Instituto Tecnológico de Aeronáutica – ITA

Matemática Computacional – CCI-22

Laboratório 9 – Problemas de Valor Inicial com Métodos de Passo Simples

Professor: Marcos Ricardo Omena de Albuquerque Maximo

21 de maio de 2019

1 Tarefas

1.1 Implementação

Implementar as seguintes funções em MATLAB (cada uma em um arquivo .m separado):

1. `y = PVIEuler(f, x, y0)`: utiliza o Método de Euler para resolver o problema de valor inicial (PVI) associado ao sistema de equações diferenciais ordinárias (EDOs) $\mathbf{p}'(t) = f(t, \mathbf{p})$ e à condição inicial $\mathbf{p}(t_0 = a) = \mathbf{y}_0$, no intervalo $[a = x_0, b = x_n]$. A solução é calculada nos pontos contidos no vetor coluna $\mathbf{x} = [x_0, x_1, x_2, \ldots, x_n]^T$, em que $h = x_{i+1} - x_i$ (i.e. considera-se pontos igualmente espaçados no intervalo $[a, b]$). A função vetorial $f(t, \mathbf{p})$ é definida como um vetor linha:

$$f(t, \mathbf{p}) = \begin{bmatrix} f_1(t, \mathbf{p}) & f_2(t, \mathbf{p}) & \cdots & f_l(t, \mathbf{p}) \end{bmatrix} \quad (1)$$

em que:

$$\mathbf{p}(t) = \begin{bmatrix} p_1(t) & p_2(t) & \cdots & p_l(t) \end{bmatrix} \quad (2)$$

Desse modo, o retorno da função é a matriz $\mathbf{y}$ definida da seguinte forma:

$$\mathbf{y} = \begin{bmatrix} \mathbf{p}(x_0) \\ \mathbf{p}(x_1) \\ \mathbf{p}(x_2) \\ \vdots \\ \mathbf{p}(x_n) \end{bmatrix} = \begin{bmatrix} p_1(x_0) & p_2(x_0) & \cdots & p_l(x_0) \\ p_1(x_1) & p_2(x_1) & \cdots & p_l(x_1) \\ p_1(x_2) & p_2(x_2) & \cdots & p_l(x_2) \\ \vdots & \vdots & \ddots & \vdots \\ p_1(x_n) & p_2(x_n) & \cdots & p_l(x_n) \end{bmatrix} \quad (3)$$

2. `y = PVIHeun(f, x, y0)`: utiliza o Método de Heun para resolver o problema de valor inicial (PVI) associado ao sistema de equações diferenciais ordinárias (EDOs) $\mathbf{p}'(t) = f(t, \mathbf{p})$ e à condição inicial $\mathbf{p}(t_0 = a) = \mathbf{y}_0$, no intervalo $[a = x_0, b = x_n]$. A solução é calculada nos pontos contidos no vetor coluna $\mathbf{x} = [x_0, x_1, x_2, \ldots, x_n]^T$,

em que $h = x_{i+1} - x_i$ (i.e. considera-se pontos igualmente espaçados no intervalo $[a, b]$).

3. `y = PVIRungeKutta4(f, x, y0)`: utiliza o Método de Runge-Kutta de 4ª ordem para resolver o problema de valor inicial (PVI) associado ao sistema de equações diferenciais ordinárias (EDOs) $\mathbf{p}'(t) = f(t, \mathbf{p})$ e à condição inicial $\mathbf{p}(t_0 = a) = \mathbf{y}_0$, no intervalo $[a = x_0, b = x_n]$. A solução é calculada nos pontos contidos no vetor coluna $\mathbf{x} = [x_0, x_1, x_2, \ldots, x_n]^T$, em que $h = x_{i+1} - x_i$ (i.e. considera-se pontos igualmente espaçados no intervalo $[a, b]$).

1.2 Análise

1. Considere a equação diferencial ordinária do oscilador de Van der Pol:

$$\frac{d^2y}{dx^2} - \mu \left(1 - y^2\right) \frac{dy}{dx} + y = 0 \tag{4}$$

Fixando $\mu = 1$, resolva PVIs associados a esta EDO utilizando o Método de Runge-Kutta de 4ª ordem. Considere o seguinte conjunto de condições iniciais: $\{[-2, -2], [-2, 2], [-1, -1], [1, 1], [2, 2], [2 - 2]\}$. Calcule a solução no intervalo $[0, 100]$ com passo $h = 0,01$. Trace num único gráfico os retratos de fase (gráfico de $y \times y'$) das soluções dos PVIs para cada condição inicial. Analisando os gráficos, comente o que acontece com as soluções desta EDO quando $x \to \infty$, independentemente da condição inicial.

Observação: a função `VanDerPol`, fornecida juntamente com o roteiro, traça o gráfico pedido neste item, desde que a função `PVIRungeKutta4` esteja implementada.

2. Considere o seguinte PVI:

$$y'' + y = 0, [y(0), y'(0)] = [1, 1] \tag{5}$$

cuja solução analítica é:

$$y^*(x) = \cos(x) + \sin(x) \tag{6}$$

Resolva este PVI numericamente no intervalo $[0, 100]$ com $h = 0,05$, utilizando os 3 métodos de passo simples implementados. Então, compare os resultados obtidos numericamente com a solução analítica. Para isto, calcule o erro absoluto entre cada solução numérica e a analítica, em cada ponto. Então, trace os erros obtidos num único gráfico, utilizando escala logarítmica para o eixo Y. O uso de escala logarítmica é motivado pelo fato de que um método possui erros ordens de grandeza menores que outro. O resultado é o esperado? Comente.

Observação: a função `ComparacaoPassoSimples`, fornecida juntamente com o roteiro, traça o gráfico pedido neste item, desde que as funções `PVIEuler`, `PVIHeun` e `PVIRungeKutta4` estejam implementadas.

2 Instruções

- A primeira etapa do processo de correção consistirá em submeter as funções implementadas a vários casos de teste de forma automatizada. Assim, os cabeçalhos das funções devem ser seguidos **rigorosamente**. Arquivos .m com os nomes destas funções e os cabeçalhos já implementados foram fornecidos juntamente com este roteiro. Dê preferência a implementar seu laboratório a partir destes arquivos .m fornecidos para evitar erros.

- **Não** é permitido o uso de funções ou comandos prontos do MATLAB que realizem toda a funcionalidade atribuída a uma certa função. Entretanto, o uso destas funções para verificação das implementações realizadas é encorajado. Em caso de dúvida quanto à permissão de uso de alguma função ou comando, recomenda-se consultar o professor.

- Não é necessário reimplementar métodos de laboratórios anteriores. Assim, funções implementadas em laboratórios anteriores podem ser utilizadas. Caso prefira, também é permitido utilizar funções equivalentes do MATLAB.

- Não é necessário se preocupar com verificação dos dados de entrada.

- Os arquivos .m implementados devem ser entregues juntamente com um relatório.

- No relatório, não é necessário demonstrar que as funções implementadas funcionam corretamente (isto será verificado separadamente). Basta incluir resultados e conclusões relativos à **Análise**.

- Para facilitar a correção da Análise, inclua os gráficos diretamente no relatório. Nos gráficos, coloque títulos, legendas e nomes nos eixos.

3 Dicas:

- Utilize o operador `@` para definir a função derivada passa como argumento nas funções a serem implementadas.

- Defina `f` como função de duas variáveis, em que o primeiro argumento é um escalar e o segundo um vetor, por exemplo:

  ```
  f(x,y) = @(x,y) [y(2), mu * (1 - y(1)^2) * y(2) - y(1)]
  ```

- Cuidado com o formato requerido para vetores e matrizes para evitar problemas na correção automática.

- Para criar gráficos com alta qualidade em formato PNG para inclusão em arquivos do Microsoft Word, utilize o comando: `print -dpng -r300 grafico.png`.

- Se utilizar LaTeX, dê preferência para incluir gráficos em formato vetorizado. No Linux, utilizando `pdflatex`, você pode gerar um gráfico em formato EPS usando "`print -depsc2 grafico.eps`" e depois convertê-lo para PDF usando o comando de terminal "`epstopdf grafico.eps`". O arquivo PDF é aceito pelo `pdflatex`.