Maple

Exact solutions

The command dsolve encountered in Section 3.7 can be used for systems too. A convenient way to find the exact solution of a system or of an initial value problem is to first write each equation separately and then apply the command dsolve, with or without initial conditions.

EXAMPLE 1 To solve the initial value problem

$$\begin{cases} x' = x + y - z \\ y' = -x + y + z, \\ z' = x - y + z \end{cases} \qquad x(0) = 1,\ y(0) = -1,\ z(0) = 0,$$

proceed as shown below. (Here Maple has been set to echo input and show results in typeset format.) Write first the three equations

```
> e1:=diff(x(t),t)=x(t)+y(t)-z(t);
```

$$e1 := \frac{\partial}{\partial t}x(t) = x(t) + y(t) - z(t)$$

```
> e2:=diff(y(t),t)=-x(t)+y(t)+z(t);
```

$$e2 := \frac{\partial}{\partial t}y(t) = -x(t) + y(t) + z(t)$$

```
> e3:=diff(z(t),t)=x(t)-y(t)+z(t);
```

$$e3 := \frac{\partial}{\partial t}z(t) = x(t) - y(t) + z(t)$$

and then use the command dsolve as follows:

```
> dsolve({e1,e2,e3,x(0)=1,y(0)=-1,z(0)=0},
  {x(t),y(t),z(t)});
```

Maple will display the solution

$$\{z(t) = \frac{2}{3}e^t\sqrt{3}\sin(t\sqrt{3}),\ y(t) = \frac{1}{3}e^t(-3\cos(t\sqrt{3}) - \sqrt{3}\sin(t\sqrt{3})),$$
$$x(t) = -\frac{1}{3}e^t(-3\cos(t\sqrt{3}) + \sqrt{3}\sin(t\sqrt{3}))\}$$

erical solutions

With the help of Maple we will numerically solve the initial value problem

$$\begin{cases} x' = tx - y + z \\ y' = 3x + y - z, \\ z' = x + y - tz \end{cases} \qquad x(0) = y(0) = 1,\ z(0) = 0, \tag{1}$$

in the interval [0, 1.5] with step size 0.1. We will first use Euler's method and then the second-order Runge-Kutta method.

Euler's method To numerically solve the initial value problem (1) with the help of Euler's method, we first write the three functions defining the vector field and then assign the initial conditions t_0, x_0, y_0, z_0, the step size h, and the number of steps n. The do command then uses Euler's formula to iterate all the steps.

```
> f1:=(t,x,y,z)->t*x-y+z;
```

$$f1 := (t, x, y, z) \to tx - y + z$$

```
> f2:=(t,x,y,z)->3*x+y-z;
```

$$f2 := (t, x, y, z) \to 3x + y - z$$

```
> f3:=(t,x,y,z)->x+y-t*z;
```

$$f3 := (t, x, y, z) \to x + y - tz$$

```
> t0:=0: x0:=1: y0:=1: z0:=0:
> h:=0.1:
> n:=15:
> t:=t0: x:=x0: y:=y0: z:=z0:
> for i from 1 to n do
  u:=f1(t,x,y,z): v:=f2(t,x,y,z):   w:=f3(t,x,y,z):
  x:=x+h*u: y:=y+h*v: z:=z+h*w:
  t:=t+h:
  print(t,x,y,z);
  od:
```

Maple displays the numerical result below:

```
 .1, .9, 1.4, .2
 .2, .789, 1.79, .428
 .3, .66858, 2.1629, .67734
 .4, .5400814, 2.512030, .9401678
 .5, .404498436, 2.83124064, 1.207772228
 .6, .2623765166, 3.114937012, 1.470957524
 .7, .1137211588, 3.358047916, 1.720431426
 .8, -.0420800091, 3.555925913, 1.947178134
 .9, -.2063211877, 3.704176688, 2.142788474
1.0, -.3810289160, 3.798419153, 2.299723061
1.1, -.5690014168, 3.833980087, 2.411489779
1.2, -.7738406035, 3.805528693, 2.472723770
1.3, -.9999819682, 3.706657004, 2.479165727
1.4, -1.252728752, 3.529411541, 2.427541686
1.5, -1.538297763, 3.263779901, 2.315354129
```

The second-order Runge-Kutta method In case of the second-order Runge-Kutta method, the first part of the program is the same as for Euler's method. The difference appears in using the command do. The changes reflect formula (5) of Section 4.5. The program works as follows.

```
> for i from 1 to n do
  a:=f1(t,x,y,z): b:=f2(t,x,y,z): c:=f3(t,x,y,z):
```

```
u:=x+h*a: v:=y+h*b: w:=z+h*c:
  x:=x+h*(a+f1(t+h,u,v,w))/2:
  y:=y+h*(b+f2(t+h,u,v,w))/2:
  z:=z+h*(c+f3(t+h,u,v,w))/2:
  t:=t+h:
  print(t,x,y,z);
  od:
```

Maple displays the numerical result below:

```
 .1, .8945000000, 1.395000000, .2140000000
 .2, .7807439500, 1.773058750, .4513366500
 .3, .6603398859, 2.127972638, .7042301039
 .4, .5344876539, 2.454202842, .9641548238
 .5, .4038917521, 2.746862558, 1.222215625
 .6, .2686896235, 3.001637640, 1.469518363
 .7, .1283964925, 3.214640732, 1.697508652
 .8, -.0181336987, 3.382201772, 1.898255436
 .9, -.1727331441, 3.500599904, 2.064661404
1.0, -.3379339723, 3.565743174, 2.190588593
1.1, -.5169751448, 3.572802772, 2.270894318
1.2, -.7138073142, 3.515808053, 2.301379180
1.3, -.9331009066, 3.387207087, 2.278654656
1.4, -1.180262830, 3.177395277, 2.199942240
1.5, -1.461467158, 2.874211673, 2.062818978
```

To draw graphs of x, y, and z using the Runge-Kutta method, we can proceed as follows:

```
> f1:=(t,x,y,z)->t*x-y+z;
```

$$f1 := (t, x, y, z) \rightarrow tx - y + z$$

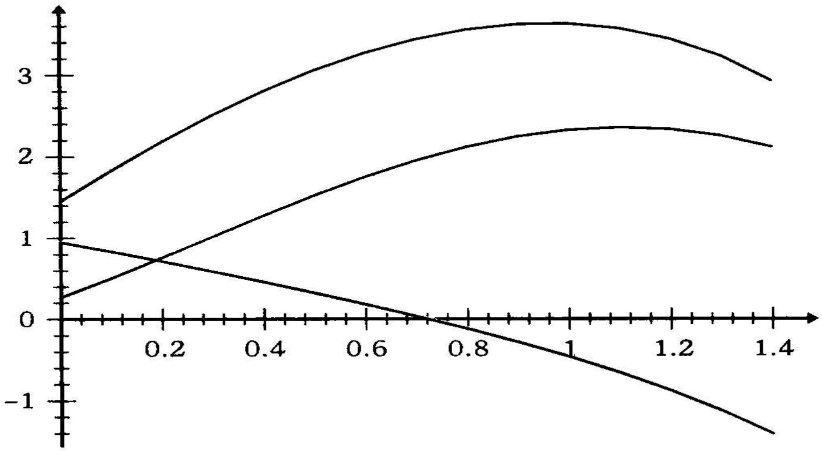
```
> f2:=(t,x,y,z)->3*x+y-z;
```

$$f2 := (t, x, y, z) \rightarrow 3x + y - z$$

```
> f3:=(t,x,y,z)->x+y-t*z;
```

$$f3 := (t, x, y, z) \rightarrow x + y - tz$$

```
> t0:=0: x0:=1: y0:=1: z0:=0:
> M0:=[t0,x0]: N0:=[t0,y0]: P0:=[t0,z0]:
> h:=0.1:
> n:=15:
> t:=t0: x:=x0: y:=y0: z:=z0: M:=M0: N:=N0: P:=P0:
> for i from 1 to n do
  a:=f1(t,x,y,z): b:=f2(t,x,y,z): c:=f3(t,x,y,z):
  u:=x+h*a: v:=y+h*b: w:=z+h*c:
  x:=x+h*(a+f1(t+h,u,v,w))/2:
  y:=y+h*(b+f2(t+h,u,v,w))/2:
  z:=z+h*(c+f3(t+h,u,v,w))/2:
  M:=(M,[t,x]): N:=(N,[t,y]): P:=(P,[t,z]):
  t:=t+h:
  od:
PLOT(CURVES([M],[N],[P]));
```



The graphs of the solutions are drawn in Figure 4.6.1. Compare these graphs with the numerical data obtained with the help of the Runge-Kutta method.

Maple is also capable of finding the eigenvalues and eigenvectors of a given matrix. For this it is endowed with a linear algebra package, which we must call with a specific command. Then we need to define the matrix and finally use the commands `eigenvalues` and `eigenvectors`.

EXAMPLE 2 To obtain the eigenvalues and eigenvectors of the matrix

$$\mathbf{A} = \begin{pmatrix} 1 & -3 & 3 \\ 3 & -5 & 3 \\ 6 & -6 & 4 \end{pmatrix},$$

we can proceed as follows. We first call the linear algebra package with the command `with(linalg)`. We then use the command `matrix` to explain to Maple that the matrix we define is three-dimensional (i.e., three rows and three columns), and then write the elements of the matrix in a sequence starting with the first row, then the second row, etc. The command `eigenvalues` will find that the characteristic polynomial has the simple root 4 and the double root -2. The command `eigenvectors` will then obtain for the eigenvalue -2 the two eigenvectors that it writes down, and for the eigenvalue 4 the one eigenvector that it writes down.

```
> with(linalg):
> A := matrix(3,3, [1,-3,3,3,-5,3,6,-6,4]);
```

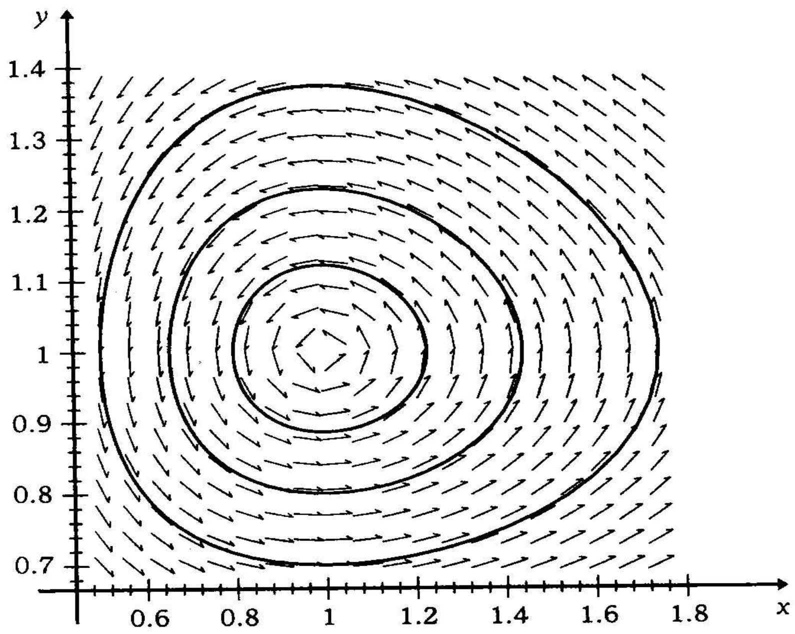
$$A := \begin{bmatrix} 1 & -3 & 3 \\ 3 & -5 & 3 \\ 6 & -6 & 4 \end{bmatrix}$$

```
> eigenvalues(A);
```

$$4, -2, -2$$

```
> eigenvectors(A);
```

$$[\ -2, 2,\ \{[1, 1, 0], [-1, 0, 1]\}\ ],\quad [\ 4, 1,\ \{[1, 1, 2]\}\ ]$$



With the help of the DEplot command we can draw the direction field and the flow of a two-dimensional linear or nonlinear system in the phase plane. To obtain the curves, we need to specify initial conditions and Maple will draw a curve for each specified initial condition.

EXAMPLE 3 To draw the direction field and the solution curves through (1.2, 1.2), (1, 0.7), and (0.8, 0.1) for the two-dimensional system

$$\begin{cases} x' = x(1-y) \\ y' = 3y(x-1), \end{cases}$$

proceed as follows. First call the package DEtools and then use the command DEplot as shown below. Since the curve is obtained numerically, a step size is also specified. Maple then draws the picture in Figure 4.6.2.

```
> with(DEtools):
> DEplot([diff(x(t),t)=x(t)*(1-y(t)),
  diff(y(t),t)=.3*y(t)*(x(t)-1)],[x(t),y(t)],t=-7..7,
  [[x(0)=1.2,y(0)=1.2],[x(0)=1,y(0)=.7],
  [x(0)=0.8,y(0)=1]],stepsize=.2);
```