

```

1  /*
2   Projeto de Cálculo Numérico
3   Prof. Alexandre Roma (IME-USP)
4
5       IAG-USP
6       2015
7
8   Fábio Oliveira - 7978417
9   Jessé Stenico - 9051932
10
11  Resolução de E.D.O. pelos métodos:
12  Euler Explícito,
13  Euler Implícito e
14  Runge-Kutta Clássico.
15
16  Os resultados obtidos serão comparados com a solução analítica conhecida previamente.
17
18  Caso de calibragem: Decaimento Radioativo
19
20   $y' = -\text{Lambda} \cdot y$ 
21   $y = \exp(-\text{Lambda} \cdot t)$ 
22
23  onde "Lambda" depende do material radioativo em questão.
24  */
25  #include <stdio.h>
26  #include <math.h>
27
28  double SOL_ANALITICA (double t){ /*UTILIZADO PARA ANÁLISE DE CONVERGÊNCIA DO MÉTODO NUMÉRICO*/
29  /*CASO A SOLUÇÃO ANALÍTICA SEJA DESCONHECIDA, PREENCHER COM "return 1.0"
30  E DESCONSIDERAR OS RESULTADOS OBTIDOS PARA OS ARQUIVOS yanalitico.txt E convergencia*/
31
32      return exp(-0.5*t);
33  }
34
35  double F_XY(double t, double y){ /*EDO*/
36
37      return -0.5*y;
38  }
39
40  double dFdY(double t, double y){ /*DERIVADA DA FUNÇÃO FORNECIDA ACIMA EM RELAÇÃO A Y*/
41
42      return -0.5;
43  }
44
45  #define t_MIN 0.0
46  #define t_MAX 5.0
47  #define y_contorno SOL_ANALITICA (t_MIN)
48
49  /*UTILIZADO NO MÉTODO DE NEWTON-RAPHSON PARA EULER-IMPLICITO*/
50  #define EPSILON 1e-8
51  #define MAXIMO_ITERACAO 30
52
53  #define TETO 10 /* nPassos = 2^k , com 1<k<=TETO*/
54
55
56
57  /*NÃO ALTERAR O CÓDIGO A PARTIR DESTES PONTOS*/
58  /*NÃO ALTERAR O ARQUIVO funcoes.h*/
59
60
61  /* ** ATENÇÃO! ** */
62  /*É NECESSÁRIO QUE O ARQUIVO funcoes.h ESTEJA NA MESMA PASTA QUE O CÓDIGO-FONTE.*/
63  /*NELES ESTÃO CONTIDAS TODAS AS FUNÇÕES DE AJUSTE UTILIZADAS NO PROJETO*/
64
65  #include "funcoes.h"
66

```

```

67  int main (void){
68
69      double passos[TETO];          /*QUANTIDADE DE PASSOS (nPassos) A SER DADO*/
70      double deltaT[TETO];          /*deltaT CORRESPONDENTE A CADA nPassos*/
71
72      double (*analitico)(double);
73      double (*fxy)(double, double);
74      double (*df)(double, double);
75
76      analitico=&SOL_ANALITICA;
77      fxy=&F_XY;
78      df=&dFdY;
79
80      verifica_info( );
81
82      printf("Inicializando vetores..\n");
83
84      inicializa_vetores(passos, deltaT);/*INICIALIZA OS VETORES COM AS QUANTIDADES DE PASSOS E OS VALORES DE
dT*/
85
86                                     /*PARA CADA nPassos CORRESPONDENTE*/
87      resolve_analitico (analitico, deltaT); /*CRIA ARQUIVO COM A SOLUÇÃO ANALÍTICA*/
88
89      printf("\nCalculando aproximacoes numericas...\n");
90
91      /*AS TRÊS FUNÇÕES ABAIXO SÃO OS METODOS UTILIZADOS NO TRABALHO*/
92      euler_explicito (fxy, passos, deltaT);
93
94      euler_implicito (fxy, df, passos, deltaT);
95
96      runge_kutta (fxy, passos, deltaT);
97
98      return 0;
99  }

```