



**IFSC UNIVERSIDADE
DE SÃO PAULO**
Instituto de Física de São Carlos

Projeto 03

Métodos básicos de integração e derivação

Jefter Santiago Mares
n° USP:12559016

12 de novembro de 2022

Conteúdo

1	Tarefa A - Cálculo de derivada	2
2	Tarefa B - Cálculo de integral	7

Tarefa A - Cálculo de derivada

Por definição a derivada de uma função é

$$\frac{df}{dx} = f'(x) = \lim_{\Delta x \rightarrow 0} \frac{f(x + \Delta x) - f(x)}{\Delta x} \quad (1)$$

Essa definição de derivada é chamada de derivada à direita, a derivada à esquerda é dada por

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

Para uma função real e contínua o limite pela esquerda e direita são iguais, o valor numérico no entanto, para valores de Δx a derivada à esquerda e à direita podem se aproximar a valores muito diferentes, por isso é necessário encontrar o melhor termo Δx .

A derivada da forma centrada (**derivada simétrica**) apresenta melhor aproximação e é dada por

$$f'(x) = \lim_{\Delta x \rightarrow 0} \frac{f(x + \Delta x) - f(x - \Delta x)}{2\Delta x} \quad (3)$$

Para fazer o cálculo numérico de uma derivada primeiramente discretizamos a variável x , ou seja, fazemos $x_n \equiv x_0 + nh$ onde $n = (0, \pm 1, \pm 2, \dots)$ temos então $f(x) = f(x_n) = f_n = f(x_0 + nh)$.

Para realizar o cálculo da derivada fazemos uma expansão de Taylor da função discretizada, então

$$f_n = f(x_n) = \sum_{n=0}^{\infty} \frac{f(x_0)}{n!} (x_n - x_0)^n \quad (4)$$

fazendo $x_n \equiv x_0 + nh \Rightarrow x_n - x_0 \equiv nh$ e substituindo na expansão de Taylor temos

$$f(x_n) = \sum_{n=0}^{\infty} \frac{f(x_0)}{n!} (nh)^n \quad (5)$$

usando a expansão para calcular a derivada primeira de x à direita, temos

$$f(x_1) = f(x_0) + 1 \cdot h f'(x_1) \Rightarrow f'(x_1) = \frac{f(x_1) - f(x_0)}{h}$$

$$f' = \frac{f_1 - f_0}{h} + \mathcal{O}(h) \quad (6)$$

e a primeira derivada de x à esquerda é dada por

$$f' = \frac{f_0 - f_{-1}}{h} + \mathcal{O}(h) \quad (7)$$

A derivada simétrica em três pontos, ou seja, a derivada centrada (3) que como foi dito é a que apresenta melhor aproximação pode ser escrita como

$$f' = \frac{f_1 - f_{-1}}{2h} + \mathcal{O}(h^2) \quad (8)$$

Queremos também calcular a primeira derivada simétrica em 5 pontos, temos então

$$f'_5 = \frac{f(x - 2h) - 8f(x - h) + 8f(x + h) - f(x + 2h)}{12h}$$

$$f' = \frac{f_{-2} - 8f_{-1} + 8f_1 - f_2}{12h} + \mathcal{O}(h^4) \quad (9)$$

é esperado que as derivadas escritas nas formas (8) e (9) tenha uma convergência mais rápida para o valor encontrado analiticamente pois a função é calculada em mais pontos nessas fórmulas, isso reduz o erro esperado para o cálculo numérico dessas derivadas.

Além das derivadas primeiras também precisamos fazer o cálculo numérica das segunda e terceira derivada de função utilizando as fórmulas simétrica e anti-simétrica em 5 pontos, respectivamente.

$$f'' = \frac{-f_{-2} + 16f_{-1} - 30f_0 + 16f_1 - f_2}{12h^2} + \mathcal{O}(h^4) \quad (10)$$

$$f''' = \frac{-f_{-2} + 2f_{-1} - 2f_1 + f_2}{2h^3} + \mathcal{O}(h^2) \quad (11)$$

Resolução

Queremos implementar um programa que calcule a derivada da função $f(x) = \cosh(3x) \sin\left(\frac{x}{2}\right)$ no ponto $x = \frac{1}{2}$, com precisão dupla.

$$\frac{d}{dx}f(x) = \frac{d}{dx} \cosh(3x) \frac{d}{dx} (3x) \sin\left(\frac{x}{2}\right) + \frac{1}{2} \cosh(3x) \cos\left(\frac{x}{2}\right)$$

$$\frac{d}{dx}f = 3 \sinh(3x) \sin\left(\frac{x}{2}\right) + \frac{1}{2} \cosh(3x) \cos\left(\frac{x}{2}\right) \quad (12)$$

$$\frac{d^2}{dx^2}f(x) = 3 \sinh(3x) \cos\left(\frac{x}{2}\right) + \frac{35}{4} \cosh(3x) \sin\left(\frac{x}{2}\right) \quad (13)$$

$$\frac{d^3}{dx^3}f(x) = \frac{1}{8} \left[198 \sinh(3x) \sin\left(\frac{x}{2}\right) + 107 \cosh(3x) \cos\left(\frac{x}{2}\right) \right] \quad (14)$$

Código em Fortran

```

1  !      Cálculo das derivadas de uma função f(x)
2      implicit real*8 (a-h, o-z)
3      parameter (n = 14)
4      dimension h(n)
5      h = (/5.0e-1, 2.0e-1, 1.0e-1, 5.0e-2, 1.0e-2, 5.0e-3, 1.0e-3,
6      $      5.e-4, 1.0e-4, 5.0e-5, 1.0e-5, 1.0e-6, 1.0e-7, 1.0e-8/)
7
8      x = 0.5d0
9
10     df = 3*sin(x/2)*sinh(3*x)+(0.5)*cos(x/2)*cosh(3*x)
11     df2 = 3*sinh(3*x)*cos(x/2)+(8.75)*cosh(3*x)*sin(x/2)
12     df3 = (0.125)*(198*sinh(3*x)*sin(x/2)+107*cosh(3*x)*cos(x/2))
13
14     open(unit = 10, file="saida.dat")
15
16     do i = 1, n
17
18         fr = f(x+h(i))

```

```

19         fl = f(x-h(i))
20
21         fr2 = f(x + 2*h(i))
22         fl2 = f(x - 2*h(i))
23
24         df_r = (fr-f(x))/h(i)
25         df_l = (f(x)-fl)/h(i)
26         df_sm3 = (fr-fl)/(2*h(i))
27
28         df_sm5 = (fl2 + 8*(fr-fl) - fr2)/(12*h(i))
29
30         df2_sm5=(-fl2-fr2+16*(fr+fl)-30*f(x))/(12*h(i)**2)
31
32         df3_asm5 = (fr2-fl2+2*(fl-fr))/(2*h(i)**3)
33
34         err_r = abs(df - df_r)
35         err_l = abs(df - df_l)
36         err_sm3 = abs(df - df_sm3)
37         err_sm5 = abs(df - df_sm5)
38         err_asm5 = abs(df2 - df2_sm5)
39         err2_asm5 = abs(df3 - df3_asm5)
40
41         write(10, *) h(i), err_sm3, err_r, err_l, err_sm5, err_asm5,
42         $      err2_asm5
43     end do
44
45     write(10, *) "Valores exatos", df, df, df, df, df2, df3
46     close(10)
47 end
48
49 function f(x)
50     implicit real * 8 (a-h, o-z)
51     f = cosh(3 * x) * sin(x / 2)
52     return
53 end

```

Resultados

Na tabela abaixo estão os resultados para as fórmulas de derivação dadas.

Note que, para as fórmulas para cálculo da primeira derivada o aumento de pontos calculados aumenta a precisão do cálculo, então como esperado o método que oferece maior precisão e estabilidade é o da derivada simétrica em 5 pontos. No entanto, essa precisão decai para h a partir da ordem 10^{-5} podemos dizer então que o método é melhor para h até esta ordem, já que o calculo a partir dele converge para o valor mais preciso em menos iterações que os outros métodos (3 pontos à esquerda, direita e simétrica).

O caso da derivada terceira anti-simétrica em 5 pontos expõe um problema computacional relacionado aos pontos flutuantes em que para h muito pequeno se torna impossível saber o erro atrelado ao cálculo. Isso acontece porque o computador pode estar alocando valores aleatórios (ou lixo) no espaço que a variável do erro não estava preenchendo.

Tabela 1: Erros calculados para cada método para diferentes h .

H	Derivada simétrica (3 pontos)	Derivada à esquerda (2 pontos)	Derivada à direita (2 pontos)	Derivada simétrica (5 pontos)	Derivada segunda simétrica (5 pontos)	Derivada terceira anti-simétrica (5 pontos)
0.5	2.106678323581137	5.7693816931927806	1.5560250460305065	1.4953176671794182	1.3930924069170949	42.924442381871593
0.2	0.2972794239134368	1.4753863116237089	0.88082746379683474	0.0296455144898608	0.0297818925915418	5.5152779026445273
0.1	0.0729809779543329	0.6432406862433164	0.49727873033465064	0.0017851706987017	0.0018137229953048	1.3362264585161228
0.05	0.0181623416769949	0.3009748744740963	0.26465019112010646	0.0001105370821177	0.0001126244404687	0.3314463185106291
0.01	0.0007254350693797	0.0571401661611141	0.05568929602235517	0.0000001763316666	0.000000179821562	0.0132246126315749
0.005	0.0001813505025749	0.0283864091768912	0.02802370817174138	0.0000000110196931	0.0000000112304867	0.0033058937863472
0.001	0.0000072539153306	0.0056481184098053	0.00563361057914457	0.0000000000176307	0.0000000000962678	0.0001322371209724
0.0005	0.0000018134779331	0.0028222434185711	0.00281861646270531	0.0000000000011995	0.0000000012805045	0.0000336493304331
0.0001	0.0000000725387008	0.0005641583393649	0.00056401326196331	0.0000000000005901	0.00000000703183751	0.0001155361312044
0.00005	0.0000000181361059	0.0002820610344951	0.00028202476228323	0.00000000000019078	0.00000001295302745	0.0013832650656269
0.00001	0.0000000007289187	0.0000564093137036	0.00005640785586624	0.00000000000044981	0.00000036822441327	0.0582266882343915
0.000001	0.0000000000062448	0.0000056409384248	0.00000564092593524	0.00000000000122586	0.0002222532222138	67.498841907047023
0.0000001	0.0000000000117452	0.000000564536656	0.00000056456014663	0.00000000000117452	0.0018504836939516	43.523461396381741
0.00000001	0.0000000000107425	0.000000055218941	0.00000005550040915	0.0000000000107425	0.5606629230281115	43.523461396381741
Exatos	2.7200159512296831	2.7200159512296831	2.7200159512296831	2.7200159512296831	11.281716150250322	43.523461396381741

No gráfico abaixo podemos observar o comportamento de todos os métodos para vários h cada vez menores. A partir desse gráfico compilei os melhores h para cada método.

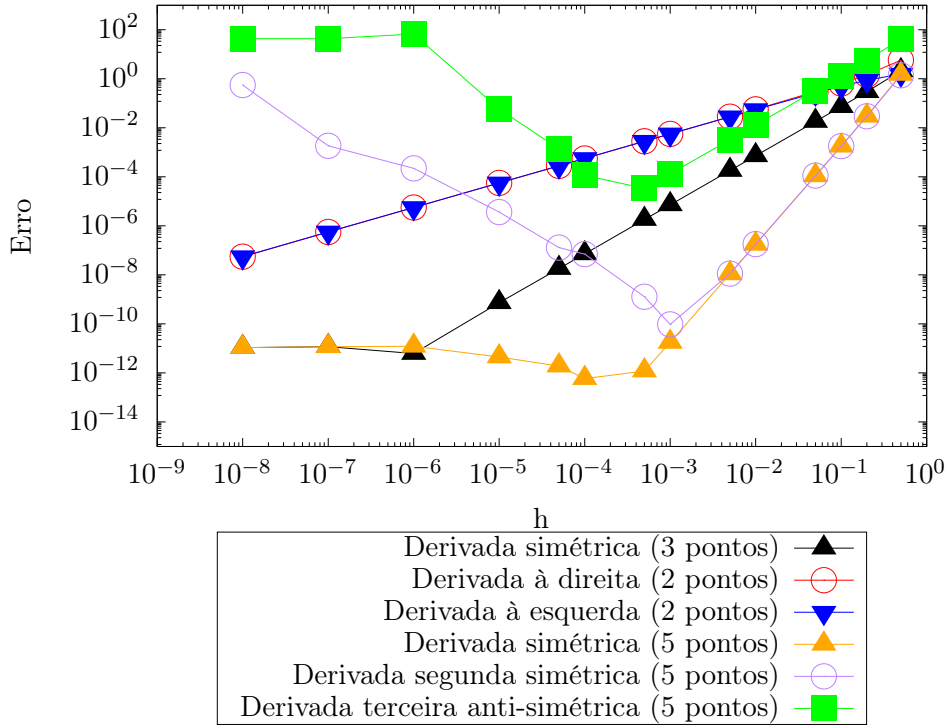


Figura 1: Comportamento do erro em relação ao h para todas as derivdas calculadas.

Tabela 2: Melhores ordens de grandeza de h para cada um dos métodos.

	Derivada simétrica (3 pontos)	Derivada à esquerda (2 pontos)	Derivada à direita (2 pontos)	Derivada simétrica (5 pontos)	Derivada segunda simétrica (5 pontos)	Derivada terceira anti-simétrica (5 pontos)
$h \approx$	10^{-7}	10^{-9}	10^{-9}	10^{-4}	10^{-4}	10^{-5}

O desvio apresentado pela derivada à esquerda e à direita em 2 pontos é igual e apresentam

uma linearidade relacionada ao valor de h . Utilizando valores menores de h podemos aproximar a derivada bem até bastante casas decimais, porém, os métodos usando derivada simétrica ainda assim são melhores para aproximação dado um h correto conseguem melhor precisão em menos iterações.

Tarefa B - Cálculo de integral

Foi implementado um código que calcula a integral

$$\int_0^1 e^{\frac{x}{4}} \sin \pi x dx$$

usando os métodos: regra do trapézio, regra de Simpson e regra de Boole.

$$\int_a^b f(x) dx = \frac{h}{2} (f_{-1} + 2f_0 + f_1) + \mathcal{O}(h^3) \quad (15)$$

$$\int_a^b f(x) dx = \frac{h}{3} (f_1 + 4f_0 + f_{-1}) + \mathcal{O}(h^5) \quad (16)$$

$$\int_a^b f(x) dx = \frac{2h}{45} (7f_0 + 32f_1 + 12f_2 + 32f_3 + 7f_4) + \mathcal{O}(h^7) \quad (17)$$

Código em Fortran

```
1  !      Cacula integral da funcao f(x) = exp(x/4)sin(pi x)
2  !      em [a, b] = [0, 1]
3      implicit real * 8 (a-h, o-z)
4      n = 12
5
6      open(unit = 11, file="saida.dat")
7
8      write(11, 3)
9  3      format(12(' '), 'N', 16(' '), 'h', 16(' '), 'Regra do Trapezio',
10 $      9(' '), 'Regra de Simpson', 10(' '), 'Regra de Boole')
11
12      f_int = f_integral(1.0d0) - f_integral(0.0d0)
13
14      do j = 1, 10
15
16          trap = 0.0d0
17          simp = 0.0d0
18          boole = 0.0d0
19
20          h = 1.0d0 / n
21
22          do i = 1, n, 2
23
24              x = i * h
25
26              f0 = f(x)
27              f1_ = f(x - h)
28              f1 = f(x + h)
29
30              trap = trap + (h/2) * (f1_ + 2 * f0 + f1)
```

```

31         simp = simp + (h/3) * (f1_ + 4 * f0 + f1)
32     end do
33
34     do i = 0, n-4, 4
35
36         x = i * h
37
38         f0 = f(x)
39         f1 = f(x + h)
40         f2 = f(x + 2*h)
41         f3 = f(x + 3*h)
42         f4 = f(x + 4*h)
43
44         boole = boole + (7*f0+32*f1+12*f2+32*f3+7*f4)
45     end do
46     boole = boole * (2*h/45)
47
48     err1 = abs(trap - f_int)
49     err2 = abs(simp - f_int)
50     err3 = abs(boole - f_int)
51
52     write(11, *) n, h, err1, err2, err3
53
54     n = 2*n
55 end do
56
57 write(11, *) "Valores exatos"
58 write(11, 6) f_int, f_int, f_int
59 6 format(28(' '), d25.18, d25.18, d25.18)
60
61 close(11)
62 end
63
64 function f(x)
65 implicit real * 8 (a-h, o-z)
66 f = exp(x / 4) * sin(acos(-1.0d0) * x)
67 return
68 end
69
70 function f_integral(x)
71 implicit real * 8 (a-h, o-z)
72 pi = acos(-1.d0)
73 f_integral = 4*exp(x/4)*(sin(pi*x)-4*pi*cos(pi*x))/(1+16*pi**2)
74 return
75 end

```


Resultados

Tabela 3: Erros calculados para a integral $\int_0^1 e^{\frac{x}{4}} \sin(\pi x) dx$ usando a regra do Trapézio, de Simpson e de Boole, respectivamente.

N	h	Regra do Trapezio	Regra de Simpson	Regra de Boole
12	0.0833333333333333	0.0041571359729468	0.0000187593023729	0.0000004800312709
24	0.0416666666666666	0.0010384097928567	0.0000011656005067	0.0000000073129510
48	0.0208333333333333	0.0002595478905307	0.0000000727435781	0.0000000001135505
96	0.0104166666666666	0.0000648835640231	0.0000000045448130	0.0000000000017713
192	0.0052083333333333	0.0000162206779871	0.0000000002840249	0.0000000000000277
384	0.0026041666666666	0.0000040551561835	0.0000000000177518	0.0000000000000004
768	0.0013020833333333	0.0000010137882134	0.0000000000011098	0.0000000000000001
1536	0.0006510416666666	0.0000002534470014	0.00000000000000691	0.00000000000000008
3072	0.0003255208333333	0.0000000633617467	0.00000000000000049	0.00000000000000003
6144	0.0001627604166666	0.0000000158404362	0.00000000000000004	0.00000000000000007
Exatos	[HTML]330001	0.722452884092878111		

Nota-se a grande diferença entre os métodos empregados, sobretudo o de Boole, que consegue na sua oitava iteração a mesma precisão o de Simpson consegue na ultima. Entretanto o método de Boole também possui o maior complexidade computacional que os outros métodos implementados.

Além disso, os métodos de quadratura apresentam uma maior estabilidade que os métodos de derivação, conforme o valor de h foi diminuído o valor da integral calculada não teve oscilações muito grandes no seu desvio.

Pelo gráfico podemos notar que o método de Boole apresenta melhor precisão para quase todos valores de h e os outros dois métodos possuem comportamento linear.

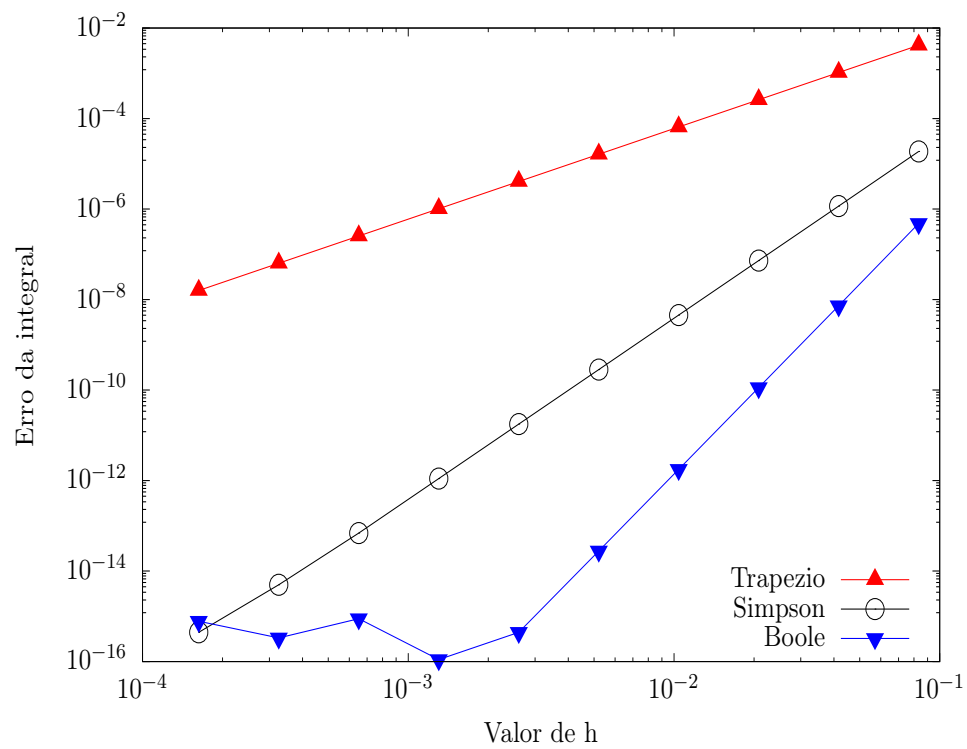


Figura 2: Comportamento do erro em relação ao h para todas as derivdas calculadas.