos igual à dimensão do vetor r . Se r tem dimensão L ela retorna $r(1), r(2) \dots r(L)$. Em C++, a partir do padrão conhecido como C++11, diversos algoritmos para gerar números aleatórios foram integrados à biblioteca padrão. Nas tabelas 1 e 2 vemos exemplos de seu uso.

Algoritmo 2.3

```

#include <random>
#define <random>
...
double aleat( )
{
  //static std::mt19937_64 e{seed};
  //static std::mt19937 e{seed};
  //static std::knuth_b e{seed};
  //static std::ranlux48 e{seed};
  static std::uniform_real_distribution<double> d{0.0, 1.0};
  return d(e);
}

```

Exercícios - 2.4

1. Use o método de Monte Carlo para refazer o cálculo da área do quarto de círculo do exercício 1.
2. Calcule o volume sob a curva $e^{0.1(x^2+y^2)}$ usando Monte Carlo e compare com o exercício (2).
3. Calcule a integral $\int_0^2 dx dy dz dw e^{-(x^2+xy+y^2+yz+z^2+zw+w^2)}$. Você pode usar o método de trapézios e Monte Carlo para comparar.

2.4 Cálculo do erro na integração por Monte Carlo

Em qualquer aproximação a estimativa do erro cometido é um dos itens mais importantes. Por exemplo, se estamos fazendo um experimento para determinar a aceleração da gravidade em um determinado local, uma resposta como $10m/s^2$ não faz qualquer sentido. Poderia ser $12m/s^2$ ou $9m/s^2$? É necessário que incluamos na resposta o erro cometido na medida. Uma resposta correta seria algo como $(10.00 \pm 0.22)m/s^2$. Como fazer este cálculo usando o método de Monte Carlo. Como não temos controle sobre o erro em uma única medida (é um processo estatístico), o correto é fazermos vários cálculos e tomarmos a média, calculando então o erro cometido. Para exemplificar vamos tomar a integral do exercício 3.3. Podemos fazer o cálculo um número M de vezes, gerando um conjunto de medidas $\{I_1, I_2 \dots I_M\}$. O valor médio será $\bar{I} = \sum_{m=1}^M I_m$. De posse do valor médio calculamos o desvio, em relação à média, de cada medida como $\Delta I = \frac{1}{M} \sum |\bar{I} - I_m|$. Deste modo o resultado será $I = \bar{I} \pm \Delta I$. Este cálculo, apesar de simples, funciona na maior parte dos casos. Problemas começam a aparecer quando os cálculos das integrais são feitos usando-se geradores de números pseudoaleatórios de baixa qualidade o que resulta em resultados correlacionados para as integrais.

Exercícios - 2.5

1. Usando a função do exercício 1.1 calcule o valor de π com pelo menos 3 algarismos significativos.
2. Continuando o exercício 4.1, inclua barras de erros.