Atividades de apoio para implementações e/ou simulações de métodos numéricos com o auxílio do software GNU Octave

BOLSISTAS DO PROJETO: PAULO HENRIQUE CARDOSO DE NOVAIS e YURE MORAES PIRES
ORIENTADOR DO PROJETO: PROFESSOR GISLAN SILVEIRA SANTOS

Soluções de equações diferenciais ordinárias

1 Método de Runge-Kutta de ordem 3

O método de Runge-Kutta de ordem 3 é um método o qual tem menor erro de aproximação quando comparado ao método de Runge-Kutta de ordem 2, visto que há mais um termo (k_3) que será desenvolvido pela série de Taylor. Dessa forma, tem-se que o método de Runge-Kutta de ordem 3 pode ser descrito por:

$$y_{n+1} = y_n + h(c_1 k_1 + c_2 k_2 + c_3 k_3), \tag{1}$$

onde,

$$\begin{gathered} k_1 = f(x, y), \\ k_2 = f(x + a_2 h, y + h b_{21} k_1), \\ k_3 = f(x + h a_3, y + h(a_3 - b_3 2) k_1 + b_3 2 k_2), \\ a_2 = b_{21}, \\ a_3 = b_{31} + b_{32}. \end{gathered} \tag{2}$$

Existem infinitos métodos de Runge-Kutta de ordem 3, porém, será demonstrado em (4) uma variação chamada de **método de Heun**, onde $c_1 = \frac{1}{4}$ e $c_2 = 0$.

Obs: Para determinar as demais incógnitas de (2) é necessário resolver o sistema abaixo:

$$\left\{ \begin{array}{ccccccc} c_1 & + & c_2 & + & c_3 & = & 1 \\ c_2 a_2 & + & c_3 a_3 & = & \frac{1}{2} & & \\ c_2 a_2^2 & + & c_3 a_3^2 & = & \frac{1}{3} & & \\ c_3 b_{32} a_2 & = & \frac{1}{3} & & & & \end{array} \right. , \tag{3}$$

$$\begin{aligned} y_{n+1} &= y_n + \frac{h}{4}(k_1 + 3k_3), \\ k_1 &= f(x_n, y_n), \\ k_2 &= f(x_n + \tfrac{1}{3}h, y_n + \tfrac{1}{3}h k_1), \\ k_3 &= f(x_n + \tfrac{2}{3}h, y_n + \tfrac{2}{3}h k_2). \end{aligned} \tag{4}$$

1.1 Implementação

Apos definir o método na sessão anterior, o algoritmo será representado da seguinte forma.

```
function solucao=rk3(f,a,b,h1,y0)
  %"f" deve ser do tipo function handle @(t,y)
  %"a"    o in cio do intervalo
  %"b"    o final do intervalo
```

```
%"y0"    o pvi (y(a)=y0)
%"h1"    o passo (dever  ser um n mero inteiro , caso contr rio ser  definido um
%novo h)
%o m todo soluciona a edo para o ponto "b" (y(b))
%existem infinitos m todos de Runge-Kutta de ordem 3, o implementado aqui
%   conhecido como m todo de Heun, onde c1=1/4, c2=0 e c3=3/4.
  s=ceil((b-a)/h1);
  h=(b-a)/s
  t=(a:h:b)';
  y=[y0; zeros(length(t)-1,1)];
  k1=zeros(length(t),1);
  k2=zeros(length(t),1);
  k3=zeros(length(t),1);
  for i=1:length(t)-1
    k1(i)=f(t(i,1),y(i,1));
    k2(i)=f(t(i,1)+h/3,y(i,1)+h*k1(i)/3);
    k3(i)=f(t(i,1)+2*h/3,y(i,1)+2*h*k2(i)/3);
    y(i+1)=y(i)+h*(k1(i)+3*k3(i))/4;
  endfor
  solucao=[t y];
  plot(s,y)
```

2 Referências

1. CHAPRA, Steven C.; CANALE, Raymond P.. **Métodos Numéricos para Engenharia, 5ª Edição**. São Paulo, McGraw-Hill, 2011. 809p.

2. EATON, John W.; BATEMAN, David; HAUBERG, Soren; WEHBRING, Rik. **GNU Octave: A high-level interactive language for numerical computations, 5ª edição**. 2020. 1077p.

3. FRANCO, Neide Maria Bertoldi. **Cálculo Numérico**. São Paulo: Universidade de São Paulo, Instituto de Ciências Matemáticas e de Computação, 2006. 489 p.

4. Todos os Colaboradores. **Cálculo Numérico Um Livro Colaborativo Versão Octave**. Porto Alegre: Projeto REAMAT da UFRGS, 2020. Disponível em:
https://www.ufrgs.br/reamat/CalculoNumerico/livro-oct/main.html