In[•]:=
```
(*Q1*)
```

In[•]:=
```
Exit[]
```

In[1]:=
```
sol = Quiet[DSolve[{P'[t] == r * P[t] * (1 - P[t]), P[0] == P0}, P[t], t]]
```

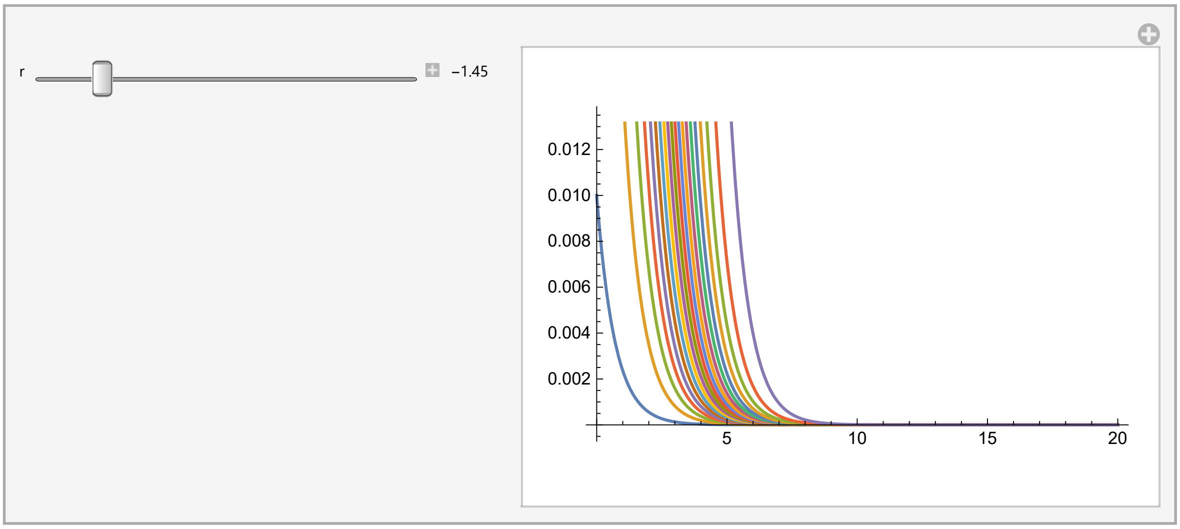
Out[1]= $\left\{\left\{P[t] \to \frac{e^{r t} P0}{1 - P0 + e^{r t} P0}\right\}\right\}$

In[2]:=
```
p[t_, p0_, r_] = Exp[r * t] * P0 / (1 - P0 + Exp[r * t] * P0)
```

Out[2]= $\frac{e^{r t} P0}{1 - P0 + e^{r t} P0}$

In[3]:=
```
Manipulate[Plot[Evaluate@Table[p[t, p0, r] /. P0 → n, {n, 0.01, 1, 0.05}], {t, 0, 20}],
 {r, -2, 2, Appearance → "Labeled"}]
```

Out[3]=



In[•]:=
```
Limit[sol[[1, 1, 2]] /. {P0 → 1, r → 2}, t → ∞]
```

Out[•]= 1

In[•]:=
```
(*Q2*)
```

In[•]:=
```
sol = Quiet[DSolve[{P'[t] == r * P[t] * (1 - P[t] / K), P[0] == P0}, P[t], t]]
```

Out[•]= $\left\{\left\{P[t] \to \frac{e^{r t} K P0}{K - P0 + e^{r t} P0}\right\}\right\}$

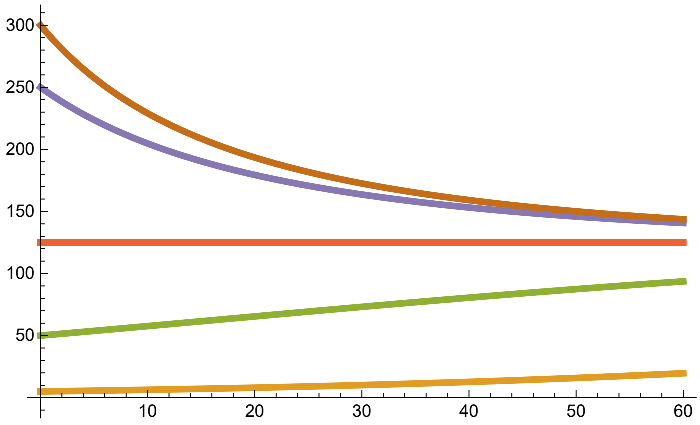
In[•]:=
```
K = 125; r = 0.025;
```

In[•]:=
```
data = Table[sol[[1, 1, 2]] /. P0 → n, {n, {n, 5, 50, 125, 250, 300}}]
```

Out[•]= $\left\{\frac{125\, e^{0.025 t}\, n}{125 - n + e^{0.025 t}\, n}, \frac{625\, e^{0.025 t}}{120 + 5\, e^{0.025 t}}, \frac{6250\, e^{0.025 t}}{75 + 50\, e^{0.025 t}}, 125., \frac{31\,250\, e^{0.025 t}}{-125 + 250\, e^{0.025 t}}, \frac{37\,500\, e^{0.025 t}}{-175 + 300\, e^{0.025 t}}\right\}$

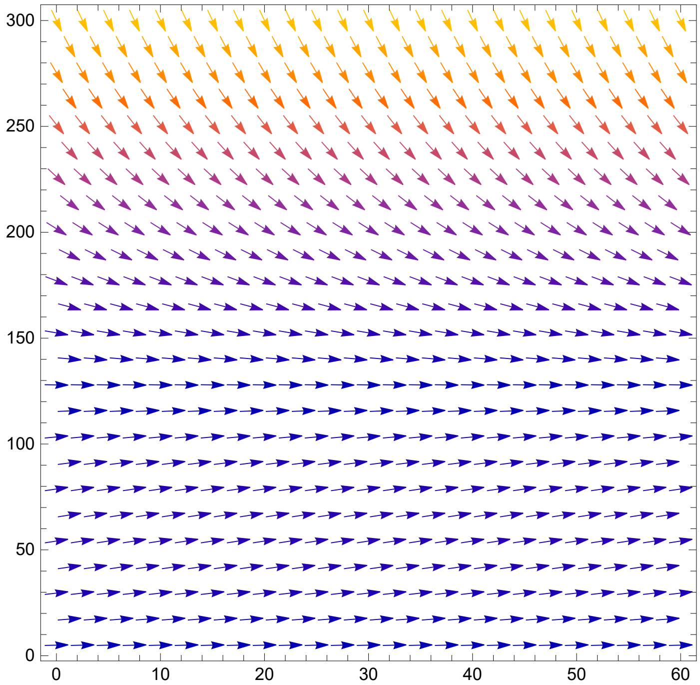
In[ ]:=
```
p1 = Plot[data, {t, 0, 60}, PlotStyle → Thickness[0.01]]
```

Out[ ]=



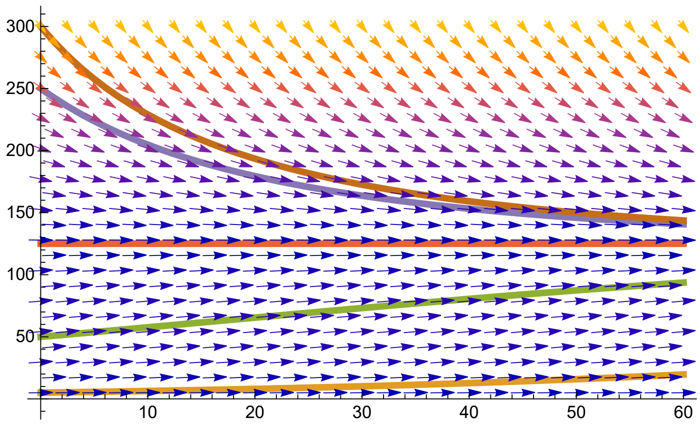
In[ ]:=
```
p2 = VectorPlot[{1, r * P * (1 - P / K)}, {t, 0, 60}, {P, 5, 300}, VectorPoints → Fine]
```

Out[ ]=



In[ ]:=
```
Show[p1, p2]
```

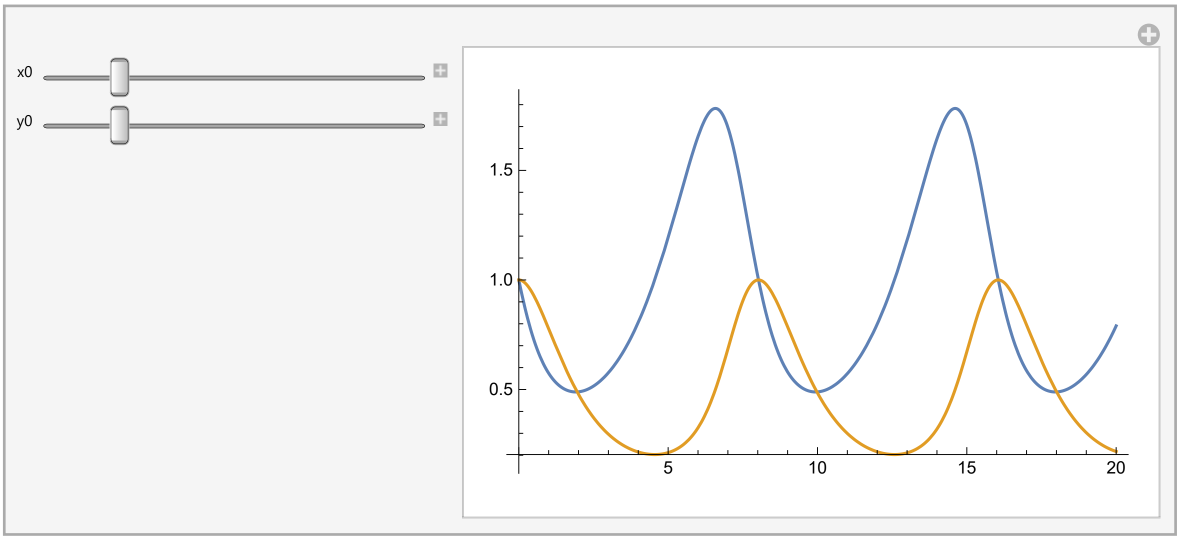
Out[ ]=



In[ ]:=
```
(*Q3*)
```

In[ ]:=
```
a = 2 / 3; b = 4 / 3; c = 1; d = 1;
```

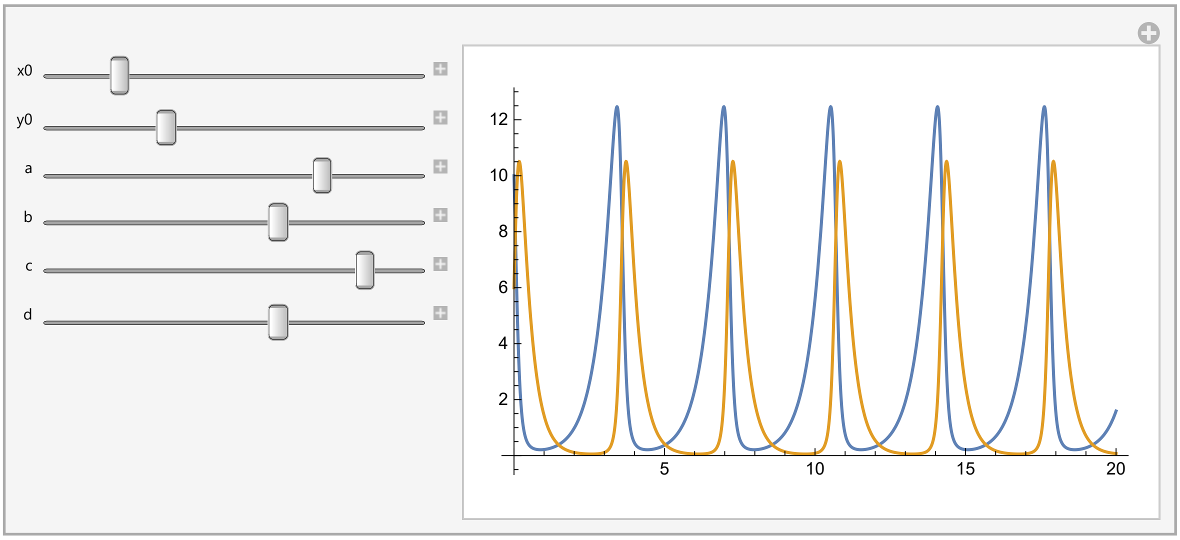
In[∘]:=
```
Manipulate[
 sol = NDSolve[{x'[t] == a * x[t] - b * x[t] * y[t], y'[t] == -c * y[t] + d * x[t] * y[t],
    x[0] == x0, y[0] == y0}, {x[t], y[t]}, {t, 0, 20}];
 Plot[Evaluate[{x[t], y[t]} /. sol], {t, 0, 20}], {{x0, 1}, 0.9, 1.5, .1},
 {{y0, 1}, 0.9, 1.5, 0.1}]
```

Out[∘]=


In[∘]:= (*Q4*)

In[∘]:= Exit[]

In[∘]:=
```
Manipulate[
 sol = NDSolve[{x'[t] == a * x[t] - b * x[t] * y[t], y'[t] == -c * y[t] + d * x[t] * y[t],
    x[0] == x0, y[0] == y0}, {x[t], y[t]}, {t, 0, 20}];
 Plot[Evaluate[{x[t], y[t]} /. sol], {t, 0, 20}], {{x0, 10}, 0, 60},
 {{y0, 6}, 0, 20}, {{a, 2}, -4, 4}, {{b, 1}, -4, 4}, {{c, 3}, -4, 4}, {{d, 1}, -4, 4}]
```

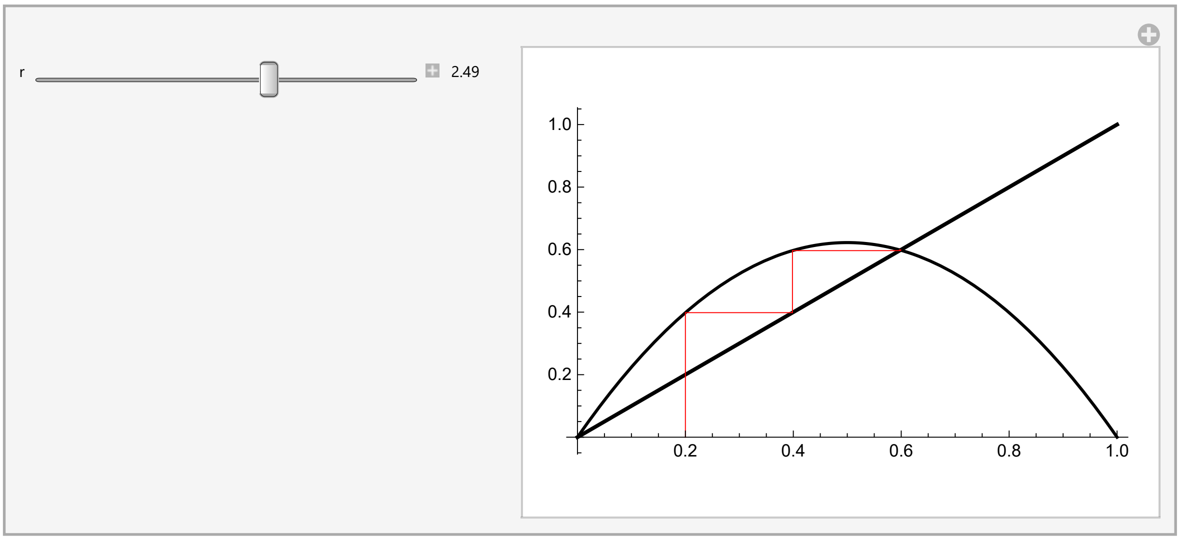
Out[∘]=


In[∘]:= (*Q5*)

In[∘]:= (*a*)

In[ ]:=
```
Manipulate[CobwebPlot[x ↦ r * x * (1 - x), 0.2, 10, {0, 1}],
 {{r, 2}, 0, 4, Appearance → "Labeled"}]
```

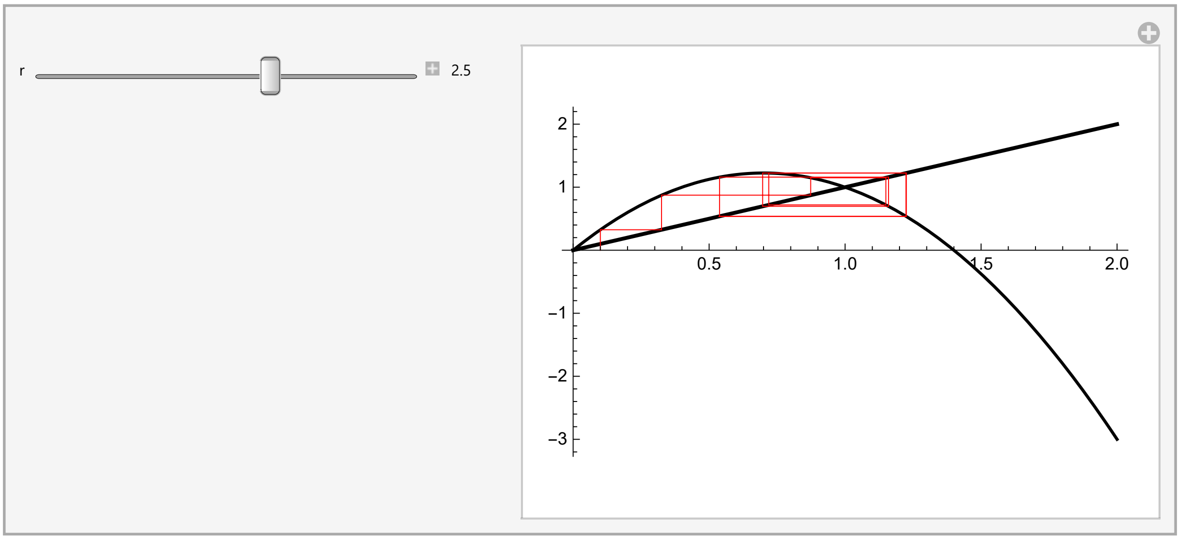
Out[ ]=



In[ ]:=
```
(*b*)
```

In[ ]:=
```
Manipulate[CobwebPlot[x ↦ (x + r * x * (1 - x)), 0.1, 10, {0, 2}],
 {{r, 2.5}, 0, 4, Appearance → "Labeled"}]
```

Out[ ]=

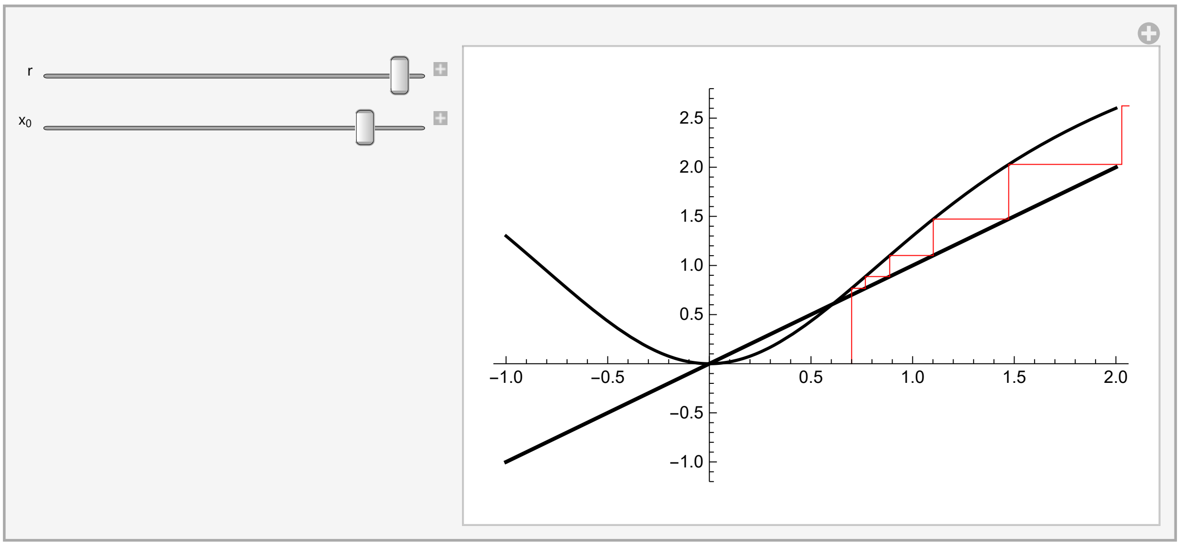


In[ ]:=
```
(*c*)
```

In[ ]:=
```
A = 2;
```

In[ ]:=
```
Manipulate[CobwebPlot[x ↦ (r * x^2) / (x^2 + 2), x0, 10, {-1, 2}],
 {{r, 3.9}, 0, 4}, {{x0, 0.7}, 0, 0.8}]
```

Out[ ]=



In[ ]:=
```
(*Q6*)
```

In[ ]:=
```
β = 0.5; γ = 0.0714;
```

In[ ]:=
```
sol = NDSolve[{s'[t] == -β * s[t] * i[t], i'[t] == β * s[t] * i[t] - γ * i[t], r'[t] == γ * i[t],
   s[0] == 0.9999, i[0] == 0.0001, r[0] == 0}, {s[t], i[t], r[t]}, {t, 0, 100}]
```

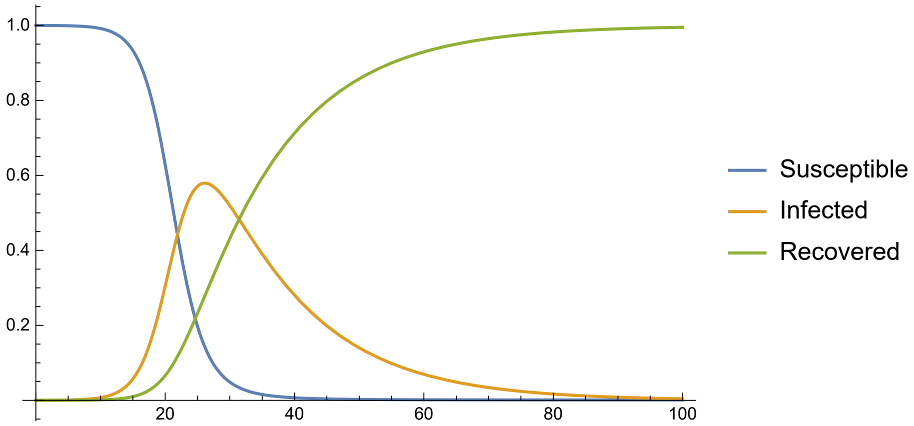
Out[ ]= {{s[t] → InterpolatingFunction[Domain: {{0., 100.}} Output: scalar][t],

i[t] → InterpolatingFunction[Domain: {{0., 100.}} Output: scalar][t],

r[t] → InterpolatingFunction[Domain: {{0., 100.}} Output: scalar][t]}}

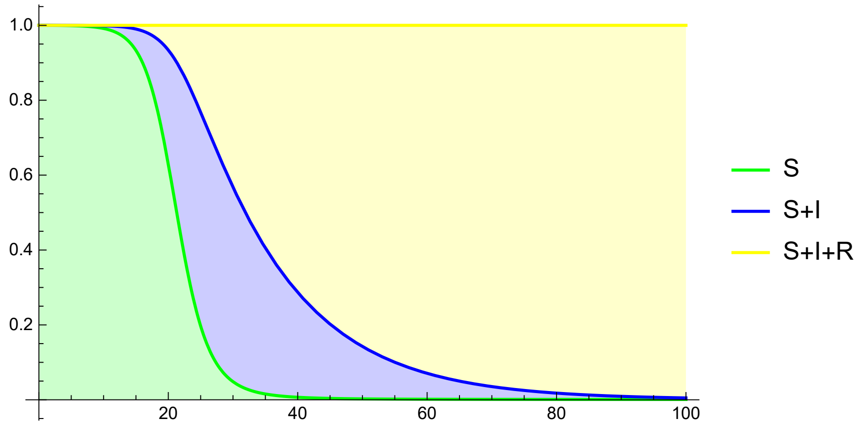
In[ ]:=
```
Plot[{sol[[1, 1, 2]], sol[[1, 2, 2]], sol[[1, 3, 2]]},
 {t, 0, 100}, PlotLegends → {"Susceptible", "Infected", "Recovered"}]
```

Out[ ]=



In[∘]:=
```
Plot[{sol[[1, 1, 2]], sol[[1, 1, 2]] + sol[[1, 2, 2]],
  sol[[1, 1, 2]] + sol[[1, 2, 2]] + sol[[1, 3, 2]]}, {t, 0, 100},
 Filling → {1 → Axis, 2 → {1}, 3 → {2}}, PlotStyle → {Green, Blue, Yellow},
 PlotLegends → {"S", "S+I", "S+I+R"}]
```

Out[∘]=



In[∘]:=
```
(*Q7*)
```

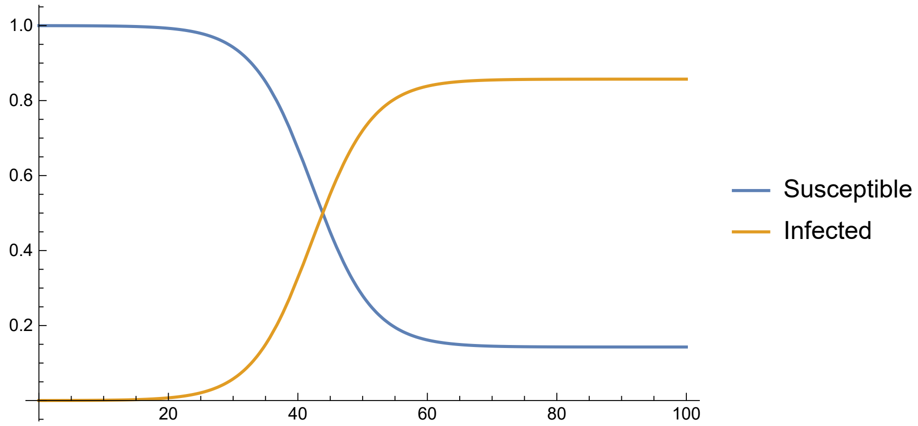
In[∘]:=
```
β = 0.25; γ = 0.0357;
```

In[∘]:=
```
sol = NDSolve[{s'[t] == -β * s[t] * i[t] + γ * i[t], i'[t] == β * s[t] * i[t] - γ * i[t],
   s[0] == 0.9999, i[0] == 0.0001}, {s[t], i[t]}, {t, 0, 100}]
```

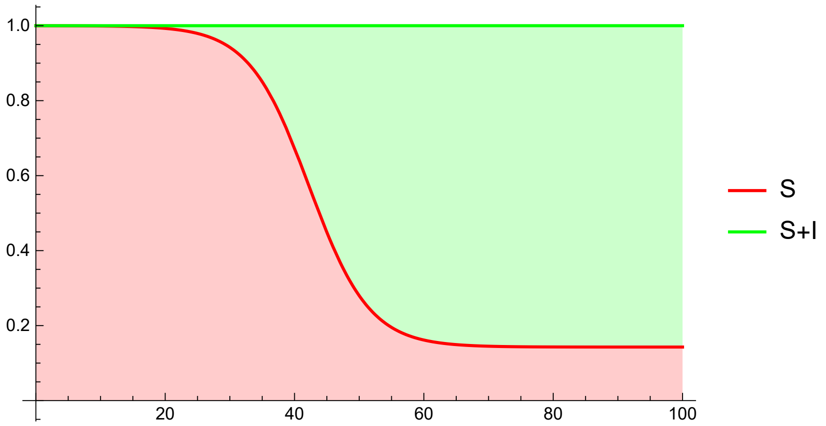
Out[∘]=
```
{{s[t] → InterpolatingFunction[ Domain: {{0., 100.}} Output: scalar ][t],
  i[t] → InterpolatingFunction[ Domain: {{0., 100.}} Output: scalar ][t]}}
```

In[∘]:=
```
Plot[{sol⟦1, 1, 2⟧, sol⟦1, 2, 2⟧}, {t, 0, 100}, PlotLegends → {"Susceptible", "Infected"}]
```

Out[∘]=



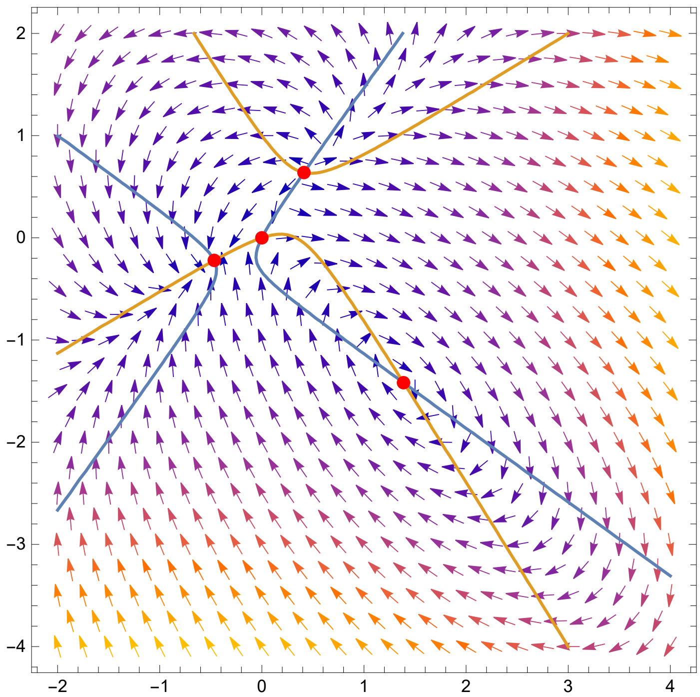
In[ ]:=
```
Plot[{sol[[1, 1, 2]], sol[[1, 1, 2]] + sol[[1, 2, 2]]}, {t, 0, 100},
 Filling → {1 → Axis, 2 → {1}}, PlotStyle → {Red, Green}, PlotLegends → {"S", "S+I"}]
```

Out[ ]=


In[ ]:=
```
(*Q8*)
```

In[ ]:=
```
(*i*)
```

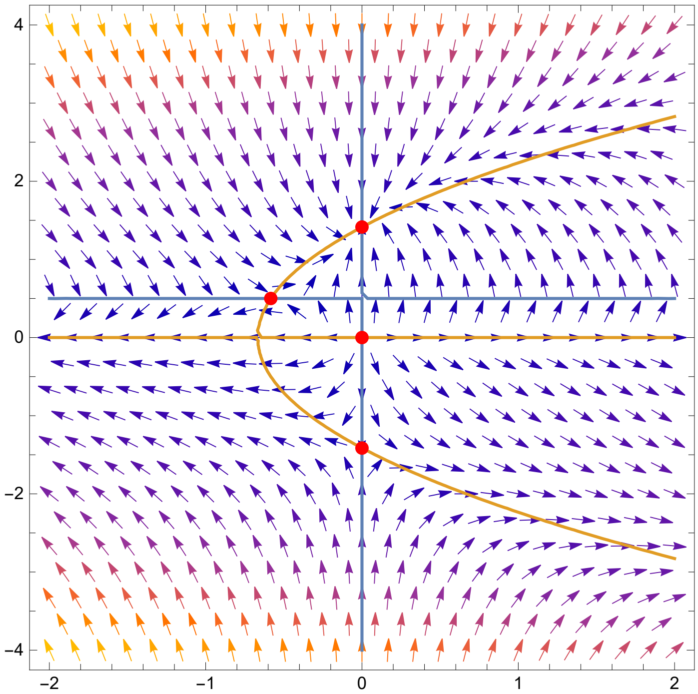
In[ ]:=
```
f[x_, y_] = 2 * x - y + 3 * (x^2 - y^2) + 2 * x * y;
g[x_, y_] = x - 3 * y - 3 * (x^2 - y^2) + 3 * x * y;
a = VectorPlot[{f[x, y], g[x, y]}, {x, -2, 4}, {y, -4, 2}, VectorPoints → Fine];
b = ContourPlot[{f[x, y] == 0, g[x, y] == 0}, {x, -2, 4}, {y, -4, 2}];
soln = NSolve[{f[x, y] == 0, g[x, y] == 0}, {x, y}];
Show[a, b, Graphics[{Red, PointSize[Large], Point[{x, y}] /. soln}]]
```

Out[ ]=


In[ ]:=
```
(*ii*)
```

In[∘]:=
```
f[x_, y_] = x - 2 * x * y;
g[x_, y_] = 2 * y - y^3 + 3 * x * y;
p1 = VectorPlot[{f[x, y], g[x, y]}, {x, -2, 2}, {y, -4, 4}, VectorPoints → Fine];
p2 = ContourPlot[{f[x, y] == 0, g[x, y] == 0}, {x, -2, 2}, {y, -4, 4}];
soln = NSolve[{f[x, y] == 0, g[x, y] == 0}, {x, y}];
Show[p1, p2, Graphics[{Red, PointSize[Large], Point[{x, y}] /. soln}]]
```

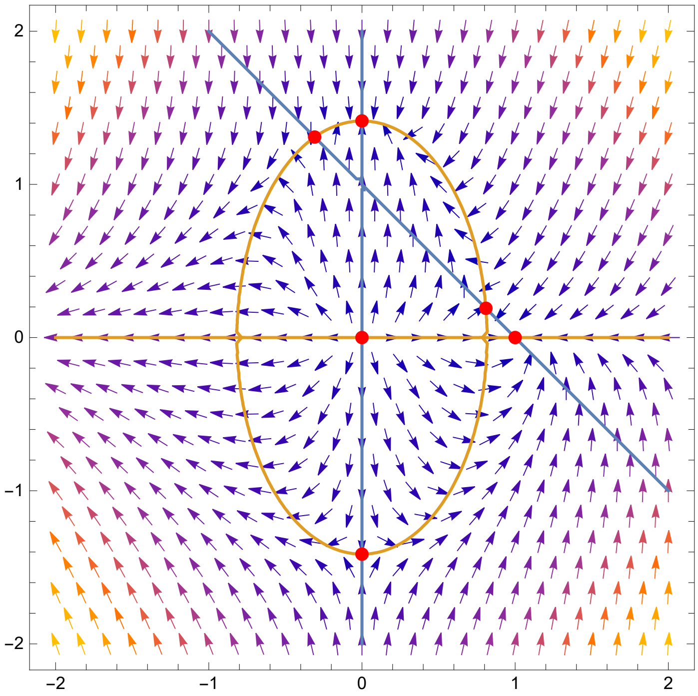
Out[∘]=



In[∘]:=
```
(*iii*)
```

In[∘]:=
```
f[x_, y_] = x - x^2 - x * y;
g[x_, y_] = 2 * y - y^3 - 3 * x^2 * y;
a = VectorPlot[{f[x, y], g[x, y]}, {x, -2, 2}, {y, -2, 2}, VectorPoints → Fine];
b = ContourPlot[{f[x, y] == 0, g[x, y] == 0}, {x, -2, 2}, {y, -2, 2}];
soln = NSolve[{f[x, y] == 0, g[x, y] == 0}, {x, y}];
Show[a, b, Graphics[{Red, PointSize[Large], Point[{x, y}] /. soln}]]
```

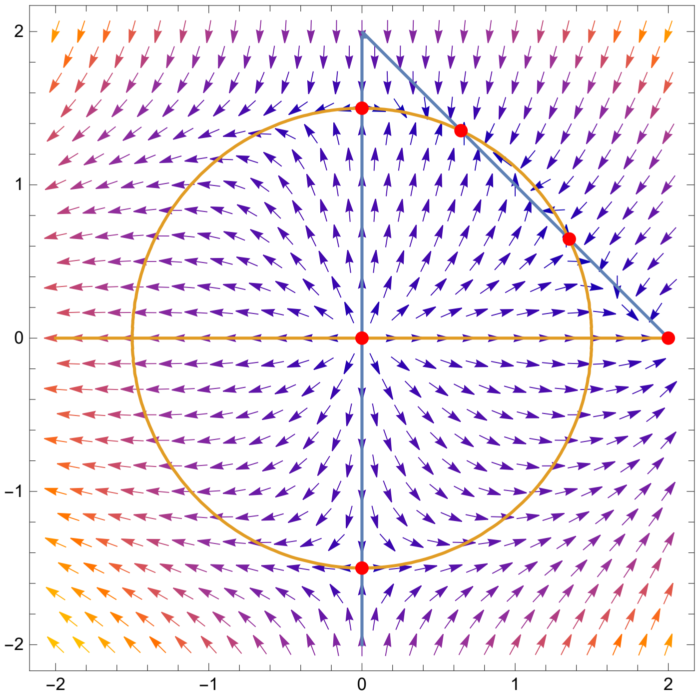
Out[∘]=



In[∘]:=
```
(*iv*)
```

In[ ]:=
```
f[x_, y_] = 2 * x - x^2 - x * y;
g[x_, y_] = y * (9 / 4 - y^2) - x^2 * y;
a = VectorPlot[{f[x, y], g[x, y]}, {x, -2, 2}, {y, -2, 2}, VectorPoints → Fine];
b = ContourPlot[{f[x, y] == 0, g[x, y] == 0}, {x, -2, 2}, {y, -2, 2}];
soln = NSolve[{f[x, y] == 0, g[x, y] == 0}, {x, y}];
Show[a, b, Graphics[{Red, PointSize[Large], Point[{x, y}] /. soln}]]
```

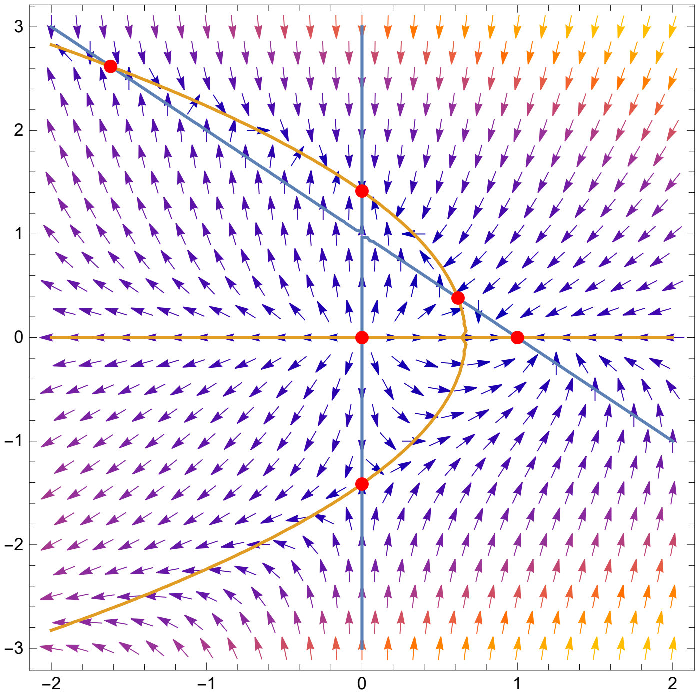
Out[ ]=



In[ ]:=
```
(*V*)
```

In[∘]:=
```
f[x_, y_] = x - x^2 - x * y;
g[x_, y_] = 2 * y - y^3 - 3 * x * y;
a = VectorPlot[{f[x, y], g[x, y]}, {x, -2, 2}, {y, -3, 3}, VectorPoints → Fine];
b = ContourPlot[{f[x, y] == 0, g[x, y] == 0}, {x, -2, 2}, {y, -3, 3}];
soln = NSolve[{f[x, y] == 0, g[x, y] == 0}, {x, y}];
Show[a, b, Graphics[{Red, PointSize[Large], Point[{x, y}] /. soln}]]
```

Out[∘]=



In[∘]:=
```
(*vi*)
```

In[∘]:=
```
f[x_, y_] = 5 * (y + x - x^3 / 3);
g[x_, y_] = 0.2 * (x + 0.7 - 0.5 * y);
a = VectorPlot[{f[x, y], g[x, y]}, {x, -4, 4}, {y, -7, 8}, VectorPoints → Fine];
b = ContourPlot[{f[x, y] == 0, g[x, y] == 0}, {x, -4, 4}, {y, -7, 8}];
soln = NSolve[{f[x, y] == 0, g[x, y] == 0}, {x, y}];
Show[a, b, Graphics[{Red, PointSize[Large], Point[{x, y}] /. soln}]]
```

Out[∘]=