

```

1  program integracaonumerica
2
3  ! Projeto 2 - Cálculo numérico
4  ! Nome: Henrique Krastins Okuti
5  ! Contato: henrique.okuti@usp.br
6
7  implicit none
8
9  real*8 integral, integraltrapezio, integralsimpson, f_a, f_b, funcao, aux2
10 real*8 diferenca_integral, logerro_trapezio, logerro_simpson, logh
11 real*8 xi, soma, funcao_montecarlo, diferenca_montecarlo, mediafuncao
12 real*8 desvio_padrao, somaquadrado, mediasomaquadrado
13 real*8 h(13), precisao_montecarlo(4)
14 integer i, aux1, n, nr
15
16 ! Começamos calculando o valor exato da integral
17
18 integral = (0.5)*(exp(1.d0)*(sin(1.d0)+cos(1.d0))-1)
19 write(*,*)"A integral deve valer: ",integral
20
21 ! Agora definimos os valores dos intervalos h a serem utilizados
22
23 i = 1          ! Variável de contagem para gerarmos o vetor h
24 n = 13         ! Final da contagem para gerarmos o vetor h
25 do while (i.LE.n)
26     h(i) = 2.d0**(-i)      ! Aqui nós iremos gerar a sequência 1/2 , 1/4, 1/8, ...
27     e armazenar no vetor h
28     i = i+1
29 enddo
30
31 ! Agora começamos os cálculos da integral pelo método do trapézio
32
33 ! A primeira etapa consiste em calcular o valor no extremos pois eles possuem pesos
34 diferentes
35
36 write(*,*)"Valores nos extremos:"
37 f_a = exp(0.d0)*cos(0.d0)      ! Aqui apenas aplicamos x = 0 para obter f(0)
38 f_b = exp(1.d0)*cos(1.d0)      ! Aqui apenas aplicamos x = 1 para obter f(1)
39 write(*,*)"f(0):",f_a
40 write(*,*)"f(1):",f_b
41 f_a = f_a/2.d0                 ! Os valores dos extremos devem ser divididos por 2,
42 f_b = f_b/2.d0                 ! então nós atualizamos os valores dividindo ambos por 2
43
44 ! A segunda etapa consiste em calcular a integral em si, vamos definir alguns
45 valores iniciais
46
47 i = 1          ! Início do contador para o vetor h(i) que contém a largura que estamos
48 integrando
49 n = 13         ! Final do contador para o vetor h(i)
50 aux1 = 1       ! Variavel auxiliar que irá serve para indicar qual trapézio estamos
51 calculando
52 integraltrapezio = 0
53
54 !Vamos gerar um arquivo para armazenar os valores de h, integral calculada e
55 diferença entre o resultado obtido e o resultado esperado
56
57 write(14,*)"          h(i)          ", "          Integral Trapezio          ", "
58 Diferença(It-I)      ", "          Log(h)          ", "          Log(Dif_Trap)          "
59 do while(i.LE.n)
60
61     aux2 = h(i)**(-1.d0) - 1.d0      ! Variável auxiliar que serve para indicar o
62     último trapézio que iremos calcular
63
64     do while(aux1.LE.int(aux2))
65         funcao = exp(float(aux1)*h(i))*cos(float(aux1)*h(i))      ! Calculamos a
66         função no trapézio indicado
67         integraltrapezio = integraltrapezio + funcao                ! A integral é a
68         soma de todos os trapézios
69     enddo
70     i = i+1
71 enddo

```

```

60         aux1 = aux1 + 1                                ! Devemos
           incrementar o contador para passar para o trapézio seguinte
61
62     enddo
63
64     integraltrapezio = h(i)*(integraltrapezio + f_a + f_b) ! Multiplicamos todos
           os termos pelo tamanho do trapézio
65     diferenca_integral = abs(integraltrapezio - integral) ! Tomamos a diferença
           absoluta entre o resultado "experimental" e o resultado "teórico"
66     logh = dlog10(h(i))
67     loggerro_trapezio = dlog10(abs(diferenca_integral))
68     write(14,*)h(i),integraltrapezio,diferenca_integral,logh,loggerro_trapezio
           ! Escrevemos os resultados relevantes no arquivo fort.14
69     write(42,*)logh,loggerro_trapezio ! Este arquivo servirá para o exercício 4
70
71     integraltrapezio = 0 ! Reiniciamos o valor da integral para a próxima largura
72     aux1 = 1 ! Reiniciamos o valor que começaremos a calcular as
           integrais
73     i = i + 1 ! Passamos para a próxima largura do trapézio
74 enddo
75
76
77 ! Agora começamos os cálculos da integral pelo método de Simpson
78
79 write(*,*)"Valores nos extremos:"
80 f_a = exp(0.d0)*cos(0.d0) ! Aqui apenas aplicamos x = 0 para obter f(0)
81 f_b = exp(1.d0)*cos(1.d0) ! Aqui apenas aplicamos x = 1 para obter f(1)
82 write(*,*)"f(0):",f_a
83 write(*,*)"f(1):",f_b
84
85 ! Vamos definir alguns valores iniciais
86
87 i = 1 ! Inicio do contador para o vetor h(i) que contém a largura que estamos
           integrando
88 n = 13 ! Final do contador para o vetor h(i)
89 aux1 = 1 ! Variavel auxiliar que irá serve para indicar qual trapézio estamos
           calculando
90 integralsimpson = 0
91
92 !Vamos gerar um arquivo para armazenar os valores de h, integral calculada e
           diferença entre o resultado obtido e o resultado esperado
93 write(15,*)"          h(i)          ","          Integral Simpson          ","
           Diferença (It-I)          ","          Log(h)          ","          Log(Dif_Simp)          "
94 do while(i.LE.n)
95
96     aux2 = h(i)**(-1.d0) - 1.d0 ! Variável auxiliar que serve para indicar o
           último trapézio que iremos calcular
97
98     do while(aux1.LE.int(aux2))
99
100         if(dmod(dfloat(aux1),2.d0).EQ.0.d0)
101             then ! Condicional para o
102                 caso par
103                     funcao = 2.d0*exp(float(aux1)*h(i))*cos(float(aux1)*h(i)) !
104                         Calculamos a função no trapézio indicado
105                     integralsimpson = integralsimpson + funcao ! A integral é
106                         a soma de todos os trapézios
107                     aux1 = aux1 + 1 ! Devemos
108                         incrementar o contador para passar para o trapézio seguinte
109
110                 else ! Condicional para
111                     o caso ímpar
112                     funcao = 4.d0*exp(float(aux1)*h(i))*cos(float(aux1)*h(i)) !
113                         Calculamos a função no trapézio indicado
114                     integralsimpson = integralsimpson + funcao ! A integral é
115                         a soma de todos os trapézios
116                     aux1 = aux1 + 1 ! Devemos
117                         incrementar o contador para passar para o trapézio seguinte
118                 endif

```

```

110     enddo
111
112     integralsimpson = (h(i)/3.d0)*(integralsimpson + f_a + f_b) ! Multiplicamos
113     todos os termos pelo tamanho do trapézio
114     diferenca_integral = abs(integralsimpson - integral) ! Tomamos a diferença
115     absoluta entre o resultado "experimental" e o resultado "teórico"
116     logh = dlog10(h(i))
117     loggerro_simpson = dlog10(diferenca_integral)
118     write(15,*)h(i),integralsimpson,diferenca_integral,logh,loggerro_simpson !
119     Escrevemos os resultados relevantes no arquivo fort.15
120     write(43,*)logh,loggerro_simpson ! Este arquivo servirá para o exercício 4
121
122     integralsimpson = 0 ! Reiniciamos o valor da integral para a próxima largura
123     aux1 = 1 ! Reiniciamos o valor que começaremos a calcular as
124     integrais
125     i = i + 1 ! Passamos para a próxima largura do trapézio
126 enddo
127
128 ! Agora vamos realizar o cálculo da integral pelo método de Monte Carlo
129
130 nr = 1
131 precisao_montecarlo(1) = 1.4453421215421258E-007
132 precisao_montecarlo(2) = 3.6133555036954590E-008
133 precisao_montecarlo(3) = 9.0333955871102489E-009
134 precisao_montecarlo(4) = 2.2583446224189174E-009
135 i = 1
136 n = 4
137 funcao_montecarlo = 0
138 mediafuncao = 0
139 desvio_padrao = 0
140 somaquadrado = 0
141 diferenca_montecarlo = 10
142
143 write(16,*)" h(i) ", " integral ", " iteracoes ", " desvio_padrao "
144 do while (i.LE.n)
145     do while (diferenca_montecarlo.GE.precisao_montecarlo(i))
146         call Random_Number(xi)
147         funcao_montecarlo = (exp(xi)*cos(xi)+ funcao_montecarlo)
148         mediafuncao = funcao_montecarlo / nr
149
150         somaquadrado = (exp(xi)*cos(xi))**(2.d0) + somaquadrado
151         mediasomaquadrado = somaquadrado / nr
152
153         desvio_padrao = sqrt((mediasomaquadrado + (mediafuncao)**(2.d0))/nr)
154
155         diferenca_montecarlo = abs(mediafuncao - integral)
156         nr = nr + 1
157     enddo
158
159     write(16,*)int(h(i+9)**(-1.d0)),mediafuncao, nr, desvio_padrao
160     i = i+1
161     funcao_montecarlo = 0 ! Reiniciamos os valores das variáveis de interesse
162     mediafuncao = 0
163     somaquadrado = 0
164     desvio_padrao = 0
165     nr = 1
166 enddo
167 end program integracaonumerica

```