



Capítulo 24

Equações Diferenciais

Resumo do Capítulo

- **MATLAB** resolve numericamente uma grande classe de equações diferenciais incluindo equações ordinárias de várias ordens e equações diferenciais parciais, com valores iniciais e valores de contorno.
- Aqui nos restringiremos a métodos para resolver equações diferenciais ordinárias, dados valores iniciais da função incógnita em um ponto.
- Veja *Help* → **MATLAB** → *Mathematics* → *Differential Equations* → *Initial Value Problems for ODEs and DAEs*

O Problema

- Sistemas de equações diferenciais ordinárias têm a forma

$$\begin{array}{ll} y_1' = f_1(t, y_1, y_2, \dots, y_n) & y_1(t_0) = y_{1_0} \\ y_2' = f_2(t, y_1, y_2, \dots, y_n) & y_2(t_0) = y_{2_0} \\ \vdots & \vdots \\ y_n' = f_n(t, y_1, y_2, \dots, y_n) & y_n(t_0) = y_{n_0} \end{array}$$

onde $y_i(t)$ são funções na variável t das quais queremos estimar valores, y_i' , $1 \leq i \leq n$, são suas derivadas dy_i/dt , e y_{i_0} é o valor inicial da i -ésima função.

- Em notação vetorial

$$y' = f(t, y), \quad y(t_0) = y_0.$$

Um exemplo

- Considere a equação diferencial ordinária de segunda ordem conhecida como equação de van der Pol:

$$x'' - \mu(1 - x^2)x' + x = 0,$$

onde μ é um parâmetro positivo.

- Podemos transformá-la no formato descrito fazendo $y_1 = x$ e $y_2 = dx/dt$. A equação é reescrita como o sistema de duas equações

$$\begin{aligned}y_1' &= y_2 \\y_2' &= \mu(1 - y_1^2)y_2 - y_1.\end{aligned}$$

Codificando o sistema

- Os algoritmos (*solver*) padrão do **MATLAB** para resolver tais sistemas de equações diferenciais são *ode45* e *ode15s*, nessa ordem.
- Em ambos os casos, o sistema deve ser codificado como uma função $y_{linha} = odefile(t, y)$ em um arquivo M *odefile.m*, onde *ylinha* é um vetor-coluna de n elementos.
- Essa função recebe valores para t e y , seus argumentos, e devolve no vetor *ylinha* o valor das n derivadas y'_i nesses valores.
- Veja o arquivo *vdpol.m* com a descrição do sistema de equações correspondente à função de van der Pol.

Resolvendo o sistema

- O sistema é finalmente resolvido fazendo-se uma chamada à função `ode45(odefun,tspan,y0)`, que integra o sistema descrito em `odefun` no intervalo `tspan = [t0, tf]`, com valores iniciais em `y0`.
- Veja e execute o arquivo `mm2401.m` no repositório de arquivos do livro. Note que o `ode45` escolheu automaticamente dividir o intervalo `[0, 20]` em 333 pontos.
- É possível dizer ao `solver` em quais pontos queremos resolver o sistema, especificando em `tspan` os pontos intermediários e não somente os extremos do intervalo. Entretanto, esses valores podem ser interpolações dos valores calculados sobre os pontos que o `solver` de fato escolheu para fazer os cálculos. Essa decisão é do `solver`, se nada for passado como opção.
- Há um sem-número de opções que podem ser passadas ao `solver`. Veja `help odeset`.