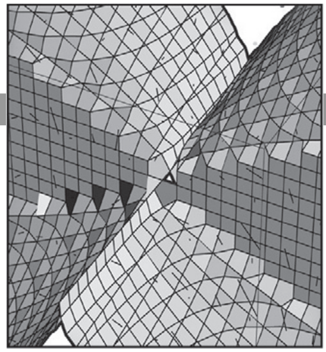
First-Order Ordinary Differential Equations

2



We will devote a considerable amount of time in this text to developing explicit, implicit, numerical, and graphical solutions of differential equations. In this chapter we introduce frequently encountered forms of first-order ordinary differential equations and methods to construct explicit, numerical, and graphical solutions of them. Several of the equations along with the methods of solution discussed here will be used in subsequent chapters of the text.

Charles Emile Picard (July 24, 1856, Paris, France-December 11, 1941, Paris, France) According to O'Connor, "Picard and his wife had three children, a daughter and two sons, who were all killed in World War I. His grandsons were wounded and captured in World War II."
In regards to Picard's teaching, the famous French mathematician Jacques Salomon Hadamard (1865–1963) wrote in Picard's obituary "A striking feature of Picard's scientific personality was the perfection of his teaching, one of the most marvelous, if not the most marvelous that I have known."
See texts like [6], [7], or [3].

2.1 Theory of First-Order Equations: A Brief Discussion

In order to understand the types of first-order initial-value problems that have a unique solution, the Picard-Lindelöf Existence and Uniqueness theorem is stated.

Theorem 1 (Existence and Uniqueness). *Consider the initial-value problem*

$$\begin{cases} dy/dx = f(x,y) \\ y(x_0) = y_0 \end{cases}. \tag{2.1}$$

If f and $\partial f/\partial y$ are continuous functions on the rectangular region R,

$$R = \{(x,y) | a < x < b,\ c < y < d\},$$

containing the point (x_0, y_0), there exists an interval $|x - x_0| < h$ centered at x_0 on which there exists one and only one solution to the differential equation that satisfies the initial condition.

Differential Equations with Mathematica. http://dx.doi.org/10.1016/B978-0-12-804776-7.00002-4

Often, we can use the command

```
DSolve[{y'[x]==f[x,y[x]],y[x0]==y0},y[x],x]
```

to solve the initial-value problem (2.1).

When exact solutions are not possible or not desired, try to use `NDSolve` to generate a numerical solution. The command

```
NDSolve[{y'[x]==f[x,y[x]],y[x0]==y0},y[x],{x,x0,x1}]
```

attempts to generate a numerical solution of $dy/dx = f(x, y)$, $y(x_0) = y_0$ valid for $x_0 \le x \le x_1$.

EXAMPLE 2.1.1: Solve the initial-value problem

$$\begin{cases} dy/dx = x/y \\ y(0) = 0 \end{cases}.$$

Does this result contradict the Existence and Uniqueness Theorem?

SOLUTION: We begin by using `StreamPlot` to graph the direction field associated with the equation in Figure 2-1(a).

```
p1 = StreamPlot[{1, x/y}, {x, −5, 5}, {y, −5, 5}, StreamStyle
  → Gray, Frame → False, Axes → Automatic, AxesOrigin
```

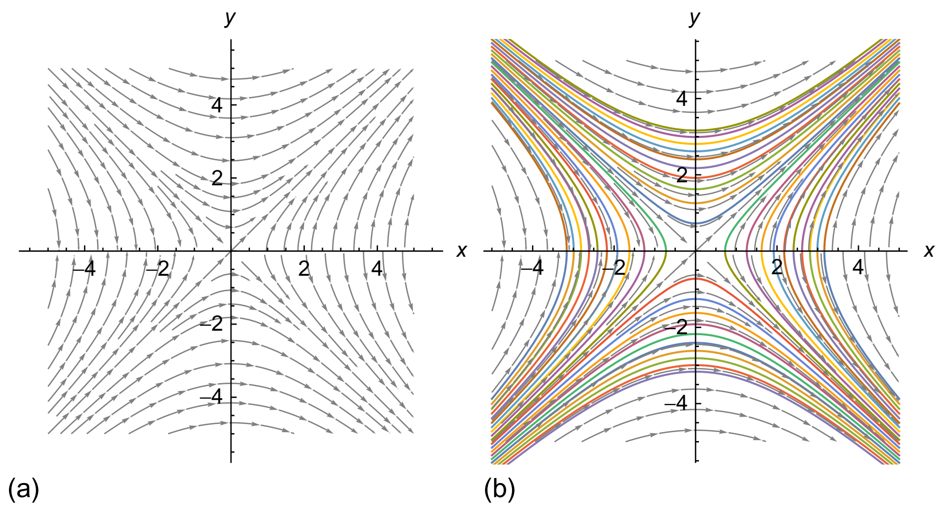


Figure 2-1 (a) Direction field for $dy/dx = x/y$. (b) Various solutions of $dy/dx = x/y$ with the direction field

```
→ {0, 0}, AxesLabel → {x, y}, AxesStyle → Black, PlotLabel
→ "(a)"];
```

This equation is solved with DSolve to determine the family of solutions $y = -\sqrt{x^2 + C}$ and $y = \sqrt{x^2 + C}$.

```
gensol = DSolve[y'[x] == x/y[x], y[x], x]
```

$$\left\{\left\{y[x] \to -\sqrt{x^2 + 2C[1]}\right\}, \left\{y[x] \to \sqrt{x^2 + 2C[1]}\right\}\right\}$$

```
gensol[[1, 1, 2]]
```

$$-\sqrt{x^2 + 2C[1]}$$

```
gensol[[2, 1, 2]]
```

$$\sqrt{x^2 + 2C[1]}$$

We extract the formulas for the solutions using Part ([[...]]). In this case, there are two formulas.

```
gensol[[1, 1, 2]]
```

$$-\sqrt{x^2 + 2C[1]}$$

```
gensol[[2, 1, 2]]
```

$$\sqrt{x^2 + 2C[1]}$$

To graph the solutions for various values of the arbitrary constant, Table is used to create two lists of solutions to the differential equation. Generally, Table[f[x],{x,a,b,n}] creates a list of $f(x)$ for x-values from a to b in steps of n. These are then graphed with Plot and shown together with the direction field in Figure 2-1(b).

```
toplot1 = Table[gensol[[1, 1, 2]]/.C[1] → i, {i, −5, 5, 10/19}];
toplot2 = Table[gensol[[2, 1, 2]]/.C[1] → i, {i, −5, 5, 10/19}];

p2a = Plot[{toplot1, toplot2}, {x, −5, 5}];
p2 = Show[p2a, p1, Frame → False, Axes → Automatic,
AxesOrigin → {0, 0},
AxesLabel → {x, y}, AxesStyle → Black, PlotRange →
  {{−5, 5}, {−5, 5}}, AspectRatio → 1, PlotLabel → "(b)"];

Show[GraphicsRow[{p1, p2}]]
```

Application of the initial condition yields $0^2 - 0^2 = C$, so $C = 0$. Therefore, solutions that pass through $(0, 0)$, satisfy $y^2 - x^2 = 0$, so there are four solutions, $y = x$, $y = -x$, $y = |x|$, and $y = -|x|$ that satisfy the differential equation and the initial condition.

```
partsol = DSolve[{y'[x] == x/y[x], y[0] == 0}, y[x], x]
```

$$\left\{\left\{y[x] \to -\sqrt{x^2}\right\}, \left\{y[x] \to \sqrt{x^2}\right\}\right\}$$

The solution is graphed in Figure 2-2.

```
Plot[{Abs[x], −Abs[x]}, {x, −5, 5}, PlotRange → {{−5, 5},
  {−5, 5}}, AspectRatio → 1, PlotStyle → Black, AxesStyle
  → Black, AxesLabel → {x, y}]
```

Although more than one solution satisfies this initial-value problem, the Existence and Uniqueness Theorem is *not* contradicted because

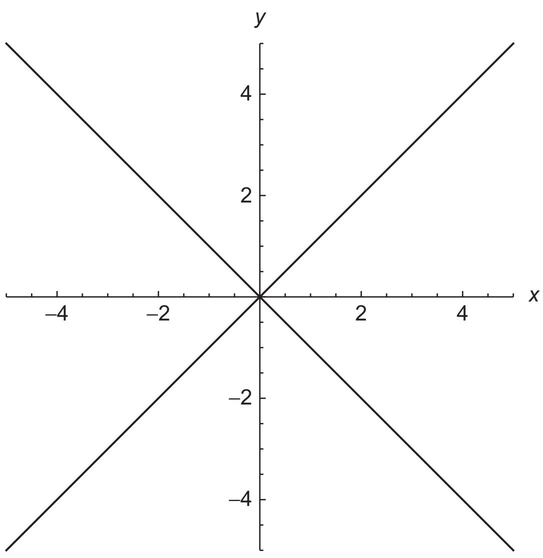


Figure 2-2 Solutions of $dy/dx = x/y$, $y(0) = 0$

the function $f(x, y) = x/y$ is not continuous at the point $(0, 0)$; the requirements of the theorem are not met.

■

EXAMPLE 2.1.2: Verify that the initial-value problem $\{dy/dx = y,\ y(0) = 1\}$ has a unique solution.

SOLUTION: In this case, $f(x, y) = y$, $x_0 = 0$, and $y_0 = 1$. Hence, both f and $\partial f/\partial y = 1$ are continuous on all rectangular regions containing the point $(x_0, y_0) = (0, 1)$. Therefore by the Existence and Uniqueness Theorem, there exists a unique solution to the differential equation that satisfies the initial condition $y(0) = 1$.

We can verify this by solving the initial-value problem. The unique solution is $y = e^x$, which is computed with `DSolve` and then graphed with `Plot` in Figure 2-3(a). Notice that the graph passes through the point $(0, 1)$, as required by the initial condition. We show the graph together with the direction field in Figure 2-3(b).

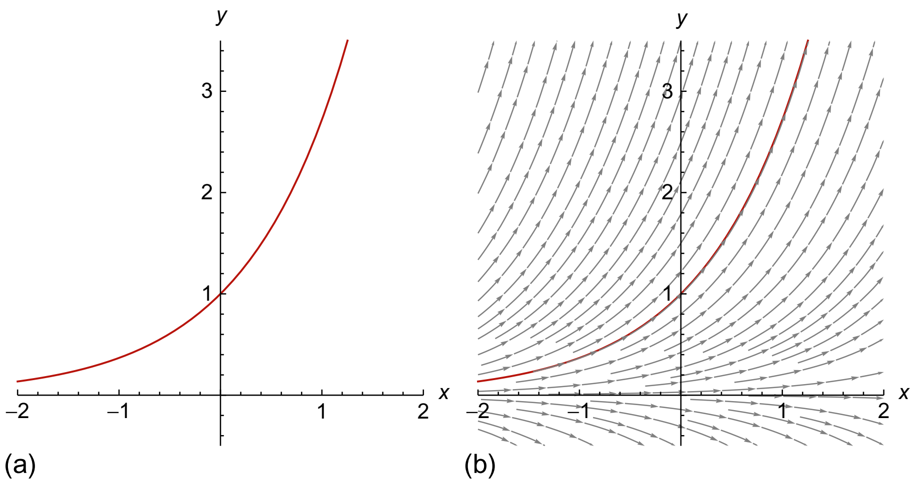


Figure 2-3 (a) Plot of $y = e^x$. (b) The solution together with the direction field for $dy/dx = y$

```
sol = DSolve[{y'[x] == y[x], y[0] == 1}, y[x], x]
```

$\{\{y[x] \to e^x\}\}$

```
p1 = Plot[sol[[1, 1, 2]], {x, −2, 2}, PlotStyle->CMYKColor
  [0, 0.89, 0.94, 0.28], PlotRange → {{−2, 2}, {−.5, 3.5}},
  AspectRatio → 1, AxesStyle → Black, AxesLabel → {x, y},
  PlotLabel → "(a)"];

p2a = StreamPlot[{1, y}, {x, −2, 2}, {y, −.5, 3.5},
  StreamStyle → Gray, Frame → False, Axes → Automatic,
  AxesOrigin→{0,0},AxesLabel →{x,y},AxesStyle→Black];
  p2 = Show[p1, p2a, PlotLabel → "(b)"];

Show[GraphicsRow[{p1, p2}]]
```

■

EXAMPLE 2.1.3: Show that the initial-value problem

$$\begin{cases} x\dfrac{dy}{dx} - y = x^2 \cos x \\ y(0) = 0 \end{cases}$$

has infinitely many solutions.

SOLUTION: Writing $xy' - y = x^2 \cos x$ in the form $y' = f(x, y)$ results in

$$\frac{dy}{dx} = \frac{x^2 \cos x + y}{x}$$

and because $f(x, y) = \left(x^2 \cos x + y\right)/x$ is not continuous on an interval containing $x = 0$, the Existence and Uniqueness theorem does not guarantee the existence or uniqueness of a solution. In fact, using `DSolve` we see that a general solution of the equation is $y = x \sin x + Cx$ and for every value of C, $y(0) = 0$.

```
sol = DSolve[{xy'[x] − y[x] == x^2Cos[x], y[0] == 0},
y[x], x]
```

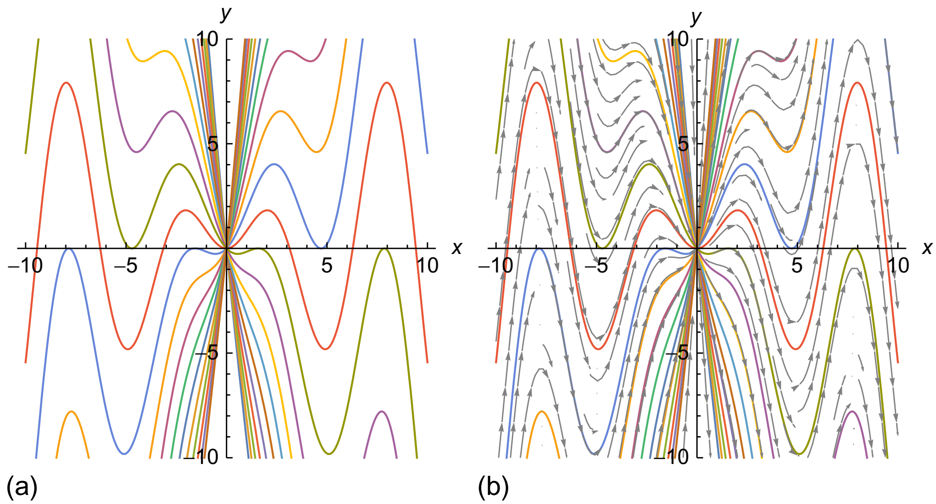


Figure 2-4 (a) Every solution satisfies $y(0) = 0$. (b) Solutions with the direction field

```
{{y[x] → xC[1] + xSin[x]}}
```

We confirm this graphically by graphing several solutions. First, we use `Table` to define `toplot` to be a set of functions obtained by replacing the arbitrary constant in $y(x)$ by $-10, -9, \ldots, 9, 10$ (Figure 2-4).

```
toplot = Table[sol[[1, 1, 2]]/.C[1] → i, {i, −10, 10}];

p1 = Plot[toplot, {x, −10, 10}, PlotRange → {−10, 10},
  AxesLabel → {x, y}, PlotLabel → "(a)", AspectRatio → 1,
  AxesStyle → Black];

p2a = StreamPlot[{1, (x^2Cos[x] + y)/x}, {x, −10, 10},
{y, −10, 10}, StreamStyle → Gray,
  Frame → False, Axes → Automatic, AxesOrigin → {0, 0},
  AxesLabel → {x, y}, AxesStyle → Black];
p2 = Show[p1, p2a, PlotLabel → "(b)"];

Show[GraphicsRow[{p1, p2}]]
```

■

2.2 Separation of Variables

Definition 5 (Separable Differential Equation). *A differential equation that can be written in the form $g(y)y' = f(x)$ or $g(y)\,dy = f(x)\,dx$ is called a* ***separable differential equation****.*

Separable differential equations are solved by collecting all the terms involving y on one side of the equation, all the terms involving x on the other side of the equations and integrating:

$$g(y)\,dy = f(x)\,dx \Longrightarrow \int g(y)\,dy = \int f(x)\,dx + C,$$

where C is a constant.

EXAMPLE 2.2.1: Show that the equation

$$\frac{dy}{dx} = \frac{2\sqrt{y} - 2y}{x}$$

is separable, and solve by separation of variables.

SOLUTION: As with previous examples, we start by graphing the direction field with `StreamPlot` in Figure 2-5(a).

```
p1 = StreamPlot[{1, (2Sqrt[y] − 2y)/x}, {x, −1, 1}, {y, 0, 2},
  PlotRange → {{−1, 1}, {0, 2}},
  AxesLabel → {x, y}, PlotLabel → "(a)", AspectRatio → 1,
  AxesStyle → Black, StreamStyle → Black, Frame → False,
  Axes, → Automatic, AxesOrigin → {0, 0}];
```

The equation $y' = \left(2\sqrt{y} - 2y\right)/x$ is separable because it can be written in the form

$$\frac{1}{2\sqrt{y} - 2y}dy = \frac{1}{x}dx.$$

To solve the equation, we integrate both sides and simplify. Observe that we can write this equation as

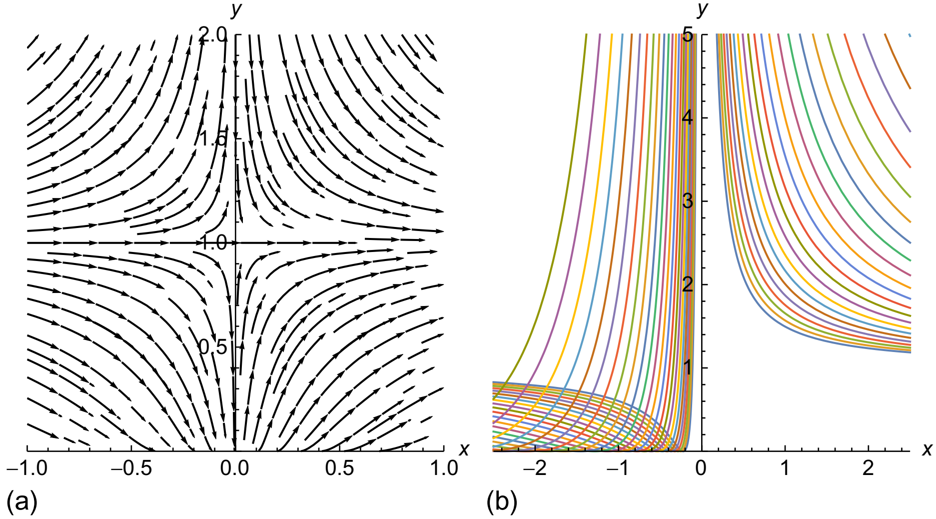


Figure 2-5 (a) Direction field. (b) Various solutions of $y' = \left(2\sqrt{y} - 2y\right)/x$

$$\int \frac{1}{2\sqrt{y}}\frac{1}{1-\sqrt{y}}dy = \frac{1}{x}dx + C.$$

To evaluate the integral on the left-hand side, let $u = 1 - \sqrt{y}$ so $-du = \dfrac{1}{2\sqrt{y}}dy$. We then obtain

$$-\int \frac{1}{u}du = \int \frac{1}{x}dx + C_1$$

so that $-\ln|u| = \ln|x| + C_1$. Recall that $-\ln|u| = \ln|u|^{-1}$, so we have

$$\ln\frac{1}{|u|} = \ln|x| + C_1.$$

Using Mathematica, we use `Integrate`.

```
step1 = Integrate[1/(2Sqrt[y] − 2y), y]
```

$-\mathrm{Log}\left[1 - \sqrt{y}\right]$

The integral on the right-hand side of the equation is computed in the same way.

Note that `Log[x]` represents the **natural logarithm function**, $y = \ln x$.

```
step2 = Integrate[1/x, x]
```

Log[x]

Simplification yields

$$\frac{1}{|u|} = e^{\ln|x|+C_1} = C_2|x|,$$

where $C_2 = e^{C_1}$. Resubstituting we find that

$$\frac{1}{\left|1-\sqrt{y}\right|} = C_2|x| \quad \text{or} \quad x = \frac{C_3}{1-\sqrt{y}}.$$

Solving for y shows us that

$$\begin{aligned}\sqrt{y}-1 &= \frac{C_3}{x}\\ \sqrt{y} &= \frac{x+C_3}{x}\\ y &= \left(\frac{x+C_3}{x}\right)^2\end{aligned}$$

is a general solution of the equation $y' = \left(2\sqrt{y}-2y\right)/x$. We obtain the same results with Mathematica,

We use constant to represent the arbitrary constant C to avoid ambiguity with the built-in symbol C.

```
step3 = Solve[step1 == step2 + constant, y]
```

$$\left\{\left\{y \to \frac{e^{-2\text{constant}}\left(-1+e^{\text{constant}}x\right)^2}{x^2}\right\}\right\}$$

where $e^{-\text{constant}}$ represents the arbitrary constant in the solution. We obtain an equivalent result with DSolve. Entering

```
Clear[x, y]
gensol = DSolve[y'[x] == (2Sqrt[y[x]] - 2y[x])/x, y[x], x]
```

$$\left\{\left\{y[x] \to \frac{\left(e^{\frac{C[1]}{2}}+x\right)^2}{x^2}\right\}\right\}$$

finds a general solution of the equation which is equivalent to the one we obtained by hand and names the result gensol. The formula for the solution, which is the second part of the first part of the first part of `gensol`, is extracted from `gensol` with `gensol[[1,1,2]]`. Alternatively, if you are using Version 10, you can select, copy, and paste the result to any location in the notebook.

To graph the solution for various values of `C[1]`, which represents the arbitrary constant in the formula for the solution, we use `Table` together with `ReplaceAll` (`/.`) to generate a set of functions obtained by replacing `C[1]` in the formula for the solution by i for $i = -3$, -2.75, ..., 2.75, and 3, naming the resulting set of functions `toplot`.

```
toplot = Table[gensol[[1, 1, 2]]/.C[1] → i,

  {i, −3, 3, .25}];
```

We then graph the set of functions `toplot` with `Plot` in Figure 2-5(b).

```
p2 = Plot[toplot, {x, −2.5, 2.5}, PlotRange → {{−2.5, 2.5},

  {0, 5}}, AxesLabel → {x, y}, PlotLabel → "(b)",

  AspectRatio → 1, AxesStyle → Black, Frame → False,

  Axes → Automatic, AxesOrigin → {0, 0}];

Show[GraphicsRow[{p1, p2}]]
```

■

An initial-value problem involving a separable equation is solved through the following steps.

1. Find a general solution of the differential equation using separation of variables.
2. Use the initial condition to determine the unknown constant in the general solution.

EXAMPLE 2.2.2: Solve (a) $y \cos x \, dx - \left(1 + y^2\right) dy = 0$ and (b) the initial-value problem $\left\{y \cos x \, dx - \left(1 + y^2\right) dy = 0,\ y(0) = 1\right\}$.

SOLUTION: As in the previous examples, we begin by graphing the direction field in Figure 2-7(a) with `StreamPlot`.

```
p1 = StreamPlot[{1, yCos[x]/(1 + y^2)}, {x, −Pi, 2Pi},
  {y, 0, 3Pi}, AxesLabel → {x, y}, PlotLabel → "(a)",
  AspectRatio→1,AxesStyle→Black,StreamStyle→Black,
  Frame → False, Axes → Automatic, AxesOrigin → {0, 0}];
```

(a) Note that this equation can be rewritten as $dy/dx = (y \cos x) / \left(1 + y^2\right)$. We first use DSolve to solve the equation.

```
gensol1 = DSolve[y'[x]==y[x]Cos[x]/(1 + y[x]^2), y[x], x]
```

$$\left\{\left\{y[x] \to -\sqrt{\text{ProductLog}\left[e^{2C[1]+2\text{Sin}[x]}\right]}\right\},\right.$$

$$\left.\left\{y[x] \to \sqrt{\text{ProductLog}\left[e^{2C[1]+2\text{Sin}[x]}\right]}\right\}\right\}$$

In this case, we see that DSolve is able to solve the nonlinear equation, although the result contains the ProductLog function. Given z, the **Product Log function** returns the principal value of w that satisfies $z = we^w$. See Figure 2-6. A more familiar form of the solution is found

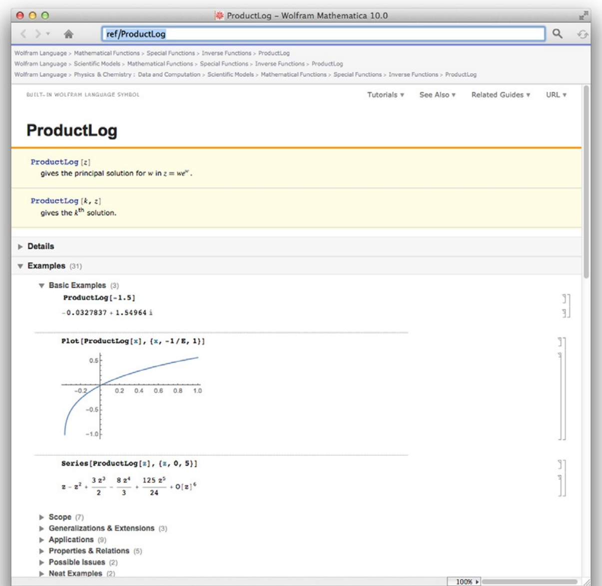


Figure 2-6 Mathematica's help for ProductLog

using traditional techniques. Separating variables and integrating gives us

$$\begin{aligned}\frac{1+y^2}{y}\,dy &= \cos x\,dx\\ \left(\frac{1}{y}+y\right)dy &= \cos x\,dx\\ \ln|y|+\frac{1}{2}y^2 &= \sin x + C.\end{aligned}$$

We can also use Mathematica to implement the steps necessary to solve the equation by hand. To solve the equation, we must integrate both the left- and right-hand sides which we do with `Integrate`, naming the resulting output `lhs` and `rhs`, respectively.

```
lhs = Integrate[(1 + y^2)/y, y]

rhs = Integrate[Cos[x], x]
```

$\frac{y^2}{2}+\mathrm{Log}[y]$

`Sin[x]`

Therefore, a general solution to the equation is $\ln|y|+\frac{1}{2}y^2=\sin x+C$. We now use `ContourPlot` to graph $\ln|y|+\frac{1}{2}y^2=\sin x+C$ in Figure 2-7(b) for various values of C by observing that the level curves of $f(x,y)=\ln|y|+\frac{1}{2}y^2-\sin x$ correspond to the graph of $\ln|y|+\frac{1}{2}y^2=\sin x+C$ for various values of C.

```
p2a = ContourPlot[lhs − rhs, {x, −Pi, 2Pi}, {y, 0, 3Pi},
  AxesLabel → {x, y}, AspectRatio → 1, ContourShading →
  False, Contours → 50, ContourStyle → {{Thickness[.001],
  Black}}, AxesStyle → Black, Frame → False, Axes →
  Automatic, AxesOrigin → {0, 0}];

p2 = Show[p1, p2a, AxesLabel → {x, y}, PlotLabel → "(b)",
  AspectRatio → 1, AxesStyle → Black, Frame → False,
  Axes → Automatic, AxesOrigin → {0, 0}];
```

By substituting $y(0)=1$ into this equation, we find that $C=1/2$, so the implicit solution is given by $\ln|y|+\frac{1}{2}y^2=\sin x+1/2$.

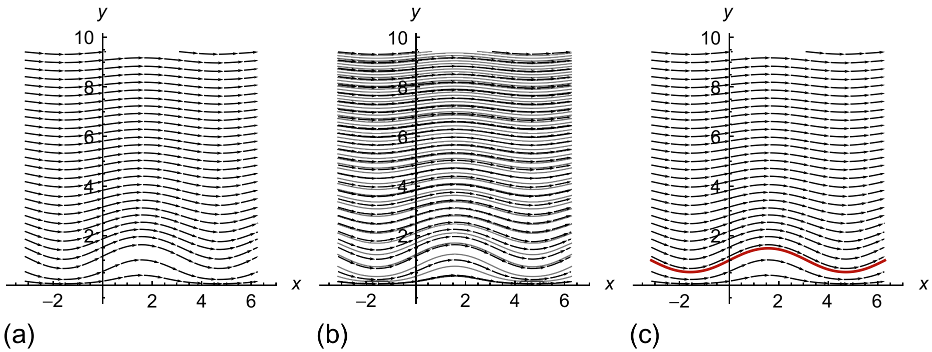


Figure 2-7 (a) Direction field for the equation. (b) Plot of $\ln|y| + \frac{1}{2}y^2 = \sin x + C$ for various values of C. (c) The solution that satisfies $y(0) = 1$ is highlighted

We can also use DSolve to solve the initial value problem as well. The solution is then graphed in Figure 2-7(c) with Plot.

```
sol2 = DSolve[{y'[x]==y[x]Cos[x]/(1 + y[x]^2), y[0] == 1},
y[x], x]
```

$$\left\{\left\{y[x] \to \sqrt{\texttt{ProductLog}\left[e^{1+2\texttt{Sin}[x]}\right]}\right\}\right\}$$

```
p3a = Plot[y[x]/.sol2, {x, −Pi, 2Pi}, AxesStyle → Black,
  AspectRatio → 1, PlotRange → {{−Pi, 2Pi}, {0, 3Pi}},
PlotStyle → {{Thickness[.01], CMYKColor[0, 0.89, 0.94,
0.28]}}];
p3 = Show[p1, p3a, AxesLabel → {x, y}, PlotLabel → "(c)",
  AspectRatio → 1, AxesStyle → Black, Frame → False,
  Axes → Automatic, AxesOrigin → {0, 0}];
Show[GraphicsRow[{p1, p2, p3}]]
```

■

EXAMPLE 2.2.3: Solve each of the following equations. (a) $y' - y^2 \sin t = 0$, (b) $y' = \alpha y\left(1 - \dfrac{1}{K}y\right)$, $K, \alpha > 0$ constant.

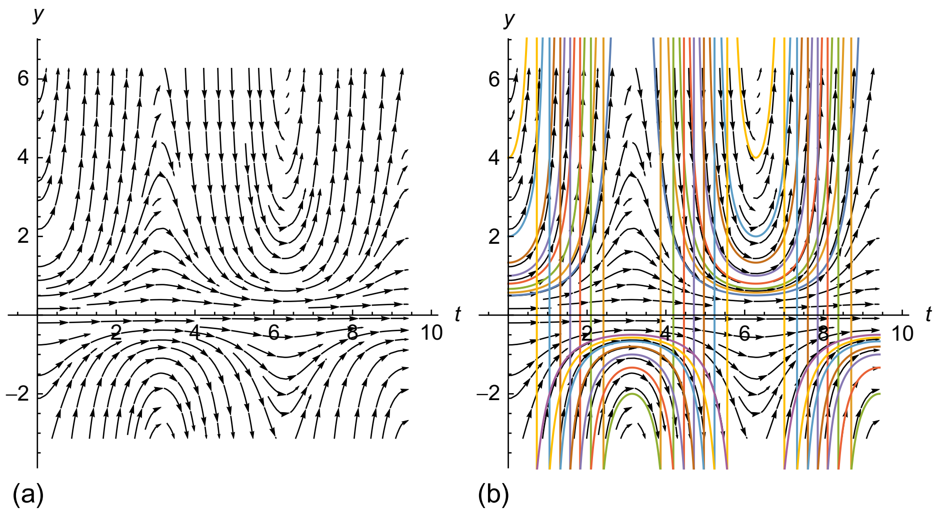


Figure 2-8 (a) Direction field for the equation. (b) Several solutions of $y' - y^2 \sin t = 0$ with the direction field

SOLUTION: (a) The direction field is shown in Figure 2-8(a).

```
p1 = StreamPlot[{1, y^2Sin[t]}, {t, 0, 3Pi}, {y, −Pi, 2Pi},
  AxesLabel → {t, y}, PlotLabel → "(a)", AspectRatio → 1,
  AxesStyle → Black, StreamStyle → Black, Frame → False,
  Axes → Automatic, AxesOrigin → {0, 0}];
```

The equation is separable:

$$\begin{aligned}\frac{1}{y^2}dy &= \sin t\,dt\\ \int \frac{1}{y^2}dy &= \int \sin t\,dt\\ -\frac{1}{y} &= -\cos t + C\\ y &= \frac{1}{\cos t + C}.\end{aligned}$$

We check our result with `DSolve`.

```
gensola = DSolve[y'[t] − y[t]^2Sin[t] == 0, y[t], t]
```

$$\left\{\left\{y[t] \to \frac{1}{-C[1] + \text{Cos}[t]}\right\}\right\}$$

Observe that the result is given as a list. The formula for the solution is the second part of the first part of the first part of `sola`.

```
gensola[[1, 1, 2]]
```

$$\frac{1}{-C[1]+\text{Cos}[t]}$$

We then graph the solution for various values of C with `Plot` in Figure 2-8(b).

```
toplot = Table[gensola[[1, 1, 2]]/.C[1] → i, {i, −1, 1, .25}];

p2a = Plot[toplot, {t, 0, 3Pi}];

p2 = Show[p1, p2a, AxesLabel → {t, y}, PlotLabel → "(b)",
AspectRatio → 1,
  AxesStyle → Black, Frame → False, Axes → Automatic,
  AxesOrigin → {0, 0}];

Show[GraphicsRow[{p1, p2}]]
```

`expression /. x->y` replaces all occurrences of x in `expression` by y.
`Table[a[k],{k,n,m}]` generates the list a_n, a_{n+1}, $\ldots$, a_{m-1}, a_m.
To graph the list of functions `list` for $a \le x \le b$, enter `Plot[list,{x,a,b}]`.

(b) After separating variables, we use partial fractions to integrate.

$$y' = \alpha y\left(1-\frac{1}{K}y\right)$$
$$\frac{1}{\alpha y\left(1-\frac{1}{K}y\right)}dy = dt$$
$$\frac{1}{\alpha}\left(\frac{1}{y}+\frac{1}{K-y}\right) = dt$$
$$\frac{1}{\alpha}\left(\ln|y|-\ln|K-y|\right) = C_1 t$$
$$\frac{y}{K-y} = Ce^{\alpha t}$$
$$y = \frac{CKe^{\alpha t}}{Ce^{\alpha t}-1}.$$

We check the calculations with Mathematica. First, we use `Apart` to find the partial fraction decomposition of $\dfrac{1}{\alpha y\left(1-\frac{1}{K}y\right)}$.

```
step1 = Apart[1/(αy(1 − 1/ky)), y]
```

$$\frac{1}{y\alpha}-\frac{1}{(-k+y)\alpha}$$

Then, we use `Integrate` to check the integration.

```
step2 = Integrate[step1, y]
```

$$k\left(\frac{\text{Log}[y]}{k\alpha}-\frac{\text{Log}[-k+y]}{k\alpha}\right)$$

Last, we use `Solve` to solve $\frac{1}{\alpha}\left(\ln|y|-\ln|K-y|\right)=ct$ for y.

```
step3 = Solve[step2 == constant, y]
```

$$\left\{\left\{y\to\frac{e^{\text{constant}\,\alpha}k}{-1+e^{\text{constant}\,\alpha}}\right\}\right\}$$

We can use `DSolve` to find a general solution of the equation

```
gensol = DSolve[y'[t] == αy[t](1 - 1/ky[t]), y[t], t]
```

$$\left\{\left\{y[t]\to\frac{e^{t\alpha+kC[1]}k}{-1+e^{t\alpha+kC[1]}}\right\}\right\}$$

as well as find the solution that satisfies the initial condition $y(0)=y_0$.

```
initsol = DSolve[{y'[t] == αy[t](1 - 1/ky[t]), y[0] == y0},
y[t], t]
```

$$\left\{\left\{y[t]\to\frac{e^{t\alpha}k\,\text{y0}}{k-\text{y0}+e^{t\alpha}\text{y0}}\right\}\right\}$$

The equation $y'=\alpha y\left(1-\frac{1}{K}y\right)$ is called the **Logistic equation** (or **Verhulst equation**) and is used to model the size of a population that is not allowed to grow in an unbounded manner. Assuming that $y(0)>0$, then all solutions of the equation have the property that $\lim_{t\to\infty}y(t)=K$.

To see this, we set $\alpha=K=1$ and use `StreamPlot` to graph the direction field associated with the equation in Figure 2-9(a).

Logistic growth is discussed in more detail in Section 3.2.2.

```
p1 = StreamPlot[{1, y(1 - y)}, {t, 0, 3}, {y, 0, 3}, AxesLabel
  → {t, y}, PlotLabel → "(a)", AspectRatio → 1,
  AxesStyle → Black, StreamStyle → Black, Frame → False,
  Axes, → Automatic, AxesOrigin → {0, 0}];
```

The property is more easily seen when we graph various solutions along with the direction field as done next in Figure 2-9(b).

```
step2 = initsol[[1, 1, 2]]/.{k → 1, α → 1}
```

$$\frac{e^{t}\text{y0}}{1-\text{y0}+e^{t}\text{y0}}$$

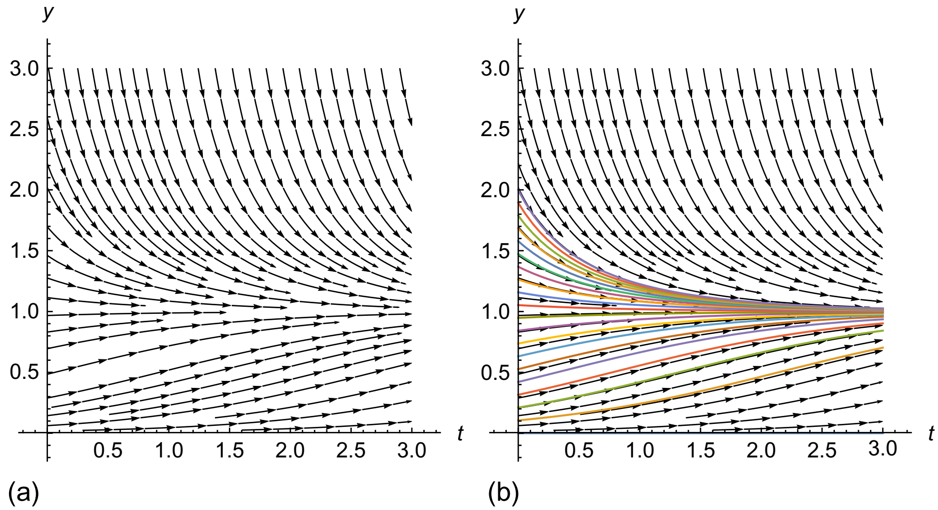


Figure 2-9 (a) A typical direction field for the Logistic equation. (b) A typical direction field for the Logistic equation along with several solutions

```
toplot = Table[step2/.y0 → i, {i, 0, 2, 2/19}];
p2a = Plot[toplot, {t, 0, 3}, AxesLabel → {t, y},
  AspectRatio → 1, AxesStyle → Black, Frame → False,
  Axes → Automatic, AxesOrigin → {0, 0}];
p2 = Show[p1, p2a, PlotLabel → "(b)"];
Show[GraphicsRow[{p1, p2}]]
```

■

Sources: D. N. Burghes and M. S. Borrie, *Modeling with Differential Equations*, Ellis Horwood Limited, pp. 41–45. Joyce M. Black and Esther Matassarin-Jacobs, *Luckman and Sorensen's Medical-Surgical Nursing: A Psychophysiologic Approach*, Fourth Edition, W. B. Saunders Company (1993), pp. 1509–1519, 1775–1808.

Application: Kidney Dialysis

The primary purpose of the kidney is to remove waste products, like urea, creatinine, and excess fluid, from blood. When kidneys are not working properly, wastes accumulate in the blood; when toxic levels are reached, death is certain. The leading causes of chronic kidney failure in the United States are hypertension (high blood pressure) and diabetes mellitus. In fact, one-quarter of all patients requiring **kidney dialysis** have diabetes. Fortunately, **kidney dialysis** removes waste products from the blood of patients with improperly working kidneys. During the hemodialysis process, the patient's blood is pumped through a

dialyser, usually at a rate of 1–3 deciliters per minute. The patient's blood is separated from the "cleaning fluid" by a semi-permeable membrane, which permits wastes (but not blood cells) to diffuse to the cleaning fluid; the cleaning fluid contains some substances beneficial to the body which diffuse to the blood. The "cleaning fluid," called the **dialysate**, is flowing in the *opposite* direction as the blood, usually at a rate of 2–6 deciliters per minute. Waste products from the blood diffuse to the dialysate through the membrane at a rate proportional to the difference in concentration of the waste products in the blood and dialysate. If we let $u(x)$ represent the concentration of wastes in blood, $v(x)$ represent the concentration of wastes in the dialysate, where x is the distance along the dialyser, Q_D represent the flow rate of the dialysate through the machine, and Q_B represent the flow rate of the blood through the machine, then

$$\begin{cases} Q_B u' = -k(u-v) \\ -Q_D v' = k(u-v) \end{cases},$$

where k is the proportionality constant.

If we let L denote the length of the dialyser and the initial concentration of wastes in the blood is $u(0) = u_0$ while the initial concentration of wastes in the dialysate is $v(L) = 0$, then we must solve the initial-value problem

$$\begin{cases} Q_B u' = -k(u-v) \\ -Q_D v' = k(u-v) \\ u(0) = u_0,\ v(L) = 0 \end{cases}.$$

Solving the first equation for u' and the second equation for $-v'$, we obtain the equivalent system

$$\begin{cases} u' = -\dfrac{k}{Q_B}(u-v) \\ -v' = \dfrac{k}{Q_D}(u-v) \\ u(0) = u_0,\ v(L) = 0 \end{cases}.$$

Adding these two equations results in a separable (and linear) equation in $u - v$,

$$\begin{aligned} u' - v' &= -\frac{k}{Q_B}(u-v) + \frac{k}{Q_D}(u-v) \\ (u-v)' &= -\left(\frac{k}{Q_B} - \frac{k}{Q_D}\right)(u-v) \end{aligned}$$

Let $\alpha = k/Q_B - k/Q_D$ and $y = u - v$. Then we must solve the separable equation $y' = -\alpha y$, which is done with `DSolve`, naming the resulting output `step1`. We then name `y` the result obtained in `step1` by extracting the formula for `y[x]` from `step1` with `Part` (`[[...]]`) and replacing `C[1]` by `c` with `ReplaceAll` (`/.`).

```
Clear[x, y]

step1 = DSolve[y'[x] == -αy[x], y[x], x]
```

$$\{\{y[x] \to e^{-x\alpha} C[1]\}\}$$

```
y = step1[[1, 1, 2]]/.C[1] → c
```

$$c e^{-x\alpha}$$

Using the facts that $u' = -\frac{k}{Q_B}(u - v)$ and $v = u - y$, we are able to use `DSolve` to find $u(x)$.

```
step2 = DSolve[{u'[x] == -k/Q_S cExp[-αx], u[0] == u0},
u[x], x]
```

$$\left\{\left\{u[x] \to \frac{e^{-x\alpha}\left(ck - ce^{x\alpha}k + e^{x\alpha}\mathrm{u0}\,\alpha Q_S\right)}{\alpha Q_S}\right\}\right\}$$

Note that we use `cap1` to represent L.

Because $y = u - v$, $v = u - y$. Consequently, because $v(L) = 0$ we are able to compute `c`.

```
leftside = step2[[1, 1, 2]] - y/.x->cap1
```

$$-ce^{-\mathrm{cap1}\alpha} + \frac{e^{-\mathrm{cap1}\alpha}\left(ck - ce^{\mathrm{cap1}\alpha}k + e^{\mathrm{cap1}\alpha}\mathrm{u0}\,\alpha Q_S\right)}{\alpha Q_S}$$

```
cval = Solve[leftside == 0, c]
```

$$\left\{\left\{c \to \frac{e^{\mathrm{cap1}\alpha}\mathrm{u0}\,\alpha Q_S}{-k + e^{\mathrm{cap1}\alpha}k + \alpha Q_S}\right\}\right\}$$

and determine u and v. Next, we substitute the value of C into the formula for u and v.

```
u = Simplify[step2[[1, 1, 2]]/.cval[[1]]]
```

$$\frac{\mathrm{u0}\left(\left(-1 + e^{(\mathrm{cap1}-x)\alpha}\right)k + \alpha Q_S\right)}{\left(-1 + e^{\mathrm{cap1}\alpha}\right)k + \alpha Q_S}$$

```
v = Simplify[u - y/.cval[[1]]]
```

$$\frac{e^{-x\alpha}\left(e^{\mathrm{cap1}\alpha} - e^{x\alpha}\right)\mathrm{u0}\,(k - \alpha Q_S)}{\left(-1 + e^{\mathrm{cap1}\alpha}\right)k + \alpha Q_S}$$

For example, in healthy adults, typical urea nitrogen levels are 11–23 milligrams per deciliter, while serum creatinine levels range from 0.6 to 1.2 milligrams per deciliter and the total volume of blood is 4–5 L.

Suppose that hemodialysis is performed on a patient with urea nitrogen level of 34 mg/dL and serum creatinine level of 1.8 using a dialyser with $k = 2.25$ and $L = 1$. If the flow rate of blood, Q_B, is 2 dL/minute while the flow rate of the dialysate, Q_D, is 4 dL/minute, will the level of wastes in the patient's blood reach normal levels after dialysis is performed?

After defining the appropriate constants, we evaluate u and v

```
α = k/Q_S − k/Q_O ;
k = 2.25;
cap1 = 1;
u0 = 34 + 1.8;
Q_S = 2; Q_O = 4;
u//Simplify
v//Simplify
```

$-14.2623 + 50.0623e^{-0.5625x}$

$-14.2623 + 25.0312e^{-0.5625x}$

and then graph u and v on the interval [0,1] with `Plot` in Figure 2-10. Remember that the dialysate is moving in the direction opposite the blood. Thus, we see from the graphs that as levels of waste in the blood decrease, levels of waste in the

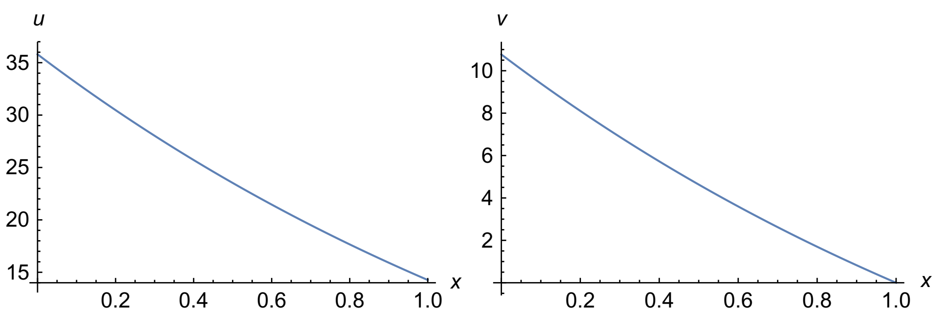


Figure 2-10 Remember that the dialysate moves in the opposite direction as the blood

dialysate increase and at the end of the dialysis procedure, levels of waste in the blood are within normal ranges.

```
p1 = Plot[u, {x, 0, 1}, AxesLabel → {x, "u"}, AxesStyle → Black];
p2 = Plot[v, {x, 0, 1}, AxesLabel → {x, "v"}, AxesStyle → Black];
Show[GraphicsRow[{p1, p2}]]
```

Typically, hemodialysis is performed 3–4 hours at a time 3 or 4 times per week. In some cases, a kidney transplant can free patients from the restrictions of dialysis. Of course, transplants have other risks not necessarily faced by those on dialysis; the number of available kidneys also affects the number of transplants performed. For example, in 1991 over 130,000 patients were on dialysis while only 7000 kidney transplants had been performed.

2.3 Homogeneous Equations

Definition 6 (Homogeneous Differential Equation). *A differential equation that can be written in the form*

$$M(x, y)\,dx + N(x, y)\,dy = 0,$$

where

$$M(tx, ty) = t^n M(x, y) \quad \textit{and} \quad N(tx, ty) = t^n N(x, y)$$

is called a ***homogeneous differential equation of degree*** *n.*

It is a good exercise to show that an equation is homogeneous if we can write it in either of the forms $dy/dx = F(y/x)$ or $dy/dx = G(x/y)$.

EXAMPLE 2.3.1: Show that the equation $\left(x^2 + xy\right) dx - y^2\,dy = 0$ is homogeneous.

SOLUTION: Let $M(x, y) = x^2 + xy$ and $N(x, y) = -y^2$. Because $M(tx, ty) = (tx)^2 + (tx)(ty) = t^2\left(x^2 + xy\right) = t^2 M(x, y)$ and $N(tx, ty) = -t^2 y^2 = t^2 N(x, y)$, the equation $\left(x^2 + xy\right) dx - y^2\,dy = 0$ is homogeneous of degree two.

■

Homogeneous equations can be reduced to separable equations by either of the substitutions

$$y = ux \quad \text{or} \quad x = vy.$$

Generally, use the substitution $y = ux$ if $N(x, y)$ is less complicated than $M(x, y)$ and use $x = vy$ if $M(x, y)$ is less complicated than $N(x, y)$. If a difficult integration problem is encountered after a substitution is made, try the other substitution to see if it yields an easier problem.

EXAMPLE 2.3.2: Solve the equation $\left(x^2 - y^2\right) dx + xy\, dy = 0$.

SOLUTION: For this example, $M(x, y) = x^2 - y^2$ and $N(x, y) = xy$. Then, $M(tx, ty) = t^2 M(x, y)$ and $N(tx, ty) = t^2 N(x, y)$, which means that $\left(x^2 - y^2\right) dx + xy\, dy = 0$ is a homogeneous equation of degree two. Assume $x = vy$. Then, $dx = v\, dy + y\, dv$ and substituting into the equation and simplifying yields

$$\begin{aligned}
0 &= \left(x^2 - y^2\right) dx + xy\, dy \\
0 &= \left(v^2 y^2 - y^2\right) (v\, dy + y\, dv) + vy \cdot y\, dy \\
0 &= \left(v^2 - 1\right) (v\, dy + y\, dv) + v\, dy \\
0 &= v^3\, dy + y \left(v^2 - 1\right) dv.
\end{aligned}$$

We solve this equation by rewriting it in the form

$$\frac{1}{y} dy = \frac{1 - v^2}{v^3} dv = \left(\frac{1}{v^3} - \frac{1}{v}\right) dv$$

and integrating. This yields

$$\ln |y| = -\frac{1}{2v^2} - \ln |v| + C_1,$$

which can be simplified as

$$\ln |vy| = -\frac{1}{2v^2} + C_1, \quad \text{so} \quad vy = Ce^{-1/(2v^2)}, \text{ where } C = \pm e^{C_1}.$$

Because $x = vy$, $v = x/y$, and resubstituting into the above equation yields

$$x = Ce^{-y^2/(2x^2)}$$

as a general solution of the equation $\left(x^2 - y^2\right) dx + xy\, dy = 0$. We see that `DSolve` is able to solve the equation as well.

```
Clear[x, y]

gensol = DSolve[x^2 - y[x]^2 + xy[x]y'[x] == 0, y[x], x]
```

$\{\{y[x] \to -x\sqrt{C[1] - 2\text{Log}[x]}\}, \{y[x] \to x\sqrt{C[1] - 2\text{Log}[x]}\}\}$

The result means that a general solution of the equation is $y^2 = x^2\,(C - 2\ln|x|)$. We can graph this implicit solution for various values of C by solving this equation for C

```
f = Solve[y==gensol[[1, 1, 2]], C[1]]
```

$$\left\{\left\{C[1] \to \frac{y^2 + 2x^2\text{Log}[x]}{x^2}\right\}\right\}$$

```
f[[1, 1, 2]]
```

$$\frac{y^2 + 2x^2\text{Log}[x]}{x^2}$$

and then noting that graphs of the equation $y^2 = x^2\,(C - 2\ln|x|)$ for various values of C are the same as the graphs of the level curves of the function $f(x, y) = \left(y^2 + 2x^2 \ln|x|\right) / \left(2x^2\right)$.

The `ContourPlot` command graphs several level curves $z = f(x, y)$, C a constant, of the function $z = f(x, y)$. We may instruct Mathematica to graph the level curves of $z = f(x, y)$ for particular values of C by including the `Contours` option. For example, the level curves of $f(x, y) = \left(y^2 + 2x^2 \ln|x|\right) / \left(2x^2\right)$ that intersect the x-axis at $x = 1, 3/2, 2, \ldots, 19/2$, and 10 are the contours with values obtained by replacing each occurrence of y in $f(x, y)$ by 0 and x by 1, 3/2, 2, ..., 19/2, and 10 which we do now with `Table` and `ReplaceAll` (`/.`), naming the resulting set of ten numbers `contourvals`.

```
contourvalues = Table[f[[1, 1, 2]]/.{x → i, y → 0},
  {i, 1, 10, .5}];

cp1 = ContourPlot[f[[1, 1, 2]], {x, 0.01, 10}, {y, -5, 5},
  PlotPoints → 150, Frame → False, Contours →
    contourvalues, Axes → Automatic, ContourStyle →
```

```
  {{Gray, Thickness[.01]}}, ContourShading → False,
  AxesOrigin → {0, 0}, AxesStyle → Black, AxesLabel →
{x, y}, PlotLabel → "(a)"];
```

graphs several level curves of $z = f(x, y)$ for $0.01 \leq x \leq 10$ and $-5 \leq y \leq 5$ and names the resulting graphics object cp1. cp1 is not displayed because we include a semi-colon (;) at the end of the ContourPlot command. The option Contours->contourvals instructs Mathematica to draw contours with values given in the list of numbers contourvals. We use StreamPlot to graph the direction field associated with the equation on the same rectangle, $[0.01, 10 \times [-5, 5]$

We avoid $x = 0$ because $f(x, y)$ is undefined if $x = 0$.

```
dirf = StreamPlot[{1, (y^2 − x^2)/(xy)}, {x, 0.01, 10},
  {y, −5, 5}, Frame → False, StreamStyle → Black];
```

and then display cp1 (Figure 2-11(a)) and the direction field together with Show together with GraphicsRow in Figure 2-11(b).

```
p2 = Show[cp1, dirf, PlotLabel → "(b)"];
Show[GraphicsRow[{cp1, p2}]]
```

■

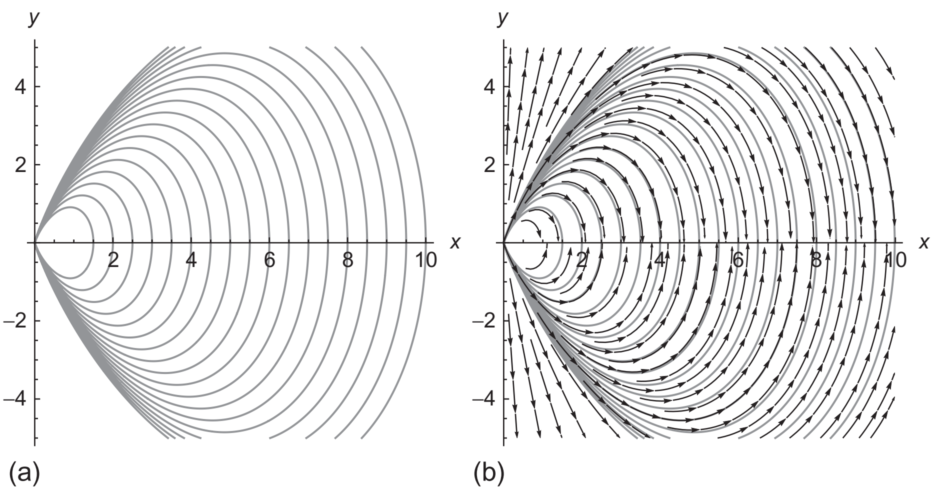


Figure 2-11 (a) Various solutions for the homogeneous equation $\left(x^2 - y^2\right) dx + xy\, dy = 0$. (b) Various solutions and direction field for the homogeneous equation $\left(x^2 - y^2\right) dx + xy\, dy = 0$

EXAMPLE 2.3.3: Solve $\left(y^2 + 2xy\right) dx - x^2 dy = 0$.

SOLUTION: In this case, letting $F(t) = t^2 + 2t$, we note that $dy/dx = F(y/x) = (y/x)^2 + 2(y/x)$ so the equation is homogeneous.

Let $y = ux$. Then, $dy = u\,dx + x\,du$. Substituting into $\left(y^2 + 2xy\right) dx - x^2 dy = 0$ and separating gives us

$$\begin{aligned}
\left(y^2 + 2xy\right) dx - x^2 dy &= 0 \\
\left(u^2x^2 + 2ux^2\right) dx - x^2(u\,dx + x\,du) &= 0 \\
\left(u^2 + 2u\right) dx - (u\,dx + x\,du) &= 0 \\
\left(u^2 + u\right) dx &= -x\,du \\
\frac{1}{u\,(u+1)} du &= -\frac{1}{x} dx.
\end{aligned}$$

Integrating the left- and right-hand sides of this equation with `Integrate`,

```
Integrate[1/(u(u + 1)), u]
```

```
Log[u] − Log[1 + u]
```

```
Integrate[1/x, x]
```

```
Log[x]
```

exponentiating, resubstituting $u = y/x$, and solving for y gives us

$$\begin{aligned}
\ln|u| - \ln|u+1| &= -\ln|x| + C \\
\frac{u}{u+1} &= \frac{C}{x} \\
\frac{\frac{y}{x}}{\frac{y}{x}+1} &= \frac{C}{x} \\
y &= \frac{Cx}{x-C}.
\end{aligned}$$

```
Solve[(y/x)/(y/x + 1) == cx, y]
```

$$\left\{\left\{y \to -\frac{cx^2}{-1 + cx}\right\}\right\}$$

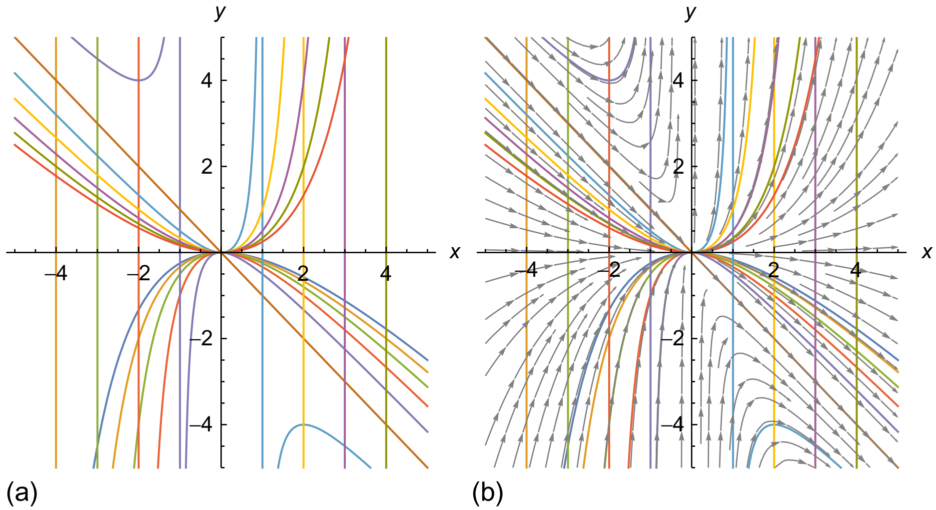


Figure 2-12 (a) Graphs of several solutions of $\left(y^2 + 2xy\right) dx - x^2 dy = 0$. (b) Graphs of several solutions together with the direction field

We confirm this result with `DSolve` and then graph several solutions with `Plot` in Figure 2-12(a).

```
sol = DSolve[y[x]^2 + 2xy[x] − x^2y′[x] == 0, y[x], x]
```

$$\left\{\left\{y[x] \rightarrow -\frac{x^2}{x - C[1]}\right\}\right\}$$

```
Solve[y[x]^2 + 2xy[x] − x^2y′[x] == 0, y′[x]]
```

$$\left\{\left\{y'[x] \rightarrow \frac{2xy[x] + y[x]^2}{x^2}\right\}\right\}$$

```
toplot = Table[sol[[1, 1, 2]]/.C[1] → i, {i, −5, 5}];
p1 = Plot[toplot, {x, −5, 5}, PlotRange → {−5, 5},
  AxesStyle → Black, AxesLabel → {x, y}, PlotLabel → "(a)",
  AspectRatio → 1];
```

We use `StreamPlot` to graph the direction field and then display the direction field together with the solutions in Figure 2-12(b).

```
p2a = StreamPlot[{1, (2xy + y^2)/x^2}, {x, −5, 5},
  {y, −5, 5}, StreamStyle → Gray];
p2 = Show[p1, p2a, PlotLabel → "(b)", AspectRatio → 1];
Show[GraphicsRow[{p1, p2}]]
```

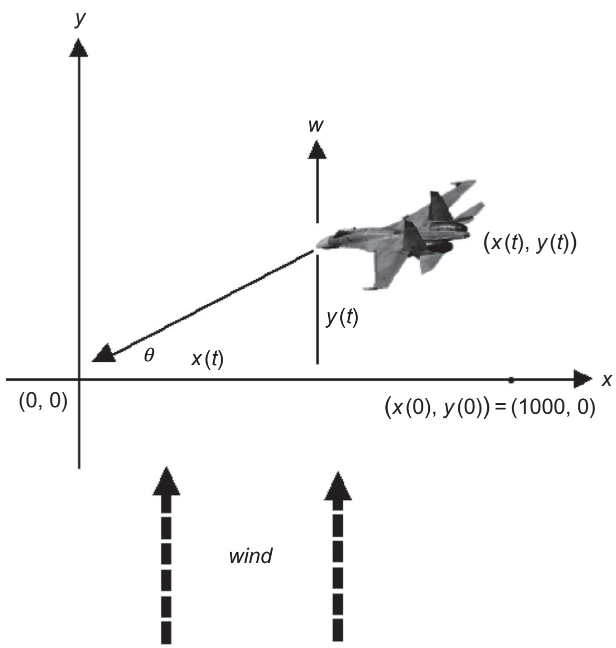
In Figure 2-12, observe that the vertical lines correspond to discontinuities in the solution and are not graphs of the solutions to the equations. *Vertical lines are never a portion or part of the graph of a real-valued function of a single variable.* With careful use of the `Exclusion` option, the vertical lines can be deleted from the plots.

■

Sources: A particularly interesting and fun-to-read discussion of flight paths and models of pursuit can be found in *Differential Equations: A Modeling Perspective* by Robert L. Borrelli and Courtney S. Coleman and published by John Wiley & Sons.

Application: Models of Pursuit

Suppose that one object pursues another whose motion is known by a predetermined strategy. For example, suppose that an airplane is positioned at $B(1000, 0)$ to fly to another airport $A(0, 0)$ that is 1000 miles directly west of its position B, as illustrated in the following figure. Assume that the airplane aims toward A at all times. If the wind goes from south to north at a constant speed, w, and the airplane's speed in still air is b, determine conditions on b so that the airplane eventually arrives at A and describe its path.



As described, the speed of the airplane, b, must be greater than the speed of the wind, w: $b > w$, in order for the plane to arrive at A. Observe that dx/dt describes the airplane's velocity in the x direction:

$$\frac{dx}{dt} = -b\cos\theta = \frac{-bx}{\sqrt{x^2+y^2}},$$

because from right-triangle trigonometry we know that $\cos\theta = \text{adjacent}/\text{hypotenuse} = x/\sqrt{x^2+y^2}$. Similarly,

$$\frac{dy}{dt} = -b\sin\theta + w = \frac{-by}{\sqrt{x^2+y^2}} + w,$$

so

$$\frac{dy}{dx} = \frac{dy/dt}{dx/dt} = \frac{\dfrac{-by}{\sqrt{x^2+y^2}} + w}{\dfrac{-bx}{\sqrt{x^2+y^2}}} = \frac{by - w\sqrt{x^2+y^2}}{bx}.$$

This is a homogeneous equation (of degree one) because it can be written in the form $dy/dx = F(y/x)$:

$$\frac{dy}{dx} = \frac{by - w\sqrt{x^2+y^2}}{bx} = \frac{y}{x} - \frac{w}{b}\sqrt{1+\left(\frac{y}{x}\right)^2}.$$

Therefore, we must solve the initial-value problem

$$\begin{cases} \dfrac{dy}{dx} = \dfrac{by - w\sqrt{x^2+y^2}}{bx} \\ y(1000) = 0 \end{cases}.$$

In this case, we see `DSolve` is both able to find a general solution of the equation

```
Clear[x, y, w, b]
DSolve[y'[x] == (by[x] − wSqrt[x^2 + y[x]^2])/(bx),
  y[x], x]
```

$$\left\{\left\{y[x] \to x\mathrm{Sinh}\left[\frac{bC[1] - w\mathrm{Log}[x]}{b}\right]\right\}\right\}$$

as well as solve the initial-value problem.

```
Clear[x, y, w, b]
DSolve[{y'[x] == (by[x] − wSqrt[x^2 + y[x]^2])/(bx), y[1000] == 0},
  y[x], x]
```

$$\left\{\left\{y[x] \to x\mathrm{Sinh}\left[\frac{w\mathrm{Log}[1000] - w\mathrm{Log}[x]}{b}\right]\right\}\right\}$$

Alternatively, letting $y = ux$, differentiating to obtain $dy = u\,dx + x\,du$, and substituting into the homogeneous equation results in the separable equation

$$\frac{dy}{dx} = \frac{du}{dx}x + u = \frac{bux - w\sqrt{x^2 + u^2x^2}}{bx}$$
$$\frac{du}{dx}x + u = u - \frac{w}{b}\sqrt{1+u^2}$$
$$\frac{1}{\sqrt{1+u^2}}du = -\frac{w}{b}\frac{1}{x}dx$$

```
y = ux;

eqn = Dt[y] == (by - wSqrt[x^2 + y^2])/(bx);

step1 = PowerExpand[Simplify[eqn]]
```

$$\frac{\sqrt{1+u^2}\,w}{b} + x\mathrm{Dt}[u] + u\mathrm{Dt}[x] == u$$

Integrating the left-hand side of this equation yields $\int \frac{1}{\sqrt{1+u^2}}du = \ln\left|u + \sqrt{1+u^2}\right| + C_1$

```
leftint = Integrate[1/Sqrt[1 + u^2], u]

ArcSinh[u]

leftint = TrigToExp[leftint]
```

$$\mathrm{Log}\left[u + \sqrt{1+u^2}\right]$$

and integrating the right results in $-\frac{w}{b}\int \frac{1}{x}dx = -\frac{w}{b}\ln|x| + C_2$. Note that absolute value bars are not necessary because x and y are both positive and, hence, u is nonnegative. Thus, $\ln\left(u + \sqrt{1+u^2}\right) = -\frac{w}{b}\ln x + C$.

```
rightint = Integrate[-w/(bx), x] + c
```

$$c - \frac{w\mathrm{Log}[x]}{b}$$

Because $y(1000) = 0$, $C = \frac{w}{b}\ln 1000$

```
cval = Solve[leftint == rightint/.{x → 1000, u → 0}, c]
```

$$\left\{\left\{c \to \frac{w\mathrm{Log}[1000]}{b}\right\}\right\}$$

and $\ln\left(u + \sqrt{1+u^2}\right) = -\frac{w}{b}\ln x + \frac{w}{b}\ln 1000$.

```
step2 = leftint == rightint/.cval[[1]]
```

$$\text{Log}\left[u+\sqrt{1+u^2}\right] == \frac{w\text{Log}[1000]}{b} - \frac{w\text{Log}[x]}{b}$$

Solving for u gives us

$$\begin{aligned}
\ln\left(u+\sqrt{1+u^2}\right) &= \ln\left(\frac{x}{1000}\right)^{-w/b} \\
u+\sqrt{1+u^2} &= \left(\frac{x}{1000}\right)^{-w/b} \\
\sqrt{1+u^2} &= \left(\frac{x}{1000}\right)^{-w/b} - u \\
1+u^2 &= \left(\frac{x}{1000}\right)^{-2w/b} - 2u\left(\frac{x}{1000}\right)^{-w/b} + u^2 \\
2u\left(\frac{x}{1000}\right)^{-w/b} &= \left(\frac{x}{1000}\right)^{-2w/b} - 1 \\
u &= \frac{1}{2}\left[\left(\frac{x}{1000}\right)^{-w/b} - \left(\frac{x}{1000}\right)^{w/b}\right].
\end{aligned}$$

```
step3 = Solve[step2, u]
```

$$\left\{\left\{u \to \frac{1}{2}\left(10^{\frac{3w}{b}} x^{-\frac{w}{b}} - 10^{-\frac{3w}{b}} x^{\frac{w}{b}}\right)\right\}\right\}$$

We solve for y by resubstituting $u = y/x$ and multiplying by x:

$$\begin{aligned}
\frac{y}{x} &= \frac{1}{2}\left[\left(\frac{x}{1000}\right)^{-w/b} - \left(\frac{x}{1000}\right)^{w/b}\right] \\
y &= \frac{1}{2}x\left[\left(\frac{x}{1000}\right)^{-w/b} - \left(\frac{x}{1000}\right)^{w/b}\right] \\
y &= \frac{1}{2^x}\left[\left(x\left(10^{-3}\right)\right)^{-w/b} - \left(x\left(10^{-3}\right)\right)^{w/b}\right].
\end{aligned}$$

```
Clear[y]
```

```
y[x_] = xstep3[[1, 1, 2]]
```

$$\frac{1}{2}x\left(10^{\frac{3w}{b}} x^{-\frac{w}{b}} - 10^{-\frac{3w}{b}} x^{\frac{w}{b}}\right)$$

We graph y for various values of w/b by setting $b = 1$ and then using `Table` to generate the value of y for $w = 0.25, 0.50, \ldots, 2.0$. These functions are then graphed with `Plot` in Figure 2-13. Notice that the airplane never arrives at A if $w/b \geq 1$.

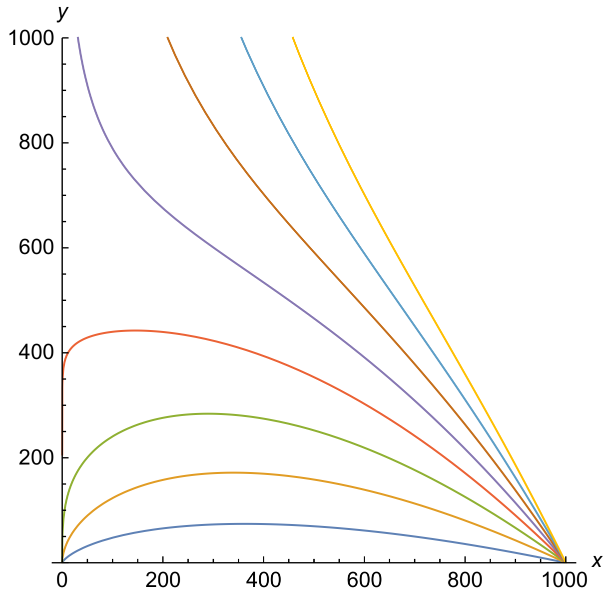


Figure 2-13 If $w/b \geq 1$, the airplane never reaches its destination

```
b = 1;
toplot = Table[y[x], {w, 0.2, 2.0, .25}];
Plot[toplot, {x, 0, 1000}, PlotRange → {0, 1000},
  AxesLabel → {x, y}, AxesStyle → Black,
  AspectRatio → 1]
```

2.4 Exact Equations

Definition 7 (Exact Differential Equation). *A differential equation that can be written in the form*

$$M(x,y)\,dx + N(x,y)\,dy = 0$$

where

$$M(x,y)\,dx + N(x,y)\,dy = \frac{\partial f}{\partial x}(x,y)\,dx + \frac{\partial f}{\partial y}(x,y)\,dy$$

*for some function $z = f(x,y)$ is called an **exact differential equation**.*

We can show that the differential equation $M(x, y)\,dx + N(x, y)\,dy = 0$ is exact if and only if $\partial M/\partial y = \partial N/\partial x$.

EXAMPLE 2.4.1: Show that the equation $2xy^3\,dx + \left(1 + 3x^2y^2\right) dy = 0$ is exact and that the equation $x^2y\,dx + 5xy^2\,dy = 0$ is not exact.

SOLUTION: Because

$$\frac{\partial}{\partial y}\left(2xy^3\right) = 6xy^2 = \frac{\partial}{\partial x}\left(1 + 3x^2y^2\right),$$

the equation $2xy^3\,dx + \left(1 + 3x^2y^2\right) dy = 0$ is an exact equation. On the other hand, the equation $x^2y\,dx + 5xy^2\,dy = 0$ is not exact because

$$\frac{\partial}{\partial y}\left(x^2y\right) = x^2 \neq 5y^2 = \frac{\partial}{\partial x}\left(5xy^2\right).$$

(However, the equation $x^2y\,dx + 5xy^2\,dy = 0$ is separable.)

■

If an equation is exact, we can find a function $z = f(x, y)$ such that $M(x, y) = \frac{\partial f}{\partial x}(x, y)$ and $N(x, y) = \frac{\partial f}{\partial y}(x, y)$.

1. Assume that $M(x, y) = \frac{\partial f}{\partial x}(x, y)$ and $N(x, y) = \frac{\partial f}{\partial y}(x, y)$.
2. Integrate $M(x, y)$ with respect to x. (Add an arbitrary function of y, $g(y)$.)
3. Differentiate the result in Step 2 with respect to y and set the result equal to $N(x, y)$. Solve for $g'(y)$.
4. Integrate $g'(y)$ with respect to y to obtain an expression for $g(y)$. (There is no need to include an arbitrary constant.)
5. Substitute $g(y)$ into the result obtained in Step 2 for $f(x, y)$.
6. A general solution is $f(x, y) = C$ where C is a constant.
7. If given an initial-value problem, apply the initial condition to determine C.

Remark. A similar algorithm can be stated so that in Step 2 $N(x, y)$ is integrated with respect to y.

EXAMPLE 2.4.2: Solve $2x \sin y\,dx + \left(x^2 \cos y - 1\right) dy = 0$ subject to $y(0) = 1/2$.

SOLUTION: The equation $2x \sin y \, dx + \left(x^2 \cos y - 1\right) dy = 0$ is exact because

$$\frac{\partial}{\partial y}(2x \sin y) = 2x \cos y = \frac{\partial}{\partial x}\left(x^2 \cos y - 1\right).$$

Let $z = f(x, y)$ be a function with $\partial f/\partial x = 2x \sin y$ and $\partial f/\partial y = x^2 \cos y - 1$. Then, integrating $\partial f/\partial x$ with respect to x yields

$$f(x, y) = \int 2x \sin y \, dx = x^2 \sin y + g(x).$$

Notice that the arbitrary function $g = g(y)$ of y serves as a "constant" of integration with respect to x. Because we have $\partial f/\partial y = x^2 \cos y - 1$ from the differential equation, and

$$\frac{\partial f}{\partial y}(x, y) = x^2 \cos y + g'(y)$$

from differentiation of $f(x, y)$ with respect to y, $g'(y) = -1$. Integrating $g'(y)$ with respect to y gives us $g(y) = -y$. Therefore, $f(x, y) = x^2 \sin y - y$, so a general solution of the exact equation is $x^2 \sin y - y = C$, where C is a constant. Because our solution requires that $y(0) = 1/2$, we must find the solution in the family of solutions that passes through the point $(0, 1/2)$. Substituting these values of x and y into the general solution, we obtain $0^2 \cdot \sin(1/2) - 1/2 = C$ so that $C = -1/2$. Therefore, the desired solution is $x^2 \sin y - y = -1/2$. We are able to use `DSolve` to solve the initial-value problem implicitly as well.

Notice that we do not have to include the constant in calculating $g(y)$ because we combine it with the constant in the general solution.

```
Clear[x, y]

partsol = DSolve[{2xSin[y[x]] + (x^2Cos[y[x]] - 1)y'[x] == 0,
  y[0] == 1/2}, y[x], x]
```

$\texttt{Solve}\left[x^2\texttt{Sin}[y[x]] - y[x] == -\frac{1}{2}, y[x]\right]$

```
partsol[[1]]
```

$x^2\texttt{Sin}[y[x]] - y[x] == -\frac{1}{2}$

```
partsol[[1, 1]]
```

$x^2\texttt{Sin}[y[x]] - y[x]$

To graph the function `partsol[[1,1,2]]`, we use `ContourPlot` to graph the equation for $-10 \le x \le 10$ and $-10 \le y \le 10$ in

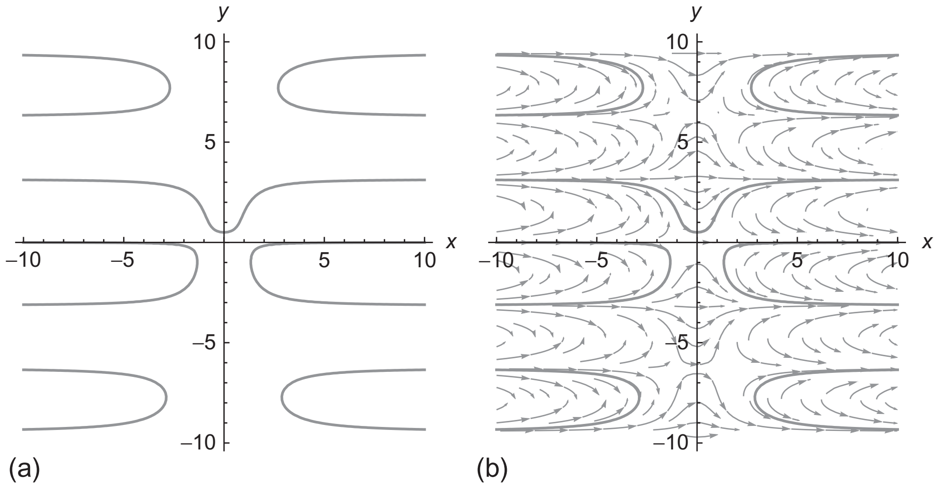


Figure 2-14 Plot of $x^2 \sin y - y = -1/2$

Figure 2-14(a). We include the option `AxesOrigin->{0,0}` to specify that the axes intersect at the point $(0, 0)$. We use `StreamPlot` to graph the direction field and show the direction field together with several solutions in Figure 2-14(b).

```
p1 = ContourPlot[Evaluate[partsol[[1, 1]]/.y[x] → y],
  {x, −10, 10}, {y, −10, 10}, Frame → False, ContourShading
  → False, Contours → {−1/2}, ContourStyle → {{Black,
  Thickness[.01]}}, Axes → Automatic, AxesOrigin → {0, 0},
  AxesLabel → {x, y}, PlotLabel → "(a)", AxesStyle
  → Black];
p2a = StreamPlot[{1, −2xSin[y]/(x^2Cos[y] − 1)},
  {x, −10, 10}, {y, −10, 10}, StreamStyle → Gray];
p2 = Show[p1, p2a, PlotLabel → "(b)", AspectRatio → 1];
Show[GraphicsRow[{p1, p2}]]
```

■

The following example illustrates how we can use Mathematica to assist us in carrying out the necessary steps encountered when solving an exact equation.

EXAMPLE 2.4.3: Solve
$\left(2x - y^2 \sin(xy)\right) dx + (\cos(xy) - xy \sin(xy))\, dy = 0.$

SOLUTION: We begin by identifying $M(x,y) = 2x - y^2 \sin(xy)$ and $N(x,y) = \cos(xy) - xy\sin(xy)$. We then define `capm`, corresponding to M, and `capn`, corresponding to N. We then see that the equation is exact because $\partial M/\partial y = \partial N/\partial x$.

```
capm[x_, y_] = 2x - y^2Sin[xy];

capn[x_, y_] = Cos[xy] - xySin[xy];

D[capm[x, y], y] == D[capn[x, y], x]

True
```

Next, we compute $\int M(x,y)\, dx$ and add an arbitrary function of y, `g[y]`, to the result.

```
f = Integrate[capm[x, y], x] + g[y]

x^2 + yCos[xy] + g[y]
```

Differentiating f with respect to y gives us

```
D[f, y] == capn[x, y]
```

and because we must have that $\partial f/\partial y = N(x,y)$, we obtain the equation

```
Cos[xy] - xySin[xy] + g'[y] == Cos[xy] - xySin[xy]
```

which we solve for $g'(y)$ with `Solve`.

```
Solve[D[f, y] == capn[x, y], g'[y]]

{{g'[y] → 0}}

f = f/.g[y] → 0
```

Thus, $g(y)$ is a (real-valued) constant and a general solution of the equation is $x^2 + y\cos(xy) = C$. We can graph this general solution for various values of C by observing that the level curves of the function $z = x^2 + y\cos(xy)$ correspond to the graphs of the equation $x^2 + y\cos(xy) = C$ for various values of C.

$$x^2 + y\text{Cos}[xy]$$

We now use `ContourPlot` to graph several level curves of $z = x^2 + y\cos(xy)$ on the rectangle $[0, 3\pi] \times [0, 3\pi]$ in Figure 2-15. In this case, the option `Frame->False` instructs Mathematica to not place a frame around the resulting graphics object, the option `Axes->Automatic` specifies that axes are to be placed on the graph, `AxesOrigin->{0,0}` specifies that the axes are to intersect at the point $(0, 0)$, `AxesStyle->GrayLevel[.5]` specifies that the axes are to be drawn in a light gray, `ContourShading->False` specifies that the region between contours is to not be shaded and the option `PlotPoints->150` helps assure that the resulting contours appear smooth. `Contours->n` instructs Mathematica to graph n contours. If you prefer that the contours correspond to specific function values us `Contours->{list of function values}`. Use `ContourStyle` to specify the style of your contours, such as their thickness and color.

```
cp1 = ContourPlot[f, {x, 0, 3Pi}, {y, 0, 3Pi},

  Frame → False, Axes → Automatic, AxesOrigin → {0, 0},
```

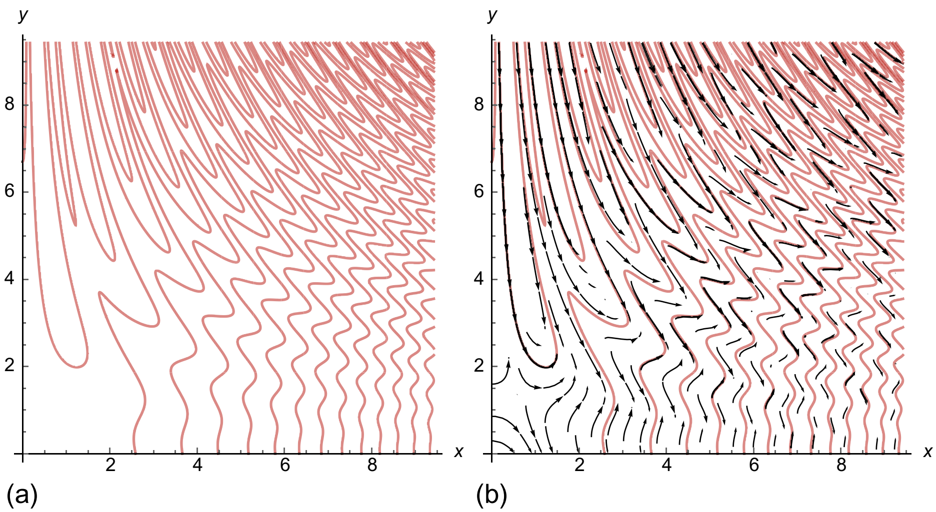


Figure 2-15 (a) Level curves of $z = x^2 + y\cos(xy)$. (b) Level curves of $z = x^2 + y\cos(xy)$ together with the direction field

```
    AxesLabel → {x, y}, ContourShading → False,
    Contours → 15, AxesStyle → Black,
    ContourStyle → {{CMYKColor[0, 0.89, 0.94, 0.28],
    Thickness[.0075]}}, PlotPoints → 150,
    PlotLabel → "(a)"];

p2 = StreamPlot[{1, −capm[x, y]/capn[x, y]},
    {x, 0, 3Pi}, {y, 0, 3Pi}, StreamStyle → Black];

p3 = Show[cp1, p2, PlotLabel → "(b)"];

Show[GraphicsRow[{cp1, p3}]]
```

We see that DSolve is able to find an implicit solution of the equation after we rewrite it in the form $\left(2x - y^2 \sin(xy)\right) + (\cos(xy) - xy\sin(xy))\, y' = 0$.

```
gensol = DSolve[capm[x, y[x]] + capn[x, y[x]]y'[x] == 0,
y[x], x]

Solve[x^2 + Cos[xy[x]]y[x] == C[1], y[x]]

step2 = gensol[[1, 1]]

x^2 + Cos[xy[x]]y[x]

implicitplot = step2/.y[x] → y

x^2 + yCos[xy]
```

■

2.5 Linear Equations

Definition 8 (First-Order Linear Equation). *A differential equation of the form*

$$a_1(x)\frac{dy}{dx} + a_0(x)y = f(x), \tag{2.2}$$

where $a_1(x)$ is not identically the zero function, is a first-order ***linear differential equation****.*

Assuming that $a_1(x)$ is not identically the zero function, dividing equation (2.2) by $a_1(x)$ gives us the **standard form** of the first-order linear equation:

$$\frac{dy}{dx} + p(x)y = q(x). \tag{2.3}$$

If $q(x)$ is identically the zero function, we say that the equation is **homogeneous**. The **corresponding homogeneous equation** of equation (2.3) is

$$\frac{dy}{dx} + p(x)y = 0. \tag{2.4}$$

2.5.1 Integrating Factor Approach

Multiplying equation (2.3) by $e^{\int p(x)\,dx}$ yields

$$e^{\int p(x)\,dx}\frac{dy}{dx} + e^{\int p(x)\,dx}p(x)y = e^{\int p(x)\,dx}q(x).$$

By the product rule and the Fundamental Theorem of Calculus,

$$\frac{d}{dx}\left(e^{\int p(x)\,dx}y\right) = e^{\int p(x)\,dx}\frac{dy}{dx} + e^{\int p(x)\,dx}p(x)y$$

so equation (2.3) becomes

$$\frac{d}{dx}\left(e^{\int p(x)\,dx}y\right) = e^{\int p(x)\,dx}q(x).$$

Integrating and dividing by $e^{\int p(x)\,dx}$ yields a general solution of $y' + p(x)y = q(x)$:

$$\begin{aligned} e^{\int p(x)\,dx}y &= \int e^{\int p(x)\,dx}q(x)\,dx \\ y &= \frac{1}{e^{\int p(x)\,dx}}\int e^{\int p(x)\,dx}q(x)\,dx = e^{-\int p(x)\,dx}\int e^{\int p(x)\,dx}q(x)\,dx. \end{aligned}$$

The term $\mu(x) = e^{\int p(x)\,dx}$ is called an **integrating factor** for the linear equation (2.3).

Thus, first-order linear equations can always be solved, although the resulting integrals may be difficult or impossible to evaluate exactly.

As we see with the following command, `DSolve` is always able to solve first-order linear differential equations, although the result might contain unevaluated integrals.

```
Clear[x, y, p, q]
DSolve[y'[x] + p[x]y[x] == q[x], y[x], x]
```

$$\left\{\left\{y[x] \to e^{\int_1^x -p[K[1]]\,dK[1]} C[1] + e^{\int_1^x -p[K[1]]\,dK[1]} \int_1^x e^{-\int_1^{K[2]} -p[K[1]]\,dK[1]} q[K[2]]\,dK[2]\right\}\right\}$$

EXAMPLE 2.5.1: Solve $x\,dy/dx + y = x\cos x$, $x > 0$.

SOLUTION: First, we place the equation in the form used in the derivation above. Dividing the equation by x yields

$$\frac{dy}{dx} + \frac{1}{x}y = \cos x, \tag{2.5}$$

where $p(x) = 1/x$ and $q(x) = \cos x$. Then, an integrating factor is

$$\mu(x) = e^{\int \frac{1}{x}dx} = e^{\ln|x|} = x, \quad \text{for } x > 0,$$

and multiplying equation (2.5) by the integrating factor gives us

$$\frac{d}{dx}(xy) = x\frac{dy}{dx} + y = x\cos x.$$

Integrating once we have

$$xy = \int x\cos x\,dx.$$

Using the integration by parts formula, $\int u\,dv = uv - \int v\,du$, with $u = x$ and $dv = \cos x\,dx$, we obtain $du = dx$ and $v = \sin x$ so

$$xy = \int x\cos x\,dx = x\sin x - \int \sin x\,dx = x\sin x + \cos x + C.$$

Therefore, a general solution of the equation $x\,dy/dx + y = x\cos x$ for $x > 0$ is $y = (x\sin x + \cos x + C)/x$. We see that `DSolve` is also successful in finding a general solution of the equation.

If we want to solve the equation for $x < 0$, then we would have $e^{\int \frac{1}{x}dx} = e^{\ln|x|} = -x$ for $x < 0$.

```
Clear[x, y]
gensol = DSolve[xy'[x] + y[x] == xCos[x], y[x], x]
```

$$\left\{\left\{y[x] \to \frac{C[1]}{x} + \frac{\text{Cos}[x] + x\text{Sin}[x]}{x}\right\}\right\}$$

As we have seen in previous examples, we can graph the solution for various values of the arbitrary constant by generating a set of functions obtained by replacing the arbitrary constant with numbers using `Table` and `ReplaceAll` (`/.`).

```
toplot = Table[gensol[[1, 1, 2]]/.C[1] → i, {i, −4, 4}];
```

In this case, Mathematica generates several error messages, which are not displayed here because the solution is undefined if $x = 0$. Nevertheless, the resulting graph shown in Figure 2-16(a) is displayed correctly.

```
q1 = Plot[toplot, {x, 0, 4Pi}, PlotRange → {−2Pi, 2Pi},
  AspectRatio → 1, AxesStyle → Black, AxesLabel → {x, y},
  PlotLabel → "(a)"];
```

To graph the direction field, we use `StreamPlot`. First, we write the equation in the form $dy/dx = f(x, y)$.

```
step2 = Solve[xy′[x] + y[x] == xCos[x], y′[x]]/.y[x] → y
```

$$\left\{\left\{y'[x] \to \frac{-y + x\mathrm{Cos}[x]}{x}\right\}\right\}$$

We then use `StreamPlot` to graph the direction field and display the result in Figure 2-16(b).

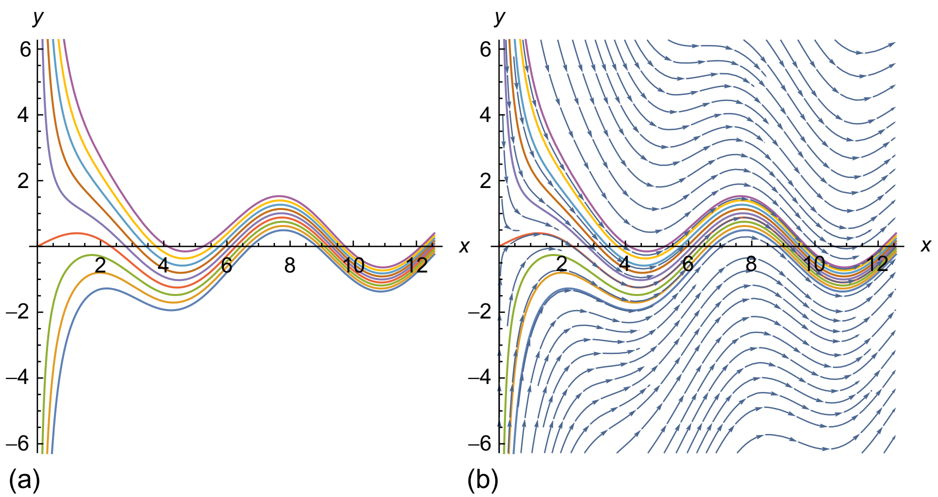


Figure 2-16 (a) Various solutions of $x\,dy/dx + y = x\cos x$, $x > 0$. (b) Various solutions with the direction field

```
q2a = StreamPlot[{1, step2[[1, 1, 2]]}, {x, 0, 4Pi}, {y, −2Pi, 2Pi},
  PlotRange → {−2Pi, 2Pi}, AspectRatio → 1,
  AxesStyle → Black, AxesLabel → {x, y}, PlotLabel → "(b)"];

q2 = Show[{q1, q2a}, PlotRange → {{0, 4Pi}, {−2Pi, 2Pi}}, PlotLabel
→ "(b)"];

Show[GraphicsRow[{q1, q2}]]
```

■

As with other types of equations, we solve initial-value problems by first finding a general solution of the equation and then applying the initial condition to determine the value of the constant.

EXAMPLE 2.5.2: Solve the initial-value problem $\begin{cases} dy/dx + 5x^4 y = x^4 \\ y(0) = -7 \end{cases}$.

SOLUTION: As we have seen in many previous examples, `DSolve` can be used to find a general solution of the equation and the solution to the initial-value problem, as done in `gensol` and `partsol`, respectively.

```
Clear[x, y, gensol]

gensol = DSolve[y'[x] + 5x^4y[x] == x^4, y[x], x]
```

$$\left\{\left\{y[x] \to \frac{1}{5} + e^{-x^5} C[1]\right\}\right\}$$

```
partsol = DSolve[{y'[x] + 5x^4y[x] == x^4, y[0] == −7}, y[x], x]
```

$$\left\{\left\{y[x] \to \frac{1}{5} e^{-x^5}\left(-36 + e^{x^5}\right)\right\}\right\}$$

We now graph various solutions to the differential equation with `Table` followed by `Plot` in Figure 2-17(a).

```
toplot1 = Table[gensol[[1, 1, 2]]/.C[1] → c, {c, −20, 20, 2}];

p1 = Plot[toplot1, {x, −1, 2}, PlotStyle → Gray,
```

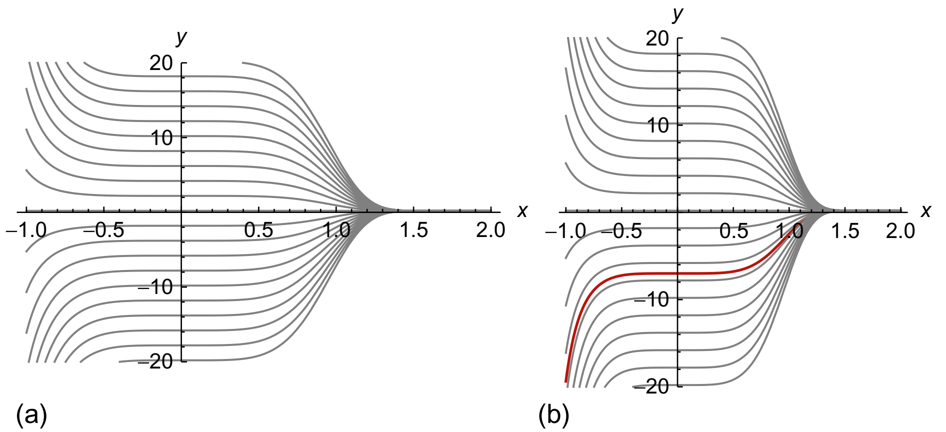


Figure 2-17 (a) Various solutions of $dy/dx+5x^4y=x^4$. (b) The solution of $dy/dx+5x^4y=x^4$ that satisfies $y(0)=-7$

```
    AxesStyle → Black, AxesLabel → {x, y}, PlotRange → {−20, 20},
    PlotLabel → "(a)", AspectRatio → 1];
```

Next we graph the solution to the initial-value problem obtained in `partsol` with `Plot` in Figure 2-17(b).

```
p2a = Plot[partsol[[1, 1, 2]], {x, −1, 2},
  PlotStyle → {{Thickness[.01], CMYKColor[0, 0.89, 0.94, 0.28]}},
  AxesStyle → Black, AxesLabel → {x, y}, PlotRange → {−20, 20},
  PlotLabel → "(b)", AspectRatio → 1];
p2 = Show[p2a, p1];

Show[GraphicsRow[{p1, p2}]]
```

We can also use Mathematica to carry out the steps necessary to solve first-order linear equations. We begin by identifying the integrating factor $e^{\int 5x^4\,dx}=e^{x^5}$, computed as follows with `Integrate`.

```
intfac = Exp[Integrate[5x^4, x]]
```

e^{x^5}

Therefore, the equation can be written as

$$\frac{d}{dx}\left(e^{x^5}y\right)=x^4e^{x^5}$$

so that integration of both sides of the equation yields

$$e^{x^5} y = \frac{1}{5} e^{x^5} + C.$$

```
rightside = Integrate[intfacx^4, x]
```

$$\frac{e^{x^5}}{5}$$

Hence, a general solution is $y = \frac{1}{5} + Ce^{-x^5}$. Note that we compute y by using `Solve` to solve the equation $e^{x^5} y = \frac{1}{5} e^{x^5} + C$ for y.

```
step1 = Solve[Exp[x^5]y == rightside + c, y]
```

$$\left\{\left\{y \to \frac{1}{5} e^{-x^5} \left(5c + e^{x^5}\right)\right\}\right\}$$

We find the unknown constant C by substituting the initial condition $y(0) = -7$ into the general solution and solving for C.

```
findc = Solve[-7==step1[[1, 1, 2]]/.x → 0]
```

$$\left\{\left\{c \to -\frac{36}{5}\right\}\right\}$$

Therefore, the solution to the initial-value problem is $y = \frac{1}{5} - \frac{36}{5} e^{-x^5}$.

```
step1[[1, 1, 2]]/.findc[[1]]
```

$$\frac{1}{5} e^{-x^5} \left(-36 + e^{x^5}\right)$$

■

We can use `DSolve` to solve a first-order linear equation even if the coefficient functions are discontinuous or piecewise-defined. In such situations, it is often useful to take advantage of the *unit step function*. The **unit step function**, $\mathcal{U}(t)$, is defined by

$$\mathcal{U}(t) = \begin{cases} 1, & t \geq 0 \\ 0, & t < 0 \end{cases}.$$

The Mathematica command `UnitStep[t]` returns $\mathcal{U}(t)$.

EXAMPLE 2.5.3 (Drug Concentration): If a drug is introduced into the bloodstream in dosages $D(t)$ and is removed at a rate proportional to the concentration, the concentration $C(t)$ at time t is given by

$$\begin{cases} dC/dt = D(t) - kC \\ C(0) = 0 \end{cases},$$

where $k > 0$ is the constant of proportionality.

Suppose that over a 24-hour period, a drug is introduced into the bloodstream at a rate of $24/t_0$ for exactly t_0 hours and then stopped so that $D_{t_0}(t) = \begin{cases} 24/t_0, \ 0 \le t \le t_0 \\ 0, \ t > t_0 \end{cases}$. Calculate and then graph $C(t)$ on the interval $[0, 30]$ if $k = 0.05, 0.10, 0.15, 0.20$, and 0.25 for $t_0 = 4, 8, 12, 16$, and 25. How does increasing t_0 affect the concentration of the drug in the bloodstream? Then consider the effect of increasing k.

See J. D. Murray's *Mathematical Biology*, Springer-Verlag, 1990, pp. 645–649.

SOLUTION: To compute $C(t)$, we must keep in mind that $D_{t_0}(t)$ is a piecewise defined function. In terms of the unit step function, $\mathcal{U}(t)$, $D_{t_0}(t) = \dfrac{24}{t_0}\mathcal{U}(t_0 - t)$

```
d[t_, t0_] = 24/t0UnitStep[t0 − t];
```

Note that we use lower-case letters to avoid any ambiguity with built-in objects like `C` and `D`.

For example, entering `d(t,4)` returns $D_4(t) = \begin{cases} 6, \ 0 \le t \le 4 \\ 0, \ t > 4 \end{cases}$.

```
d[t, 4]

6UnitStep[4 − t]
```

Given k and t_0, the function `sol` returns the solution to the initial-value problem $\begin{cases} dC/dt = D_{t_0}(t) - kC \\ C(0) = 0 \end{cases}$.

```
Clear[sol, k, c, t, t0]

sol[k_, t0_]:=DSolve[{c′[t] == d[t, t0] − kc[t], c[0] == 0}, c[t], t]
[[1, 1, 2]]
```

Then, for $k = 0.05$ we solve the initial-value problem $\begin{cases} dC/dt = D_{t_0}(t) - kC \\ C(0) = 0 \end{cases}$ for $t = 4, 8, 12, 16$, and 20 by applying `sol` to the list `{4,8,12,16,20}` using `Map`. These solutions are graphed with `Plot` in Figure 2-18.

`Map[f,list]` computes $f(x)$ for each element x of `list`.

```
toplot05 = Map[sol[0.05, #]&, {4, 8, 12, 16, 20}];

p1 = Plot[toplot05, {t, 0, 30}, PlotRange → {0, 30},
  AspectRatio → 1, AxesStyle → Black,
  AxesLabel → {t, c}]
```

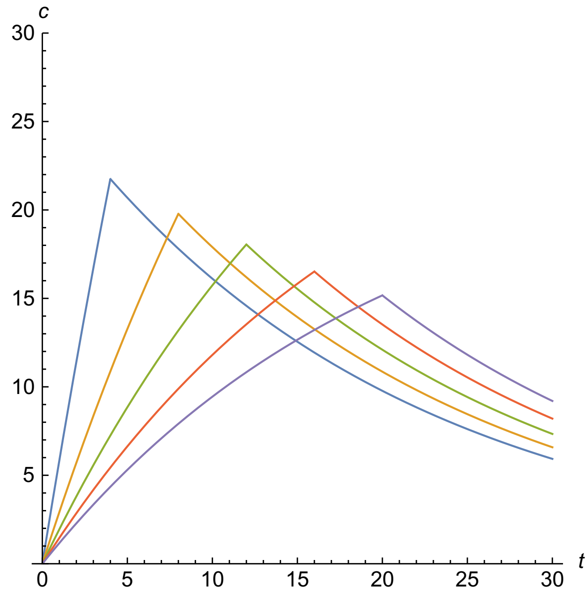


Figure 2-18 As t_0 increases, the maximum concentration of the drug decreases

Similar steps are repeated for $k = 0.10$, 0.15, 0.20, and 0.25 by defining the function `toplot`. Given k, `toplot[k]` solves the initial-value problem $\begin{cases} dC/dt = D_{t_0}(t) - kC \\ C(0) = 0 \end{cases}$ for $t_0 = 4$, 8, 12, 16, and 20.

```
Clear[toplot, sols]

toplot[k_]:=Map[sol[k, #]&, {4, 8, 12, 16, 20}];
```

We then apply `toplot` to the list `{0.1, 0.15, 0.20, 0.25}` naming the resulting list of lists of functions `sols`.

```
sols = Map[toplot, {0.1, 0.15, 0.2, 0.25}];
```

Each list of functions in `sols` is then graphed with `Plot` by applying the pure function

```
Plot[Evaluate[#], {t, 0, 30}, PlotRange -> {0, 30},
AspectRatio -> 1, AxesStyle -> Black,
AxesLabel->{t,c}] &
```

to each element of `sols` with `Map`.

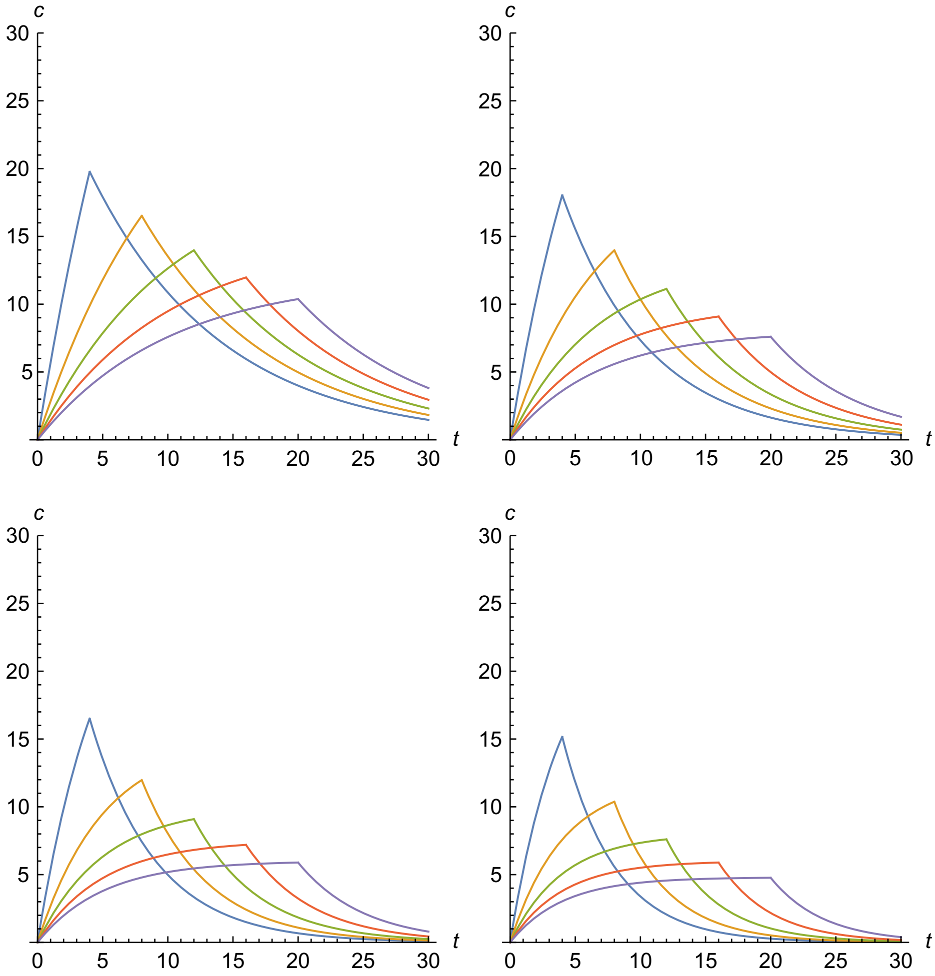


Figure 2-19 $C(t)$ for various values of t_0 and k

```
toshow = Map[Plot[#, {t, 0, 30}, PlotRange → {0, 30},
  AspectRatio → 1, AxesStyle → Black,
  AxesLabel → {t, c}]&, sols];
```

Finally, all four graphs are shown together as a graphics array using `Show` and `GraphicsGrid` in Figure 2-19.

```
Show[GraphicsGrid[Partition[toshow, 2]]]
```

From the graphs, we see that as t_0 is increased, the maximum concentration level decreases and occurs at later times, while increasing k increases the rate at which the drug is removed from the bloodstream.

Another way to see how t_0 and k affect the concentration of the drug is to use `Manipulate`. To avoid some computational errors, we use `NDSolve` rather than `DSolve` in the `Manipulate` function. In Figure 2-20, use the sliders to experiment with different t_0 and k values.

```
Manipulate[Plot[Evaluate[NDSolve[{c′[t] ==
  d[t, t0] − kc[t], c[0] == 0},
  c[t], {t, 0, 30}][[1, 1, 2]]], {t, 0, 30},
  PlotRange → {{0, 30}, {0, 30}},
```

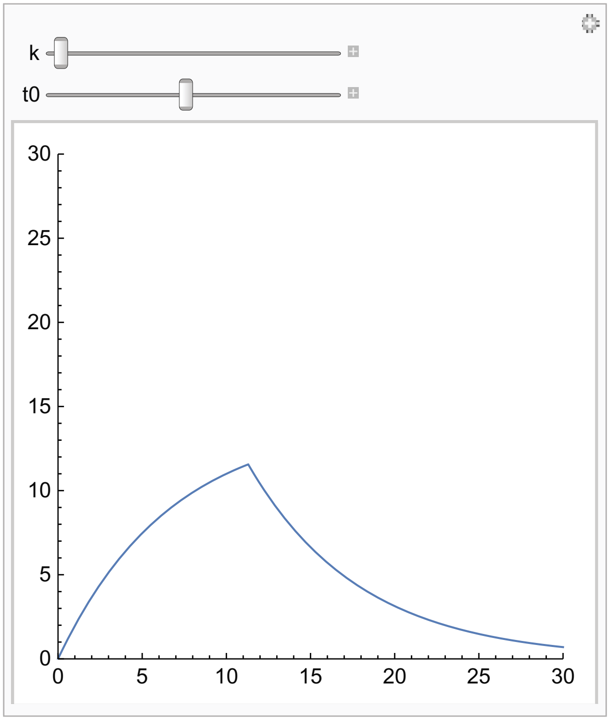


Figure 2-20 Use the sliders to change k and t_0 to see how the solution to the initial value problem changes. To see the specific values k and t_0, click on the + button to expand the sliders

```
AspectRatio → 1, AxesStyle → Black],
{{k, 1}, 0, 25}, {{t0, 12}, 0, 24}]
```

■

If the integration cannot be carried out, the solution can often by approximated numerically by taking advantage of numerical integration techniques. Generally,

```
NDSolve[{y'[t]==f[t,y[t]],y[t0]=y0},y[t],{t,tmin,tmax}]
```

attempts to numerically solve $dy/dt = f(t, y)$, $y(t_0) = y_0$ for $t_{\min} \leq t \leq t_{\max}$. Observe that the syntax for `NDSolve` is nearly identical to that of the `DSolve` command except that we must specify an interval on which the solution is to be accurate (Figure 2-21).

EXAMPLE 2.5.4: Graph the solution to the initial-value problem $y' - y\sin(2\pi x) = 1$, $y(0) = 1$ on the interval $[0, 2\pi]$.

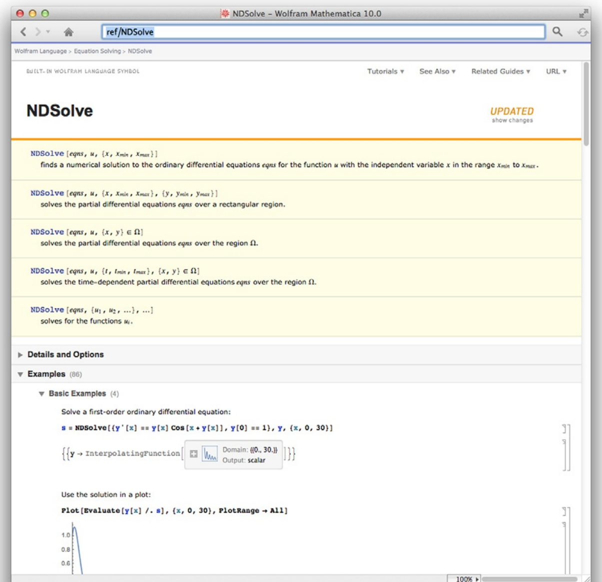


Figure 2-21 Mathematica's help for `NDSolve`

SOLUTION: Note that `DSolve` is successful in finding the solution to the initial-value problem even though the result contains unevaluated integrals.

```
Clear[x, y, partsol]

partsol = DSolve[{y'[x] − Sin[2Pix]y[x] == 1,
  y[0] == 1}, y[x], x]
```

$$\left\{\left\{y[x] \to e^{-\frac{\mathrm{Cos}[2\pi x]}{2\pi}}\left(e^{\frac{1}{2}/\pi} + \mathrm{BesselI}\left[0, \frac{1}{2\pi}\right] + \int_1^x e^{\frac{\mathrm{Cos}[2\pi K[1]]}{2\pi}} dK[1]\right)\right\}\right\}$$

We can evaluate the result for particular numbers. For example, entering

```
partsol[[1, 1, 2]]/.x → 1
```

$$e^{-\frac{1}{2}/\pi}\left(e^{\frac{1}{2}/\pi} + \mathrm{BesselI}\left[0, \frac{1}{2\pi}\right]\right)$$

returns the value of the solution to the initial-value problem if $x = 1$. This result is a bit complicated to understand so we use `N` to obtain a numerical approximation.

```
N[%]

1.85827
```

Before graphing the solution to the initial value problem, we use `StreamPlot` to graph the direction field for the equation in Figure 2-22(a).

```
p1 = StreamPlot[{1, 1 + Sin[2Pix]y}, {x, 0, 8}, {y, 0, 8},
  Axes → Automatic, Frame → False, AxesStyle → Black,
  AxesLabel → {x, y}, PlotLabel → "(a)"];
```

To graph the solution on the interval $[0, 2\pi]$, we use `NDSolve` to generate a numerical solution to the initial-value problem valid for $0 \leq x \leq 2\pi$. As stated previously, the command

```
NDSolve[{deq,ics},fun,{var,varmin,varmax}]
```

returns a numerical solution `fun` (which is a function of the variable `var`) of the differential equation `deq` that satisfies the initial conditions `ics` valid on the interval `[varmin,varmax]`. In some cases, the

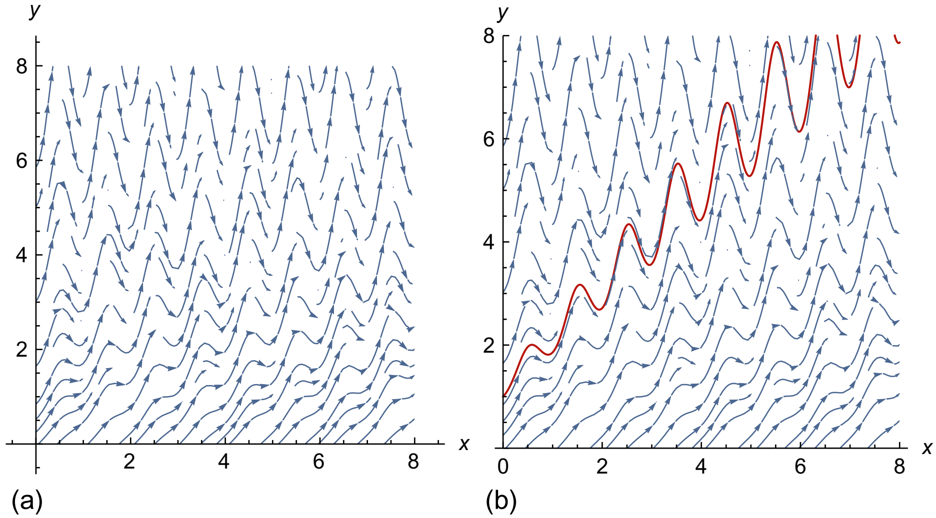


Figure 2-22 (a) Plot of the direction field. (b) Plot of a numerical solution to an initial value problem together with the direction field

interval on which the solution returned by `NDSolve` is smaller than the interval requested.

We see that the syntax for the `NDSolve` command is nearly the same as the syntax of the `DSolve` command although we must specify an interval on which we want the approximation to be valid. In this case, including $\{x, 0, 2\pi\}$ in the `NDSolve` command instructs Mathematica to (try to) make the resulting numerical solution valid for $0 \le x \le 2\pi$.

Note that the number of initial conditions in `ics` must equal the order of the differential equation `deq`.

```
numsol = NDSolve[{y'[x] - Sin[2Pix]y[x] == 1, y[0] == 1}, y[x],
{x, 0, 8}]
```

```
{{y[x] → InterpolatingFunction[][x]}}
```

The resulting output is an `InterpolatingFunction` which represents an approximate function obtained through interpolation. We can evaluate the result for particular values of x as long as $0 \le x \le 2\pi$. For example, entering

```
numsol/.x → 1
```

```
{{y[1] → 1.85827}}
```

approximates the value of the solution to the initial-value problem if $x = 1$. In this case, the result means that $y(1) \approx 1.85828$. We can graph the result returned by `NDSolve` in the same way as we graph results returned by `DSolve`: entering

```
p2a = Plot[numsol[[1, 1, 2]], {x, 0, 8},
  PlotStyle→{{Thickness[.01], CMYKColor[0, 0.89, 0.94,
  0.28]}}, AxesStyle → Black, AxesLabel → {x, y}, PlotRange
  → {0, 8}, PlotLabel → "(b)", AspectRatio → 1];
p2 = Show[p2a, p1];

Show[GraphicsRow[{p1, p2}]]
```

graphs the solution to the initial-value problem on the interval $[0, 2\pi]$ as shown in Figure 2-22(b). Note that we obtain the same graph by entering `Plot[y[x] /. numsol,{x,0,2Pi}]`.

■

2.5.2 Variation of Parameters and the Method of Undetermined Coefficients

Observe that equation (2.4) is separable:

$$\begin{aligned}\frac{dy}{dx} + p(x)y &= 0\\ \frac{1}{y}dy &= -p(x)\,dx\\ \ln|y| &= -\int p(x)\,dx + C\\ y &= Ce^{-\int p(x)\,dx}.\end{aligned}$$

A **particular solution** is a specific function that is a solution to the equation that does not contain any arbitrary constants.

Notice that any constant multiple of a solution to a linear homogeneous equation is also a solution. Now suppose that y is any solution of equation (2.3) and y_p is a particular solution of equation (2.3). Then,

$$\begin{aligned}\left(y - y_p\right)' + p(x)\left(y - y_p\right) &= y' + p(x)y - \left({y_p}' + p(x)y_p\right)\\ &= q(x) - q(x) = 0.\end{aligned}$$

Thus, $y - y_p$ is a solution to the corresponding homogeneous equation of equation (2.3). Hence,

$$\begin{aligned}y - y_p &= Ce^{-\int p(x)\,dx}\\ y &= Ce^{-\int p(x)\,dx} + y_p\\ y &= y_h + y_p,\end{aligned}$$

where $y_h = Ce^{-\int p(x)\,dx}$. That is, a general solution of equation (2.3) is

$$y = y_h + y_p,$$

where y_p is a particular solution to the nonhomogeneous equation and y_h is a general solution to the corresponding homogeneous equation. Thus, to solve equation (2.3), we need to first find a general solution to the corresponding homogeneous equation, y_h, which we can accomplish through separation of variables, and then find a particular solution, y_p, to the nonhomogeneous equation.

If y_h is a solution to the corresponding homogeneous equation of equation (2.3) then for any constant C, Cy_h is also a solution to the corresponding homogeneous equation. Hence, it is impossible to find a particular solution to equation (2.3) of this form. Instead, we search for a particular solution of the form $y_p = u(x)y_h$, where $u(x)$ is *not* a constant function. Assuming that a particular solution, y_p, to equation (2.3) has the form $y_p = u(x)y_h$, differentiating gives us

$$y_p{}' = u' y_h + u y_h{}'$$

and substituting into equation (2.3) results in

$$y_p{}' + p(x)y_p = u' y_h + u y_h{}' + p(x)u y_h = q(x).$$

Because $u y_h{}' + p(x)u y_h = u\left[y_h{}' + p(x)y_h\right] = u \cdot 0 = 0$, we obtain

y_h is a solution to the corresponding homogeneous equation so $y_h{}' + p(x)y_h = 0$.

$$\begin{aligned} u' y_h &= q(x) \\ u' &= \frac{1}{y_h} q(x) \\ u' &= e^{\int p(x)\,dx} q(x) \\ u &= \int e^{\int p(x)\,dx} q(x)\,dt \end{aligned}$$

so

$$y_p = u(x)\,y_h = Ce^{-\int p(x)\,dx} \int e^{\int p(x)\,dx} q(x)\,dx.$$

Because we include an arbitrary constant of integration when evaluating $\int e^{\int p(x)\,dx} q(x)\,dx$, it follows that we can write a general solution of equation (2.3) as

$$y = e^{-\int p(x)\,dx} \int e^{\int p(x)\,dx} q(t)\,dt. \tag{2.6}$$

Exponential growth is discussed in more detail in Section 3.2.1.

EXAMPLE 2.5.5 (Exponential Growth): Let $y = y(t)$ denote the size of a population at time t. If y grows at a rate proportional to the amount present, y satisfies

$$\frac{dy}{dt} = \alpha y, \tag{2.7}$$

where α is the **growth constant**. If $y(0) = y_0$, using equation (2.6) results in $y = y_0 e^{\alpha t}$. We use `DSolve` to confirm this result.

```
Clear[t, y]
DSolve[{y'[t] == αy[t], y[0] == y0}, y[t], t]
```

$\{\{y[t] \to e^{t\alpha} y0\}\}$

$dy/dt = k(y - y_s)$ models *Newton's Law of Cooling*: the rate at which the temperature, $y(t)$, changes in a heating/cooling body is proportional to the difference between the temperature of the body and the constant temperature, y_s, of the surroundings. Newton's Law of Cooling is discussed in more detail in Section 3.3.

EXAMPLE 2.5.6: Solve each of the following equations: (a) $dy/dt = k(y - y_s)$, $y(0) = y_0$, k and y_s constant and (b) $y' - 2ty = t$.

SOLUTION: (a) By hand, we rewrite the equation and obtain

$$\frac{dy}{dt} - ky = -ky_s.$$

A general solution of the corresponding homogeneous equation

$$\frac{dy}{dt} - ky = 0$$

is $y_h = e^{kt}$. Because k and $-ky_s$ are constants, we suppose that a particular solution of the nonhomogeneous equation, y_p, has the form $y_p = A$, where A is a constant.

This will turn out to be a lucky guess. If there is not a solution of this form, we would not find one of this form.

Assuming that $y_p = A$, we have $y_p' = 0$ and substitution into the nonhomogeneous equation gives us

$$\frac{dy_p}{dt} - ky_p = -KA = -ky_s \quad \text{so} \quad A = y_s.$$

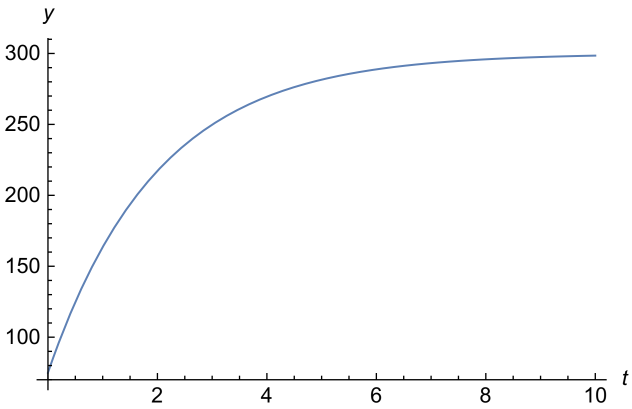


Figure 2-23 The temperature of the body approaches the temperature of its surroundings

Thus, a general solution is $y = y_h + y_p = Ce^{kt} + y_s$. Applying the initial condition $y(0) = y_0$ results in $y = y_s + (y_0 - y_s)e^{kt}$.

We obtain the same result with `DSolve`. We graph the solution satisfying $y(0) = 75$ assuming that $k = -1/2$ and $y_s = 300$ in Figure 2-23. Notice that $y(t) \to y_s$ as $t \to \infty$.

```
sola = DSolve[{y'[t] == k(y[t] - ys),
  y[0] == y0}, y[t], t]
{{y[t] → e^(kt) y0 + ys − e^(kt) ys}}

tp = sola[[1, 1, 2]]/.{k → −1/2,
  ys → 300, y0 → 75};
p1 = Plot[tp, {t, 0, 10}, AxesLabel → {t, y},
  AxesStyle → Black]
```

(b) The equation is in standard form and we identify $p(t) = -2t$. Then, the integrating factor is $\mu(t) = e^{\int p(t)\,dt} = e^{-t^2}$. Multiplying the equation by the integrating factor, $\mu(t)$, results in

$$e^{-t^2}(y' - 2ty) = te^{-t^2} \quad \text{or} \quad \frac{d}{dt}\left(ye^{-t^2}\right) = te^{-t^2}.$$

Integrating gives us

$$ye^{-t^2} = -\frac{1}{2}e^{-t^2} + C \quad \text{or} \quad y = -\frac{1}{2} + Ce^{t^2}.$$

We confirm the result with `DSolve`.

```
solb = DSolve[y'[t] - 2ty[t] == t, y[t], t]
```

$$\left\{\left\{y[t] \to -\frac{1}{2} + e^{t^2} C[1]\right\}\right\}$$

■

Application: Antibiotic Production

Source: Kevin H. Dykstra and Henry Y. Wang, "Changes in the Protein Profile of *Streptomyces Griseus* during a Cycloheximide Fermentation," *Biochemical Engineering V*, Annals of the New York Academy of Sciences, Volume 56, New York Academy of Sciences (1987), pp. 511–522.

When you are injured or sick, your doctor may prescribe antibiotics to prevent or cure infections. In the journal article "Changes in the Protein Profile of *Streptomyces Griseus* during a Cycloheximide Fermentation" we see that production of the antibiotic cycloheximide by Streptomyces is typical of antibiotic production. During the production of cycloheximide, the mass of *Streptomyces* grows relatively quickly and produces little cycloheximide. After approximately 24 hours, the mass of *Streptomyces* remains relatively constant and cycloheximide accumulates. However, once the level of cycloheximide reaches a certain level, extracellular cycloheximide is degraded (**feedback inhibited**). One approach to alleviating this problem to maximize cycloheximide production is to continuously remove extracellular cycloheximide. The rate of growth of *Streptomyces* can be described by the separable equation

$$\frac{dX}{dt} = \mu_{\max}\left(1 - \frac{1}{X_{\max}}X\right)X,$$

where X represents the mass concentration in g/L, $\mu_{\max}$ is the maximum specific growth rate, and $X_{\max}$ represents the maximum mass concentration. We now solve the initial-value problem $\begin{cases} dX/dt = \mu_{\max}\left(1 - \frac{1}{X_{\max}}X\right)X \\ X(0) = 1 \end{cases}$ with `DSolve`, naming the result `sol1`.

Note that this equation can be converted to a linear equation with the substitution $y = X^{-1}$.

```
Clear[x]

sol1 = DSolve[{x'[t] == μ(1 - x[t]/xmax)x[t],
  x[0] == 1}, x[t], t]
```

$$\left\{\left\{x[t] \to \frac{e^{t\mu}\text{xmax}}{-1 + e^{t\mu} + \text{xmax}}\right\}\right\}$$

Experimental results have shown that $\mu_{\max} = 0.3\,\text{hr}^{-1}$ and $X_{\max} = 10\,\text{g/L}$. For these values, we use `Plot` to graph $X(t)$ on the interval $[0, 24]$ in Figure 2-24(a). Then, we use `Table` and `TableForm` to determine the mass concentration at the end of 4, 8, 12, 16, 20, and 24 hours.

```
μ = 0.3; xmax = 10.;

p1 = Plot[x[t]/.sol1, {t, 0, 24}, PlotStyle → Black,
  AxesStyle → Black, AxesLabel → {t, x},
  PlotLabel → "(a)"];

TableForm[Table[{t, sol1[[1, 1, 2]]}, {t, 4, 24, 4.}]]
```

4.	2.69487
8.	5.50521
12.	8.02624
16.	9.3104
20.	9.78178
24.	9.93326

The rate of accumulation of cycloheximide is the difference between the rate of synthesis and the rate of degradation:

$$\frac{dP}{dt} = R_s - R_d.$$

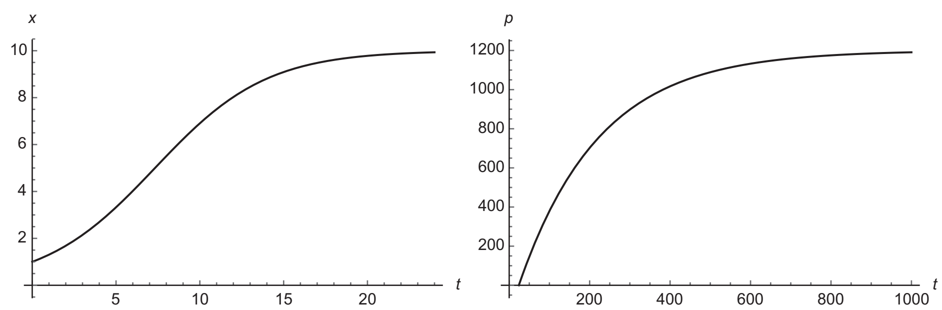


Figure 2-24 (a) Plot of the mass concentration, $x(t)$. (b) Accumulation of the antibiotic

It is known that $R_d = K_d P$, where $K_d = 5\times10^{-3}\,\mathrm{h}^{-1}$, so $dP/dt = R_s - R_d$ is equivalent to $dP/dt = R_s - K_d P$. Furthermore,

$$R_s = Q_{po} EX \left(1 + \frac{P}{K_l}\right)^{-1},$$

where Q_{po} represents the specific enzyme activity with value $Q_{po} \approx 0.6\,\mathrm{g\,CH/g}$, protein $\cdot$ h and K_l represents the inhibition constant. E represents the intracellular concentration of an enzyme which we will assume is constant. For large values of K_l and t, $X(t) \approx 10$ and $(1 + P/K_l)^{-1} \approx 1$. Thus, $R_s \approx 10 Q_{po} E$ so

$$\frac{dP}{dt} = 10 Q_{po} E - K_d P.$$

After defining K_d and Q_{po}, we solve the initial-value problem $\begin{cases} dP/dt = 10Q_{po}E - K_d P \\ p(24) = 0 \end{cases}$ and then graph $\frac{1}{E} P(t)$ on the interval $[0, 24]$ in Figure 2-24(b).

```
Clear[p]

kd = 5/1000;

qp0 = 0.6;

sol2 = DSolve[{p'[t] == 10qp0 − kdp[t],
  p[24] == 0}, p[t], t]

{{p[t] → 1200.e^(−0.005t) (−1.1275 + 1.e^(0.005t))}}

p2 = Plot[p[t]/.sol2, {t, 24, 1000}, PlotStyle → Black,
  AxesStyle → Black, AxesLabel → {t, p},
  AxesOrigin → {0, 0},
  PlotLabel → "(b)"];

Show[GraphicsRow[{p1, p2}]]
```

From the graph, we see that the total accumulation of the antibiotic approaches a limiting value, which in this case is 1200.

2.6 Numerical Approximations of Solutions to First-Order Equations

2.6.1 Built-In Methods

Numerical approximations of solutions to differential equations can be obtained with `NDSolve`, which is particularly useful when working with nonlinear equations for which `DSolve` alone is unable to find an explicit or implicit solution. The command

```
NDSolve[{y'[t]==f[t,y[t]],y[t0]==y0},y[t],{t,a,b}]
```

attempts to generate a numerical solution of

$$\begin{cases} \dfrac{dy}{dt} = f(t, y) \\ y(t_0) = y_0 \end{cases}$$

valid for $a \le t \le b$. In some cases, the interval on which the solution returned by `NDSolve` is smaller than the interval requested. You can obtain basic information regarding `NDSolve` by entering `?NDSolve` or detailed information by accessing Mathematica's on-line help facility by selecting **Help** from the Mathematica help menu.

EXAMPLE 2.6.1: Consider

$$\frac{dy}{dt} = \left(t^2 - y^2\right) \sin y, \; y(0) = -1.$$

(a) Determine $y(1)$. (b) Graph $y(t)$ for $-1 \le t \le 10$.

SOLUTION: We first use `StreamPlot` to plot the direction field for $y' = \left(t^2 - y^2\right) \sin y$ in Figure 2-25(a).

```
f[t_, y_] = (t^2 - y^2)Sin[y];
p1 = StreamPlot[{1, f[t, y]}, {t, 0, 10}, {y, 0, 10},
  AspectRatio → 1, StreamStyle → Black, AxesOrigin → {0, 0},
```

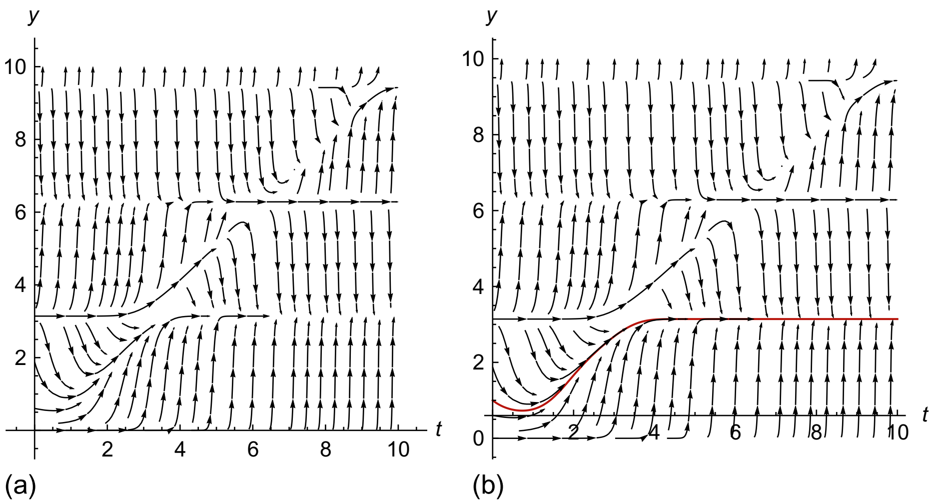


Figure 2-25 (a) The direction field for the equation. (b) Graph of the solution to $y' = \left(t^2 - y^2\right)\sin y$, $y(0) = -1$

```
Frame → False, Axes → Automatic, AxesStyle → Black,
PlotLabel → "(a)", AxesLabel → {t, y}];
```

We obtain a numerical solution valid for $0 \leq t \leq 1000$ using the `NDSolve` function.

```
sole = NDSolve[{y'[t] == f[t, y[t]], y[0] == 1}, y[t], {t, 0, 1000}]
```

```
{{y[t] → InterpolatingFunction[][t]}}
```

Entering `sole /.t->1` evaluates the numerical solution if $t = 1$.

```
sole/.x → 1
```

```
{{y[1] → −0.766}}
```

The result means that $y(1) \approx -.766$. We use the `Plot` command to graph the solution for $0 \leq t \leq 10$ in Figure 2-25.

```
p2a = Plot[y[t]/.sole, {t, 0, 10}, PlotStyle →
  {{Thickness[.01], CMYKColor[0, 0.89, 0.94, 0.28]}}];
p2 = Show[p2a, p1, AxesStyle → Black, AxesLabel → {t, y},
PlotRange → {{0, 10}, {0, 10}}, PlotLabel → "(b)",
AspectRatio → 1];
Show[GraphicsRow[{p1, p2}]]
```

■

EXAMPLE 2.6.2: Graph the solution to the initial-value problem

$$\begin{cases} dy/dx = \sin(2x - y) \\ y(0) = 0.5 \end{cases},$$

on the interval $[0, 15]$. What is the value of $y(1)$?

SOLUTION: As with the previous examples, we use `StreamPlot` to generate a plot of the direction field shown in Figure 2-26(a).

```
Clear[x, y]

f[x_, y_] = Sin[2x − y];

p1 = StreamPlot[{1, f[x, y]}, {x, 0, 10}, {y, 0, 10},
  AspectRatio → 1, StreamStyle → Black, AxesOrigin → {0, 0},
  Frame → False, Axes → Automatic, AxesStyle → Black,
  PlotLabel → "(a)", AxesLabel → {x, y}];
```

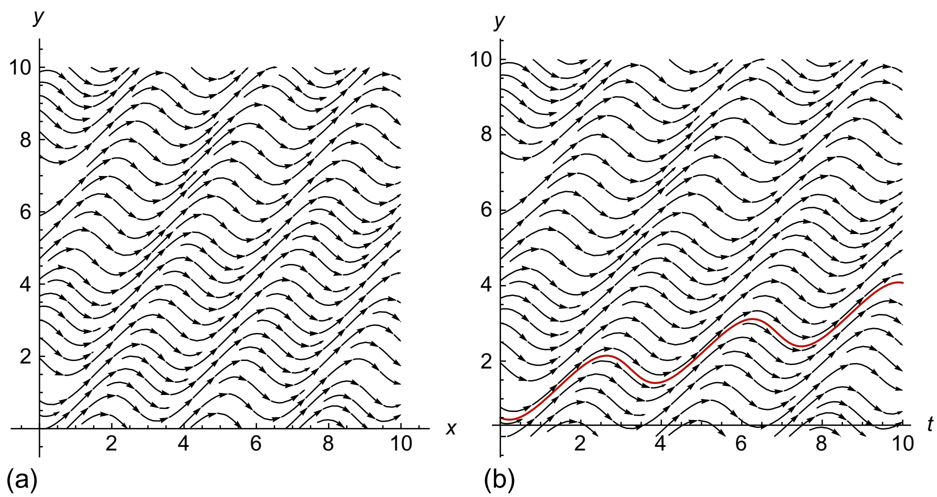


Figure 2-26 (a) Plot of the direction field. (b) Graph of the solution to $y' = \sin(2x - y)$, $y(0) = 0.5$

We use `NDSolve` to approximate the solution to the initial-value problem, naming the resulting output `numsol`. The resulting `InterpolatingFunction` is a procedure that represents an approximate function obtained through interpolation.

```
numsol = NDSolve[{y'[x] == f[x, y[x]], y[0] == .5}, y[x],
{x, 0, 15}]
{{y[x] → InterpolatingFunction[][x]}}
numsol/.x → 1
{{y[1] → 0.875895}}
```

returns a list corresponding to the value of $y(x)$ if $x = 1$. We interpret the result to mean that $y(1) \approx 0.875895$. We then graph the solution returned by `NDSolve` using `Plot` in the same way that we graph solutions returned by `DSolve`. As you probably expect, entering `Plot[numsol[[1,1,2]],{x,0,15}]` produces the same graph as the one shown in Figure 2-26(b) generated by the following `Plot` command.

```
p2a = Plot[y[x]/.numsol, {x, 0, 10},
  PlotStyle → {{Thickness[.01], CMYKColor[0, 0.89, 0.94,
  0.28]}}];
p2 = Show[p2a, p1, AxesStyle → Black, AxesLabel → {x, y},
  PlotRange → {{0, 10}, {0, 10}}, PlotLabel → "(b)",
  AspectRatio → 1];

Show[GraphicsRow[{p1, p2}]]
```

A different way to graph solutions that satisfy different initial conditions is to define a function as we do here. Given i, `sol[i]` returns a numerical solution to the initial-value problem $y' = \sin(2x-y)$, $y(0) = i$.

```
Clear[x, y, i, sol]
sol[i_]:=NDSolve[{y'[x] == Sin[2x − y[x]], y[0] == i},
  y[x], {x, 0, 15}]
```

For example, to use `sol`, we first use `Table` to define `inits` to be the list of numbers $i/2$ for $i = 1, 2, \ldots, 5$ and then use `Map` to apply `sol` to the list of numbers `inits`. The command

```
interpfunctions=Map[sol,inits]
```

computes `sol[i]` for each value of i in `inits`. The result is a nested list consisting of `InterpolatingFunction`'s.

We graph the set of `InterpolatingFunction`'s with `Plot` in the same way as we graph other sets of functions. See Figure 2-27(b).

```
inits = Table[i/2, {i, 1, 10}];

interpfunctions = Map[sol, inits];
```

Last, we show these graphs together with the direction field associated with the equation in Figure 2-27(b).

```
p3 = Plot[Evaluate[y[x]/.interpfunctions],
  {x, 0, 10}, PlotRange → {{0, 10}, {0, 10}},
  AspectRatio → 1,
  PlotStyle → {{CMYKColor[0, 0.89, 0.94, 0.28]}},
  AxesLabel → {x, y}, PlotLabel → "(a)"];

p4 = Show[p3, p1, PlotLabel → "(b)"];

Show[GraphicsRow[{p3, p4}]]
```

■

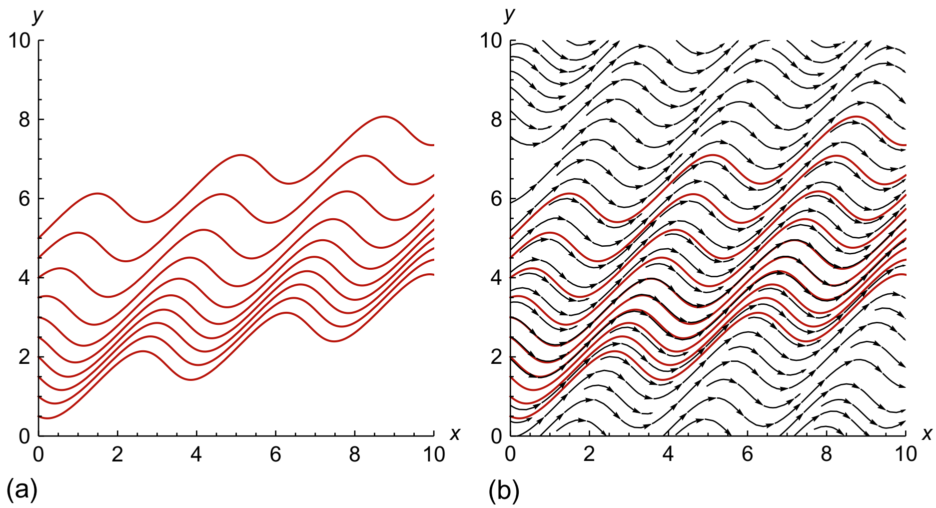


Figure 2-27 Various solutions of $y' = \sin(2x - y)$

Application: Modeling the Spread of a Disease

Source: Herbert W. Hethcote, "Three Basic Epidemiological Models," in *Applied Mathematical Ecology*, edited by Simon A. Levin, Thomas G. Hallan, and Louis J. Gross, New York, Springer-Verlag (1989), pp. 119–143.

Suppose that a disease is spreading among a population of size N. In some diseases, like chickenpox, once an individual has had the disease, the individual becomes immune to the disease. In other diseases, like most venereal diseases, once an individual has had the disease and recovers from the disease, the individual does not become immune to the disease; subsequent encounters can lead to recurrences of the infection.

Let $S(t)$ denote the percent of the population susceptible to a disease at time t, $I(t)$ the percent of the population infected with the disease, and $R(t)$ the percent of the population unable to contract the disease. For example, $R(t)$ could represent the percent of persons who have had a particular disease, recovered, and have subsequently become immune to the disease. In order to model the spread of various diseases, we begin by making several assumptions and introducing some notation.

1. Susceptible and infected individuals die at a rate proportional to the number of susceptible and infected individuals with proportionality constant μ called the **daily death removal rate**; the number $1/\mu$ is the **average lifetime** or **life expectancy**.
2. The constant λ represents the **daily contact rate**: on average, an infected person will spread the disease to λ people per day.
3. Individuals recover from the disease at a rate proportional to the number infected with the disease with proportionality constant γ. The constant γ is called the **daily recovery removal rate**; the **average period of infectivity** is $1/\gamma$.
4. The **contact number** $\sigma = \lambda/(\gamma + \mu)$ represents the average number of contacts an infected person has with both susceptible and infected persons.

If a person becomes susceptible to a disease after recovering from it (like gonorrhea, meningitis, and streptococcal sore throat), then the percent of persons susceptible to becoming infected with the disease, $S(t)$, and the percent of people in the population infected with the disease, $I(t)$, can be modeled by the system of differential equations

$$\begin{cases} \dfrac{dS}{dt} = -\lambda IS + \gamma I + \mu - \mu S \\ \dfrac{dI}{dt} = \lambda IS - \gamma I - \mu I \\ S(0) = S_0,\ I(0) = I_0,\ S(t) + I(t) = 1 \end{cases} . \tag{2.8}$$

This model is called an **SIS model** (susceptible-infected-susceptible model) because once an individual has recovered from the disease, the individual again becomes susceptible to the disease.

We can write $dI/dt = \lambda IS - \gamma I - \mu I$ as $dI/dt = \lambda I(1-I) - \gamma I - \mu I$ because $S(t) = 1 - I(t)$. Therefore, we need to solve the initial-value problem

$$\begin{cases} \dfrac{dI}{dt} = [\lambda - (\gamma + \mu)]I - \lambda I^2 \\ I(0) = I_0 \end{cases}. \tag{2.9}$$

In the following, we use `i` to represent I, thus avoiding conflict with the built-in constant `I` $= \sqrt{-1}$. After defining `eq`, we use `DSolve` to find the solution to the initial-value problem.

```
eq = i'[t] + (γ + μ − λ)i[t] == −λi[t]^2;

sol = DSolve[{eq, i[0] == i0}, i[t], t]//FullSimplify
```

$$\left\{\left\{i[t] \to \frac{e^{\frac{(i\pi + t\lambda)(\gamma+\mu)}{\gamma-\lambda+\mu}}\, \mathrm{i0}^{\frac{\gamma+\mu}{\gamma-\lambda+\mu}}(-\gamma+\lambda-\mu)}{e^{\frac{(i\pi + t\lambda)(\gamma+\mu)}{\gamma-\lambda+\mu}}\, \mathrm{i0}^{\frac{\gamma+\mu}{\gamma-\lambda+\mu}}\lambda - e^{t\gamma+t\mu+\frac{t\lambda^2}{\gamma-\lambda+\mu}+\frac{\lambda(i\pi + \mathrm{Log}[\mathrm{i0}] - \mathrm{Log}[-\gamma+\lambda-\mathrm{i0}\lambda-\mu])}{\gamma-\lambda+\mu}}(-\gamma+\lambda-\mathrm{i0}\lambda-\mu)^{\frac{\gamma+\mu}{\gamma-\lambda+\mu}}}\right\}\right\}$$

We can use this result to see how a disease might spread through a population. For example, we compute the solution to the initial-value problem, which is extracted from `sol` with `sol[[1,1,2]]`, if $\lambda = 0.50$, $\gamma = 0.75$, and $\mu = 0.65$. In this case, we see that the contact number is $\sigma = \lambda/(\gamma + \mu) \approx 0.357143$.

```
λ = 0.5;

γ = 0.75;

μ = 0.65;
σ = λ/(γ + μ)
sol[[1, 1, 2]]
```

0.357143

$$-\frac{0.9e^{1.55556(i\pi+0.5t)}\mathrm{i0}^{1.55556}}{-e^{1.67778t+0.555556(i\pi-\mathrm{Log}[-0.9-0.5\mathrm{i0}]+\mathrm{Log}[\mathrm{i0}])}(-0.9-0.5\mathrm{i0})^{1.55556}+0.5e^{1.55556(i\pi+0.5t)}\mathrm{i0}^{1.55556}}$$

Next, we use `Table` to substitute various initial conditions into `sol[[1,1,2]]`, naming the resulting set of nine functions `toplot1`. We then graph the functions in `toplot1` for $0 \le t \le 5$ in Figure 2-28(a). Apparently, regardless of the initial percent of the population infected, under these conditions, the disease is eventually removed from the population. This makes sense because the contact number is less than one.

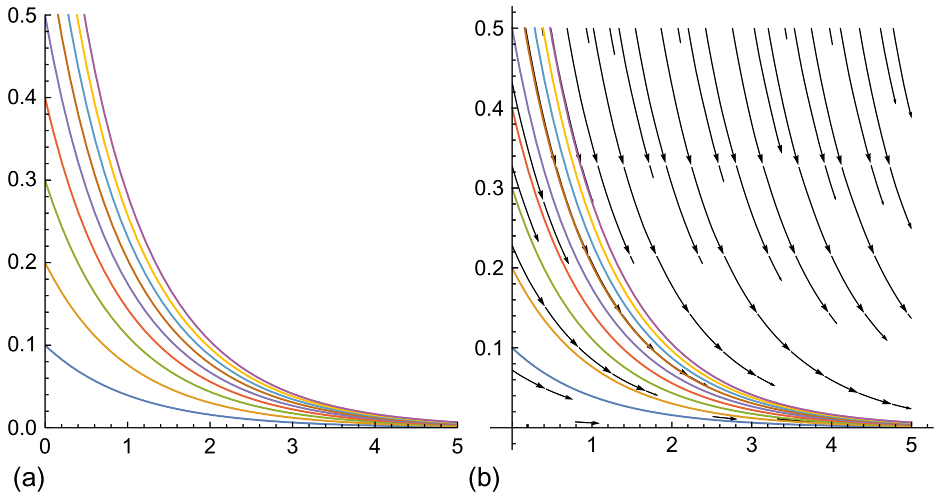


Figure 2-28 (a) The disease is removed from the population. (b) Illustrating the removal of the disease using the direction field

```
toplot1 = Table[sol[[1, 1, 2]], {i0, 0.1, 0.9, 0.1}];

p1 = Plot[Evaluate[toplot1], {t, 0, 5}, AspectRatio → 1,

  PlotLabel → "(a)", PlotRange → {{0, 5}, {0, 0.5}}];
```

After writing the equation in the form $dI/dt = f(t, I)$ in `step1`,

```
step1 = Solve[eq, i'[t]]

{{i'[t] → −0.9i[t] − 0.5i[t]^2}}

toplot2 = step1[[1, 1, 2]]/.i[t] → i

−0.9i − 0.5i^2
```

we use `Streamplot` to graph the direction field shown in Figure 2-28(b).

```
p2a = StreamPlot[{1, toplot2}, {t, 0, 5}, {i, 0, .5},

  Frame → False, Axes → Automatic, AxesStyle → Black,

  StreamStyle → Black, StreamPoints → Fine,

  PlotLabel → "(b)", AspectRatio → 1];
```

```
p2 = Show[p2a, p1, PlotRange → {{0, 5}, {0, .5}}];

Show[GraphicsRow[{p1, p2}]]
```

On the other hand, if $\lambda = 1.5$, $\gamma = 0.75$, and $\mu = 0.65$, we see that the contact number is $\sigma = \lambda/(\gamma + \mu) \approx 1.07143$.

```
λ = 1.5;

γ = 0.75;

μ = 0.65;

σ = λ/(γ + μ)
sol[[1, 1, 2]]

1.07143
```

$$\frac{0.1e^{-14.(i\pi+1.5t)}}{\left(-\frac{e^{-21.1t-15.(i\pi-\mathrm{Log}[0.1-1.5\mathrm{i0}]+\mathrm{Log}[\mathrm{i0}])}}{(0.1-1.5\mathrm{i0})^{14.}}+\frac{1.5e^{-14.(i\pi+1.5t)}}{\mathrm{i0}^{14.}}\right)\mathrm{i0}^{14.}}$$

Proceeding as before, we graph the solution using different initial conditions in Figure 2-29. In this case, we see that no matter what percent of the population is initially infected, a certain percent of the population is always infected. This makes sense because the contact number is greater than one. In fact, it is a theorem that

$$\lim_{t\to\infty} I(t) = \begin{cases} 1 - 1/\sigma, & \text{if } \sigma > 1 \\ 0, & \text{if } \sigma \le 1 \end{cases}.$$

```
toplot2 = Table[sol[[1, 1, 2]], {i0, 0.1, 0.9, 0.1}];
p1 = Plot[toplot2, {t, 0, 5}];
```

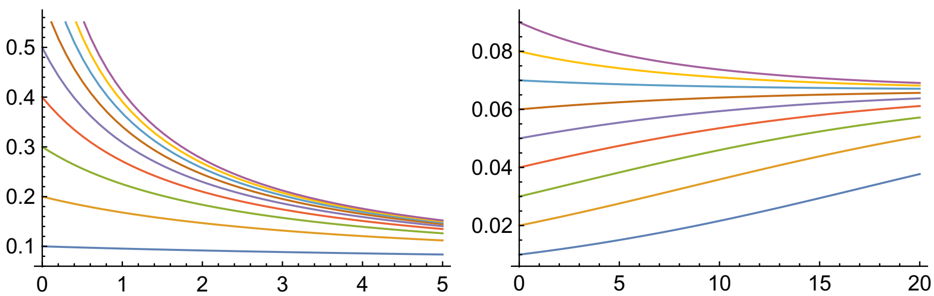


Figure 2-29 The disease persists

```
toplot3 = Table[sol[[1, 1, 2]], {i0, 0.01, 0.09, 0.01}];

p2 = Plot[toplot3, {t, 0, 20}];

Show[GraphicsRow[{p1, p2}]]
```

The incidence of some diseases, such as measles, rubella, and gonorrhea, oscillate seasonally. To model these diseases, we may wish to replace the constant contact rate λ, by a periodic function $\lambda(t)$. For example, to graph the solution to the SIS model for various initial conditions if (a) $\lambda(t) = 3 - 2.5 \sin 6t$, $\gamma = 2$, and $\mu = 1$ and (b) $\lambda(t) = 3 - 2.5 \sin 6t$, $\gamma = 1$, and $\mu = 1$ we proceed as follows. For (a), we begin by defining λ, γ, and μ, and `eq`.

```
Clear[λ, i, t, γ, μ]

λ[t_] = 3 − 2.5Sin[6t];

γ = 2;

μ = 1;

eq = i'[t]==(λ[t] − (γ + μ))i[t] − λ[t]i[t]^2

i'[t] == −i[t]^2(3 − 2.5Sin[6t]) − 2.5i[t]Sin[6t]
```

We will graph the solutions satisfying the initial conditions $I(0) = I_0$ for $I_0 = 0.1, 0.2, \ldots, 0.9$. We begin by defining `graph`. Given `i0`, `graph[i0]` graphs the solution to the initial-value problem

$$\begin{cases} \dfrac{dI}{dt} = [\lambda(t) - (\gamma + \mu)]\, I - \lambda(t) I^2 \\ I(0) = I_0 \end{cases}$$

on the interval $[0, 10]$. Next, we use `Table` to define the list of numbers `inits`, corresponding to the initial conditions, and then use `Map` to apply the function `graph` to the list of numbers `inits`. We see that the result is a list of nine graphics objects that we name `toshow`.

```
graph[i0_]:=Module[{numsol},

    numsol = NDSolve[{eq, i[0]==i0}, i[t],

  {t, 0, 10}];

  Plot[i[t]/.numsol, {t, 0, 10}, PlotRange → {{0, 10}, {0, 1}},

  AspectRatio → 1]]

inits = Table[i, {i, 0.1, 0.9, 0.1}];

toshow = Map[graph, inits];
```

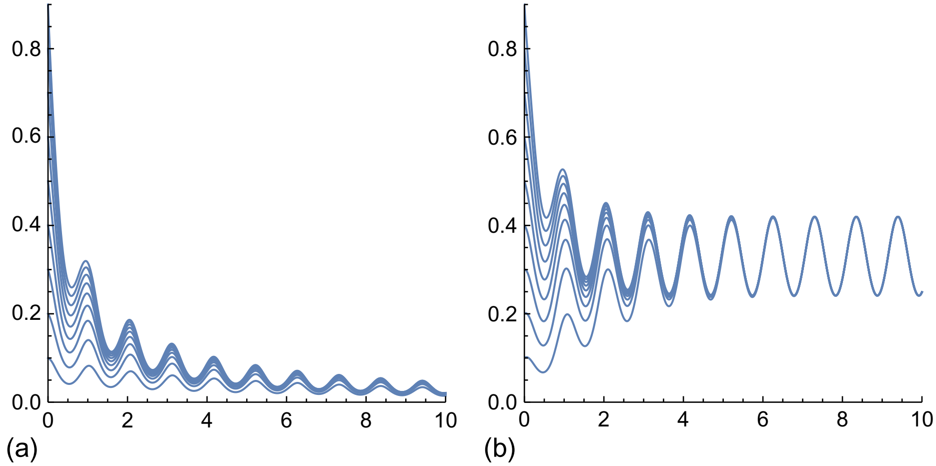
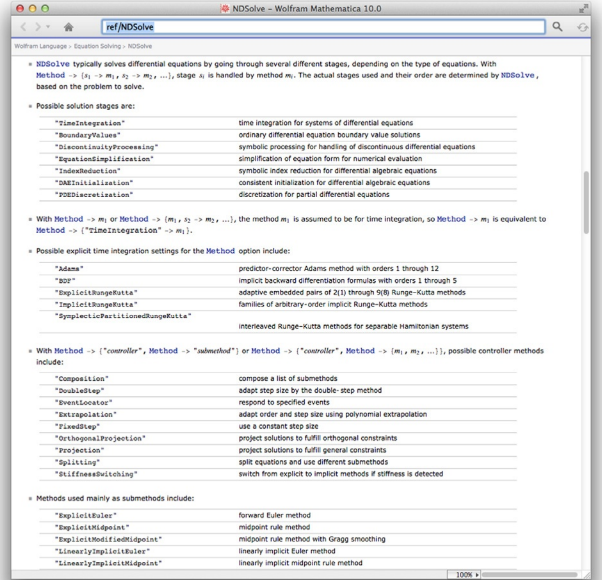


Figure 2-30 The disease persists periodically in the population

Finally, we use `Show` to view the list of nine graphs `toshow` in Figure 2-30(a). For (b), we proceed in the same manner as in (a). See Figure 2-30(b).

```
p1 = Show[toshow, PlotRange → {{0, 10}, All}, PlotLabel → "(a)"];

Clear[λ, i, t, γ, μ]

λ[t_] = 3 − 2.5Sin[6t];

γ = 1;

μ = 1;

eq = i′[t]==(λ[t] − (γ + μ))i[t] − λ[t]i[t]^2

i′[t] == i[t](1 − 2.5Sin[6t]) − i[t]^2(3 − 2.5Sin[6t])

graph[i0_]:=Module[{numsol},
numsol = NDSolve[{eq, i[0]==i0}, i[t], {t, 0, 10}];
  Plot[i[t]/.numsol, {t, 0, 10}, PlotRange → {{0, 10}, {0, 1}},
  AspectRatio → 1]]

inits = Table[i, {i, 0.1, 0.9, 0.1}];

toshow = graph/@inits;
```


NDSolve - Wolfram Mathematica 10.0

ref/NDSolve

Wolfram Language > Equation Solving > NDSolve

- NDSolve typically solves differential equations by going through several different stages, depending on the type of equations. With Method -> {s_1 -> m_1, s_2 -> m_2, ...}, stage s_i is handled by method m_i. The actual stages used and their order are determined by NDSolve, based on the problem to solve.
- Possible solution stages are:

"TimeIntegration"	time integration for systems of differential equations
"BoundaryValues"	ordinary differential equation boundary value solutions
"DiscontinuityProcessing"	symbolic processing for handling of discontinuous differential equations
"EquationSimplification"	simplification of equation form for numerical evaluation
"IndexReduction"	symbolic index reduction for differential algebraic equations
"DAEInitialization"	consistent initialization for differential algebraic equations
"PDEDiscretization"	discretization for partial differential equations

- With Method -> m_1 or Method -> {m_1, s_2 -> m_2, ...}, the method m_1 is assumed to be for time integration, so Method -> m_1 is equivalent to Method -> {"TimeIntegration" -> m_1}.
- Possible explicit time integration settings for the Method option include:

"Adams"	predictor-corrector Adams method with orders 1 through 12
"BDF"	implicit backward differentiation formulas with orders 1 through 5
"ExplicitRungeKutta"	adaptive embedded pairs of 2(1) through 9(8) Runge-Kutta methods
"ImplicitRungeKutta"	families of arbitrary-order implicit Runge-Kutta methods
"SymplecticPartitionedRungeKutta"	interleaved Runge-Kutta methods for separable Hamiltonian systems

- With Method -> {"*controller*", Method -> "*submethod*"} or Method -> {"*controller*", Method -> {m_1, m_2, ...}}, possible controller methods include:

"Composition"	compose a list of submethods
"DoubleStep"	adapt step size by the double- step method
"EventLocator"	respond to specified events
"Extrapolation"	adapt order and step size using polynomial extrapolation
"FixedStep"	use a constant step size
"OrthogonalProjection"	project solutions to fulfill orthogonal constraints
"Projection"	project solutions to fulfill general constraints
"Splitting"	split equations and use different submethods
"StiffnessSwitching"	switch from explicit to implicit methods if stiffness is detected

- Methods used mainly as submethods include:

"ExplicitEuler"	forward Euler method
"ExplicitMidpoint"	midpoint rule method
"ExplicitModifiedMidpoint"	midpoint rule method with Gragg smoothing
"LinearlyImplicitEuler"	linearly implicit Euler method
"LinearlyImplicitMidpoint"	linearly implicit midpoint rule method

100%

Figure 2-31 NDSolve can implement a wide range of numerical methods when solving differential equations

```
p2 = Show[toshow, PlotRange → {{0, 10}, All}, PlotLabel → "(b)"];

Show[GraphicsRow[{p1, p2}]]
```

2.6.2 Other Numerical Methods

In other cases, you may wish to implement your own numerical algorithms to approximate solutions of differential equations. We briefly discuss three familiar methods (Euler's method, the improved Euler's method, and the Runge-Kutta method) and illustrate how to implement these algorithms using Mathematica. Details regarding these and other algorithms, including discussions of the error involved in implementing them, can be found in most numerical analysis texts or other references like the Zwillinger's *Handbook of Differential Equations*, [29]. Also, note that NDSolve has numerous options for implementing different numerical methods when numerically solving differential equations. Use the **Help** menu and explore the NDSolve **Options** for a comprehensive discussion of NDSolve's capabilities. See Figure 2-31.

Euler's Method

In many cases, we cannot obtain an explicit formula for the solution to an initial-value problem of the form

$$\begin{cases} \dfrac{dy}{dx} = f(x, y) \\ y(x_0) = y_0 \end{cases}$$

but we can approximate the solution using a numerical method like **Euler's method**, which is based on tangent line approximations. Let h represent a small change, or **step size**, in the independent variable x. Then, we approximate the value of y at the sequence of x-values, $x_1, x_2, \ldots, x_n$, where

$$\begin{aligned} x_1 &= x_0 + h \\ x_2 &= x_1 + h = x_0 + 2h \\ x_3 &= x_2 + h = x_0 + 3h \\ &\vdots \\ x_n &= x_{n-1} + h = x_0 + nh. \end{aligned}$$

The slope of the tangent line to the graph of y at each value of x is found with the differential equation $y' = dy/dx = f(x, y)$. For example, at $x = x_0$, the slope of the tangent line is $f(x_0, y(x_0)) = f(x_0, y_0)$. Therefore, the tangent line to the graph of y is

$$y - y_0 = f(x_0, y_0)(x - x_0) \quad \text{or} \quad y = f(x_0, y_0)(x - x_0) + y_0.$$

Using this line to find the value of y, which we call y_1, at x_1 then yields

$$y_1 = f(x_0, y_0)(x_1 - x_0) + y_0 = hf(x_0, y_0) + y_0.$$

Therefore, we obtain the approximate value of y at x_1. Next, we use the point (x_1, y_1) to estimate the value of y when $x = x_2$. Using a similar procedure, we approximate the tangent line at $x = x_1$ with

$$y - y_1 = f(x_1, y_1)(x - x_1) \quad \text{or} \quad y = f(x_1, y_1)(x - x_1) + y_1.$$

Then, at $x = x_2$,

$$y_2 = f(x_1, y_1)(x_2 - x_1) + y_1 = hf(x_1, y_1) + y_1.$$

Continuing with this procedure, we see that at $x = x_n$,

$$y_n = hf(x_{n-1}, y_{n-1}) + y_{n-1}. \tag{2.10}$$

Using this formula, we obtain a sequence of points of the form (x_n, y_n), $n = 1, 2, \ldots$ where y_n is the approximate value of $y(x_n)$.

```
f[x_, y_] =?;
h =?;
x0 =?;
y0 =?;

xe[n_] = x0 + nh;
ye[n_]:=ye[n] = hf[xe[n − 1], ye[n − 1]] + ye[n − 1];
ye[0] = y0;
```

EXAMPLE 2.6.3: Use Euler's method with (a) $h = 0.1$ and (b) $h = 0.05$ to approximate the solution of $y' = xy$, $y(0) = 1$ on $0 \le x \le 1$. Also, determine the exact solution and compare the results.

SOLUTION: Because we will be considering this initial-value problem in subsequent examples, we first determine the exact solution with `DSolve` and graph the result with `Plot`, naming the graph `p1`.

```
exactsol = DSolve[{y'[x]==xy[x], y[0]==1}, y[x], x]
```

$$\left\{\left\{y[x] \to e^{\frac{x^2}{2}}\right\}\right\}$$

```
p1 = Plot[E^(x^2/2), {x, 0, 1}, PlotStyle → GrayLevel[0.4]];
```

To implement Euler's method (2.10), we note that $f(x, y) = xy$, $x_0 = 0$, and $y_0 = 1$. (a) With $h = 0.1$, we have the formula

$$y_n = hf(x_{n-1}, y_{n-1}) + y_{n-1} = 0.1x_{n-1}y_{n-1} + y_{n-1}.$$

For $x_1 = x_0 + h = 0.1$, we have

$$y_1 = 0.1x_0y_0 + y_0 = 0.1 \cdot 0 \cdot 1 + 1 = 1.$$

Similarly, for $x_2 = x_0 + 2h = 0.2$,

$$y_2 = 0.1x_1y_1 + y_1 = 0.1 \cdot 0.1 \cdot 1 + 1 = 1.01.$$

In the following, we define f, h, x, and y to calculate y_n given by equation (2.10). We define `ye` using the form

```
ye[n_]:=ye[n]=...
```

so that Mathematica "remembers" the values of `ye` computed, and thus, when computing `ye[n]`, Mathematica need not recompute `ye[n-1]` if `ye[n-1]` has previously been computed.

```
f[x_, y_] = xy;
h = 0.10;
x0 = 0;
y0 = 1;

xe[n_] = x0 + nh;
ye[n_]:=ye[n] = h * f[xe[n - 1], ye[n - 1]] + ye[n - 1];
ye[0] = y0;
```

Next, we use `Table` to calculate the set of ordered pairs (x_n, y_n) for $n = 0, 1, 2, \ldots, 9, 10$, naming the result `first`, and then `TableForm` to view first in traditional row-and-column form.

```
first = Table[{xe[n], ye[n]}, {n, 0, 10}];
TableForm[first]
```

0.	1
0.1	1.
0.2	1.0025
0.3	1.00751
0.4	1.01507
0.5	1.02522

```
0.6   1.03803
0.7   1.05361
0.8   1.07204
0.9   1.09348
1.    1.11809
```

To compare these results to the exact solution, we use `ListPlot` to graph the list of ordered pairs first in `t2` and display `t2` together with `p1` with `Show` in Figure 2-32.

```
lp = Map[Point, first]
```

```
{Point[{0., 1}], Point[{0.1, 1.}], Point[{0.2, 1.0025}],
Point[{0.3, 1.00751}], Point[{0.4, 1.01507}], Point[{0.5,
1.02522}], Point[{0.6, 1.03803}], Point[{0.7, 1.05361}]
Point[{0.8, 1.07204}], Point[{0.9, 1.09348}],
Point[{1., 1.11809}]}
```

Alternatively, we can produce Figure 2-32(a) using `ListPlot` together with the option `PlotStyle->PointSize[.02]` with the following command.

```
t2 = Graphics[{PointSize[0.02], lp}];

q1 = Show[p1, t2, PlotLabel → "(a)"];
```

(b) For $h = 0.05$, we use

$$y_n = hf(x_{n-1}, y_{n-1}) + y_{n-1} = 0.05x_{n-1}y_{n-1} + y_{n-1}$$

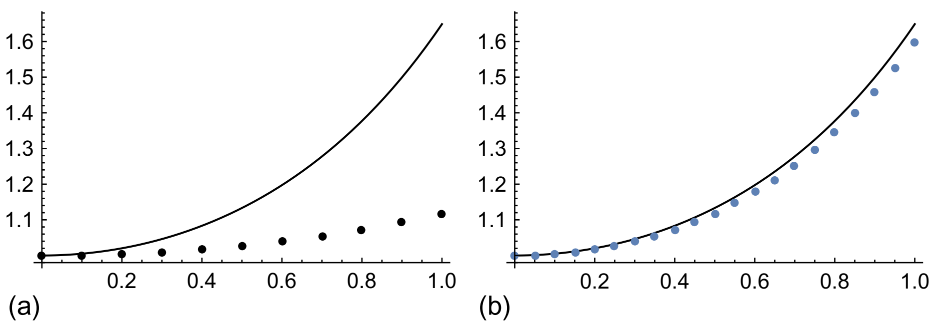


Figure 2-32 (a) Comparison of Euler's method to the exact solution using $h = 0.1$. (b) Comparison of Euler's method to the exact solution using $h = 0.05$

to obtain an approximation. In the same manner as in (a), we define f, h, x, and y to calculate y_n given by equation (2.10). Then, we use `Table` to calculate the set of ordered pairs for $n = 0, 1, 2, \ldots, 19, 20$, naming the result `second`, followed by `TableForm` to view `second` in traditional row-and-column form.

```
Remove[x, y, f]
f[x_, y_] = xy;
h = 0.05;
x0 = 0;
y0 = 1;

xe[n_] = x0 + nh;
ye[n_]:=ye[n] = h * f[xe[n - 1], ye[n - 1]] + ye[n - 1];
ye[0] = y0;

second = Table[{xe[n], ye[n]}, {n, 0, 20}];
TableForm[second]
```

0.	1
0.05	1.
0.1	1.0025
0.15	1.00751
0.2	1.01507
0.25	1.02522
0.3	1.03803
0.35	1.05361
0.4	1.07204
0.45	1.09348
0.5	1.11809
0.55	1.14604
0.6	1.17756

```
0.65  1.21288
0.7   1.2523
0.75  1.29613
0.8   1.34474
0.85  1.39853
0.9   1.45796
0.95  1.52357
1.    1.59594
```

We graph the approximation obtained with $h = 0.05$ together with the graph of $y = e^{x^2/2}$ in Figure 2-32(b). Notice that the approximation is more accurate when h is decreased.

```
t3 = ListPlot[second, PlotStyle->PointSize[0.02]];

q2 = Show[p1, t3, PlotLabel → "(b)"];

Show[GraphicsRow[{q1, q2}]]
```

■

Improved Euler's Method

Euler's method can be improved by using an average slope over each interval. Using the tangent line approximation of the curve through (x_0, y_0), $y = f(x_0, y_0)(x - x_0) + y_0$, we find the approximate value of y at $x = x_1$ which we now call $y_1{}^*$:

$$y_1{}^* = hf(x_0, y_0) + y_0.$$

With the differential equation $y' = f(x, y)$, we find that the approximate slope of the tangent line at $x = x_1$ is $f\left(x, y_1{}^*\right)$. Then, the average of the two slopes, $f(x_0, y_0)$ and $f(x_1, y_1{}^*)$, is $\frac{1}{2}(f(x_0, y_0) + f(x_1, y_1{}^*))$, and an equation of the line through (x_0, y_0) with slope $\frac{1}{2}(f(x_0, y_0) + f(x_1, y_1{}^*))$ is

$$y = \frac{1}{2}\left(f\left(x_0, y_0\right) + f\left(x_1, y_1{}^*\right)\right)\left(x - x_0\right) + y_0.$$

Therefore, at $x = x_1$, we find the approximate value of f with

$$y_1 = \frac{1}{2}\left(f\left(x_0, y_0\right) + f\left(x_1, y_1{}^*\right)\right)\left(x_1 - x_0\right) + y_0 = \frac{1}{2}h\left(f\left(x_0, y_0\right) + f\left(x_1, y_1{}^*\right)\right) + y_0.$$

Continuing in this manner, the approximation at each step of the **improved Euler's method** depends on the following two calculations:

$$\begin{aligned} y_n{}^* &= hf\left(x_{n-1}, y_{n-1}\right) + y_{n-1} \\ y_n &= \frac{1}{2}h\left(f\left(x_{n-1}, y_{n-1}\right) + f\left(x_n, y_n{}^*\right)\right) + y_{n-1}. \end{aligned} \tag{2.11}$$

```
f[x_, y_] =?;

h =?;

x0 =?;

y0 =?;

xi[n_] = x0 + nh;

yi[n_]:=
yi[n] = 1/2 h (f [xi[n − 1], yi[n − 1]] + f [xi[n], hf [xi[n − 1], yi[n − 1]]
 +yi[n − 1]]) + yi[n − 1]; yi[0] = y0;
```

EXAMPLE 2.6.4: Use the improved Euler's method to approximate the solution of $y' = xy$, $y(0) = 1$ on $0 \le x \le 1$ for $h = 0.1$. Also, compare the results to the exact solution.

SOLUTION: In this case, $f(x, y) = xy$, $x_0 = 0$, and $y_0 = 1$ so equation (2.11) becomes

$$\begin{aligned} y_n{}^* &= hx_{n-1}y_{n-1} + y_{n-1} \\ y_n &= \frac{1}{2}h\left(x_{n-1}y_{n-1} + x_n y_n{}^*\right) + y_{n-1} \end{aligned}$$

for $n = 1, 2, \ldots, 10$. For example, if $n = 1$, we have

$$y_1{}^* = hx_0y_0 + y_0 = 0.1 \cdot 0 \cdot 1 + 1 = 1$$

and

$$y_1 = \frac{1}{2}h\left(x_0y_0 + x_1y_1{}^*\right) + y_0 = \frac{1}{2} \cdot 0.1 \cdot (0 \cdot 1 + 0.1 \cdot 1) + 1 = 1.005.$$

Similarly,

$$y_2{}^* = hx_1y_1 + y_1 = 0.1 \cdot 0.1 \cdot 1.005 + 1.005 = 1.01505$$

and

$$\begin{aligned} y_2 &= \frac{1}{2}h\left(x_1y_1 + x_2y_2{}^*\right) + y_1 \\ &= \frac{1}{2} \cdot 0.1 \cdot (0.1 \cdot 1.005 + 0.2 \cdot 1.01505) + 1.005 = 1.0201755. \end{aligned}$$

In the same way as in the previous example, we define f, x, h, and y. We define `yi` using the form

`yi[n_]:=yi[n]=...,`

so that Mathematica "remembers" the values of `ystar` and `y` computed. Thus, to compute `yi[n]`, Mathematica need not recompute `yi[n-1]` if `yi[n-1]` has previously been computed.

```
Remove[f, x, y]

f[x_, y_] = xy;

h = 0.1;

x0 = 0;

y0 = 1;

xi[n_] = x0 + nh;

yi[n_]:=
yi[n] = N[1/2 h (f[xi[n - 1], yi[n - 1]]
        +f[xi[n], hf[xi[n - 1], yi[n - 1]] + yi[n - 1]])
        +yi[n - 1]];

yi[0] = y0;
```

We then compute (x_n, y_n) for $n = 0, 1, \ldots, 10$ and name the resulting list of ordered pairs `third`.

```
third = Table[{x_i[n], y_i[n]}, {n, 0, 10}];
TableForm[third]
```

0.	1
0.1	1.005
0.2	1.02018
0.3	1.04599
0.4	1.08322
0.5	1.13305
0.6	1.19707
0.7	1.27739
0.8	1.37677
0.9	1.49876
1.	1.64788

We graph the approximation obtained using the improved Euler's method together with the graph of the exact solution in Figure 2-33. From the results, we see that the approximation using the improved Euler's method results in a slight improvement from that obtained in the previous example.

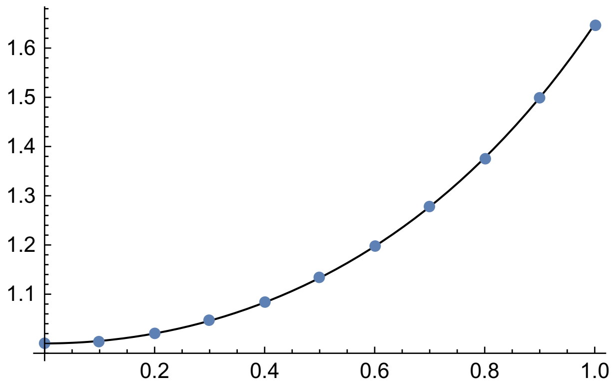


Figure 2-33 Comparison of the improved Euler's method to the exact solution using $h = 0.1$

```
t4 = ListPlot[third, PlotStyle->PointSize[0.02]];
Show[p1, t4]
```

■

The Runge-Kutta Method

In an attempt to improve on the approximation obtained with Euler's method as well as avoid the analytic differentiation of the function $f(x, y)$ to obtain $y'', y''', \ldots,$ the *Runge-Kutta method* is introduced. Let us begin with the *Runge-Kutta method of order two*. Suppose that we know the value of y at x_n. We now use the point (x_n, y_n) to approximate the value of y at a nearby value $x = x_n + h$ by assuming that

$$y_{n+1} = y_n + Ak_1 + Bk_2,$$

where

$$k_1 = hf\left(x_n, y_n\right) \quad \text{and} \quad k_2 = hf\left(x_n + ah, y_n + bk_1\right).$$

We can use the Taylor series expansion of y to obtain another representation of $y_{n+1} = y\left(x_n + h\right)$ as follows:

$$y\left(x_n + h\right) = y\left(x_n\right) + hy'\left(x_n\right) + \frac{h^2}{2!}y''\left(x_n\right) + \cdots = y_n + hy'\left(x_n\right) + \frac{h^2}{2!}y''\left(x_n\right) + \cdots$$

Now, because

$$y_{n+1} = y_n + Ak_1 + Bk_2 = y_n + Ahf\left(x_n, y_n\right) + Bhf\left(x_n + ah, y_n + bhf\left(x_n, y_m\right)\right),$$

we wish to determine values of A, B, a, and b such that these two representations of y_{n+1} agree. Notice that if we let $A = 1$ and $B = 0$, then the relationships match up to order h. However, we can choose these parameters more wisely so that agreement occurs up through terms of order h^2. This is accomplished by considering the Taylor series expansion of a function $z = F(x, y)$ of two variables about (x_0, y_0) which is given by

$$F\left(x_0, y_0\right) + \frac{\partial F}{\partial x}\left(x_0, y_0\right)\left(x - x_0\right) + \frac{\partial F}{\partial y}\left(x_0, y_0\right)\left(y - y_0\right) + \cdots$$

In our case, we have

$$\begin{aligned} f\left(x_n + ah, y_n + bhf\left(x_n, y_m\right)\right) = & f\left(x_n, y_n\right) + ah\frac{\partial f}{\partial x}\left(x_n, y_n\right) \\ & + bhf\left(x_n, y_n\right)\frac{\partial f}{\partial y}\left(x_n, y_n\right)\left(y - y_0\right) + O\left(h^2\right). \end{aligned}$$

The power series is then substituted into the following expression and simplified to yield:

$$\begin{aligned} y_{n+1} &= y_n + Ahf\left(x_n, y_n\right) + Bhf\left(x_n + ah, y_n + bhf\left(x_n, y_m\right)\right) \\ &= y_n + (A+B)hf\left(x_n, y_n\right) + aBh^2 \frac{\partial f}{\partial x}\left(x_n, y_n\right) + bBh^2 f\left(x_n, y_n\right) \frac{\partial f}{\partial x}\left(x_n, y_n\right) + O\left(h^3\right). \end{aligned}$$

Comparing this expression to the following power series obtained directly from the Taylor series of y,

$$y\left(x_n + h\right) = y\left(x_n\right) + hf\left(x_n, y_n\right) + \frac{1}{2}h^2 \frac{\partial f}{\partial x}\left(x_n, y_n\right) + \frac{1}{2}h^2 \frac{\partial f}{\partial y}\left(x_n, y_n\right) + O\left(h^3\right)$$

or

$$y_{n+1} = y_n + hf\left(x_n, y_n\right) + \frac{1}{2}h^2 \frac{\partial f}{\partial x}\left(x_n, y_n\right) + \frac{1}{2}h^2 \frac{\partial f}{\partial y}\left(x_n, y_n\right) + O\left(h^3\right),$$

we see that A, B, a, and b must satisfy the following system of nonlinear equations:

$$A + B = 1, \quad aA = \frac{1}{2}, \quad \text{and} \quad bB = \frac{1}{2}.$$

Therefore, choosing $a = b = 1$, the Runge-Kutta method of order two uses the equation:

$$\begin{aligned} y_{n+1} &= y\left(x_n + h\right) = y_n + \frac{1}{2}hf\left(x_n, y_n\right) + \frac{1}{2}hf\left(x_n + h, y_n + hf\left(x_n, y_n\right)\right) \\ &= y_n + \frac{1}{2}\left(k_1 + k_2\right), \end{aligned} \tag{2.12}$$

where $k_1 = hf\left(x_n, y_n\right)$ and $k_2 = hf\left(x_n + h, y_n + k_1\right)$.

```
f[x_, y_] =?;

h =?;

x0 =?;

y0 =?;

xr[n_] = x0 + nh;

yr[n_]:=
yr[n] = yr[n − 1] + 1/2 hf[xr[n − 1], yr[n − 1]] +
1/2 hf[xr[n − 1] + h, yr[n − 1] + hf[xr[n − 1], yr[n − 1]]]
yr[0] = y0;
```

EXAMPLE 2.6.5: Use the Runge-Kutta method of order two with $h = 0.1$ to approximate the solution of the initial-value problem $y' = xy$, $y(0) = 1$ on $0 \le x \le 1$.

SOLUTION: As with the previous examples, $f(x, y) = xy$, $x_0 = 0$, and $y_0 = 1$. Therefore, on each step we use the three equations

$$k_1 = hf(x_n, y_n) = 0.1 x_n y_n,$$
$$k_2 = hf(x_n + h, y_n + k_1) = 0.1(x_n + 0.1)(y_n + k_1),$$

and

$$y_{n+1} = y_n + \frac{1}{2}(k_1 + k_2).$$

For example, if $n = 0$, then

$$k_1 = 0.1 x_0 y_0 = 0.1 \cdot 0 \cdot 1 = 0,$$
$$k_2 = 0.1(x_0 + 0.1)(y_0 + k_1) = 0.1 \cdot 0.1 \cdot 1 = 0.01,$$

and

$$y_1 = y_0 + \frac{1}{2}(k_1 + k_2) = 1 + \frac{1}{2} \cdot 0.01 = 1.005.$$

Therefore, the Runge-Kutta method of order two approximates that the value of y at $x = 0.1$ is 1.005.

In the same manner as in the previous two examples, we define a function `yr` to implement the Runge-Kutta method of order two and use `Table` to generate a set of approximations for $n = 0, 1, \ldots, 10$.

```
Remove[f, x, y]

f[x_, y_] = xy;

h = 0.1;

x0 = 0;

y0 = 1;

xr[n_] = x0 + nh;

yr[n_]:=
yr[n] = yr[n - 1] + 1/2 hf[xr[n - 1], yr[n - 1]] +
```

$\frac{1}{2}hf\left[x_r[n-1]+h, y_r[n-1]+hf\left[x_r[n-1], y_r[n-1]\right]\right]$
$y_r[0] = y_0;$

$\texttt{rktable1} = \texttt{Table}\left[\{x_r[i], y_r[i]\}, \{i, 0, 10\}\right];$

```
TableForm[rktable1]
```

0.	1
0.1	1.005
0.2	1.02018
0.3	1.04599
0.4	1.08322
0.5	1.13305
0.6	1.19707
0.7	1.27739
0.8	1.37677
0.9	1.49876
1.	1.64788

We then use `ListPlot` to graph the set of points determined in `rktable1`. The graphs in `p1` and `p2` are shown together with `Show` in Figure 2-34.

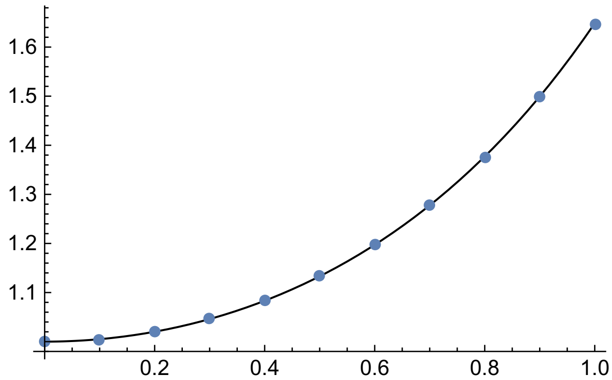


Figure 2-34 Comparison of the Runge-Kutta method of order two to the exact solution using $h = 0.1$

```
p2 = ListPlot[rktable1, PlotStyle->PointSize[0.02]];
Show[p1, p2]
```

■

The terms of the power series expansions used in the derivation of the Runge-Kutta method of order two can be made to match up to order four. These computations are rather complicated, so they will not be discussed here. However, after much work, the **fourth-order Runge-Kutta method** approximation at each step is found to be made with

$$y_{n+1} = y_n + \frac{1}{6}h\left(k_1 + 2k_2 + 2k_3 + k_4\right),\ n = 0,\ 1,\ 2,\ \ldots$$

where

$$\begin{aligned} k_1 &= f\left(x_n, y_n\right) \\ k_2 &= f\left(x_n + \frac{1}{2}h, y_n + \frac{1}{2}hk_1\right) \\ k_3 &= f\left(x_n + \frac{1}{2}h, y_n + \frac{1}{2}hk_2\right) \end{aligned} \tag{2.13}$$

and

$$k_4 = f\left(x_{n+1}, y_n + hk_3\right).$$

```
f[x_, y_] =?;
h =?;
x0 =?;
y0 =?;

xr[n_] = x0 + nh;
yr[n_]:=yr[n] = yr[n - 1] + 1/6 h (k1[n - 1] + 2k2[n - 1] + 2k3[n - 1] + k4[n - 1]);
yr[0] = y0;
k1[n_]:=k1[n] = f[xr[n], yr[n]];
k2[n_]:=k2[n] = f[xr[n] + h/2, yr[n] + 1/2 hk1[n]];
k3[n_]:=k3[n] = f[xr[n] + h/2, yr[n] + 1/2 hk2[n]];
k4[n_]:=k4[n] = f[xr[n + 1], yr[n] + hk3[n]]
```

EXAMPLE 2.6.6: Use the fourth-order Runge-Kutta method with $h = 0.1$ to approximate the solution of the problem $y' = xy$, $y(0) = 1$ on $0 \le x \le 1$.

SOLUTION: With $f(x, y) = xy$, $x_0 = 0$, and $y_0 = 1$, using equation (2.13), the formulas are

$$y_{n+1} = y_n + \frac{0.1}{6}\left(k_1 + 2k_2 + 2k_3 + k_4\right), \; n = 0, \; 1, \; 2, \; \ldots$$

where

$$\begin{aligned}
k_1 &= f\left(x_n, y_n\right) = x_n y_n \\
k_2 &= f\left(x_n + \frac{1}{2}h, y_n + \frac{1}{2}hk_1\right) = \left(x_n + \frac{1}{2}\cdot 0.1\right)\left(y_n + \frac{1}{2}\cdot 0.1k_1\right) \\
k_3 &= f\left(x_n + \frac{1}{2}h, y_n + \frac{1}{2}hk_2\right) = \left(x_n + \frac{1}{2}\cdot 0.1\right)\left(y_n + \frac{1}{2}\cdot 0.1k_2\right)
\end{aligned}$$

and

$$k_4 = f\left(x_{n+1}, y_n + hk_3\right) = x_{n+1}\left(y_n + 0.1k_3\right).$$

For $n = 0$, we have

$$\begin{aligned}
k_1 &= x_0 y_0 = 0 \cdot 1 = 0 \\
k_2 &= \left(x_0 + \frac{1}{2}\cdot 0.1\right)\left(y_0 + \frac{1}{2}\cdot 0.1k_1\right) = 0.05 \cdot 1 = 0.05 \\
k_3 &= \left(x_0 + \frac{1}{2}\cdot 0.1\right)\left(y_0 + \frac{1}{2}\cdot 0.1k_2\right) = 0.05 \cdot (1 + 0.0025) = 0.050125
\end{aligned}$$

and

$$k_4 = x_1\left(y_0 + 0.1k_3\right) = 0.1 \cdot (1 + 0.0050125) = 0.10050125.$$

Therefore,

$$y_1 = y_0 + \frac{0.1}{6}\left(k_1 + 2k_2 + 2k_3 + k_4\right) = 1.005012521.$$

We list the results for the Runge-Kutta method of order four and compare these results to the exact solution in Figure 2-35. Notice that this method yields the most accurate approximation of the methods used to this point.

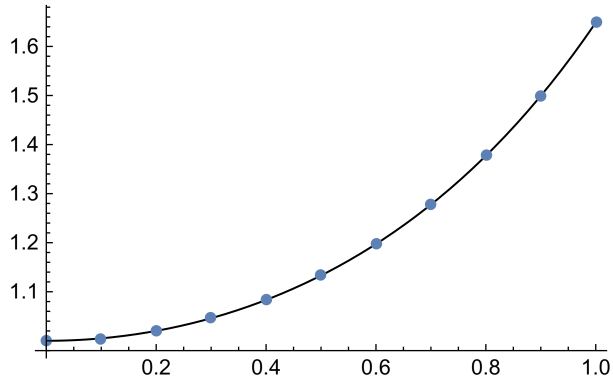


Figure 2-35 Comparison of the fourth-order Runge-Kutta method to the exact solution using $h = 0.1$

```
Remove[f, x, y]

f[x_, y_] = xy;

h = 0.1;

x0 = 0;

y0 = 1;

xr[n_] = x0 + nh;
yr[n_]:=yr[n] = yr[n - 1] + 1/6 h(k1[n - 1] + 2k2[n - 1] + 2k3[n - 1]
+k4[n - 1]);
yr[0] = y0;
k1[n_]:=k1[n] = f[xr[n], yr[n]];
k2[n_]:=k2[n] = f[xr[n] + h/2, yr[n] + 1/2 hk1[n]];
k3[n_]:=k3[n] = f[xr[n] + h/2, yr[n] + 1/2 hk2[n]];
k4[n_]:=k4[n] = f[xr[n + 1], yr[n] + hk3[n]]
rktable2 = Table[{xr[i], yr[i]}, {i, 0, 10}];
TableForm[rktable2]
```

0.	1
0.1	1.00501

```
 0.2  1.0202
 0.3  1.04603
 0.4  1.08329
 0.5  1.13315
 0.6  1.19722
 0.7  1.27762
 0.8  1.37713
 0.9  1.4993
 1.   1.64872
```

```
p3 = ListPlot[rktable2, PlotStyle->PointSize[0.02]];
Show[p1, p3]
```

■