UFES - CENTRO TECNOLÓGICO
DEPARTAMENTO DE INFORMÁTICA
Prof. Thomas W. Rauber

1º Trabalho de Algoritmos Numéricos (INF 09269) 2016/1

Data de entrega: veja `www.inf.ufes.br/~thomas`

Linguagem de Programação para Implementação
Octave/Matlab, preferencialmente código que seja compatível com os dois ambientes
Grupo de até dois alunos

Estudo Comparativo e Aplicação de Técnicas Numéricas de Solução de Equações Diferenciais Ordinárias.

Problema: Dado um problema de valor inicial (PVI) em uma Equação Diferencial Ordinária (EDO) de primeira ordem,

$$\begin{cases} y' = f(x, y) \\ y(x_0) = y_0, \end{cases}$$

o objetivo é criar uma sequência de estimativas $y_1, \ldots, y_n, \ldots$ dos valores verdadeiros $y(x_1), \ldots, y(x_n), \ldots$ da função $y(x)$ em pontos equidistantes $x_1, \ldots, x_n = x_0 + nh, x_{n+1} = x_n + h, \ldots$.

Se a variável dependente x não está presente no lado direito da equação, a EDO simplifica para um *sistema autônomo de primeira ordem* da forma

$$y' = f(y). \tag{1}$$

1 Teoria

Os métodos numéricos Euler, Euler Melhorado, Euler Modificado e Runge-Kutta devem ser comparados relativo ao erro entre o valor verdadeiro e estimado.

Para conseguir medir o erro, a solução analítica tem que estar disponível. Para obtê-la deve-se usar uma ferramenta de processamento matemático simbólico, neste caso o 'Maxima'. O código em seguida mostra como obter a solução, dada a função $f(x, y)$ como lado direito da EDO na forma canônica $y' = f(x, y)$. Três funções `func(x,y)` diferentes estão definidos como exemplo (a última é válida, pois sobrescreve as anteriores com o mesmo nome). A atribuição de valores (neste caso zero) às variáveis x_0 e y_0 aparece nos comandos `x0:0` e `y0:0`.

```
func(x,y) := (3*y+x)/(x-1)$
func(x,y) := (2*x+y)/(x+1)$
func(x,y) := (x+y)/(x+1)$

x0:0$
y0:0$

print("PVI: y'=",func(x,y),", y(",x0,")=",y0)$

disp("Solução analítica de PVI:")$
y(x):=''(rhs(''(ic1(ode2('diff(y,x)=func(x,y),y,x),x=x0,y=y0))));
```

Após a obtenção da expressão analítica, a função tem que ser codificada em Octave/Matlab. Neste exemplo criou-se uma função com o nome `f1` que deve ser colocada em um arquivo separado com o nome `f1.m`. Todas as outras funções usadas como exemplos devem ser colocadas em outros arquivos (`f2.m`, `f3.m`, `f4.m`, etc.)

```
function fx = f1(x)
        fx = (x+1).*log(x+1)-x;
end
```

Uma alternativa de implementação de funções são *funções anônimas* que têm a forma `@(lista_arg) expressão`, onde `@(lista_arg)` são as variáveis (parâmetros da função) e `expressão` é o lado direito da equação que define a função. Por exemplo, para implementar a função com um argumento $f(x) = \frac{1}{2}e^{2x}$, a função anônima se define como `@(x) 0.5*exp(2*x)` e para a função com dois argumentos $f(x,y) = x^2 + y^2 - 1$, a função anônima se define como `@(x,y) x*x+y*y-1`. Essa última verifica pelo sinal do resultado, se um ponto (x, y) no sistema de coordenadas Cartesianas está dentro ou fora do círculo unitário As funções anônimas podem ser colocadas no mesmo arquivo que o programa principal. Veja o exemplo anexado `exemplo1.m`.

O programa principal em Octave deve gerar a sequência de aproximações para cada método implementado de resolução numérica de EDO. Os métodos implementados devem ser pelo menos

- Euler
- Euler Melhorado
- Euler Modificado
- Um método de Runge-Kutta explícito: Runge-Kutta de ordem 3
- Um método de Runge-Kutta embutido: Bogacki-Shampine

Dois gráficos e duas tabelas devem ser gerados como saída do programa. Para gerar o gráfico, use o comando `plot` do Octave e para gerar uma tabela use os strings de formatação (similar em Octave e C). Por exemplo, o código a seguir produz uma tabela básica.

```
fprintf('%5s | %15s | %15s | %15s | %15s | %15s\n',...
        't','Afluente','Efluente','Vazamento','Ruido','Fluxo total');
for i=1:96 fprintf('='); end; fprintf('\n');
for t=0:tmax

    %%% CALCULAR afluente, efluente, vaza, ruido, Q ...

    fprintf('%5d | %15.2f | %15.2f | %15.2f | %15.2f | %15.2f\n',t,...
            afluente(t+1),efluente(t+1), vaza(t+1), ruido(t+1), Q(t+1) );
end
```

O primeiro grupo mostra a função exata $y(x)$ junto com as funções aproximadas. O segundo gráfico e tabela mostram para cada ponto x_n uma comparação do erro absoluto de cada método implementado. O exemplo a seguir mostra o gráfico da função exata e as aproximações dos métodos.

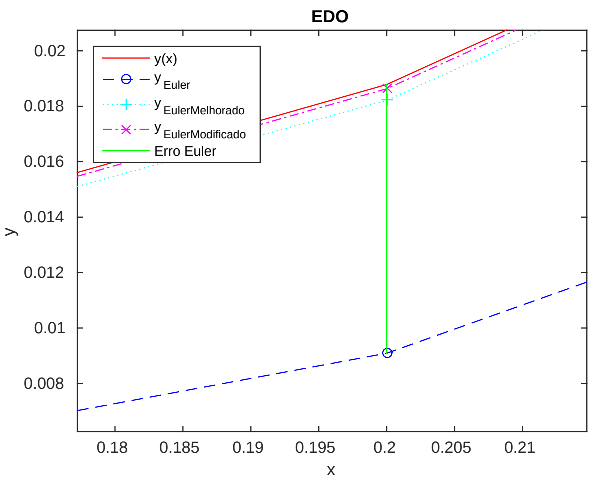

Figura 1: Gráfico comparativo (ampliação de área de interesse) que mostra a função analítica, e três variações do método de Euler. Além disso as linhas verticais mostram o erro entre a solução exata e o método de Euler simples.

Pelo menos cinco funções diferentes devem ser testadas.

2 Problema Prático

Como exemplo prático de uso de equações diferenciais ordinárias deve-se calcular o volume de líquido em um reservatório em instantes de tempo discretos. A variação[1] temporal $\dot{V} = \frac{dV}{dt}$ do volume $V\left[\mathrm{m}^3\right]$ de um líquido incompressível em um sistema fechado onde existem p fluxos $Q\left[\frac{\mathrm{m}^3}{\mathrm{s}}\right]$ de entrada Q_i^+ e q fluxos de saída Q_j^-, é simplesmente o balanço destes fluxos, definido como

$$\dot{V} = \frac{dV}{dt} = \sum_{i=1}^{p} Q_i^+ - \sum_{j=1}^{q} Q_j^- . \tag{2}$$

Note que (2) é um sistema autônomo de primeira ordem, definido em (1). Seja dada uma estação de tratamento de resíduos experimental com inicialmente 500 metros cúbicos de líquido. Considera-se o sistema durante um período de 100 unidades temporais (minutos, horas, dias, etc.). Nas definições seguintes, omitem-se as unidades. Há um fluxo afluente descrito por uma função por partes

$$Q_1(t) = \begin{cases} 110, & \text{se } t \in [0, 20) \\ 100, & \text{se } t \in [20, 40) \\ 95, & \text{se } t \in [40, 80) \\ 100, & \text{se } t \in [80, 100], \end{cases} \tag{3}$$

um fluxo efluente

$$Q_2(t) = \begin{cases} 100, & \text{se } t \in [0, 30) \\ 95, & \text{se } t \in [30, 40) \\ 105, & \text{se } t \in [40, 70) \\ 85, & \text{se } t \in [70, 100], \end{cases} \tag{4}$$

um vazamento

$$\begin{aligned} v(t) &= v_{\text{lim}} - (v_{\text{lim}} - v_{\text{nominal}}) \exp\left[\tau\theta(t_{\text{atraso}} - t)\right], \\ \theta(x) &= \begin{cases} 0, & \text{se } x \geq 0 \\ x, & \text{c.c.} \end{cases} \end{aligned} \tag{5}$$

onde $v_{\text{nominal}} = 0$ é valor do vazamento em condições de operação normal, $v_{\text{lim}} = 10$ é o valor máximo que o vazamento possa atingir assintoticamente, $t_{\text{atraso}} = 50$ é o instante de tempo em que o vazamento começa e $\tau = 0.05$ é um parâmetro temporal que controla a velocidade da evolução do vazamento a partir do valor inicial até atingir o máximo. Além disso tem um fluxo aleatório uniformemente distribuído $\epsilon(t) \in [-3, 3]$. Ajuda: Use a função `rand` do Octave para obter um número aleatório entre zero e um, e aplique um escalamento linear apropriado para forçar para o intervalo especificado.

Portanto neste cenário, a variação do volume do reservatório de (2) se instancia para

$$\dot{V} = Q_1 - Q_2 - v + \epsilon. \tag{6}$$

Tarefa: Use um ou mais métodos de resolução de uma EDO da parte teórica deste trabalho para determinar o volume do reservatório de (6). Considere somente um único passo do instante de tempo t_n para o próximo instante de tempo t_{n+1} para estimar $V(t_{n+1})$, dados o volume atual $V(t_n)$ e os fluxos atuais $Q_1(t_n), Q_2(t_n), v(t_n), \epsilon(t_n)$. O resultado, usando Euler simples, é apresentado na figura 2.

Referências

- Maxima
  `http://maxima.sourceforge.net`
- Métodos de Runge-Kutta
  `https://pt.wikipedia.org/wiki/M%C3%A9todo_de_Runge-Kutta`
- Lista de Métodos de Runge-Kutta
  `https://en.wikipedia.org/wiki/List_of_Runge%E2%80%93Kutta_methods`

[1] Para uma variável y, a sua derivada temporal, ou seja, a taxa de mudança ao longo do tempo na notação de Leibniz dy/dt pode ser expressa alternativamente na notação de Newton $\dot{y}$.

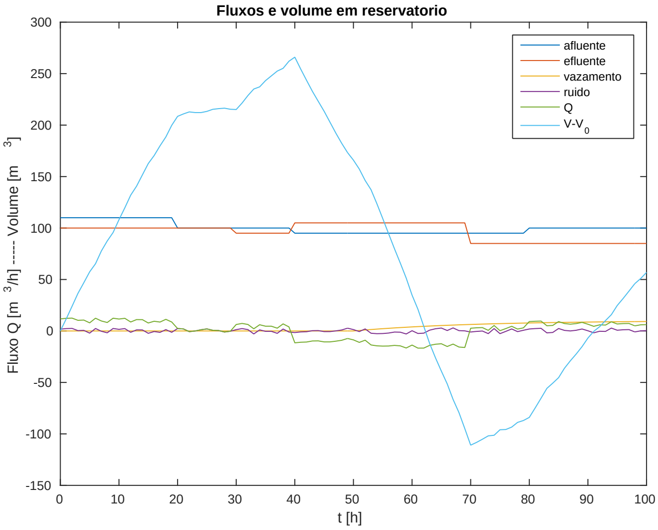

Figura 2: Evolução temporal dos fluxos e do volume do reservatório. Mostra-se o volume $V - V_0$ relativo ao volume inicial $V_0 = V(t = 0) = 500$.

- Sistema Autônomo
  `https://pt.wikipedia.org/wiki/Sistema_aut%C3%B4nomo_%28matem%C3%A1tica%29`

Elaboração: O resultado deve ser código em Octave que produz as tabelas e gráficos para o problema apresentado. Os alunos obrigatoriamente devem fornecer exemplos testados. O usuário (neste caso o professor) não deve ter o trabalho de digitar nada, além da linha de comando no Octave que executa o(s) programa(s) principa(l)(ais).

O produto final deve ser um arquivo no formato `zip` com a seguinte sintaxe: `aluno1+aluno2.zip`. O aluno se responsabiliza pelo envio e recepção correta. Em caso de problemas maiores de tráfego de rede (serviços UFES fora do ar), o aluno deve mandar novamente o arquivo original (encaminhamento da mensagem original) quando o serviço voltar. O arquivo `aluno1+aluno2.zip` deve conter uma **única** pasta com o nome `aluno1+aluno2`. Duas subpastas devem conter o código fonte e a documentação do projeto. O arquivo `aluno1+aluno2.zip` deve ser mandado como anexo **exclusivamente** copiando a hiperligação seguinte no browser ou cliente de E-mail:

`mailto:thomas.rauber@ufes.br?subject=Algoritmos%20Numericos%202016/2:%20Entrega%20de%20trabalho%20`

A documentação deve ser em forma de descrição de projeto com os, preferencialmente gerado por LaTeX, contendo os seguintes tópicos:

- Capa do Projeto
  - Título
  - Autoria
  - Data
  - Resumo
- Introdução
- Objetivos
- Metodologia
- Resultados e Avaliação
- Referências Bibliográficas

Última atualização: 1 de abril de 2016, 16:15

Bom trabalho!