

1.4 Direction Fields

The **method of direction fields** is a graphical method for displaying the general shape and behavior of solutions to $y' = f(x, y)$. It persists as a fundamental topic because it *does not require solving the differential equation* $y' = f(x, y)$. The uniform grid method and the isocline method are introduced, for computer and hand construction of direction fields.

What's a Direction Field?

L. Euler (1707–1783) discovered a way to draw a graphic showing the behavior of all solutions to a given differential equation, *without solving the equation*. The graphic is built from a grid of points arranged on a graph window. Paired with each grid point is a line segment centered on the grid point. The line segments are non-overlapping.

The graph window plus pairs of grid points and line segments is called a **direction field**, provided the line segment at grid point (x_0, y_0) coincides with the tangent line to the solution of the initial value problem

$$\begin{cases} y' = f(x, y), \\ y(x_0) = y_0. \end{cases}$$

This means that $y'(x_0)$ equals the slope of the line segment. We don't have to know a formula for $y(x)$, because $y'(x_0)$ can be computed from its equivalent formula $y'(x_0) = f(x_0, y_0)$, a number that depends only on the grid point (x_0, y_0) and the function $f(x, y)$. Euler's intent is to replace the differential equation model $y' = f(x, y)$ by a graphical model, a direction field.

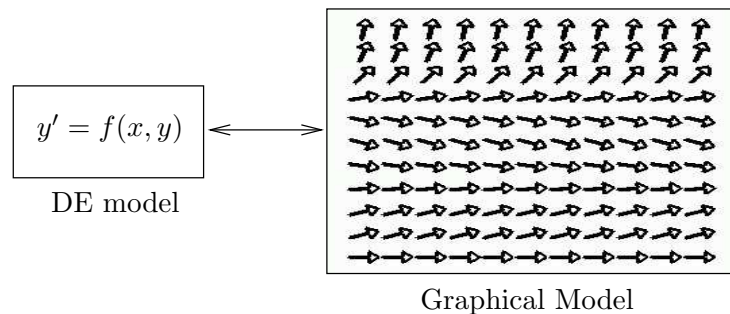


Figure 7. Model Replacement.

A differential equation model $y' = f(x, y)$ is replaced by a direction field model, a graphic consisting of pairs of grid points and line segments. A direction field **line segment** is represented by an arrow, to show the direction of the tangent. A **grid point** is the center point of an arrow shaft. Model replacement means the differential equation model is tossed aside and we deal only with the graphical model.

Solution Curves and Direction Fields

Euler's visualization idea begins with the direction field, drawn for some graph window, with pairs of grid points and line segments dense enough to cover most of the white space in the graph window. The *theory* used in Euler's idea consists of a short list of facts:

1. Solutions of $y' = f(x, y)$ don't cross.
2. A tangent to an edge-to-edge solution $y(x)$ nearly matches tangents to nearby direction field segments.
3. Direction field segments are solutions of $y' = f(x, y)$, to pixel resolution.

Details 1: If solutions $y_1(x)$, $y_2(x)$ cross at $x = x_0$, then let $y_0 = y_1(x_0) = y_2(x_0)$ and consider the initial value problem

$$\begin{cases} y' = f(x, y), \\ y(x_0) = y_0. \end{cases}$$

We assume solutions to all such initial value problems are *unique*. This implies $y_1(x) = y_2(x)$ for $|x - x_0|$ small. Hence crossings are impossible. The analysis implies that two solutions which touch must coalesce.

Direction field segments, representing solution curves, must be constructed not to touch each other. Edge-to-edge solution curves cannot cross a direction field segment, but they may coincide with a direction field segment, to pixel resolution.

Details 2: Tangent vectors for $\vec{r} = x\vec{i} + y(x)\vec{j}$ are drawn from $\vec{r}' = \vec{i} + y'(x)\vec{j} = \vec{i} + f(x, y)\vec{j}$. Continuity of f implies that the vector \vec{r}' is to pixel resolution identical for all (x, y) sufficiently close to a grid point (x_0, y_0) . This is why an edge-to-edge solution passes grid points with tangent vector *nearly* matching the tangent vector of nearby segments.

Details 3: Each segment is a tangent line $y = y_0 + m(x - x_0)$, constructed with slope $m = y'(x_0)$. It approximates the curve $y(x)$ local to the contact point (x_0, y_0) . Graphically, a short tangent line coalesces with the curve near the contact point, to pixel resolution.

The tangent line approximation is called **Euler's approximation**. Correct pronunciation is **Oiler**. To make the audience giggle, pronounce it **Yuler**.

Rules for Drawing Threaded Solutions

A direction field graphic replaces all the information supplied by the equation $y' = f(x, y)$. The equation is tossed aside and not used.

Visualization of all solutions involves drawing a small number of edge-to-edge solutions $y(x)$ onto the direction field graph window. We will use just two abbreviated rules (see **1** and **2** in Table 4).

1. Solutions don't cross.
2. Nearby tangents nearly match.

Table 4. Rules for Drawing Edge-to-Edge Solutions

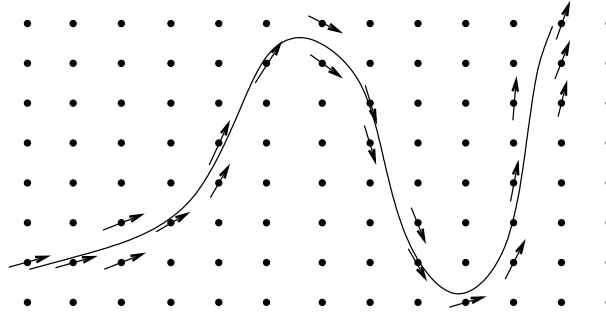


Figure 8. Threading a Solution Edge-to-Edge.

Shown in Figure 8 is a threaded solution curve for $y' = f(x, y)$ plus nearby grid points and relevant line segments (arrows). The solution threads its way through the direction field, matching tangents at nearby grid points. Arrows that touch a threaded curve must coalesce with the curve (solutions don't cross).

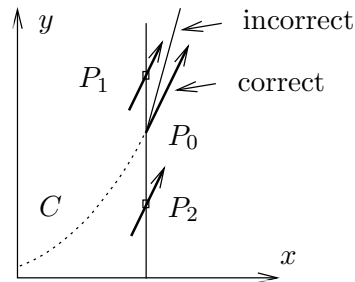


Figure 9. Threading Rules.

Solution curve C threads from the left and meets a line at point P_0 . The line contains two nearby grid points P_1, P_2 . The tangent at P_0 must nearly match direction field arrows at P_1, P_2 .

Tangent Matching Explained.

The slopes of the tangents in Figure 9 are given by $y' = f(x, y)$. For points $P_1 = (x_1, y_1)$, $P_0 = (x_0, y_0)$ and $P_2 = (x_2, y_2)$, the slopes are $f(x_1, y_1)$, $f(x_0, y_0)$, $f(x_2, y_2)$. If the points P_0, P_1, P_2 are close, then continuity of f implies all three slopes are *nearly equal*.

How to Construct a Direction Field Graphic

Window	Invent the graph x -range and y -range. 1
Grid	Plot a uniform grid of N grid points within the graph window. Invent N to populate the graphical white space, $N \approx 50$ for hand work. 2
Field	Draw at each grid point (x^*, y^*) a short tangent vector $\epsilon \vec{T}$, where $\vec{T} = \vec{i} + f(x^*, y^*)\vec{j}$. 3
Threaded Solutions	Draw additional threaded solutions on long x -intervals into the remaining white space of the graphic. 4

Construction Notes.

- 1 The window should include all significant equilibrium solutions, that is, solutions $y = \text{constant}$ of $y' = f(x, y)$, which plot as horizontal lines. Physically interesting initial conditions $(x_0, y(x_0))$ should be added.
- 2 The isocline method might also be used to select grid points. For details on both methods, see the next subsection.
- 3 The arrow shaft is a **replacement curve** for the solution of $y' = f(x, y)$ through grid point (x^*, y^*) on a small x -interval, called a **lineal element**.
- 4 Threading is educated guesswork, discussed above, in Figures 8 and 9. If possible, choose (x_0, y_0) on the left window edge, then thread the solution until it exits the window top, bottom or right.

Direction fields are studied again in Chapter 10, page 683, in connection with phase portraits of two-dimensional systems of differential equations.

Two Methods for Selecting Grid Points

There are two standard methods for selecting grid points, called the **uniform grid method** and the **isocline grid method**. The methods may be combined in some applications.

Uniform Grid Two positive increment parameters n and m are supplied along with a graph window $a \leq x \leq b$, $c \leq y \leq d$. Hand work usually starts with $n = m = 11$; computer software starts with $n = m = 21$.

The nm grid points are defined for $i = 1, \dots, n$ and $j = 1, \dots, m$ by the equations $x_i = a + (b - a)(i - 1)/(n - 1)$, $y_j = c + (d - c)(j - 1)/(m - 1)$.

Isocline Grid A graph window $a \leq x \leq b$, $c \leq y \leq d$ is given plus a list of invented slopes M_1, \dots, M_p for the lineal elements.

To define the grid points, select the number $n > 0$ of grid points to be drawn on each isocline. Construct n equally-spaced horizontal lines (or vertical lines). Define grid points as intersections of the lines with all the implicit curves $f(x, y) = M_\ell$, $\ell = 1, \dots, p$.

Along the implicit curve $f(x, y) = M_\ell$, within the graph window, mark each grid point and draw a lineal element, each element of exactly the same slope M_ℓ , for $\ell = 1, \dots, p$.

The two methods are applied in Examples 27 and 28, page 45. Illustrated for the isocline method are possibilities such as graph window clipping and fine-tuning of the slopes to allow the grid points to fill the window.

Grid points in the *isocline method* are intersections of equally-spaced lines with the implicit curves $f(x, y) = M_\ell$. Lineal elements sketched along this curve all have slope M_ℓ and therefore they can be drawn with reduced effort.

How to Make Lineal Elements

A lineal element is a line segment centered at a grid point. They should not touch, because they represent, to pixel resolution, non-crossing solution curves on a short x -interval. Choose H to be not greater than the minimum distance between pairs of grid points. Initially, one can guess the value of H , then adjust the value after seeing the result. Define

$$h = \frac{H}{2\sqrt{1 + |f(x_0, y_0)|^2}}.$$

Then a lineal element of length H is defined by the midpoint (x_0, y_0) and the two endpoints $(x_0 - h, y_0 - hM)$ and $(x_0 + h, y_0 + hM)$, where $M = f(x_0, y_0)$.

This choice insures lineal elements do not touch. It is possible to erase the line segment to the left or right of the grid point without losing much information. Arrow heads can be added to show the tangent direction.

Examples

- 25 Example (Window and Grid)** Choose a graph window for the differential equation $y' = y^2(2 - y)(1 + y)$ which includes the equilibrium solutions. Draw a 5×5 uniform grid on the graph window and plot the equilibrium solutions. Do not draw the direction field nor threaded solutions.

Solution: Let $f(x, y) = y^2(2 - y)(1 + y)$. Then $y = k$ is a constant solution of $y' = f(x, y)$ exactly when $0 = k^2(2 - k)(1 + k)$. The values $k = -1, 0, 2$ give horizontal lines $y = -1, y = 0, y = 2$. These lines are called equilibrium solutions; they are constant solutions of the differential equation. Accordingly, a graph window containing the equilibria is $-3 \leq x \leq 3, -2 \leq y \leq 3$. The 25 grid points are obtained by the formulas $x_k = -3 + 6(k/4), k = 0, \dots, 4$ and $y_j = -2 + 5(j/4), j = 0, \dots, 4$. The plot is done by hand. A computer plot appears in Figure 10.

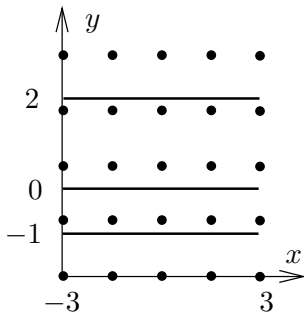


Figure 10. A graph window with uniform grid and equilibria. Three equilibrium solutions $y = -1$, $y = 0$, $y = 2$ appear plus 25 grid points on the graph window $|x| \leq 3$, $-2 \leq y \leq 3$.

26 Example (Threading a Solution) Starting at the black dots in the direction field graphic of Figure 11, thread three solution curves.

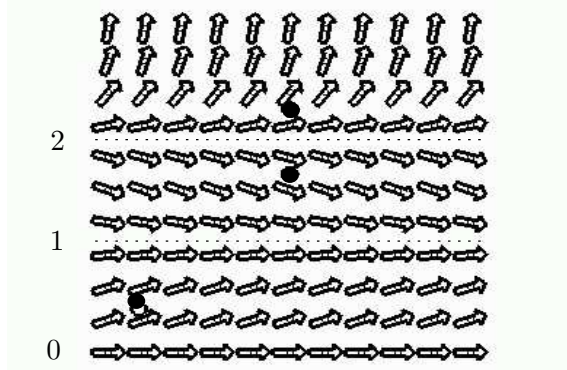


Figure 11. A direction field. A field for the differential equation $y' = y(2 - y)(1 - y)$ is plotted on graph window $0 \leq x \leq 3$, $0 \leq y \leq 3$. The black dots are at $(0.25, 0.4)$, $(1.5, 2.25)$ and $(1.5, 1.65)$.

Solution: A plot appears in Figure 12. A threaded solution matches its tangents with nearby lineal elements of the direction field in Figure 11; see page 41 for an explanation. Each threaded curve represents a solution of the differential equation through the given dot on the entire interval $0 \leq x \leq 3$, whereas the lineal elements represent solutions through the grid point on a very short x -interval.

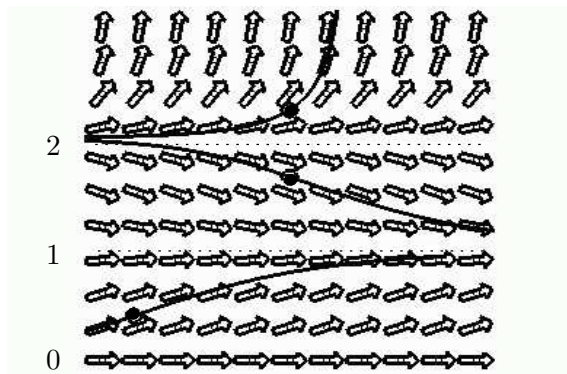


Figure 12. Threaded solutions. The graph window is $0 \leq x \leq 3$, $0 \leq y \leq 3$. Threaded curves cannot cross equilibrium solutions $y = 0$, $y = 1$ and $y = 2$.

27 Example (Uniform Grid Method) Make a direction field of 11×11 points for $y' = x + y(1 - y)$ on $-1 \leq x \leq 1$, $-2 \leq y \leq 2$.

Solution: Let $f(x, y) = x + y(1 - y)$. The 121 grid points are the pairs (x, y)

where $x = -1$ to 1 in increments of 0.2 and $y = -2$ to 2 in increments of 0.4 . The minimum distance between grid points is $H = 0.2$.

We will generate the endpoints of the lineal element at $x_0 = -0.4$, $y_0 = 1.6$. It will be shown that the first endpoint is $(-0.34076096, 1.5194349)$. This point can be located from (x_0, y_0) by traveling distance $H/2$ at slope $M = -1.36$.

$M = f(x_0, y_0)$	The line segment slope for Euler's rule.
$= x_0 + y_0(1 - y_0)$	Apply $f(x, y) = x + y(1 - y)$.
$= -1.36,$	Use the first point $x_0 = -0.4$, $y_0 = 1.6$.
$h = \frac{H}{2\sqrt{1 + M^2}}$	Apply the formula $h = (H/2)/\sqrt{1 + f(x_0, y_0)^2}$.
$= 0.059239045,$	Use $H = 0.2$ and $f(x_0, y_0) = M = -1.36$.
$X = x_0 + h$	Compute the x -coordinate of the second point.
$= -0.34076096$	Use $x_0 = -0.4$ and $h = 0.059239045$.
$Y = y_0 + hf(x_0, y_0)$	Compute the y -coordinate of the second point.
$= 1.5194349$	Use values $y_0 = 1.6$, $f(x_0, y_0) = M = -1.36$, $h = 0.059239045$.

The second endpoint $(-0.459239045, 1.6805651)$, at distance $H/2$ from the grid point, in the opposite direction, can be found by minor changes to the above calculation. Automation of this process is necessary because 121 such calculations are required. Some basic `maple` code appears below which computes the 121 pairs of points for the direction field, then plots a replica of the field. The graphic appears in Figure 13. The code adapts to numerical laboratories like `matlab`, `octave` and `scilab`, which may or may not have a suitable direction field library, depending on the version.

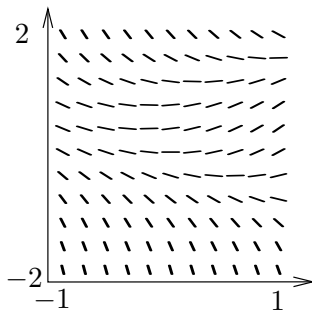


Figure 13. Direction field for the equation $y' = x + y(1 - y)$.

The uniform grid method is used on graph window $-1 \leq x \leq 1$, $-2 \leq y \leq 2$. There are 121 grid points.

```
a:=-1:b:=1:c:=-2:d:=2:n:=11:m:=11:
H:=(b-a)/(n-1):K:=(d-c)/(m-1):HH:=0.15:
f:=(x,y)->x+y*(1-y): X:=t->a+H*(t-1):Y:=t->c+K*(t-1):P:=[]:
for i from 1 to n do for j from 1 to m do
  x0:=X(i):y0:=Y(j):M:=evalf(f(x0,y0)):
  h:=evalf((HH/2)/sqrt(1+M^2)):
  Seg:=[[x0-h,y0-h*M],[x0+h,y0+h*M]]:
  if (P = []) then P:=Seg: next: fi: P:=P,Seg:
od:od:
opts:=scaling=constrained,color=black,thickness=3,axes=boxed;
plot([P],opts);
```

Versions of `maple` since V 5.1 have a `DEtools` package which simplifies the process of making a direction field. In `mathematica`, a similar command exists.

```
with(DEtools): de:=diff(y(x),x)=x+y(x)*(1-y(x)): # Maple
opts:=arrows=LINE,dirgrid=[11,11];
DEplot(de,y(x),x=-1..1,y=-2..2,opts);

<< Graphics\PlotField.m # Mathematica
PlotVectorField[1,x+y (1-y),x,-1,1,y,-2,2]
```

Resources for computer-assisted direction fields with interactive threaded solutions include `Maple`, `Mathematica`, `Matlab`. The 2014 `Matlab` interface uses open-source computer algebra system `Maxima`. Each system has a steep learning curve, well worth the investment time.

28 Example (Isocline Method) Make a direction field by the isocline method for the differential equation $y' = x + y(1 - y)$ on $-1 \leq x \leq 1$, $-1 \leq y \leq 2$.

Solution: Let $f(x, y) = x + y(1 - y)$ and let M denote the slope of a replacement lineal element. The isoclines are defined by $f(x, y) = M$. It has the standard equation $(y - 1/2)^2 = x - M + 1/4$, which is a parabola with center $(M - 1/4, 1/2)$ opening to the right. The algebra details:

$x + y(1 - y) = M$	The equation $f(x, y) = M$ expanded.
$y^2 - y = x - M$	Multiply by -1 and move $-x$ to the right side.
$y^2 - y + \frac{1}{4} = x - M + \frac{1}{4}$	Apply <i>square completion</i> : add the square of half the coefficient of y to both sides.
$(y - \frac{1}{2})^2 = x - M + \frac{1}{4}$	Write the left side as a perfect square. It has the form of the standard curve library equation $Y^2 = X$. See page 849.

The basic requirement for slope M selection is that the set of grid points obtained below fills the white space of the graph window. Briefly, some portion of each parabola has to intersect the graph window. By experiment, the slopes M to be used in the isocline method will be selected as $M = 1/4 + (-3)$ to $M = 1/4 + (1)$ in increments of 0.2 to identify 21 isoclines:

Isocline Equation	Slope M
$(y - 1/2)^2 = x - (-3)$	$0.25 + (-3)$
$(y - 1/2)^2 = x - (-2.8)$	$0.25 + (-3) + 0.2$
\vdots	\vdots
$(y - 1/2)^2 = x - (1)$	$0.25 + (-3) + 4.0$

To define the grid points, let $y = -1$ to 2 in increments of 0.3 to make 11 horizontal lines. The intersections of these 11 lines with the 21 parabolas define at least 100 grid points inside the graph window. It is possible to graph rapidly the 21 parabolas, because they are translates of the standard parabola $Y^2 = X$.

The replacement lineal elements on each parabola are sketched rapidly as follows. Using pencil and paper, graph accurately the first lineal element on the isocline curve, using associated slope M . Rotate the paper until the lineal element is vertical. Draw additional lineal elements at the remaining grid points of

the isocline curve as vertical lines. Accuracy improves with the use of a drawing easel, T -square and triangle.

A computer graphic is shown in Figure 14 which closely resembles a hand-made graphic. Compare it to the uniform grid method graphic in Figure 13, page 46.

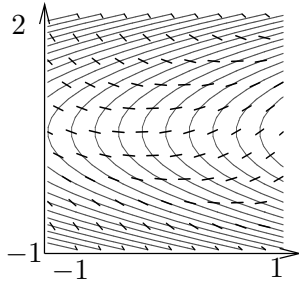


Figure 14. Direction field for the differential equation $y' = x + y(1 - y)$.

The isocline method is applied using graph window $-1 \leq x \leq 1$, $-1 \leq y \leq 2$. Parabolas are isoclines. Grid points are intersections of isoclines with equally-spaced horizontal lines.

The `maple` code that produced Figure 14 is included below to show the machine equivalents of a hand computation. The ordering of events in the code is reversed: the grid points and lineal elements are drawn first, then the isocline curves, finally the two graphics are superimposed. A key detail is solving $f(x_0, y_0) = M$ for x_0 to locate a grid point (x_0, y_0) (in this application, $x_0 = M + y_0(y_0 - 1)$). The clipping test $a \leq x_0 \leq b$ is applied to keep lineal elements inside the graph window.

```
with(plots):
a:=-1:b:=1:n:=21:c:=-1:d:=2:m:=11:
H:=0.1:F:=-3:G:=1:f:=unapply(x+y*(1-y),(x,y)):
Slope:=unapply(F+1/4+(G-F)*(t-1)/(n-1),t):
Y:=unapply(c+(d-c)*(t-1)/(m-1),t): P:=[]:
for j from 1 to n do
M:=Slope(j):h:=evalf(H*0.5/sqrt(1+M^2)):
for k from 1 to m do
y0:=Y(k): x0:=solve(f(x,y0)=M,x):
if x0<a or x0>b then next: fi:
Seg:=[[x0-h,y0-h*M],[x0+h,y0+h*M]]:
if P=[] then P:=Seg: next: fi:
P:=P,Seg:
od:od:
opts:=color=BLACK,thickness=3,axes=none,scaling=constrained:
Window:=x=a..b,y=c..d:
Plot1:=plot([P],Window,opts):
eqs:={seq((y-0.5)^2=x-Slope(j)+1/4,j=1..n)}:
Plot2:=implicitplot(eqs,Window):
display([Plot1,Plot2]);
```

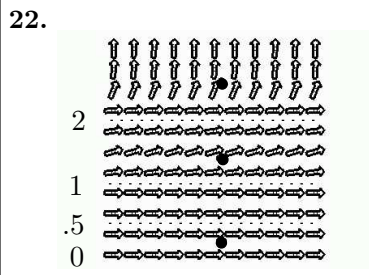
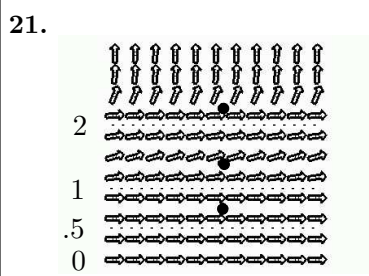
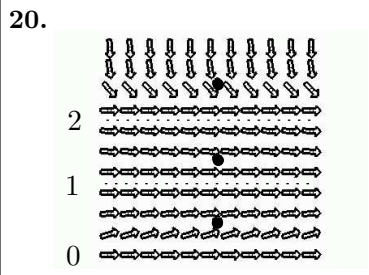
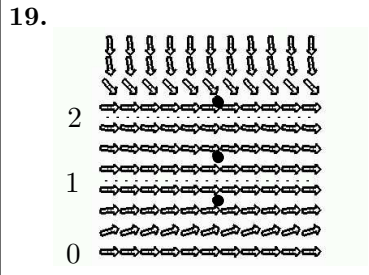
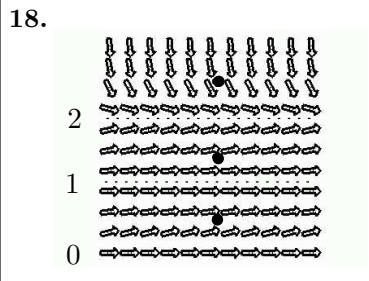
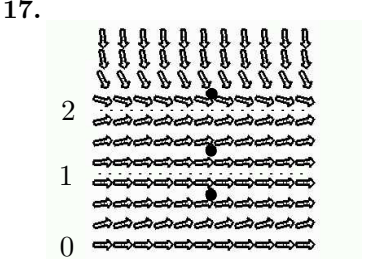
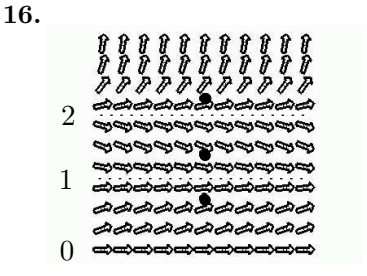
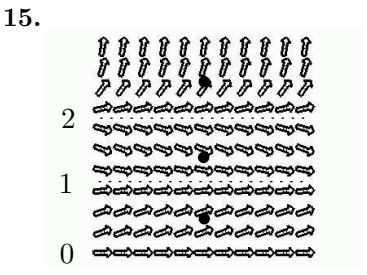
Exercises 1.4

Window and Grid. Find the equilibrium solutions, then determine a graph window which includes them and construct a 5×5 uniform grid. Follow Example 25.

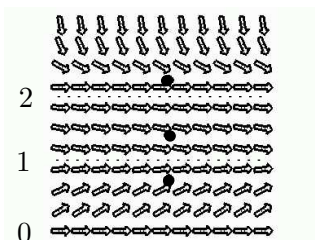
1. $y' = 2y$
2. $y' = 3y$
3. $y' = 2y + 2$

4. $y' = 3y - 2$
5. $y' = y(1 - y)$
6. $y' = 2y(3 - y)$
7. $y' = y(1 - y)(2 - y)$
8. $y' = 2y(1 - y)(1 + y)$
9. $y' = 2(y - 1)(y + 1)^2$
10. $y' = 2y^2(y - 1)^2$
11. $y' = (x + 1)(y + 1)(y - 1)y$
12. $y' = 2(x + 1)y^2(y + 1)(y - 1)^2$
13. $y' = (x + 2)y(y - 3)(y + 2)$
14. $y' = (x + 1)y(y - 2)(y + 3)$

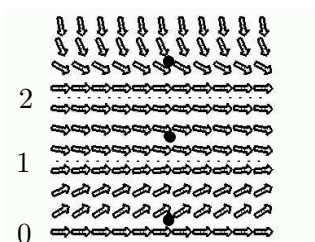
Threading Solutions. Each direction field below has window $0 \leq x \leq 3$, $0 \leq y \leq 3$. Start each threaded solution at a black dot and continue it left and right across the field. Dotted horizontal lines are equilibrium solutions. See Example 26.



23.



24.



Uniform Grid Method. Apply the uniform grid method as in Example 27, page 45 to make a direction field of 11×11 grid points for the given differential equation on $-1 \leq x \leq 1$, $-2 \leq y \leq 2$. If using a computer program, then use about 20×20 grid points.

25. $y' = 2y$

26. $y' = 3y$

27. $y' = 1 + y$

28. $y' = 2 + 3y$

29. $y' = x + y(2 - y)$

30. $y' = x + y(1 - 2y)$

31. $y' = 1 + y(2 - y)$

32. $y' = 1 + 2y(2 - y)$

33. $y' = x - y$

34. $y' = x + y$

35. $y' = y - \sin(x)$

36. $y' = y + \sin(x)$

Isocline Method. Apply the isocline method as in Example 28, page 47 to make a direction field of about 11×11 points for the given differential equation on $0 \leq x \leq 1$, $0 \leq y \leq 2$. Computer programs are used on these kinds of problems to find grid points as intersections of isoclines and lines. Graphics are expected to be done by hand. Large white spaces in the graphic should be filled by choosing a richer set of slopes.

37. $y' = x - y^2$

38. $y' = 2x - y^2$

39. $y' = 2y/(x + 1)$

40. $y' = -y^2/(x + 1)^2$

41. $y' = \sin(x - y)$

42. $y' = \cos(x - y)$

43. $y' = xy$

44. $y' = x^2y$

45. $y' = xy + 2x$

46. $y' = x^2y + 2x^2$