

```

1  program equalgebrica
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  xesquerda, xdireita, xnovο, xvelho, aleatorio_esq, aleatorio_dir, aleatorio
10 real*8  erro, tolerancia
11 real*8  funcao, derivada, modulofuncao, coeficienteT
12 integer contador, saida, contadoraux
13
14 tolerancia = 1E-9
15 coeficienteT = 0.d0
16 saida = 1000
17 call random_number(aleatorio_esq)
18 call random_number(aleatorio_dir)
19
20 write(31,*)"    iteracoes    "," coeficiente T    "," raiz    "
21 write(311,*)"    raiz    "," funcao calculada na raiz    "
22 do while (coeficienteT.LE.(2.d0))
23
24     xesquerda = aleatorio_esq*(-1.d0)
25     xdireita = 1.d0+aleatorio_dir
26     erro = 1.d0
27     contador = 0
28     do while (contador.LE.saida)
29         xnovo = (xesquerda + xdireita)/2.d0
30         funcao = (xnovo) - tanh((xnovo)/coeficienteT)
31         if (funcao.LT.(0.d0)) then
32             xesquerda = xnovo
33         else
34             xdireita = xnovo
35         endif
36         erro = abs((xesquerda - xdireita)/2.d0)
37         contador = contador + 1
38         if (erro.LE.tolerancia) exit
39     enddo
40     write(311,*)abs(xnovo), (abs(xnovo) - tanh((abs(xnovo))/coeficienteT))
41     write(31,*)contador,coeficienteT,abs(xnovo)
42     coeficienteT = coeficienteT + 1E-3
43 enddo
44
45 ! Método de Newton-Raphson + Raiz positiva como função de T
46 ! derivada: 1 - (sech^2 (x/T)/T)
47
48 tolerancia = 1E-9
49 coeficienteT = 0.d0
50 saida = 1000
51 call random_number(aleatorio)
52
53 write(32,*)"    iteracoes    "," coeficiente T    "," raiz    "," funcao na raiz
54 calculada    "
55 do while (coeficienteT.LE.(2.d0))
56     xvelho = aleatorio
57     erro = 0.d0
58     contador = 0
59     do while (contador.LE.saida)
60         funcao = abs(xvelho) - tanh(abs(xvelho)/coeficienteT)
61         derivada = 1.d0 - (1.d0/(coeficienteT *
62             (cosh(abs(xvelho)/coeficienteT)**(2.d0))))
63         xnovo = abs(abs(xvelho) - (funcao/derivada))
64         erro = abs(xnovo - xvelho)
65         modulofuncao = abs(funcao)
66         xvelho = xnovo
67         contador = contador + 1
68         if (modulofuncao.LE.tolerancia .OR. erro.LE.tolerancia) exit
69     enddo

```

```
68         write(32,*) contador,coeficienteT,abs(xnovo), (abs((abs(xnovo) -  
        tanh((abs(xnovo))/coeficienteT)))  
69         coeficienteT = coeficienteT + 1E-3  
70     enddo  
71  
72 end program eqalgebrica
```