

Projeto 3: Cálculo Numérico

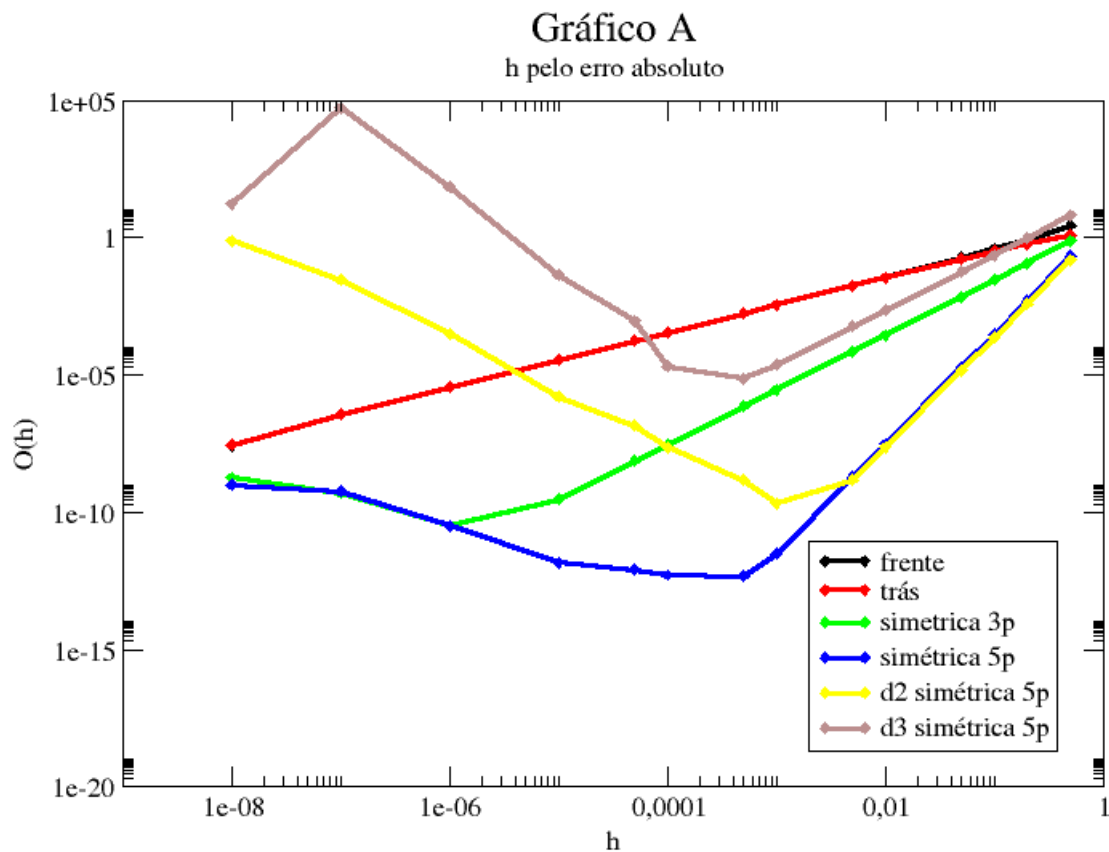
Alex Prestes, N^oUSP: 10407962

Maio 2020

Tarefa A - Derivadas Numéricas

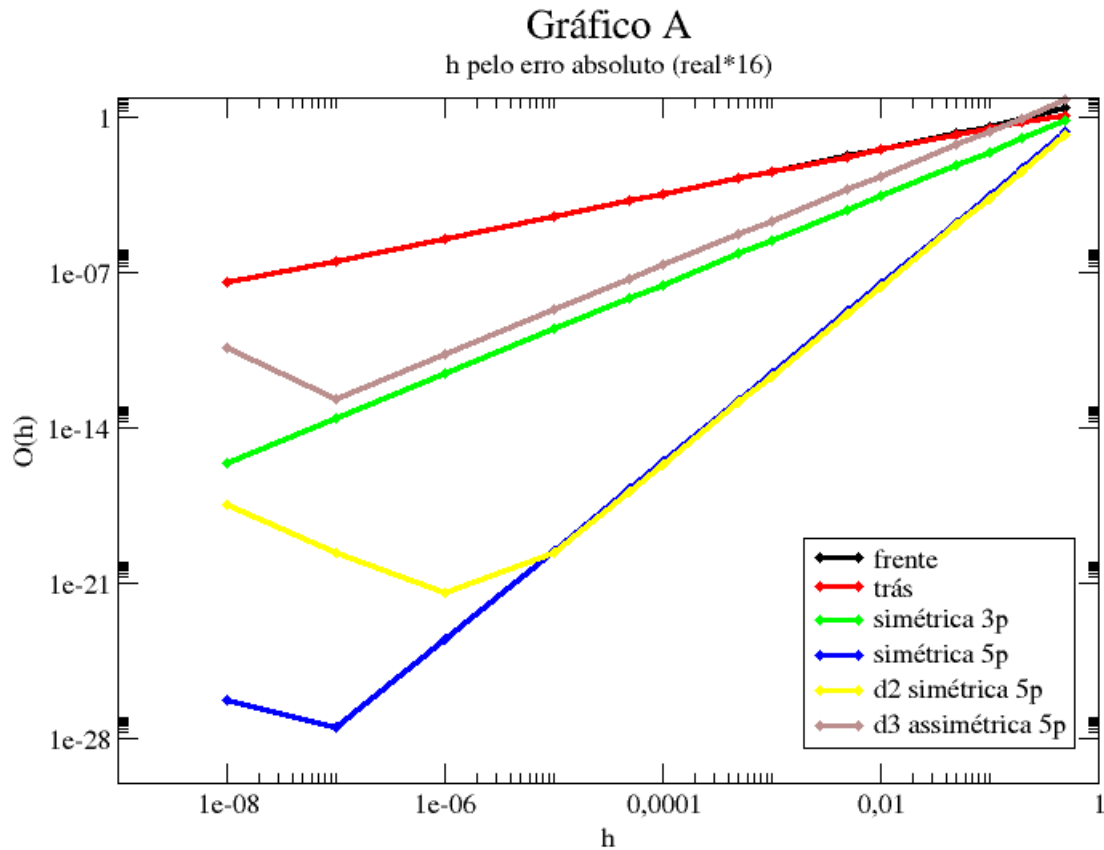
Derivadas numéricas é um método computacional para obter uma aproximação do valor de uma derivada, fazendo uso do teorema do limite, e atribuindo um valor pequeno a h , pode-se encontrar um número tão próximo quanto se queira para a aproximação das derivadas da $f(x) = \sinh(2x) \sin\left(\frac{x}{4}\right)$

```
dfrente2p(x, h_i) = ( f(x+h_i) - f(x) ) / h_i
dtras2p(x, h_i) = ( f(x) - f(x-h_i) ) / h_i
dsimetrica3p(x, h_i) = ( f(x+h_i) - f(x-h_i) ) / ( 2*h_i )
dsimetrica5p(x, h_i) = ( f(x-2*h_i) -8*f(x-h_i) +8*f(x+h_i) -f(x+2*h_i) ) / ( 12*h_i )
d2simetrica5p(x, h_i) = ( -f(x-2*h_i) +16*f(x-h_i) -30*f(x) +16*f(x+h_i) -f(x+2*h_i) ) / ( 12*h_i**2 )
d3antisimetrica5p(x, h_i) = ( -f(x-2*h_i) +2*f(x-h_i) -2*f(x+h_i) +f(x+2*h_i) ) / ( 2*h_i**3 )
```



Mas como pode-se observar, no gráfico acima, há um grande erro para valores muito pequenos de h , isso ocorre porque existe uma limitação que o computador pode escrever um número, atribuímos esse erro ao ponto flutuante, e podemos resolver definindo um tamanho maior para a variável.

Agora o mesmo código com alteração do tamanho da variável, podemos ver que obtemos valores melhores para h pequeno.



Assim observamos que para achar o melhor valor de h , é necessário um análise de h pelo erro.

Tabela 1 – Desvio das derivadas

h	Frente	Trás	Simétrica 3p	Simétrica 5p	d2 Simétrica 5p	d3 Assimétrica 5p
0.5	2.80385377815	1.2385282860256028	0.78266274606641684	0.22366629897023271	0.16163123908622890	6.5672611636207421
0.2	0.850316082316	0.61335610670733809	0.11847998780452196	5.0876718209673655E-003	3.7965495150045214E-003	0.95051247417411133
0.1	0.390044583986	0.33127352481789929	2.9385529584356362E-002	3.1262315569913213E-004	2.3438099624950581E-004	0.23425519572995768
0.05	0.187008064269	0.17234448356454912	7.3317903526151085E-003	1.9456057965605567E-005	1.4603786395284146E-005	5.8354942332513815E-002
0.01	3.61865996574E-002	3.5600429610061379E-002	2.9308502367975109E-004	3.1087345586655601E-008	2.3345486432901907E-008	2.3315305627136240E-003
0.005	1.80193752176E-002	1.7872835620077243E-002	7.3269798768205163E-005	1.9428747499716792E-009	1.4698358086207008E-009	5.8286179113409275E-004
0.001	3.59211038729E-003	3.5862488398477055E-003	2.9307737228201347E-006	3.1112890042095387E-012	2.1619683820972568E-010	2.3248294294120342E-005
0.0005	1.79532184774E-003	1.7938564615302965E-003	7.3269310973955726E-007	4.8361314952671819E-013	1.5299610467423008E-009	7.3165961609333863E-006
0.0001	3.58947069328E-004	3.5888845428733163E-004	2.9307520410526422E-008	5.1780801868517301E-013	2.3681045568935133E-008	2.1369002702442685E-005
0.00005	1.79466205730E-004	1.7945154997134338E-004	7.3278796364206755E-009	7.7760020644745964E-013	1.4395521308330217E-007	9.2232063974861944E-004
0.00001	3.58920628591E-005	3.5891473626037396E-005	2.9461633133109899E-010	1.4428458428028534E-012	1.5798437305036828E-006	4.3110798772815428E-002
0.000001	3.58904938702E-006	3.5889865745986071E-006	3.1405988920596428E-011	3.1405988920596428E-011	3.1943006411250252E-004	73.095787568973222
0.0000001	3.56997886807E-007	3.5798573261303090E-007	4.9392312462259724E-010	5.8644156197829034E-010	2.9444484074783617E-002	55493.564649214451
0.00000001	2.59754235898E-008	2.9535727641416543E-008	1.7801520257876291E-009	9.9540553577526225E-010	0.79457294053204208	17.584635917258847
Exato	2.74009174417				7.17835540972	17.58463591726

Tarefa B - Integração Numérica

Assim como a derivada, há métodos para integrais, e de forma geral seguem o todos os mesmos princípios, dividir a área abaixo do gráfico em N segmentos de tamanho $h = \frac{b-a}{N}$, a diferença entre os métodos são as formas do segmentos. Na **regra do trapézio**, os segmentos tem forma de trapézio, enquanto na **regra de Simpson** são polinômios de grau 2.

Nessa tarefa será analisada a integral definida $\int_0^1 \exp(x/2) \cos(\pi x)$

Regra do Trapézio

```
function trapezio(f, a, b, n)
    h = (b-a)/n
    trapezio = 0d0
    do i = 1, n-1, 2
        x = a + i*h
        trapezio = trapezio + ( f(x-h) +2*f(x) +f(x+h) ) * h/2
    end do
    return
end function trapezio
```

Regra de Simpson

```
function simpson(f, a, b, n)
    h = (b-a)/n
    simpson = 0d0
    do i = 1, n-1, 2
        x = a + i*h
        simpson = simpson + ( f(x-h) +4*f(x) +f(x+h) ) * h/3
    end do
    return
end function simpson
```

Regra de Bode

```
function bode(f, a, b, n)
    h = (b-a)/n
    bode = 0d0
    do i = 0, n-4, 4
        x = a + i*h
        bode = bode + ( 7*f(x) +32*f(x+h) +12*f(x+2*h) +32*f(x+3*h) +7*f(x+4*h) ) * 2*h/45
    end do
    return
end function bode
```

Graficamente pode-se observar como cada método evolui com diferentes valores de N , e também verificamos que um valor muito pequeno de h , pode acarretar em erro do ponto flutuante.

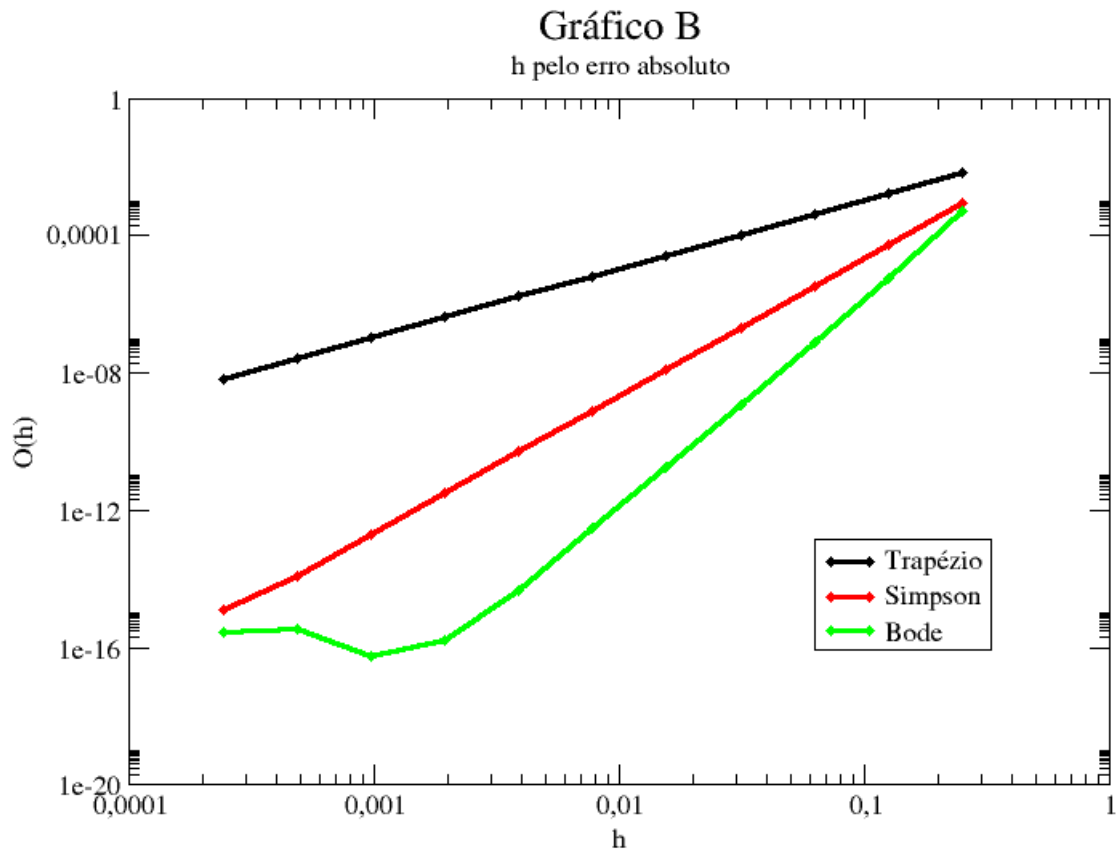


Tabela 2 – Desvio das Integrais

N	h	Trapézio	Simpson	Bode
4	0.25	7.1137027110855233E-003	9.5157185224165053E-004	5.0169532364960356E-004
8	0.125	1.7376897085270293E-003	5.4314625659163118E-005	5.5025227797544929E-006
16	0.0625	4.3193219600221044E-004	3.3203081726829176E-006	7.9312993145341792E-008
32	0.03125	1.0782826388766598E-004	2.0638015038598745E-007	1.2150510320108054E-009
64	0.015625	2.6947405185967499E-005	1.2881047978252624E-008	1.8892221120836439E-011
128	0.0078125	6.7362477045862512E-006	8.0478901320901741E-010	2.9487523534044158E-013
256	0.00390625	1.6840242050564225E-006	5.0295212439266379E-011	4.6074255521943996E-015
512	0.00195312	4.2100369387898517E-007	3.1434854719236682E-012	1.6653345369377348E-016
1024	0.00097656	1.0525077659417903E-007	1.9614865287564953E-013	5.5511151231257827E-017
2048	0.00048828	2.6312685141860470E-008	1.2517764602648640E-014	3.3306690738754696E-016
4096	0.00024414	6.5781702585088198E-009	1.1934897514720433E-015	2.7755575615628914E-016
Exato		-0.13087079127		

Tarefa C - Raízes de função

Encontrar zeros de função não é uma tarefa fácil computacionalmente, para isso há métodos que buscam raízes uma de cada vez, como o método direto que verifica a troca de sinal, no extremos do intervalo de busca, ou como o método de Newton que precisa-se conhecer a derivada previamente, e por fim o método da Secante que usa a derivada numérica no lugar da analítica.

Para iniciar a busca das raízes adicionei um trecho de código para achar o melhor intervalo.

```
do while ( f(a)*f(b) .gt. 0 )
  a = b
  b = b + h_dir
end do
```

Método Direto

Neste método divise o intervalo no ponto médio a cada iteração, e verifica para que intervalo mantém a troca de sinal, e repete-se o processo até o erro máximo desejado.

```
! verifica se o desvio já atingiu o erro
if ( desv_dir .gt. erro ) then
  ! valor medio de ab
  x_dir = (a+b)/2
  ! verifica se o produto e redefine a ou b
  if ( f(x_dir)*f(a) .gt. 0 ) then
    a = x_dir
  else
    b = x_dir
  end if
  ! atualiza o valor do desvio
  desv_dir = abs(f(a)-f(b))
else
  flag_dir = .false.
end if
```

Método Newton

O método de Newton faz uso da reta tangente que passa no ponto x_0 para buscar o próximo ponto x .

```
! verifica se o desvio já atingiu o erro
if ( desv_new .gt. erro ) then
  x_new = x0_new - f(x0_new)/df(x0_new)
  ! atualiza o valor do desvio
  desv_new = abs(f(x_new) - f(x0_new))
  ! redefine x0
  x0_new = x_new
else
  flag_new = .false.
end if
```

Método Secante

E por fim o método da Secante, que como dito anteriormente é o método de Newton, com a derivada numérica.

```
! verifica se o desvio já atingiu o erro
if ( desv_sec .gt. erro ) then
    ! derivada numerica de f
    df_sec = ( f(x0_sec + h_sec) - f(x0_sec) ) / h_sec
    x_sec = x0_sec - f(x0_sec)/df_sec
    ! atualiza o desvio
    desv_sec = abs(f(x_sec) - f(x0_sec))
    ! redefine x0
    x0_sec = x_sec
else
    flag_sec = .false.
end if
```

Na tabela abaixo é possível ver o quantas iterações cada método leva para chegar no erro desejado, todos iniciando no mesmo x_0 .

Tabela 3 – Raízes da função

(a) Primeira raiz, iniciando em -10

N	Direto	Newton	Secante
0	-5.5000000000000000	-5.2857142857142856	-5.2842848635520880
1	-4.7500000000000000	-4.3834586466165408	-4.3816682209649915
2	-4.3750000000000000	-4.0512293340610421	-4.0502080460435446
3	-4.1875000000000000	-4.0011248294319257	-4.0009752215234897
4	-4.0937500000000000	-4.0000005618729979	-3.9999982536103080
5	-4.0468750000000000	-4.0000000000001403	-4.000000038892480
6	-4.0234375000000000	-4.0000000000000000	-3.999999999913420
23	-4.0000003576278687		
Exato	-4		

(b) Segunda raiz, iniciando em -2

N	Direto	Newton	Secante
0	-0.5000000000000000	-0.4444444444444442	-0.44960811064955775
1	-1.2500000000000000	-0.97143241244874645	-0.97161655862667629
2	-0.8750000000000000	-0.99986781339573749	-0.99984688053383497
3	-1.0625000000000000	-0.99999999708816856	-0.99999986884124925
4	-0.9687500000000000	-0.9999999999999999	-0.9999999989097121
23	-0.99999988079071045		
Exato	-1		

(c) Terceira raiz, iniciando em 3

N	Direto	Newton	Secante
0	4.5000000000000000	12.333333333333334	12.263816114793780
1	5.2500000000000000	8.6647949634167105	8.6230221715744602
2	4.8750000000000000	6.4684115191192788	6.4464803067014707
3	5.0625000000000000	5.3700303573790293	5.3621292265437184
4	4.9687500000000000	5.0328974616474005	5.0320056486584601
5	5.0156250000000000	5.0002965048237744	5.0003242730590962
6	4.9921875000000000	5.0000000244178073	5.0000004790171433
7	5.0039062500000000	5.0000000000000000	5.000000000646639
8	4.9980468750000000		5.00000000000009219
24	5.0000000596046448		
Exato	5		