Chapter 1

# First-Order Differential Equations

## Project 1.3
## Direction Fields and Solution Curves

A number of specialized differential equations packages are available as freeware or shareware. Links to download sites offering such software packages are provided on the Web page **www.prenhall.com/edwards** that supports this text and manual. Such systems automate the construction of direction fields and solution curves, as do some graphing calculators.

Computer algebra systems and technical computing environments such as *Maple*, *Mathematica* and MATLAB provide extensive resources for the study of differential equations. For instance, the *Maple* command

```
with(DEtools):
DEplot( diff(y(x),x)=sin(x-y(x)), y(x),
        x=-5..5, y=-5..5 );
```

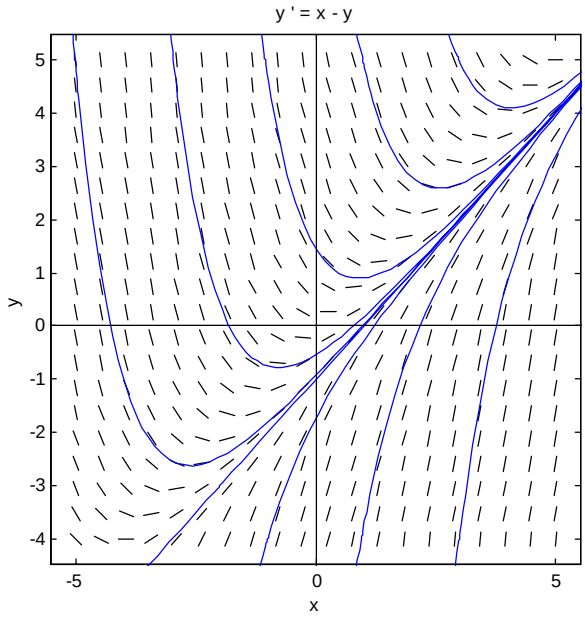
and the *Mathematica* command

```
Needs["Graphics`PlotField`"]
PlotVectorField[{1, Sin[x-y]}, {x,-5,5}, {y,-5,5}]
```

produce direction fields similar to the one shown in Fig. 1.3.4 of the text.

Figure 1.3.4 itself was generated by the MATLAB program **dfield** (John Polking and David Arnold, *Ordinary Differential Equations Using MATLAB* (2nd edition), Prentice Hall, 1999) that is available free for educational use. When a differential equation is entered in the **dfield** setup menu (as illustrated in Fig. 1.3.20 of the text), you can immediately plot a direction field and then — with a single mouse click — plot also the solution curve through any desired point.

For example, the figure at the top of page 2 shows a slope field and typical solution curves (generated using **dfield**) for the differential equation $y' = x - y$. It appears that there exists a (single) *straight line* solution curve that all other solution




curves approach as $x \to \infty$. Indeed, if we substitute the trial straight line solution $y(x) = ax + b$ in the differential equation, we get

$$a = y' = x - y = x - (ax + b) = (1 - a)x - b,$$

which is so if and only if $a = 1$ and $b = -1$. Thus $y(x) = x - 1$ is, indeed, a straight line solution of the differential equation $y' = x - y$. The figure then suggests (without proving it) that

$$y(x) - (x - 1) \ \to \ 0 \qquad \text{as} \qquad x \to \infty.$$

The following two investigations involve similar inferences from slope fields and computer-generated solution curves. You might warm up by generating the direction fields and some solution curves for Problems 1-10 in this section.

**Investigation A**
Plot a direction field and typical solution curves for the differential equation $dy/dx = \sin(x - y)$ of Example 3 in the text, but with a larger window than that of Fig. 1.3.4. With $-10 \le x, y \le 10$, for instance, a number of apparent straight line solution curves should be visible.

**(a)** Substitute $y = ax + b$ in the differential equation to determine what the coefficients $a$ and $b$ must be in order to get a solution.

**(b)** A computer algebra system gives the general solution

$$y(x) \ = \ x - 2\tan^{-1}\left(\frac{x - 2 - C}{x - C}\right).$$

Can you determine a value of the arbitrary constant $C$ that yields the linear solution $y = x - \pi/2$ corresponding to the initial condition $y(\pi/2) = 0$?

**Investigation B**
For your own personal differential equation, let $n$ be the *smallest* integer in your student ID number that is *greater* than 1, and consider the differential equation

$$\frac{dy}{dx} = \frac{1}{n}\cos(x - n\,y).$$

**(a)** First investigate (as in (a) above) the possibility of straight line solutions.

**(b)** Then generate a slope field for this differential equation, with the viewing window chosen so that you can picture some of these straight lines, plus a sufficient number of nonlinear solution curves that you can formulate a conjecture about what happens to $y(x)$ as $x \to \infty$. State your inference as plainly as you can. Given the initial value $y(0) = y_0$, it would be nice to be able to say (perhaps in terms of $y_0$) how $y(x)$ behaves as $x \to \infty$.

**(c)** A computer algebra system gives the general solution

$$y(x) = \frac{1}{n}\left[x + 2\tan^{-1}\left(\frac{1}{x - C}\right)\right]$$

Can you make a connection between this symbolic solution and your graphically generated solution curves (straight lines or otherwise)?

In the sections that follow we outline the use of *Maple*, *Mathematica*, and MATLAB to generate slope fields and solution curves.

## Using *Maple*

The differential equation $dy/dx = x - y$ is defined in *Maple* by the command

```
de := diff( y(x), x) = x - y(x);
```

$$de := \frac{\partial}{\partial x}y(x) = x - y(x)$$

Note that we write $y(x)$, not just $y$, to specify that the dependent variable $y$ is a function of the independent variable $x$, and that `:=` is used as the assignment operator. The differential equation solver in *Maple* is **`dsolve`**, and it gives the general solution

```
yg := dsolve( de, y(x) );
```

$$yg := y(x) = x - 1 + e^{(-x)}\_C1$$

Observe how *Maple* writes the (first) arbitrary constant _*C*1 that appears. We can verify that the righthand side expression in the equation *yg* actually satisfies the differential equation by showing that it yields an identity upon substitution.

```
subs( y(x) = rhs(yg), de );
```

$$\frac{\partial}{\partial x}(x - 1 + e^{(-x)}\_C1) = 1 - e^{(-x)}\_C1$$

When we enter the command **`eval(%)`** to ask that the derivative on the left be evaluated, the identity

$$1 - e^{(-x)}\_C1 = 1 - e^{(-x)}\_C1$$

that results shows that the general solution of the differential equation **`de`** is, indeed, $y(x) = x - 1 + Ce^x$. We can get a particular solution satisfying a given initial condition either by specifying it in the **`dsolve`** command,

```
y1 := dsolve( {de, y(0)=2}, y(x) );
```

$$soln1 := y(x) = x - 1 + 3e^{(-x)}$$

or by substituting a specific numerical value for the arbitrary constant in the general solution:

```
y2 := subs(_C1=0, yg );
```

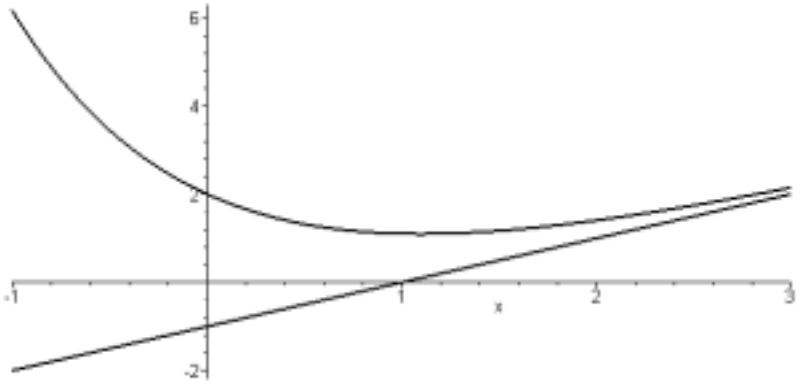
$$soln2 := y(x) = x - 1$$

We can plot both these solution curves simultaneously:

```
y1 := rhs(y1):
y2 := rhs(y2):
plot( {y1,y2}, x=-1..3 );
```

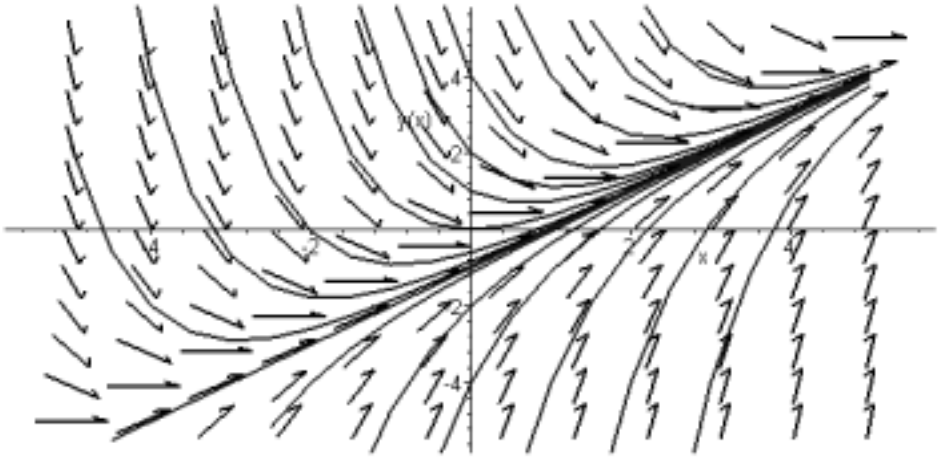
The result is shown at the top of page 5.

Any finite number of particular solutions could be plotted simultaneously (as shown in the figure at the top of page 5) by entering them as a set enclosed by braces. However, the **`DEtools`** package contains the special command **`DEplot`** for doing this sort of thing. The following rather detailed command shows how to plot simultaneously a slope field and a collection of solution curves satisfying a set of separate initial conditions, each corresponding to a different initial point $(x_0, y_0)$.




```
with(DEtools):
DEplot( diff(y(x),x) = x - y(x),y(x),
     x=-5..5,  y=-5..5,
     {[0,4],[0,2],[0,1],[0,0],[0 ,-1],[0,-2],
      [0,-4],[-4,-2],[-3,4],[-4,4],[1,4],[2,4],
      [3,4],[-2,-4],[1,-4],[2,-4],[3,-4]},
      dirgrid=[12,12],
      color = black, linecolor = blue,
      thickness = 2 );
```

We have specified the right hand side $f(x, y) = x - y$ of the differential equation $y' = f(x, y)$, the axes $[x, y]$ for plotting, and the window to be plotted.




Here we see a pleasant variety of solution curves — all appearing to funnel in on the single linear "asymptotic solution" $y = x + 1$ — together with a slope field consisting of a 12-by-12 **grid** of arrows. We could plot the slope field alone by deleting the initial points.

## Using *Mathematica*

To define the differential equation $y' = x - y$ in *Mathematica* we enter the command

```
de = D[y[x],x] == x - y[x]
```

$$y'(x) == x - y(x)$$

Note that *Mathematica* uses `=` for assignment, `==` for ordinary equality, and square brackets for functional notation. The differential equation solver in *Mathematica* is **`DSolve`**.

```
soln = DSolve[ de, y[x], x ]
```

$$\{y(x) \to x + e^{-x} c_1 - 1\}\}$$

We can define the general solution explicitly as a function of **`x`** by taking the 2nd element of the 1st element of the 1st element of the nested list **`soln`**:

```
yg[x_] = soln[[1,1,2]]
```

$$x + e^{-x} c_1 - 1$$

We can verify this general solution by substituting it for the function **`y`** in the differential equation.

```
de /. y -> yg
```

True

We can get a particular solution satisfying a given initial condition by specifying it in the **`DSolve`** command:

```
y1 = DSolve[{de, y[0] == 2}, y[x], x];
y1 = y1[[1,1,2]] // Simplify
```

$$x + 3e^{-x} - 1$$

Alternatively, we can substitute a numerical value for the arbitrary constant in the general solution:

```
y2 = yg[x] /. C[1] -> 0
```

$$x - 1$$

We can plot both these solution curves simultaneously; the command

```
Plot[ {y1,y2}, {x, -1,3} ];
```

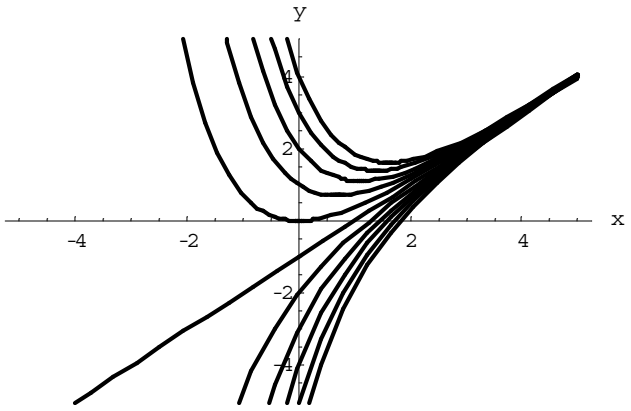
produces the same two-curve figure generated previously using *Maple*. Indeed, any finite number of particular solutions could be plotted simultaneously by entering them as a list enclosed by braces. We can construct just such a list by substituting a list (or table) of numerical values for the arbitrary constant C[1] = $c_1$ in our general solution.

```
yp = yg[x] /. C[1] -> Table[c, {c,-5,5} ]
```

$$x-5e^{-x}-1,\ x-4e^{-x}-1,\ x-3e^{-x}-1,\ x-2e^{-x}-1,\ x-e^{-x}-1,$$
$$x-1,\ x+e^{-x}-1,\ x+2e^{-x}-1,\ x+3e^{-x}-1,\ x+4e^{-x}-1,\ x+5e^{-x}-1\}$$

Here we have specified the values `C[1] = -5,-4, ....., 4,5`.

```
curves =
Plot[ Evaluate[yp], {x,-5,5}, PlotRange -> {-5,5}];
```




Here we see a family of solution curves, all appearing to funnel in on the single "asymptotic solution" $y = x + 1$. We can use the **`PlotVectorField`** function, after loading the *Mathematica* package

```
Needs["Graphics`PlotField`"]
```

to superimpose a slope field. Then the commands

```
slopes =
PlotVectorField[{1,x-y}, {x,-5,5}, {y,-5,5} ];

Show[ curves, slopes, AspectRatio -> 1 ];
```

do the job; try it for yourself.

## Using MATLAB

Figures including slope fields are generated with the greatest ease using the menu-driven **dfield** function in the *ODE Using MATLAB* package mentioned previously. Here we illustrate also the "hands on" generation of symbolic solutions and solution curves using the Student Edition of MATLAB (version 5) — or the professional edition equipped with the Symbolic Math Toolbox, which calls on an imbedded *Maple* kernel for the execution of symbolic operations.

Symbolic variables, formulas, or equations are entered as strings enclosed in single quotes. Thus we can define the differential equation $y' = x - y$ by entering

```
syms x y
de = 'Dy = x - y'

de =
Dy = x - y
```

with **D** denoting differentiation of the dependent variable **y** (which immediately follows the **D**) with respect to the independent variable **x** (the other variable in the equation). The function **dsolve** computes explicit symbolic solutions (when possible). Thus:

```
yg = dsolve(de,'x');
yg = expand(yg)

yg =
x-1+1/exp(x)*C1
```

Apparently we have a general solution involving an arbitrary constant `C1`. To verify this, we check that when we differentiate the expression **yg** with respect to the symbolic variable **x**,

```
Dy = diff(yg,x)

Dy =
1-1/exp(x)*C1
```

We get the same result as when we substitute **yg** for **y** in the differential equation,

```
subs(de, y, yg)

ans =
Dy = 1-1/exp(x)*C1
```

[The syntax for symbolic substitution of **new** for **old** in the expression **expr** is **subs(expr, old, new)**.]

We can get a particular solution satisfying a given initial condition either by specifying it in the **dsolve** command,

```
y1 = dsolve(de, 'y(0)=2', 'x');
y1 = expand(y1)

y1 =
x-1+3/exp(x)
```

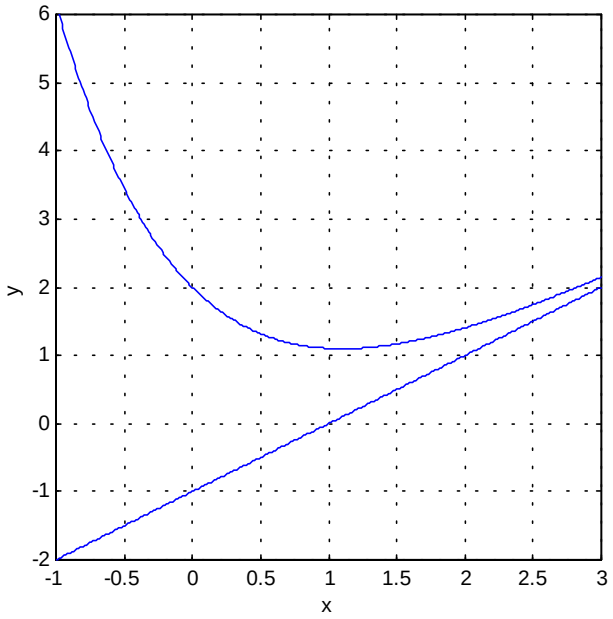
or by substituting a specific numerical value for the arbitrary constant in the general solution:

```
y2 = subs(yg,'C1',0)

y2 =
x-1
```

We can use MATLAB's **ezplot** command to plot both these two solution curves with the commands

```
ezplot(y1,[-1 3])
hold on
ezplot(y2,[-1 3])
axis([-1 3 -2 6])
grid on
```



The command **ezplot('f', [a b])** plots the expression **'f'** as a function of **x** on the interval [*a*, *b*]. The **hold on** command holds the first solution curve in place while the while the second one is plotted. The **axis** command overrides MATLAB's auto-sizing and shows the picture in the desired viewing window:

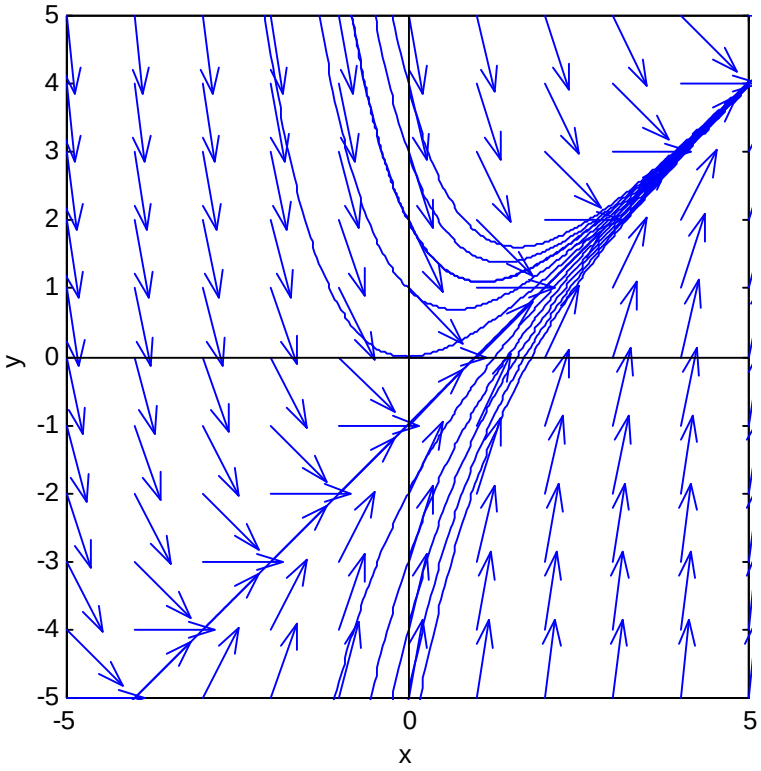
Unless the special purpose **dfield** program is used, plotting slope fields in MATLAB requiries a bit of work. The following program utilizes the built-in command **quiver(x,y,dx,dy)** that plots an $n \times n$ array of vectors with *x*- and *y*- components specified by the $n \times n$ matrices **dx** and **dy**, based at the *xy*-points in the plane whose *x*- and *y*- coordinates are specified by the $n \times n$ matrices **x** and **y**.

```
% slope field program  sfield.m
n = 10;                          % no of subintervals
a = -5; b = 5;                   % x-interval
c = -5; d = 5;                   % y-interval
h = (b-a)/n;  k = (d-c)/n;       % x- and y-step sizes
x = a : h : b;                   % x-subdivision points
y = c : k : d;                   % y-subdivision points
[x,y] = meshgrid(x,y);           % grid of points
f = x - y;                       % define f(x,y)
t = atan(f);                     % angle of inclination
dx = cos(t);   dy = sin(t);      % xy-components of arrow
quiver(x,y,dx,dy)                % plot slope field
axis([a b c d])                  % viewing window
grid on                          % draw grid lines
```

However, you need not be concerned about these matrices of coordinates and components — only the number $n$ of subintervals in each direction, the desired viewing window $a \le b, c \le d,$ and the expression **f** defining the right-hand side of the differential equation $y' = f(x,y)$ need be altered when the program **sfield** is defined by saving the commands listed above in the text file **sfield.m**. (Alternatively, these commands could be entered individually in command mode). For instance, once the slope field program **sfield** has been defined, the commands

```
for k = -5 : 5
     ezplot(subs(yg,'C1',k),[-5 5]);
     hold on
     end
sfield
```

first plot the particular solution curves corresponding to the values –5, –4, ....., 4, 5 of the arbitrary constant **C1** in the general solution **yg**, and then a $10 \times 10$ grid of direction field arrows is added. The result is shown on the next page.


5
4
3
2
1
0
-1
-2
-3
-4
-5
y
-5
0
5
x