



Introdução à Física Computacional 1S/2018

Projeto 2 — Cálculo numérico

Início: 12 de Março de 2018

Data da entrega do relatório: 02 de Abril de 2018

Prof.: Eric C. Andrade

Descrição:

Discutiremos aqui brevemente alguns conceitos básicos de cálculo numérico que são muito úteis no dia a dia da pesquisa. O ingrediente básico que permeia todos esses métodos é a expansão em série de Taylor de uma função $f(x)$ ao redor de x_0

$$f(x) = f(x_0) + f'(x_0)(x - x_0) + \frac{1}{2!}f''(x_0)(x - x_0)^2 + \frac{1}{3!}f'''(x_0)(x - x_0)^3 + \dots \quad (1)$$

Sugestão de execução:

Exercício 1: Aula 3. Exercício 2: Aulas 3 e 4. Exercício 3: Aula 4.

1) Derivada numérica

Dada uma função $f(x)$, podemos obter uma aproximação para sua derivada utilizando a definição

$$f'(x) = \lim_{h \rightarrow 0} \frac{f(x+h) - f(x)}{h}, \quad (2)$$

só que para um h finito

$$f'(x) \approx \frac{f(x+h) - f(x)}{h}. \quad (3)$$

À primeira vista, parece que tudo que temos que fazer é escolher um valor de h bem pequeno que obteremos o valor exato da derivada. Como veremos explicitamente em um exemplo, esse não é caso, pois rapidamente perdemos precisão numérica.

A primeira pergunta que nos fazemos é qual é o erro que cometemos ao passar de (2) para (3)? Para respondermos essa questão, escrevemos

$$f(x+h) = f(x) + hf'(x) + \frac{h^2}{2!}f''(x) + \dots,$$

donde obtemos a **derivada para frente de 2 pontos**

$$f'_f(x) = \frac{f(x+h) - f(x)}{h} + \mathcal{O}(h). \quad (4)$$

Aqui o símbolo $\mathcal{O}(h)$ quer dizer que o erro que cometemos ao truncar a expansão em série de Taylor para o cálculo de $f'(x)$ é da ordem de h . Analogamente, podemos calcular a **derivada para trás de 2 pontos**

$$f'_t(x) = \frac{f(x) - f(x-h)}{h} + \mathcal{O}(h). \quad (5)$$

Se combinarmos as equações (4) e (5) podemos escrever a **derivada simétrica de 3 pontos**

$$\begin{aligned} f'_{3s}(x) &= \frac{1}{2}(f'_f(x) + f'_t(x)), \\ f'_{3s}(x) &= \frac{f(x+h) - f(x-h)}{2h} + \mathcal{O}(h^2). \end{aligned} \quad (6)$$

h	$f'_f(1)$	$f'_t(1)$	$f'_{3s}(1)$	$f''_{3s}(1)$
5×10^{-1}				
1×10^{-1}				
5×10^{-2}				
1×10^{-2}				
5×10^{-3}				
1×10^{-3}				
5×10^{-4}				
1×10^{-4}				
5×10^{-5}				
1×10^{-5}				
5×10^{-6}				
1×10^{-6}				
1×10^{-7}				
1×10^{-8}				

Tabela I: Derivadas numéricas de $f(x)$ em (8) no ponto $x = 1$ por meio de diferentes aproximações em função do passo h .

Note que um há um nesse caso o erro passa agora para $\mathcal{O}(h^2)$, ou seja melhoramos nossa convergência. Isso ocorre porque o termo envolvendo a derivada segunda na expansão em Taylor é simétrico em h e assim a primeira correção vem apenas do termo cúbico. Qual é o preço que pagamos para esse ganho em precisão? Temos que agora avaliar tanto $f(x+h)$ quanto $f(x-h)$, o que aumenta o custo computacional. Naturalmente, derivadas simétricas envolvendo mais pontos podem ser construídas de modo análogo.

Podemos aplicar a mesma ideia para calcularmos numericamente a derivada segunda de uma função. Partimos da seguinte igualdade

$$f(x+h) - 2f(x) + f(x-h) = f''(x)h^2 + \mathcal{O}(h^4)$$

e chegamos assim à **derivada segunda simétrica de três pontos**

$$f''_{3s}(x) = \frac{f(x+h) - 2f(x) + f(x-h)}{h^2} + \mathcal{O}(h^2). \quad (7)$$

Baseado nessa discussão, considere agora a seguinte função

$$f(x) = e^{2x} \cos \frac{x}{4}. \quad (8)$$

(a) Escreva um programa FORTRAN que calcule as derivadas de $f(x)$ para as diferentes aproximações e valores de h no ponto $x = 1$.

(b) Preencha a tabela I com o valor absoluto dos desvios em relação aos resultados exatos $|\varepsilon|$. Aponte o valor ótimo de h em cada um dos casos e discuta seus resultados.

(c) Faça um gráfico de $\log_{10} |\varepsilon| \times \log_{10} h$ para $f'_f(1)$, $f'_{3s}(1)$ e $f''_{3s}(1)$. Verifique que a ordem da convergência das aproximações coincide com aquela esperada teoricamente e discuta seus resultados.

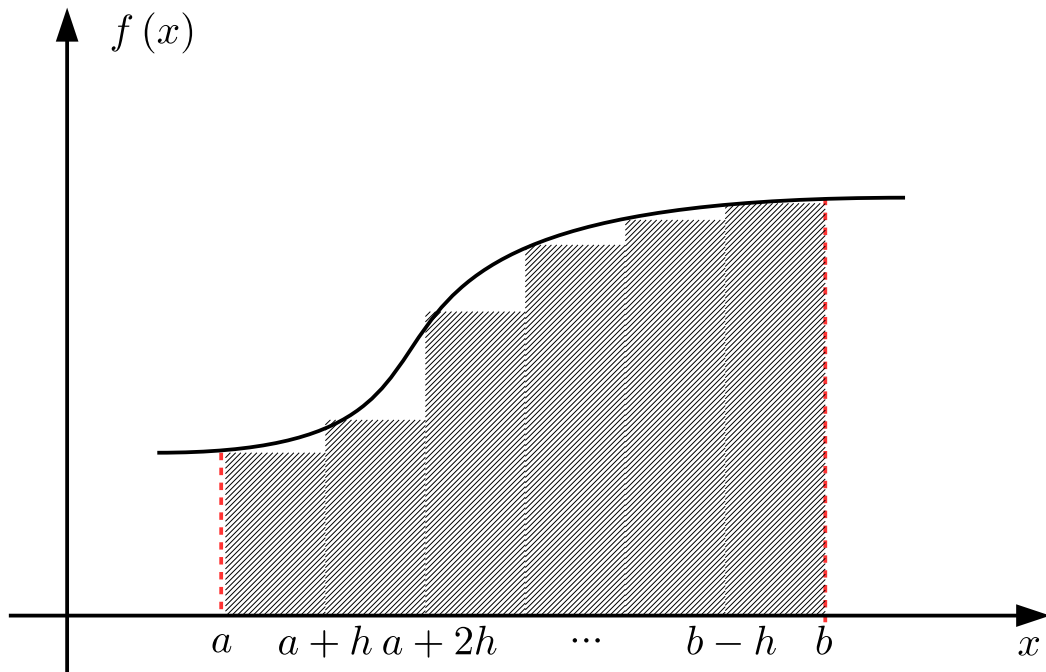


Figura 1: Representação geométrica do cálculo da integral. Dividimos o intervalo $[a, b]$ em N subintervalos de igual tamanho $h = (b - a) / N$.

2) Integração numérica

A integral

$$I = \int_a^b f(x) dx, \quad (9)$$

possui um significado geométrico muito simples: ela dá a área contida sob a curva descrita pela função $f(x)$ indo de $x = a$ até $x = b$. A ideia básica por trás dos métodos básicos de integração é dividir o intervalo $[a, b]$ em N subintervalos de igual tamanho $h = (b - a) / N$, de tal forma que a integral é agora dada por, veja figura 1

$$I = \int_a^b f(x) dx = \int_a^{a+h} f(x) dx + \int_{a+h}^{a+2h} f(x) dx + \dots + \int_{b-h}^b f(x) dx. \quad (10)$$

O que fazemos agora é aproximarmos $f(x)$ em cada um dos intervalos por funções simples, para as quais podemos fazer a integral analiticamente. Começamos com uma aproximação linear para $f(x)$ ao redor de $x = u$

$$f(x) \approx f(u) + f'(u)(x - u) \approx f(u) + \frac{f(u+h) - f(u)}{h}(x - u),$$

onde utilizamos a Eq. 4. Temos assim que

$$I = \int_u^{u+h} f(x) dx = \int_u^{u+h} \left(f(u) + \frac{f(u+h) - f(u)}{h}(x - u) \right) dx = \frac{1}{2}h(f(u+h) + f(u)), \quad (11)$$

que é conhecida como **regra do trapézio**. Seu erro local é $\mathcal{O}(h^3)$ e o erro global é de $\mathcal{O}(h^2)$.¹ A integral no intervalo

¹ O erro local é o erro em cada uma das N partições. Ele depende de $h = (b - a) / N$. Como temos N partições, temos que somar N erros locais e, por isso, o erro global é sempre uma potência de h menor do que o erro local.

h	Regra do trapézio	Regra de Simpson
1/2		
1/4		
1/8		
1/16		
1/32		
1/64		
1/128		
1/256		
1/1024		
1/2048		
1/4096		
1/8192		

Tabela II: Integral numérica de $f(x)$ em (14) no intervalo $[0, 1]$ por meio de duas diferentes aproximações como função da partição do intervalo h .

$[a, b]$ é então dada por

$$I_T \approx h [0.5f(a) + f(a+h) + f(a+2h) + f(a+3h) + \dots + f(b-h) + 0.5f(b)]. \quad (12)$$

Para escrever um código com esse método, é mais eficiente calcularmos separadamente a contribuição dos extremos, dada por $0.5h(f(a) + f(b))$, e realizarmos um laço para calcularmos as contribuições restantes das partições internas.

Podemos agora considerar uma aproximação quadrática para $f(x)$ ao redor de $x = u$

$$f(x) \approx f(u) + \left[\frac{f(u+h) - f(u)}{h} \right] (x-u) + \frac{1}{2} \left[\frac{f(u+h) - 2f(u) + f(u-h)}{h^2} \right] (x-u)^2,$$

onde utilizamos a Eq. 7. Nesse caso, precisamos de três pontos em nosso intervalo e a integral que avaliaremos é

$$I = \int_{u-h}^{u+h} f(x) dx = f(u)2h + \frac{h}{3} (f(u+h) - 2f(u) + f(u-h)) = \frac{1}{3} (f(u+h) + 4f(u) + f(u-h)), \quad (13)$$

que é conhecida como **regra de Simpson**. Seu erro global é $\mathcal{O}(h^4)$. A integral no intervalo $[a, b]$ é então dada por

$$I_S \approx \frac{h}{3} [f(a) + 4f(a+h) + 2f(a+2h) + 4f(a+3h) + 2f(a+4h) \dots + f(b)].$$

Novamente, para a confecção do código para esse método, é mais eficiente calcularmos separadamente a contribuição dos extremos, agora dada por $\frac{h}{3}(f(a) + f(b))$, e realizarmos um laço para calcularmos as contribuições restantes das partições internas. Note, contudo, que nesse os termos de ordem par e ímpar possuem pesos diferentes.

Podemos continuar esse processo e considerar cada vez mais pontos e polinômios de grau superior como aproximações para $f(x)$. Contudo, a série de Taylor converge apenas localmente e, portanto, não é imediato que um polinômio de ordem maior traduza-se em uma melhor convergência global. As duas aproximações descritas acima formam a base de métodos bem robustos para avaliarmos integrais mesmo de funções não muito suaves. O segredo do sucesso do método está em variarmos o valor das partições N até obtermos a precisão desejada.

Baseado nessa discussão, considere agora a seguinte função

$$f(x) = e^{2x} \sin x. \quad (14)$$

(a) Escreva um programa FORTRAN que calcule $I = \int_0^1 f(x) dx$ tanto pelo método do trapézio quanto pelo de Simpson para diferentes valores da partição $h = 1/N$.

(b) Preencha a tabela II com o valor absoluto dos desvios em relação ao resultado exato $|\varepsilon|$. Aponte o valor ótimo de h em cada um dos casos e discuta seus resultados.

(c) Faça um gráfico de $\log_{10} |\varepsilon| \times \log_{10} h$ para ambos os métodos. Verifique que a ordem da convergência das aproximações coincide com aquela esperada teoricamente e discuta seus resultados.

3) Equações algébricas não lineares

Seja $f(x)$ uma função qualquer, em geral não linear. Muitas vezes, estamos interessados em soluções (raízes) da equação

$$f(x) = 0. \quad (15)$$

Um exemplo clássico é dado por $f(x) = x^2 - 2bx + c$, com soluções $x = -b \pm \sqrt{b^2 - c}$. De maneira geral, contudo, não é possível encontrarmos uma solução analítica para esse problema e temos que buscar uma solução numérica. O algoritmo mais simples para resolver esse tipo de problema é o chamado **método da bisseção**. Começamos com dois palpites iniciais para a raiz x , designados x_+ e x_- tais que

$$f(x_+) > 0 \text{ e } f(x_-) < 0. \quad (16)$$

Assim a função $f(x)$ muda de sinal no intervalo $[x_+, x_-]$ e, portanto, uma de suas raízes deve estar contida nesse intervalo. Propomos então um novo valor para x , que chamamos de x_{novo} pela fórmula

$$x_{novo} = \frac{x_+ + x_-}{2}. \quad (17)$$

Se $f(x_{novo}) > (<) 0$, tomamos $x_{+(-)} = x_{novo}$ e iteramos o processo até convergirmos, por exemplo, de acordo com o critério

$$\varepsilon = \left| \frac{x_+ - x_-}{2} \right| < \text{tolerância}. \quad (18)$$

Embora pareça simplório, o método da bisseção tem sucesso garantido, mesmo que sua convergência seja lenta.

Baseado nessa discussão, considere agora a seguinte equação transcendental

$$m = \tanh(m/T). \quad (19)$$

Essa é uma equação que aparece na solução de campo médio do modelo de Ising. Nesse caso, $-1 \leq m \leq 1$ é a magnetização do problema, o parâmetro T é a temperatura e escolhemos nossas unidades tais que a temperatura crítica $T_c = 1$.

(a) Mostre graficamente que para $T > 1$ só há uma solução, $m = 0$, e que para $T < 1$ temos três soluções para essa equação: $m = 0$ e $m = \pm m_0$. Dica: Faça o gráfico das funções $f(x) = x$ e $g(x) = \tanh(x/T)$ e veja onde as curvas se cruzam.

(b) Escreva um programa FORTRAN que encontre a **raiz positiva** de $f(m) = m - \tanh(m/T)$ para $T < 1$ utilizando o método da bisseção. Discuta seu algoritmo e sua escolha dos critérios de convergência e de sua tolerância. Dicas: tome $x_- = \delta$ e $x_+ = 1 + \delta$, com δ pequeno, e acompanhe a convergência de seu programa passo a passo.

(c) Faça um gráfico da raiz positiva de $f(m)$ como função do parâmetro T no intervalo $0 \leq T \leq 1$. Discuta seus resultados.

```

Number of observations           = 100001
Mean of independent variable    = -1.434316
Mean of dependent variable      = -0.1842457
Standard dev. of ind. variable  = 0.4344598
Standard dev. of dep. variable  = 0.440337
Correlation coefficient          = 0.9999709
Regression coefficient (SLOPE)   = 1.013498
Standard error of coefficient    = 2.461248e-05
t - value for coefficient        = 41178.2
Regression constant (INTERCEPT) = 1.269431
Standard error of constant       = 3.688603e-05
t - value for constant          = 34414.94

Analysis of variance
Source      d.f      Sum of squares  Mean Square      F
Regression      1      19388.52      19388.52      1.695644e+09
Residual      99999      1.143419      1.143431e-05
Total      100000      19389.66

y = 1.2694 + 1.0135 * x
Regression of set 0 results to set 1

```

Figura 2: Exemplo de um console do Xmgrace com os resultados de uma regressão linear.

Breve discussão sobre a execução dos problemas

Nos Exercícios 1 e 2 é pedido que vocês façam um gráfico $\log_{10} |\varepsilon| \times \log_{10} h$. A ideia aqui é o erro $|\varepsilon|$ depende de uma potência de h

$$|\varepsilon| = Ah^\alpha, \quad (20)$$

onde A é uma constante numérica sem importância para nossa discussão e α é o expoente no qual estamos interessados. Por exemplo quando dizemos que a convergência de um método é linear, quer dizer que $\alpha = 1$. Uma convergência quadrática significa $\alpha = 2$ e assim por diante.

Em situações nas quais temos uma dependência do tipo lei de potência, é conveniente aplicarmos o log dos dois lados para extrairmos o valor do expoente em uma escala linear. Pela Eq. (20) temos então que

$$\begin{aligned} \log_{10} |\varepsilon| &= \log_{10} Ah^\alpha, \\ &= \log_{10} A + \alpha \log_{10} h. \end{aligned} \quad (21)$$

Ou seja, ao gerarmos a curva $\log_{10} |\varepsilon| \times \log_{10} h$ concluímos que o coeficiente angular da reta dá diretamente o expoente α . Esse é o número no qual estamos interessados para estudarmos a convergência de cada um dos métodos (o coeficiente linear pode ser ignorado nessa discussão).

Para calcular o log na base 10, basta utilizarem a função `DLOG10()`. Vocês deverão então gerar um gráfico no Xmgrace com essa curva. O próximo passo é realizar uma regressão linear para calcularmos α .

- Para ler o arquivo `foo.dat` e graficá-lo faça: `Data -> Import -> ASCII` e escolha `foo.dat`.
- Para realizar a regressão linear faça: `Data -> Transformation -> Regression`. Isso vai gerar uma janela. Basta clicar em `Accept`, pois a opção padrão é a regressão linear. Pronto, a curva já é gerada automaticamente
- Para visualizar a expressão analítica da curva, vá em `Window -> Console`, veja um exemplo na Fig. 2
- A expressão para a reta aparece, no exemplo na Fig. 2, como $y = 1.2694 + 1.0135 * x$. O valor do coeficiente angular é então 1.01350 (2) ou 1.01350 ± 2 , já com o erro. O erro pode ser encontrado na linha “Standard error of coefficient” e nesse exemplo é dado por 2.46×10^{-5} .