

Equações Diferenciais Ordinárias (EDOs)

Prof. Doherty Andrade --doherty200@hotmail.com

Introdução

O Maple tem o comando **dsolve** para determinar explicitamente a solução de edos's. Para ver outras facilidades digite ?dsolve

```
[> ?dsolve;
```

Antes de iniciar precisamos abrir o pacote de ferramentas para trabalhar com EDOS.

```
[> restart;
```

```
[> with(DEtools);
```

Exemplo 1

Vejamos um exemplo. Considere o de PVI a seguir:

$$\begin{cases} y'(x) + y = x \\ y(1) = 2. \end{cases}$$

Vamos entrar com os dados: a equação e a condição inicial.

```
[> eq1 := diff(y(x), x) = x - y(x);
```

$$eq1 := \frac{d}{dx} y(x) = x - y(x) \quad (2.1)$$

```
[> ci := y(1) = 2;
```

$$ci := y(1) = 2 \quad (2.2)$$

```
[> dsolve(eq1, y(x));
```

$$y(x) = x - 1 + e^{-x} C1 \quad (2.3)$$

Esta é a solução geral. Agora vamos incluir a condição inicial e obter a solução exata (e única) do PVI. A constante C1 vai ser ajustada.

```
[> solexata := dsolve( {eq1, ci} , y(x) );
```

$$solexata := y(x) = x - 1 + \frac{2e^{-x}}{e^{-1}} \quad (2.4)$$

Para checar se uma função é solução da EDO podemos utilizar o comando odetest.

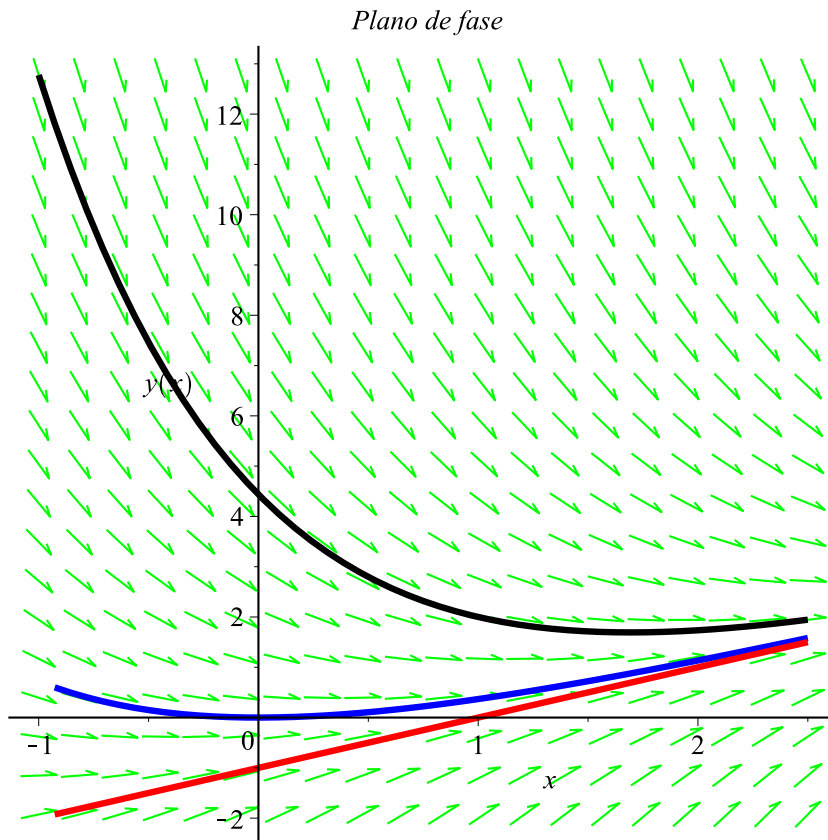
```
[> odetest(solexata, eq1);
```

$$0$$

(2.5)

Agora apresentamos o plano de fase da EDO

```
> phaseportrait(D(y)(x) = x - y(x), y(x), x = -1 ..2.5, [[y(0) = 0], [y(1) = 2], [y(0) = -1]], title = 'Plano de fase', colour = green, linecolor = [blue, black, red]);
```



Exemplo 2

Agora vamos apresentar um exemplo com EDO de ordem 2.

Considere a EDO de ordem 2 juntamente a duas condições iniciais (PVI de ordem 2) a seguir:

$$\begin{cases} y''(x) - 2y'(x) + 10y = 0 \\ y(0) = 4 \\ y'(0) = 1. \end{cases}$$

Vamos entrar com a equação e com as condições iniciais.

$$\begin{aligned} & \text{> eq2 := diff(y(x), x\$2) - 2*diff(y(x), x) + 10*y(x);} \\ & \qquad \qquad \qquad eq2 := \frac{d^2}{dx^2} y(x) - 2 \left(\frac{d}{dx} y(x) \right) + 10 y(x) \qquad (3.1) \\ & \text{> ci2 := y(0) = 4, D(y)(0) = 1;} \\ & \qquad \qquad \qquad ci2 := y(0) = 4, D(y)(0) = 1 \qquad (3.2) \\ & \text{> dsolve({eq2, ci2}, y(x));} \\ & \qquad \qquad \qquad y(x) = -e^x \sin(3x) + 4e^x \cos(3x) \qquad (3.3) \end{aligned}$$

O mesmo pode ser feito com edos de ordem superior.

Exemplo 3

Agora vamos apresentar um exemplo de PVI de ordem 3. O Maple tem dificuldades de obter explicitamente solução de EDOs de ordem superior. Neste exemplo, nenhuma solução é apresentada, mas utilizando métodos numéricos podemos obter boas aproximações da solução explícita. Este é o caso de exemplo.

$$\begin{aligned} & \text{> eq3 := diff(y(x), x\$3) - 2*diff(y(x), x\$2) - 1*diff(y(x), x) + 2*y(x) = 1;} \\ & \qquad \qquad \qquad eq3 := \frac{d^3}{dx^3} y(x) - 2 \left(\frac{d^2}{dx^2} y(x) \right) - \left(\frac{d}{dx} y(x) \right) + 2 y(x) = 1 \qquad (4.1) \\ & \text{> ci3 := y(0) = 4, D(y)(0) = 1, D@@2(y)(0) = 2;} \\ & \qquad \qquad \qquad ci3 := y(0) = 4, D(y)(0) = 1, D^{(2)} y(0) = 2 \qquad (4.2) \\ & \text{> dsolve({eq3, ci3}, y(x)); # Não retorna nada, mas podemos tentar um método numérico.} \\ & \text{> Nsol := dsolve({} \\ & \qquad (D@@3)(y)(t) = 1 - 2*y(t) + 1*D(y)(t) + 2*(D@@2)(y)(t), \\ & \qquad y(0) = 4, \\ & \qquad D(y)(0) = 1, \\ & \qquad (D@@2)(y)(0) = 2 \\ & \qquad }, 'numeric');} \\ & \qquad \qquad \qquad Nsol := \text{proc}(x_rkf45) \dots \text{end proc} \qquad (4.3) \end{aligned}$$

```

[> #Agora vamos apresentar a solução numérica em alguns pontos.
[> Nsol(0); Nsol(1.0);
       $\left[ t=0., y(t)=4., \frac{d}{dt} y(t)=1., \frac{d^2}{dt^2} y(t)=2. \right]$ 
       $\left[ t=1.0, y(t)=5.32819734477651, \frac{d}{dt} y(t)=0.397910900983833, \frac{d^2}{dt^2} y(t)= \right.$ 
       $\left. -6.25538534558275 \right]$ 
[>

```

(4.4)

Exemplo 4 (sistema 2 por 2)

```

[> restart :
[> with(DEtools) :
[> with( plots ) :
[> with( LinearAlgebra ) :

```

Consideremos o seguinte sistema de EDOs típico predador-presa.

$$\begin{cases} \frac{dx}{dt} = 3x - 1.4xy \\ \frac{dy}{dt} = -y + 0.5xy, \\ x(0) = 2, \\ y(0) = 2. \end{cases}$$

Ilustramos o campo de direções do sistema. Um campo de direções deste sistema é o gráfico no plano xy em que as flechas mostram as inclinações dadas por

$$\frac{dy}{dx} = \frac{3x - 1.4xy}{-y + 0.5xy}$$

e cujas direções são determinadas pelo crescimento de t. Assim, as flechas são tangentes às curvas

soluções.

```
> sistema2por2 := diff(x(t), t) = 3.0 · x(t) - 1.4 · x(t) · y(t), diff(y(t), t) = -1.0 · y(t) + 0.5
```

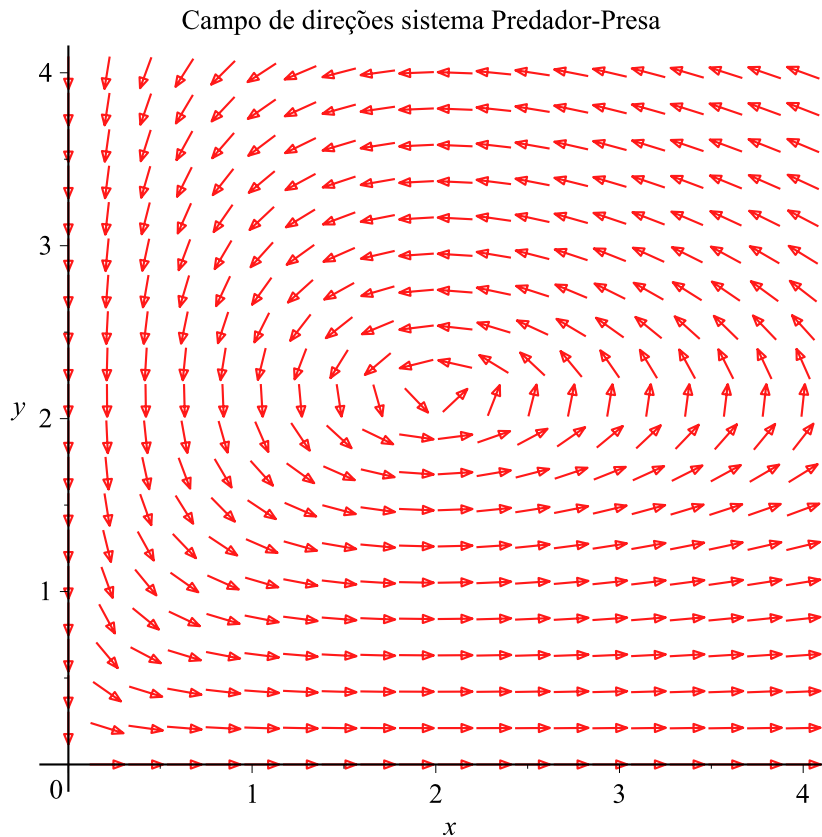
$$\text{sistema2por2} := \frac{d}{dt} x(t) = 3.0 x(t) - 1.4 x(t) y(t), \frac{d}{dt} y(t) = -1.0 y(t) + 0.5 x(t) y(t) \quad (5.1)$$

```
> fcoes := x(t), y(t);
```

$$fcoes := x(t), y(t) \quad (5.2)$$

```
> DEplot( [sistema2por2], [fcoes], t = 0 .. 1, x = 0 .. 4, y = 0 .. 4, arrows = medium, title
```

```
    = "Campo de direções sistema Predador-Presa" );
```



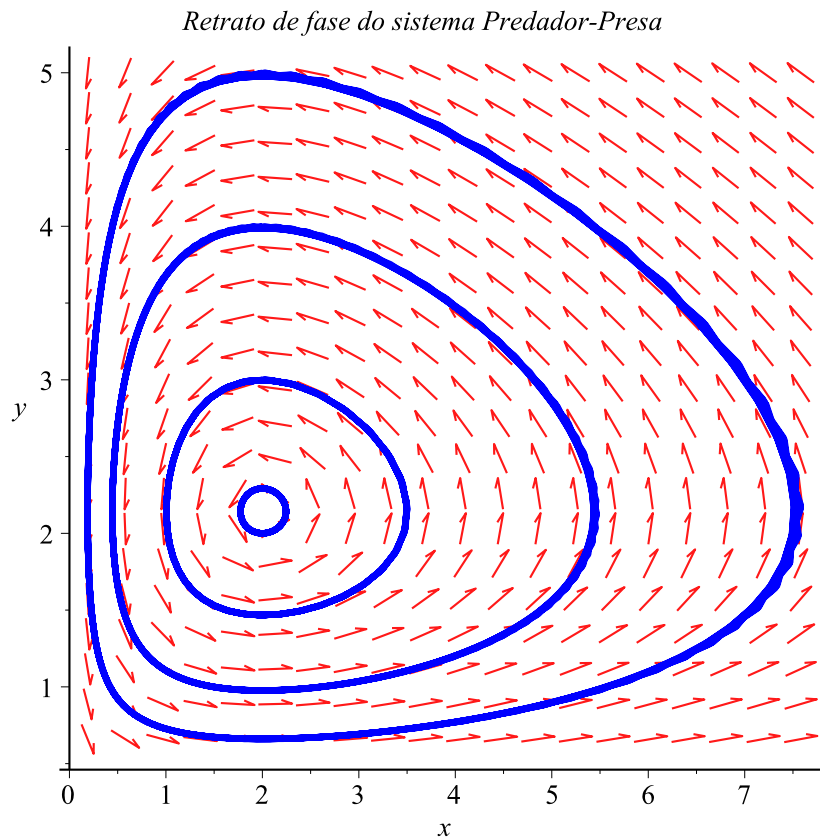
Agora vamos incluir as condições iniciais.

```
> ic_sist := [x(0) = 2.0, y(0) = i] $ i = 2..5;
```

$$ic_sist := [x(0) = 2.0, y(0) = 2], [x(0) = 2.0, y(0) = 3], [x(0) = 2.0, y(0) = 4], [x(0) = 2.0, y(0) = 5] \quad (5.3)$$

As soluções que passam por estes pontos são incorporadas no mesmo gráfico.

```
> DEplot( [ sistema2por2], [ fcoes], t=0..30, [ ic_sist], arrows = SMALL, stepsize = 0.1,
  linecolor = BLUE,
  title = `Retrato de fase do sistema Predador-Presa` );
```

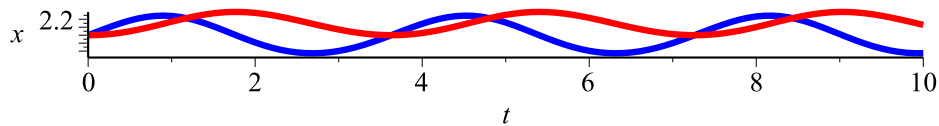


As curvas soluções para a presa (coelho) e para o predador (raposa) que passam pelos pontos apresentados pelas condições iniciais

In the time-domain, solution curves for rabbit and fox populations passing through the first of these initial points can be obtained by specifying the optional argument scene. In Figure 1.6, the rabbit population $r(t)$ is shown in blue, whereas the fox population $f(t)$ is shown in red.

```
> presa := DEplot( [ sistema2por2], [ fcoes], t=0..10, [ ic_sist[ 1 ]], stepsize = 0.1, linecolor
  = BLUE, scene = [ t, x ] ) :
> predador := DEplot( [ sistema2por2], [ fcoes], t=0..10, [ ic_sist[ 1 ]], stepsize = 0.1, linecolor
  = RED, scene = [ t, y ] ) :
> display( [ presa, predador], scaling = constrained, title = "presa (azul) e predado(vermelho)" );
```

presa (azul) e predado(vermelho)



>

Métodos Numéricos para EDOs

O comando `dsolve` com as opções `numeric` ou `type=numeric` para PVI problemas de valor inicial determina numericamente a solução da EDO ou do sistema de EDOs. Pode-se escolher o método numérico escrevendo `method=numericmethod`, onde **numericmethod** é um destes [rkf45](#), [ck45](#), [rosenbrock](#), [dverk78](#), [lsode](#), [gear](#), [taylorseries](#), ou [classical](#).

O default dos métodos numéricos para EDO é o método de Fehlberg - Runge-Kutta de ordem 4.5 (RKF45).

Vejamos um exemplo.

> `eq5 := diff(y(x), x) + 10·y(x);`

$$eq5 := \frac{d}{dx} y(x) + 10 y(x) \quad (6.1)$$

> ci5 := y(0) = 4;

$$ci5 := y(0) = 4 \quad (6.2)$$

> Sol_Euler := dsolve({eq5, ci5}, y(x), type = numeric, method = classical, stepsize = 0.1);

Sol_Euler := proc(x_classical) ... end proc (6.3)

> Sol_ABMoulton := dsolve({eq5, ci5}, y(x), type = numeric, method = classical[abmoulton], stepsize = 0.1);

Sol_ABMoulton := proc(x_classical) ... end proc (6.4)

> Sol_ABMoulton(0.5);

$$[x = 0.5, y(x) = -0.00489807128906250] \quad (6.5)$$

Mais um exemplo

> Nsol := dsolve({ (D@@3)(y)(t) = 1 - 2·y(t) + 1*D(y)(t) + 2*(D@@2)(y)(t),
y(0) = 4, D(y)(0) = 1, (D@@2)(y)(0) = 2 }, 'numeric');

Nsol := proc(x_rkf45) ... end proc (6.6)

> Nsol(0); Nsol(1.0);

$$\left[t = 0., y(t) = 4., \frac{d}{dt} y(t) = 1., \frac{d^2}{dt^2} y(t) = 2. \right]$$

$$\left[t = 1.0, y(t) = 5.32819734477651, \frac{d}{dt} y(t) = 0.397910900983833, \frac{d^2}{dt^2} y(t) = -6.25538534558275 \right] \quad (6.7)$$

>