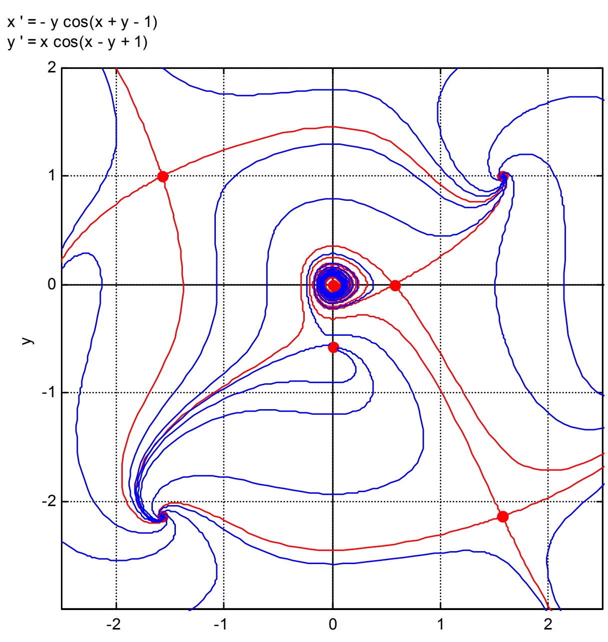
Application 9.2
Phase Plane Portraits of Almost Linear Systems

Interesting and complicated phase portraits often result from simple nonlinear perturbations of linear systems. For instance, the figure below shows a phase plane portrait for the almost linear system

$$\begin{aligned}\frac{dx}{dt} &= -y\cos(x+y-1) \\ \frac{dy}{dt} &= x\cos(x-y+1).\end{aligned} \tag{1}$$

Among the seven critical points marked with dots, we see

- Apparent spiral points in the first and third quadrants of the *xy*-plane;
- Apparent saddle points in the second and fourth quadrants, plus another one on the positive *x*-axis;
- A critical point of undetermined character on the negative *y*-axis; and
- An apparently "very weak" spiral point at the origin -- meaning one that is approached very slowly as t increases or decreases (according as it is a sink or a source).



Some ODE software systems can automatically locate and classify critical points. For instance, Fig. 9.2.22 in the text shows a screen produced by John Polking's MATLAB **`pplane`** program (cited in the Section 9.1 application). It indicates that the fourth-quadrant critical point in the figure above has approximate coordinates (1.5708, –2.1416), and that the coefficient matrix of the associated linear system has the positive eigenvalue $\lambda_1 \approx 2.8949$ and the negative eigenvalue $\lambda_2 \approx -2.3241$. It therefore follows from Theorem 2 in Section 6.2 that this critical point is, indeed, a saddle point of the almost linear system
in (1).

With a general computer algebra system, you may have to do a bit of work yourself — or tell the computer precisely what to do — in order to find and classify a critical point. In the sections below, we illustrate this procedure using *Maple*, *Mathematica*, and MATLAB. Once the critical-point coordinates $a = 1.5708$, $b = -2.1416$ indicated above have been found, the substitution $x = u + a,\ y = v + b$ yields the translated system

$$\begin{aligned} \frac{du}{dt} &= (2.1416 - v)\cos(1.5708 - u - v) = f(u,v) \\ \frac{dv}{dt} &= (1.5708 + u)\cos(4.7124 + u - v) = g(u,v). \end{aligned} \tag{2}$$

If we substitute $u = v = 0$ in the Jacobian matrix

$$\mathbf{J} = \begin{bmatrix} \dfrac{\partial f}{\partial u} & \dfrac{\partial f}{\partial v} \\ \dfrac{\partial g}{\partial u} & \dfrac{\partial g}{\partial v} \end{bmatrix} \tag{3}$$

we get the coefficient matrix

$$\mathbf{A} = \begin{bmatrix} 2.1416 & 2.1416 \\ 1.5708 & -1.5708 \end{bmatrix} \tag{4}$$

of the linear system corresponding to the almost linear system in (2).

Alternatively, one can circumvent the translated system in (2) by looking at the Taylor expansions

$$\begin{aligned} f(x,y) &= D_x f(a,b)(x-a) + D_y f(a,b)(y-b) + \cdots \\ g(x,y) &= D_x g(a,b)(x-a) + D_y g(a,b)(y-b) + \cdots \end{aligned} \tag{5}$$

of the right-hand side functions in the original system (1), and retaining only the linear terms in this expansion. We see from (5) that

$$\mathbf{A} = \begin{bmatrix} D_x f(a,b) & D_y f(a,b) \\ D_x g(a,b) & D_y g(a,b) \end{bmatrix} \tag{6}$$

is the coefficient matrix of the linearization of the system (1) that results when we substitute $u = x - a,\ v = y - b$ and retain only the terms that are linear in u and v.

In any event, we can then use our computer algebra system to find the eigenvalues $\lambda_1 \approx 2.8949$ and $\lambda_2 \approx -2.3241$ of the matrix $\mathbf{A}$, thereby verifying that the critical point (1.5708, –2.1416) of (1) is, indeed, a saddle point.

Use a computer algebra system to find and classify similarly the other critical points of (1) indicated in the figure above. Then investigate similarly an almost linear system of your own construction. One convenient way to construct such a system is to start with a linear or almost linear system and insert sine or cosine factors resembling the ones in (1). For instance:

1. $x' = x \cos y$, $\quad y' = y \sin x$

2. $x' = -y + y^2 \cos y$, $\quad y' = -x - x^2 \sin x$

3. $x' = y \cos(2x + y)$, $\quad y' = -x \sin(x - 3y)$

4. $x' = -x - y^2 \cos(x + y)$, $\quad y' = y + x^2 \cos(x - y)$

Using *Maple*

After we enter the right-hand side functions in (1),

```
f := -y*cos(x+y-1):
g :=  x*cos(x-y-1):
```

we can proceed to solve numerically for a solution near (1.5, –2):

```
soln :=
fsolve({f=0,g=0}, {x,y}, x=1..2, y=-3..-1);
```

{x = 1.570796327, y = -2.141592654}

Thus our critical point (a, b) is given approximately by

```
a := rhs(soln[1]);
b := rhs(soln[2]);
```

a := 1.570796327
b := -2.141592654

To classify this critical point, we proceed to calculate first the partial derivatives

```
fx := evalf(subs(x=a,y=b,diff(f,x))):
fy := evalf(subs(x=a,y=b,diff(f,y))):
gx := evalf(subs(x=a,y=b,diff(g,x))):
gy := evalf(subs(x=a,y=b,diff(g,y))):
```

evaluated at (a, b), and then the Jacobian matrix in (6):

```
with(linalg):
A := matrix(2,2, [fx,fy,gx,gy]);
```

Finally, its eigenvalues are given by

```
eigenvals(A);
```

2.894893108, -2.324096781

Thus the eigenvalues $\lambda_1 \approx 2.8949$ and $\lambda_2 \approx -2.3241$ are real with opposite signs, so the critical point (1.5708, –2.1416) is, indeed, a saddle point of the system in (1).

Using *Mathematica*

After we enter the right-hand side functions in (1),

```
f = -y*Cos[x+y-1];
g =  x*Cos[x-y+1];
```

we can proceed to solve numerically for a solution near (1.5, –2):

```
soln =
FindRoot[{f == 0, g == 0}, {x,1.5}, {y,-2}]

{x -> 1.5708, y -> -2.14159}
```

Thus our critical point (a, b) is given approximately by

```
a = x /. soln
b = y /. soln

1.5708
-2.14159
```

To classify this critical point, we proceed to set up the Jacobian matrix in (6),

```
A = { {D[f,x], D[f,y]},
      {D[g,x], D[g,y]}} /. {x->a, y->b};
```

evaluated at (a, b), and calculate its eigenvalues

```
Eigenvalues[A]
{2.89489, -2.3241}
```

Thus the eigenvalues $\lambda_1 \approx 2.8949$ and $\lambda_2 \approx -2.3241$ are real with opposite signs, so the critical point (1.5708, –2.1416) is, indeed, a saddle point of the system in (1).

Using MATLAB

We want to solve the equations $f(x,y)=0,\ g(x,y)=0$ where f and g are the right-hand side functions in our system (1). However, the student edition of MATLAB does not include a function for the solution of systems of equations. Our strategy is therefore to use the MATLAB function **`fminsearch`** to minimize the function

$$h(x,y) = f(x,y)^2 + g(x,y)^2 = \left(-y\cos(x+y-1)\right)^2 + \left(x\cos(x-y+1)\right)^2$$

This function is defined as a function of the vector **`v = [x;y]`** by the m-file **`h.m`** consisting of the lines

```
function   z = h(v)
x = v(1);  y = v(2);
z = (-y*cos(x+y-1))^2 + (x*cos(x-y+1))^2;
```

Evidently a minimal point where $h(x,y)=0$ will be a critical point of the system in (1). Hence the commands

```
soln = fminsearch('h',[1.5;-2])
soln =
    1.5708
   -2.1416

a = soln(1);   b = soln(2);
```

yield the approximate critical point (1.5708, –2.1416). To classify this critical point, we proceed to set up the Jacobian matrix in (6). First we define the inline functions

```
f = '-y*cos(x+y-1)';
g = 'x*cos(x-y+1)';
```

calculate their partial derivatives

```
fx = diff(f,'x');    fy = diff(f,'y');
gx = diff(g,'x');    gy = diff(g,'y');
```

and evaluate these partial derivatives at the point (a, b):

```
syms x y
fx = subs(fx, {x,y}, {a,b});
fy = subs(fy, {x,y}, {a,b});
gx = subs(gx, {x,y}, {a,b});
gy = subs(gy, {x,y}, {a,b});
```

Then the eigenvalues of the Jacobian matrix

```
A = [fx  fy
     gx  gy];
```

are given by

```
eig(A)
ans =
    2.8948
   -2.3241
```

Thus the eigenvalues $\lambda_1 \approx 2.8948$ and $\lambda_2 \approx -2.3241$ are real with opposite signs, so the critical point (1.5708, –2.1416) is, indeed, a saddle point of the system in (1).