ME 406
Assignment # 3 Solutions

```
In[552]:=
      sysid

      Mathematica  6.0.3, DynPac 11.02, 2/10/2009
```

```
In[553]:=
      plotreset; intreset;
```

Problem 1

The equations of the problem are

$$\frac{dx}{dt} = a\,y\,, \quad \frac{dy}{dt} = -x - 2\,y.$$

The eigenvalue equation is obtained in the usual way as

$$\lambda^2 + 2\,\lambda + a = 0\ .$$

We define the problem for DynPac, in preparation for the numerical part of the work.

```
In[554]:=
      setstate[{x, y}];
```

```
In[555]:=
      setparm[{a}];
```

```
In[556]:=
      slopevec = {a y, -x - 2 y};
```

Part (a)

We set the parameter value.

```
In[557]:=
      parmval = {-3};
```

(1) The eigenvalue equation in this case is $\lambda^2 + 2\lambda - 3 = 0$, with roots $\lambda = -3, 1$, so the equilibrium is a saddle and therefore unstable. We can check this with DynPac.

```
In[558]:=
      classify2D[{0, 0}];

      unstable - saddle
```

(2) To find the straight line solutions, we try $y = kx$ in the equations, which gives

$$\frac{\dot{y}}{\dot{x}} = k = \frac{-x-2\,y}{-3\,y} = \frac{1+2\,k}{3\,k}.$$

We solve the quadratic for k to get $k = -1/3, 1$. Thus the vectors defining the straight line solutions are {3, -1} and {1,1}. To find the time dependence, we use either of the original differential equations. Using the first equation and the first value of k, we get $\dot{x} = a\,y = a\,k\,x = x$, so $x = \text{const}\,e^t$, and the associated solution is $\{3,-1\}e^t$. In the same way, the second straight line solution is $\{1,1\}e^{-3\,t}$. A more direct way to get these results is to calculate the eigenvalues and eigenvectors:

```
In[559]:=
      eigsys[{0, 0}]

Out[559]=
      {{-3, 1}, {{1, 1}, {-3, 1}}}
```

(3) The general solution is given by

$$\begin{pmatrix} x \\ y \end{pmatrix} = C_1 \begin{pmatrix} 3 \\ -1 \end{pmatrix} e^{t} + C_2 \begin{pmatrix} 1 \\ 1 \end{pmatrix} e^{-3\,t}.$$

We impose the given initial conditions to get

$$\begin{pmatrix} 1 \\ 1 \end{pmatrix} = C_1 \begin{pmatrix} 3 \\ -1 \end{pmatrix} + C_2 \begin{pmatrix} 1 \\ 1 \end{pmatrix}.$$

We solve these equations for $C_1$and C_2 to get $C_1 = 0$ and $C_2 = 1$, hence

$$\begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} 1 \\ 1 \end{pmatrix} e^{-3\,t}.$$

(4) We set values for the initial condition, the initial time, the time step, and the number of time steps, and then integrate.

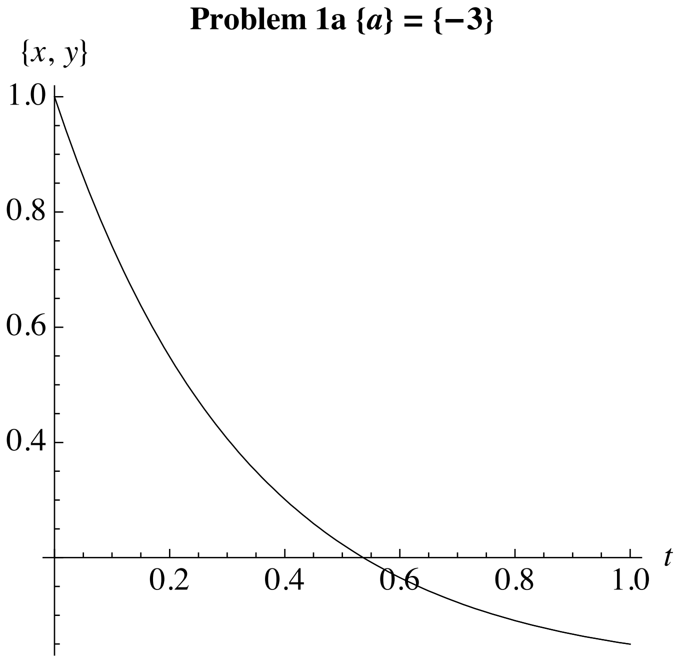
```
In[560]:=
      initvec = {1, 1}; t0 = 0.0; h = 0.01; nsteps = 100;

In[561]:=
      sysname = "Problem 1a";

In[562]:=
      sol1 = integrate[initvec, t0, h, nsteps];
```

Now we graph the results.

```
In[563]:=
      imsize = 250;
```

```
In[564]:=
      timeplot[sol1, {1, 2}]
```

```
Out[564]=
```

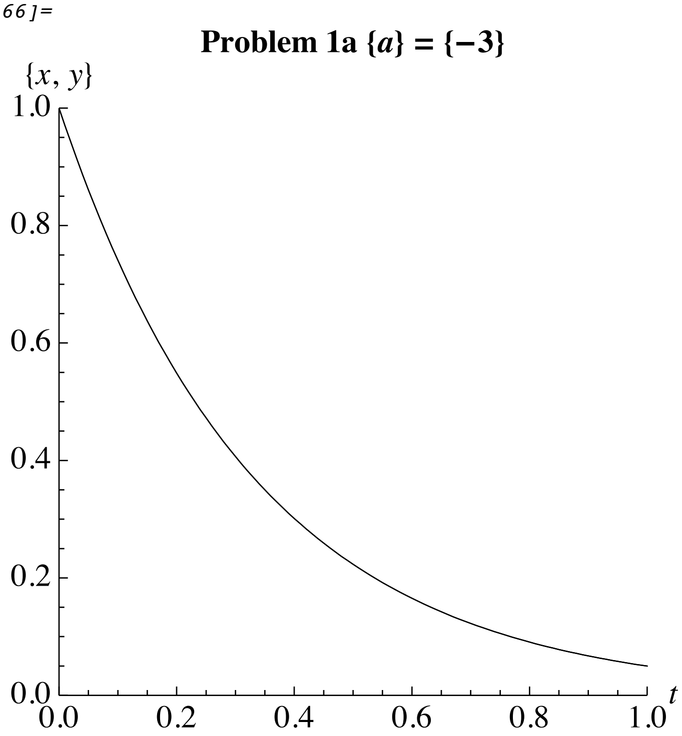


Mathematica has chosen a window and axes origin which are not optimal. We specify these and plot again.

```
In[565]:=
      plrange = {{0, 1}, {0, 1}}; axeorg = {0, 0};
```

```
In[566]:=
      timeplot[sol1, {1, 2}]
```

```
Out[566]=
```



There is only one curve because $x = y$ for this solution.

(5) We use the routine portrait to construct the phase portrait. We set a plotting window, and we use range checking to prevent runaways. We choose a set of initial conditions which includes points on the two eigenvectors and which also includes the particular case {1,1} looked at above. We also integrate both directions in time. We put a single arrow on each curve.

```
In[567]:=
      bothdirflag = True; plrange = {{-3, 3}, {-3, 3}}; rangeflag = True; ranger = plrange;
```

```
In[568]:=
      h = 0.02; nsteps = 200; t0 = 0.0;
```

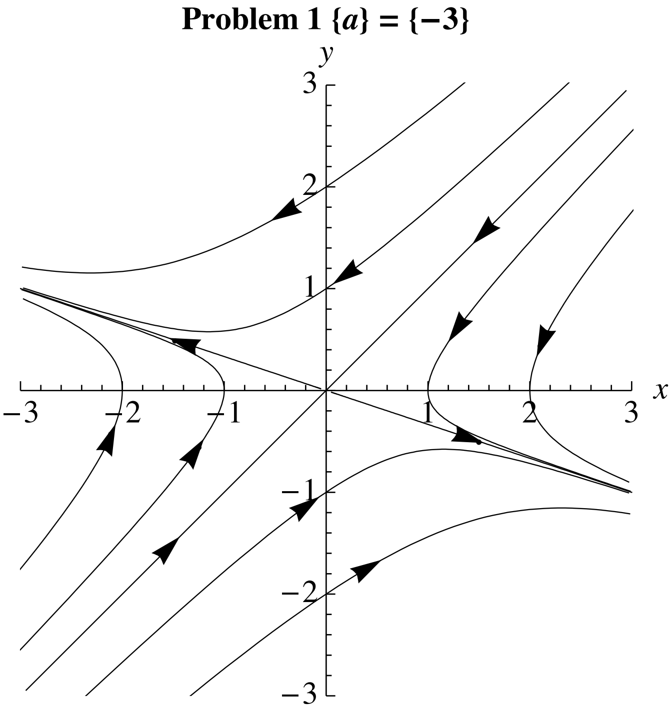
```
In[569]:=
      intlist = {{1, 1}, {-1, -1}, {3, -1}, {-3, 1},
         {1, 0}, {-1, 0}, {0, 1}, {0, -1}, {2, 0}, {-2, 0}, {0, 2}, {0, -2}};
```

```
In[570]:=
      sysname = "Problem 1";
```

```
In[571]:=
      arrowflag = True; arrowvec = {1 / 2};
```

```
In[572]:=
      portrait[intlist, t0, h, nsteps, 1, 2]
```

```
Out[572]=
```



Part (b)

We set the parameter value.

```
In[573]:=
      parmval = {3 / 4};
```

(1) The eigenvalue equation in this case is $\lambda^2 + 2\lambda + 3/4 = 0$, with roots $\lambda = -3/2, -1/2$, so the equilibrium is a strictly stable node. We check this with DynPac.

```
In[574]:=
      classify2D[{0, 0}]

      strictly stable - node
```

(2) There will be two straight line solutions. To find them we try $y = kx$ in the equations, which gives

$$\frac{\dot{y}}{\dot{x}} = k = \frac{-x - 2y}{(3/4)\,y} = \frac{-4 - 8k}{3k}.$$

We solve the quadratic for k to get $k = -2, -2/3$. Thus the vectors defining the straight line solutions are {1, -2} and {3,-2}. To find the time dependence, we use either of the original differential equations. Using the first equation and the first value of k, we get $\dot{x} = a\,y = a\,k\,x = -(3/2)\,x$, so $x = \text{const}\; e^{-(3/2)\,t}$, and the associated solution is $\{1,-2\}e^{-(3/2)\,t}$. In the same way, the second straight line solution is $\{3,-2\}e^{-t/2}$. A more direct way to get these results is to calculate the eigenvalues and eigenvectors:

```
In[575]:=
      eigsys[{0, 0}]
```

```
Out[575]=
          3     1         1          3
      {{- -, - -}, {{- -, 1}, {- -, 1}}}
          2     2         2          2
```

(3) This time we use DSolve to get the analytical solution to the initial value problem.

```
In[576]:=
      DSolve[{x'[t] == (3 / 4) y[t], y'[t] == -x[t] - 2 y[t], x[0] == 1, y[0] == 1}, {x[t], y[t]}, t]
```

```
Out[576]=
                1   -3 t/2                        1   -3 t/2
      {{x[t] -> - ⅇ       (-5 + 9 ⅇ^t), y[t] -> - - ⅇ       (-5 + 3 ⅇ^t)}}
                4                                 2
```

(4) We set values for the initial condition, the initial time, the time step, and the number of time steps, and then integrate, after turning off range checking and integration in both directions in time.

```
In[577]:=
      rangeflag = False; bothdirflag = False;
```

```
In[578]:=
      initvec = {1, 1}; t0 = 0.0; h = 0.01; nsteps = 200;
```

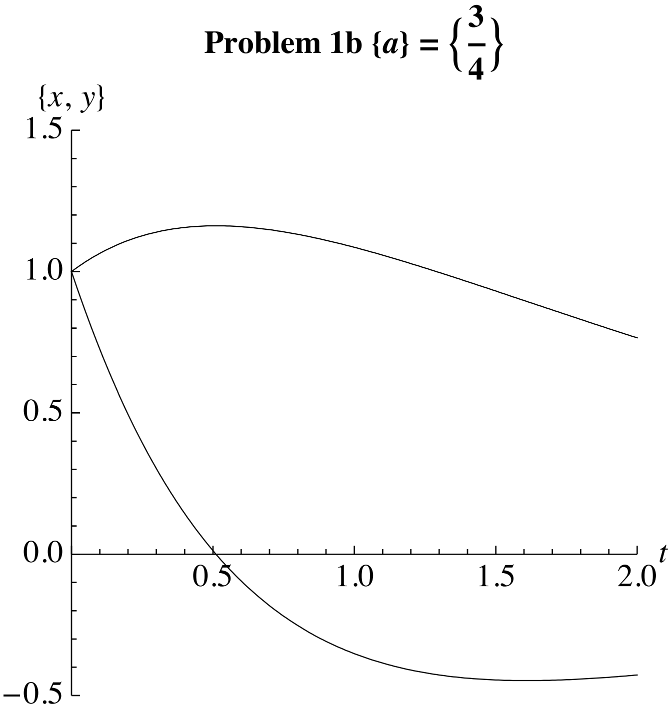
```
In[579]:=
      sol2 = integrate[initvec, t0, h, nsteps];
```

Now we graph the results, after setting the plotting window.

```
In[580]:=
      plrange = {{0, 2}, {-0.5, 1.5}};
```

```
In[581]:=
      sysname = "Problem 1b";
```

```
In[582]:=
      timeplot[sol2, {1, 2}]

Out[582]=
```



Before we construct a graph from a solution, how can we know what plotting range to choose? There is a helpful function called staterange, which scans the solution list and returns the minimum and maximum values.

```
In[583]:=
      staterange[sol2]

Out[583]=
      {{x, {0.765495, 2.}, {1.16189, 0.51}}, {y, {-0.447214, 1.61}, {1, 0.}}}
```

From here we see that *x* has a minimum value of 0.765494 which occurs at time 2, and a maximum value of 1.16189 which occurs at time 0.51. Similarly *y* has a minimum of -0.447214 at time 1.61 and a maximum of 1 at the initial time 0.

(5) We use the routine portrait to construct the phase portrait. We set a plotting window, and we use range checking to prevent runaways. We choose a set of initial conditions which includes points on the two eigenvectors and which also includes the particular case {1,1} looked at above. We also integrate both directions in time. We put a single arrow on each curve.

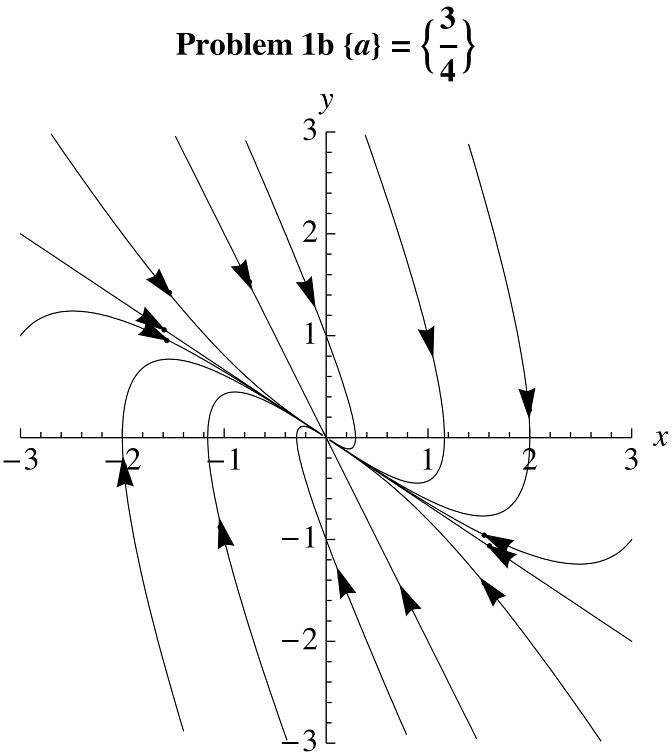
```
In[584]:=
      bothdirflag = True; plrange = {{-3, 3}, {-3, 3}}; rangeflag = True; ranger = plrange;

In[585]:=
      h = 0.02; nsteps = 300; t0 = 0.0;

In[586]:=
      intlist = {{1, 1}, {-1, -1}, {3, -1}, {-3, 1}, {-2, 2}, {2, -2},
         {0, 1}, {0, -1}, {2, 0}, {-2, 0}, {-1, 2}, {1, -2}, {-3, 2}, {3, -2}};
```

```
In[587]:=
      sysname = "Problem 1b";
```

```
In[588]:=
      arrowflag = True; arrowvec = {1 / 2};
```

```
In[589]:=
      portrait[intlist, t0, h, nsteps, 1, 2]
```

```
Out[589]=
```



▪ Part (c)

We set the parameter value.

```
In[590]:=
      parmval = {5};
```

(1) The eigenvalue equation in this case is $\lambda^2 + 2\lambda + 5 = 0$, with roots $\lambda = -1 \pm 2i$, so the equilibrium is a strictly stable spiral. We check this with DynPac.

```
In[591]:=
      classify2D[{0, 0}]

      strictly stable - spiral
```

```
In[592]:=
      eigsys[{0, 0}]

Out[592]=
      {{-1 + 2 ⅈ, -1 - 2 ⅈ}, {{-1 - 2 ⅈ, 1}, {-1 + 2 ⅈ, 1}}}
```

(2) There are no straight line solutions for a spiral.

(3) Again we use DSolve to get the analytical solution to the initial value problem.

```
In[593]:=
      ans = DSolve[{x'[t] == 5 y[t], y'[t] == -x[t] - 2 y[t], x[0] == 1, y[0] == 1}, {x[t], y[t]}, t]

Out[593]=
      {{x[t] → ⅇ^-t (Cos[2 t] + 3 Sin[2 t]), y[t] → ⅇ^-t (Cos[2 t] - Sin[2 t])}}
```

(4) We set values for the initial condition, the initial time, the time step, and the number of time steps, and then integrate, after turning off range checking and integration in both directions in time.

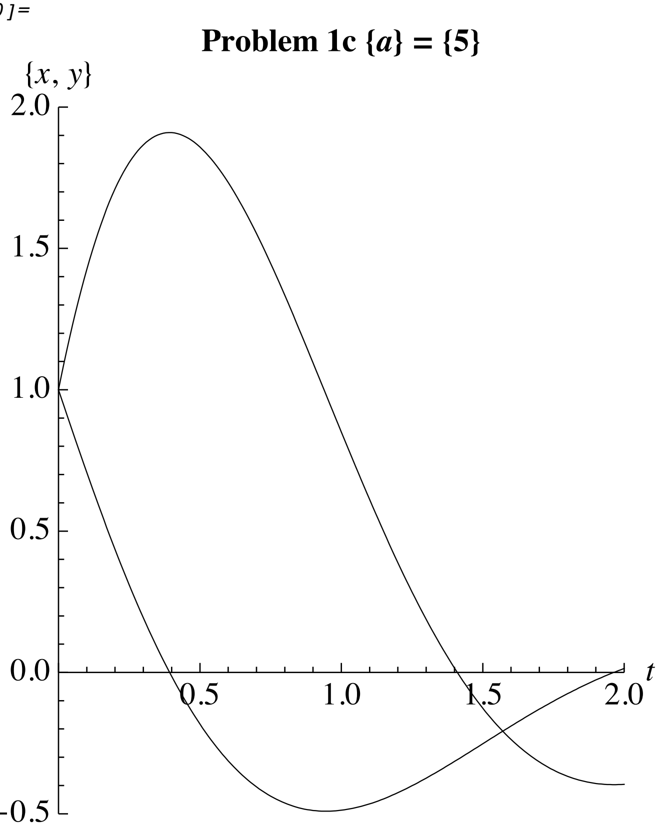
```
In[594]:=
      rangeflag = False; bothdirflag = False;

In[595]:=
      initvec = {1, 1}; t0 = 0.0; h = 0.01; nsteps = 200;

In[596]:=
      sol3 = integrate[initvec, t0, h, nsteps];
```

Now we graph the results, after checking the state range in order to set the plotting window.

```
In[597]:=
      staterange[sol3]

Out[597]=
      {{x, {-0.397005, 1.96}, {1.90981, 0.39}}, {y, {-0.491002, 0.95}, {1, 0.}}}

In[598]:=
      plrange = {{0, 2}, {-0.5, 2}};

In[599]:=
      sysname = "Problem 1c";
```

```
In[600]:=
      timeplot[sol3, {1, 2}]
```

```
Out[600]=
```



(5) We use the routine portrait to construct the phase portrait. We set a plotting window, and we use range checking to prevent runaways. We choose a set of initial conditions which includes points on the two eigenvectors and which also includes the particular case {1,1} looked at above. We also integrate both directions in time. We put a single arrow on each curve.

```
In[601]:=
      bothdirflag = True; plrange = {{-3.3, 3.3}, {-3.3, 3.3}};
      rangeflag = True; ranger = plrange;
```

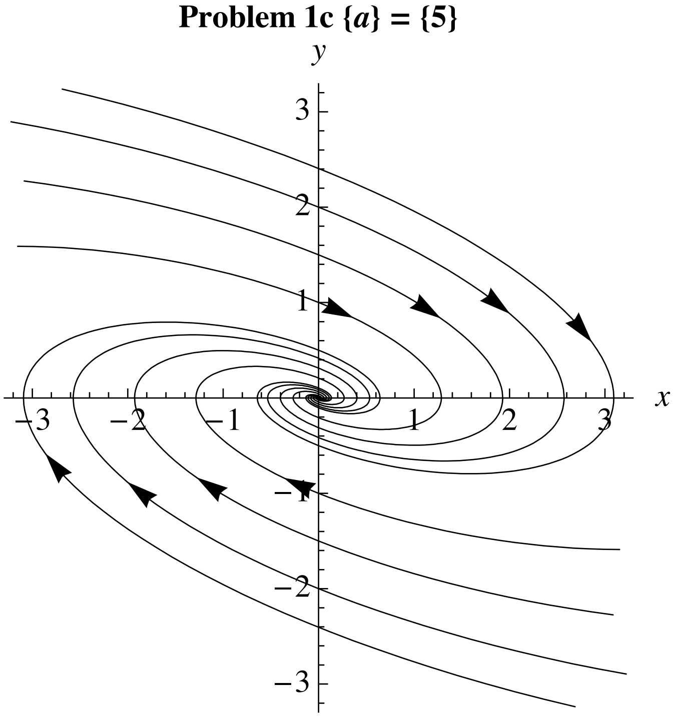
```
In[602]:=
      h = 0.02; nsteps = 300; t0 = 0.0;
```

```
In[603]:=
      intlist = {{0, 0.5}, {0, 1}, {0, 1.5}, {0, 2}, {0, -0.5}, {0, -1}, {0, -1.5}, {0, -2}};
```

```
In[604]:=
      sysname = "Problem 1c";
```

```
In[605]:=
      arrowflag = True; arrowvec = {1 / 2};
```

```
In[606]:=
      portrait[intlist, t0, h, nsteps, 1, 2]

Out[606]=
```



■ Problem 2

The given equations are

$$\dot{R} = a R + b J , \quad \dot{J} = b R + a J .$$

The eigenvalue equation is $(a - \lambda)^2 - b^2 = 0$, so $\lambda = a \pm b$. If $|b| < |a|$ both eigenvalues have the algebraic sign of a. Hence if $a > 0$ the equilibrium is unstable and all solutions run off to infinity. This is either bless or hell, depending on whether R and J are positive or negative, and that in turn will depend on the initial conditions. If $a < 0$ (still assuming $|b| < |a|$), then both eigenvalues are negative and the equilibrium is strictly stable, so anything that starts will end up at {0,0}, which in this context reasonably can be called boredom. If $|b| > |a|$, then the eigenvalues will have opposite signs, so that the equilibrium is a saddle point. Thus all solutions except those on the stable manifold run off to infinity -- again either bliss or hell, depending on the algebraic sign. Initial points on the stable manifold go to the state of boredom {0,0}, but are unstable -- any slight quarrel or other event will send the solution off to infinity.

We can be a little more precise if we look at the detailed solution. The eigenvector associated with $\lambda = a + b$ is easily shown to be {1,1}, and the eigenvector associated with $\lambda = a - b$ is {1,-1}. The general solution is then

$$C_1 \begin{pmatrix} 1 \\ 1 \end{pmatrix} e^{(a+b)\,t} + C_2 \begin{pmatrix} 1 \\ -1 \end{pmatrix} e^{(a-b)\,t} .$$

We impose the initial conditions $R(0) = R_0$, $J(0) = J_0$ to get

$$\begin{pmatrix} R\ (t) \\ J\ (t) \end{pmatrix} = \frac{1}{2}\ (R_0 + J_0)\ \begin{pmatrix} 1 \\ 1 \end{pmatrix} e^{(a+b)\ t} + \frac{1}{2}\ (R_0 - J_0)\ \begin{pmatrix} 1 \\ -1 \end{pmatrix} e^{(a-b)\ t}\ .$$

As $t \longrightarrow \infty$, the solution with the algebraically larger exponent will dominate. With the notation $\epsilon = b/|b|$, we can write the asymptotic solution as

$$\begin{pmatrix} R \\ J \end{pmatrix} \underset{t\to\infty}{\sim} \frac{1}{2}\ (R_0 + \epsilon J_0)\ \begin{pmatrix} 1 \\ \epsilon \end{pmatrix} e^{(a+|b|)\ t}\ .$$

Now we can make some simple precise statements. If $a < -|b|$, then the solution decays and the final state is the boredom at {0,0}. If $a > -|b|$, the state goes off to infinity as $t \to \infty$. This case has two sub-cases. If $b > 0$, then there is bliss for both if $R_0 + J_0 > 0$, and misery for both if $R_0 + J_0 < 0$. If $b < 0$, then in the asymptotic state, R and J always have opposite signs, so there is bliss for one, misery for the other. Specifically, for $R_0 > J_0$, there is bliss for Romeo, misery for Juliet. For $R_0 < J_0$, there is bliss for Juliet and misery for Romeo. As an exercise you might want to look at various boundary states such as $a = b$, $a = -b$, $R_0 = J_0$.

▪ Part (a)

We take $a = 1$, $b = 2$. According to the analysis above, the final state is one of bliss for both or misery for both, depending on the initial conditions. We construct the phase plane portrait and see if it is consistent with this. We start by defining the system for DynPac.

```
In[607]:=
      setstate[{R, J}];
```

```
In[608]:=
      setparm[{a, b}];
```

```
In[609]:=
      slopevec = {a R + b J, b R + a J};
```

```
In[610]:=
      parmval = {1, 2};
```

```
In[611]:=
      classify2D[{0, 0}]

      unstable - saddle
```

We integrate both directions in time, using the following set of initial conditions:

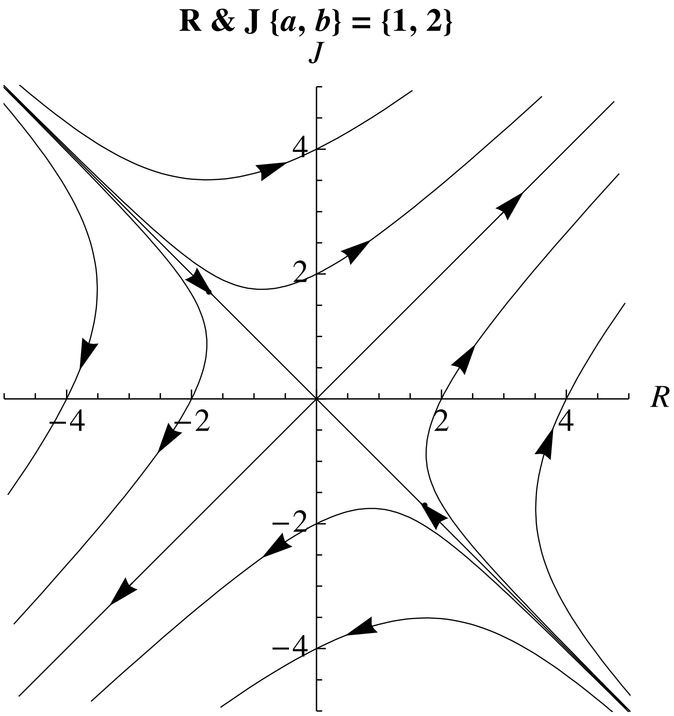
```
In[612]:=
      initset = {{2, 0}, {1, 1}, {0, 2}, {-1, 1}, {-2, 0},
         {-1, -1}, {0, -2}, {1, -1}, {4, 0}, {-4, 0}, {0, 4}, {0, -4}};
```

```
In[613]:=
      bothdirflag = True; rangeflag = True;
```

```
In[614]:=
      plrange = {{-5, 5}, {-5, 5}}; ranger = plrange;
```

```
In[615]:=
      t0 = 0.0; h = 0.02; nsteps = 300;
```

```
In[616]:=
      sysname = "R & J";

In[617]:=
      arrowflag = True; arrowvec = {2 / 3};

In[618]:=
      portrait[initset, t0, h, nsteps, 1, 2]

Out[618]=
```



This is consistent with our analysis. Initial points starting with $R+J>0$ give asymptotic states with $R=J$ and both positive. Initial points starting with $R+J<0$ give asympotic states with $R=J$ and both negative.

▪ Part (b)

We take $a=2, b=1$. According to the analysis above, the final state is one of bliss for both or misery for both, depending on the initial conditions. We construct the phase plane portrait and see if it is consistent with this.

```
In[619]:=
      parmval = {2, 1};

In[620]:=
      classify2D[{0, 0}]

      unstable - node
```

We integrate both directions in time, using the following set of initial conditions:

```
In[621]:=
      initset = {{2, 0}, {1, 1}, {0, 2}, {-1, 1}, {-2, 0},
         {-1, -1}, {0, -2}, {1, -1}, {4, 0}, {-4, 0}, {0, 4}, {0, -4}};
```

In[622]:=

```
bothdirflag = True; rangeflag = True;
```

In[623]:=

```
plrange = {{-5, 5}, {-5, 5}}; ranger = plrange;
```

In[624]:=

```
t0 = 0.0; h = 0.02; nsteps = 300;
```

In[625]:=

```
sysname = "R & J";
```

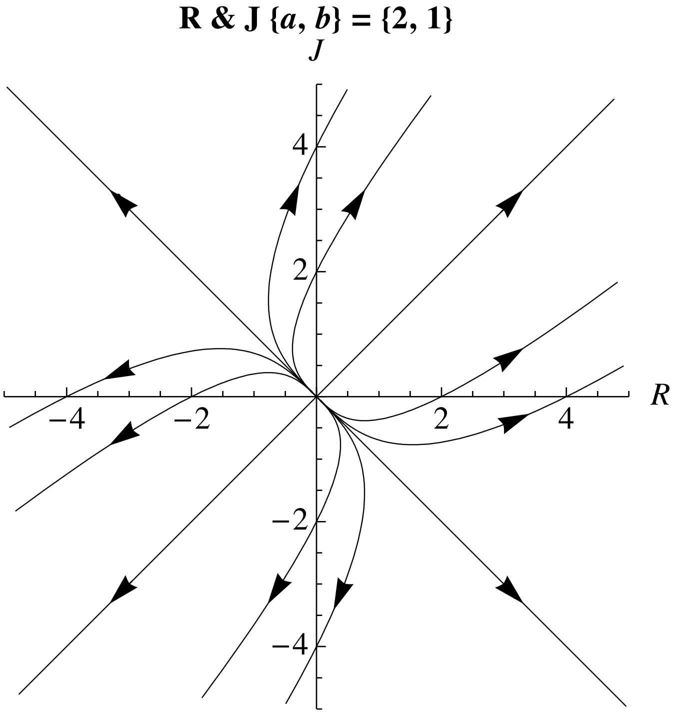
In[626]:=

```
arrowflag = True; arrowvec = {2 / 3};
```

In[627]:=

```
portrait[initset, t0, h, nsteps, 1, 2]
```

Out[627]=



This is consistent with our analysis. Initial points starting with $R + J > 0$ give asymptotic states with $R = J$ and both positive. Initial points starting with $R + J < 0$ give asympotic states with $R = J$ and both negative.

Part (c)

We take $a = -2$, $b = 1$. According to the analysis above, the final state is one of boredom for any initial condition. We construct the phase plane portrait and see if it is consistent with this.

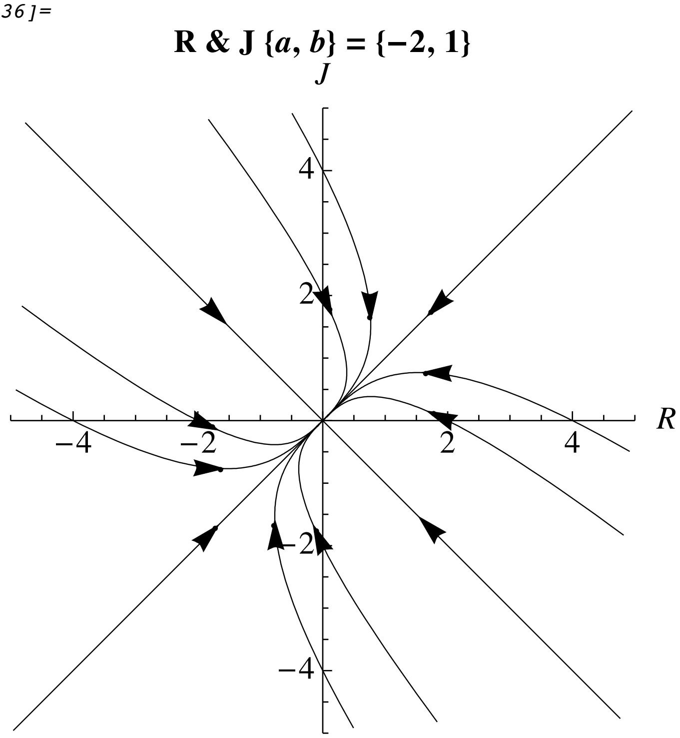
In[628]:=

```
parmval = {-2, 1};
```

```
In[629]:=
      classify2D[{0, 0}]

      strictly stable - node
```

We integrate both directions in time, using the following set of initial conditions:

```
In[630]:=
      initset = {{2, 0}, {1, 1}, {0, 2}, {-1, 1}, {-2, 0},
         {-1, -1}, {0, -2}, {1, -1}, {4, 0}, {-4, 0}, {0, 4}, {0, -4}};

In[631]:=
      bothdirflag = True; rangeflag = True;

In[632]:=
      plrange = {{-5, 5}, {-5, 5}}; ranger = plrange;

In[633]:=
      t0 = 0.0; h = 0.02; nsteps = 300;

In[634]:=
      sysname = "R & J";

In[635]:=
      arrowflag = True; arrowvec = {2 / 3};

In[636]:=
      portrait[initset, t0, h, nsteps, 1, 2]

Out[636]=
```



This is consistent with our analysis. Any initial point ultimately leads to the boredom of the origin.