Differential Equations&*Mathematica*

Produced by Bruce Carpenter Published by Math Everywhere, Inc.
www.matheverywhere.com

DE.07 Eigenvectors and Eigenvalues for Linear Systems
Basics

B.1) Linear systems are diffeq systems you can make with a coefficient matrix of numbers

□B.1.a.i)

Here's a diffeq system:

```
Clear[x, y, t]
diffeqsystem =
 ({x'[t], y'[t]} == {2.9 x[t] + 0.4 y[t], -1.4 x[t] + 0.6 y[t]});
ColumnForm[Thread[diffeqsystem]]
```

$$x'[t] == 2.9\,x[t] + 0.4\,y[t]$$
$$y'[t] == -1.4\,x[t] + 0.6\,y[t]$$

How do you tell that this diffeq system is a linear system?

□Answer:

You can tell it's a linear system because you can make it with a matrix.

Here's how:

Take another look:

```
ColumnForm[Thread[diffeqsystem]]
```

$$x'[t] == 2.9\,x[t] + 0.4\,y[t]$$
$$y'[t] == -1.4\,x[t] + 0.6\,y[t]$$

You can read off a matrix A:

```
A = {{2.9, 0.4}, {-1.4, 0.6}};
MatrixForm[A]
```

$$\begin{pmatrix} 2.9 & 0.4 \\ -1.4 & 0.6 \end{pmatrix}$$

Folks call this matrix by the name "coefficient matrix."

Now you can use this matrix to write the given linear system in this compact way:

```
linearsystem = ({x'[t], y'[t]} == A . {x[t], y[t]});
ColumnForm[Thread[linearsystem]]
```

$$x'[t] == 2.9\,x[t] + 0.4\,y[t]$$
$$y'[t] == -1.4\,x[t] + 0.6\,y[t]$$

Compare:

```
ColumnForm[Thread[diffeqsystem]]
```

$$x'[t] == 2.9\,x[t] + 0.4\,y[t]$$
$$y'[t] == -1.4\,x[t] + 0.6\,y[t]$$

Any diffeq system you can make with a coefficient matrix of numbers is a linear system.

□B.1.a.ii)

Here's another diffeq system:

```
Clear[x, y, t]
diffeqsystem =
 ({x'[t], y'[t]} == {2.9 x[t]^2 + 0.4 y[t], -1.4 x[t] y[t] + 0.6 y[t]});
ColumnForm[Thread[diffeqsystem]]
```

$$x'[t] == 2.9\,x[t]^2 + 0.4\,y[t]$$
$$y'[t] == 0.6\,y[t] - 1.4\,x[t]\,y[t]$$

How do you tell that this diffeq system is not a linear system?

□Answer:

Take another look:

```
ColumnForm[Thread[diffeqsystem]]
```

$$x'[t] == 2.9\,x[t]^2 + 0.4\,y[t]$$
$$y'[t] == 0.6\,y[t] - 1.4\,x[t]\,y[t]$$

You can tell that this is not a linear system because there is no way to get that

$$x[t]^2$$

or that

$$x[t]\,y[t]$$

term by multiplying {x[t], y[t]} by a coefficient matrix of numbers.

B.2) Eigenvectors of the coefficient matrix give the directions of propellers and suckers.

Eigenvalues of the coefficient matrix measure the strengths of propellers and suckers

□B.2.a.i) Eigenvectors: Suckers and propellers

Here's a linear system of differential equations:

```
Clear[x, y, t]
linearsystem =
 ({x'[t], y'[t]} == {1.75 x[t] + 2.17 y[t], 2.17 x[t] - 0.75 y[t]})
```

$$\{x'[t], y'[t]\} == \{1.75\,x[t] + 2.17\,y[t], 2.17\,x[t] - 0.75\,y[t]\}$$

Here it is in slightly better form:

```
ColumnForm[Thread[linearsystem]]
```

$$x'[t] == 1.75\,x[t] + 2.17\,y[t]$$
$$y'[t] == 2.17\,x[t] - 0.75\,y[t]$$

Read off the coefficient matrix A:

```
A = {{1.75, 2.17}, {2.17, -0.75}};
MatrixForm[A]
```

$$\begin{pmatrix} 1.75 & 2.17 \\ 2.17 & -0.75 \end{pmatrix}$$

Now you can use this matrix to write the given linear system in this compact way:

```
Clear[x, y, t]
matrixlinearsystem = ({x'[t], y'[t]} == A . {x[t], y[t]});
ColumnForm[Thread[matrixlinearsystem]]
```

$$x'[t] == 1.75\,x[t] + 2.17\,y[t]$$
$$y'[t] == 2.17\,x[t] - 0.75\,y[t]$$

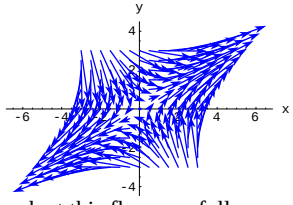
Introducing the cofficient matrix A makes it really easy to write down the flow field for this linear system:

```
Clear[Field]
Field[x_, y_] = A . {x, y}
```

$$\{1.75\,x + 2.17\,y,\ 2.17\,x - 0.75\,y\}$$

Look at the scaled flow:

```
flowplot = Table[Arrow[Field[x, y], Tail → {x, y}, VectorColor → Blue,
    ScaleFactor → 0.3, HeadSize → 0.6], {x, -3, 3, 6/12}, {y, -3, 3, 6/12}];
Show[flowplot, Axes → True, AxesLabel → {"x", "y"}];
```



Look at this flow carefully.

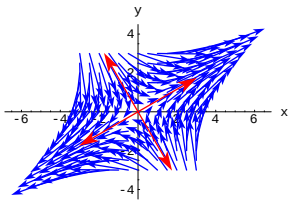
After you've got a good feel for this flow, let *Mathematica* calculate two special vectors related to the coefficient matrix A:

```
Clear[eigenvector];
{eigenvector[1], eigenvector[2]} = Eigenvectors[A]
```

$$\{\{0.865779, 0.500427\}, \{-0.500427, 0.865779\}\}$$

Plot scaled versions of these special vectors and their negatives together with the field:

```
scalefactor = 3.5;
headsize = 1;
revealer = Show[flowplot,
  Arrow[eigenvector[1], Tail → {0, 0}, VectorColor → Red,
   ScaleFactor → scalefactor, HeadSize → headsize],
  Arrow[-eigenvector[1], Tail → {0, 0}, VectorColor → Red,
   ScaleFactor → scalefactor, HeadSize → headsize],
  Arrow[eigenvector[2], Tail → {0, 0}, VectorColor → Red,
   ScaleFactor → scalefactor, HeadSize → headsize],
  Arrow[-eigenvector[2], Tail → {0, 0}, VectorColor → Red,
   ScaleFactor → scalefactor, HeadSize → headsize], Axes → True,
  AxesLabel → {"x", "y"}];
```



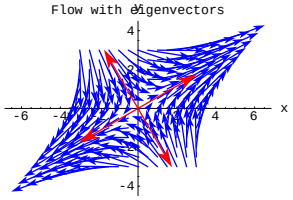
The pros call these special vectors by the name:
eigenvectors.

Hot plot.
Describe what you see.

□ **Answer:**

Take another look:

```
Show[revealer, PlotLabel → "Flow with eigenvectors"];
```



The eigenvectors tell all!

The flow is sucked toward {0, 0} by the influence of the eigenvectors on the upper left and lower right.

That's why DiffEq&*Mathematica* folks call the eigenvectors on the upper left and lower right by the name "suckers."

Then the flow is propelled away from {0, 0} by the influence of the eigenvectors on the lower left and upper right.

That's why DiffEq&*Mathematica* folks call the eigenvectors on the lower left and upper right by the name "propellers."

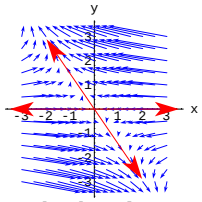
Look at this flow carefully.
After you've got a good feel for this flow, let *Mathematica* calculate the eigenvectors of the matrix A:

```
Clear[eigenvector];
{eigenvector[1], eigenvector[2]} = Eigenvectors[A]
```

```
{{1., 0.}, {-0.564684, 0.825307}}
```

Plot scaled versions of these special vectors and their negatives together with the field:

```
scalefactor = 3.5;
headsize = 1;
eigenplot = Show[flowplot, Arrow[eigenvector[1], Tail → {0, 0},
    VectorColor → Red, ScaleFactor → scalefactor, HeadSize → headsize],
  Arrow[-eigenvector[1], Tail → {0, 0}, VectorColor → Red,
    ScaleFactor → scalefactor, HeadSize → headsize],
  Arrow[eigenvector[2], Tail → {0, 0}, VectorColor → Red,
    ScaleFactor → scalefactor, HeadSize → headsize],
  Arrow[-eigenvector[2], Tail → {0, 0}, VectorColor → Red,
    ScaleFactor → scalefactor, HeadSize → headsize], Axes → True,
  AxesLabel → {"x", "y"}];
```

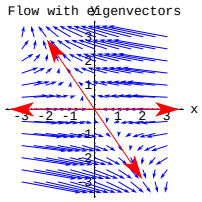


Another hot plot.
Describe what you see.

□ **Answer:**

Take another look:

```
Show[eigenplot, PlotLabel → "Flow with eigenvectors"];
```



The eigenvectors tell all!

The flow is sucked toward {0, 0} by the influence of the eigenvectors that point down the horizontal axis.

That's why DiffEq&*Mathematica* folks call the eigenvectors on the horizontal axis by the name "suckers."

Then the flow is propelled away from {0, 0} by the influence of the eigenvectors on the upper left and lower right.

That's why DiffEq&*Mathematica* folks call the eigenvectors on the upper left and lower right by the name "propellers."

□B.2.a.ii) Eigenvectors: Suckers and propellers

Try it again with this new linear system:

$$x'[t] = -1.4\,x[t] - 1.3\,y[t]$$
$$y'[t] = 0.5\,y[t].$$

The coefficient matrix is:

```
A = {{-1.4, -1.3}, {0, 0.5}};
MatrixForm[A]
```

$$\begin{pmatrix} -1.4 & -1.3 \\ 0 & 0.5 \end{pmatrix}$$

Compare with the given linear system

$$x'[t] = -1.4\,x[t] - 1.3\,y[t]$$
$$y'[t] = 0.5\,y[t]:$$

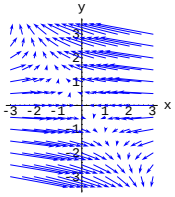
```
Clear[x, y, t]
matrixlinearsystem = ({x'[t], y'[t]} == A . {x[t], y[t]});
ColumnForm[Thread[matrixlinearsystem]]
```

```
x'[t] == -1.4 x[t] - 1.3 y[t]
y'[t] == 0.5 y[t]
```

That's the right matrix.
Look at the flow:

```
Clear[Field]
Field[x_, y_] = A . {x, y}
flowplot = Table[Arrow[Field[x, y], Tail → {x, y},
    VectorColor → Blue, ScaleFactor → 0.3, HeadSize → 0.25],
  {x, -3, 3, 6/12}, {y, -3, 3, 6/12}];
Show[flowplot, Axes → True, AxesLabel → {"x", "y"}];
```

```
{-1.4 x - 1.3 y, 0.5 y}
```



□B.2.a.iii) Eigenvectors: Suckers and no propellers

Try it again with this new linear system:

$$x'[t] = -0.57\,x[t] + 0.17\,y[t]$$
$$y'[t] = 0.17\,x[t] - 0.93\,y[t].$$

The coefficient matrix you want is:

```
A = {{-0.57, 0.17}, {0.17, -0.93}};
MatrixForm[A]
```

$$\begin{pmatrix} -0.57 & 0.17 \\ 0.17 & -0.93 \end{pmatrix}$$

Compare

$$\{x'[t], y'[t]\} = A \bullet \{x[t], y[t]\}$$

with the given linear system:

$$x'[t] = -0.57\,x[t] + 0.17\,y[t]$$
$$y'[t] = 0.17\,x[t] - 0.93\,y[t]:$$

```
Clear[x, y, t]
linearsystem = ({x'[t], y'[t]} == A . {x[t], y[t]});
ColumnForm[Thread[linearsystem]]
```

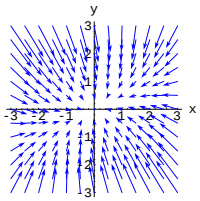
```
x'[t] == -0.57 x[t] + 0.17 y[t]
y'[t] == 0.17 x[t] - 0.93 y[t]
```

That's the right matrix.

Look at the flow:

```
Clear[Field]
Field[x_, y_] = A . {x, y}
flowplot = Table[Arrow[Field[x, y], Tail → {x, y},
   VectorColor → Blue, ScaleFactor → 0.35, HeadSize → 0.25],
  {x, -3, 3, 6/12}, {y, -3, 3, 6/12}];
Show[flowplot, Axes → True, AxesLabel → {"x", "y"}];
```

```
{-0.57 x + 0.17 y, 0.17 x - 0.93 y}
```



Look at this flow carefully.

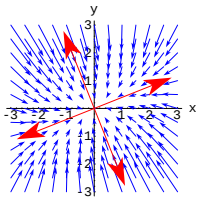
After you've got a good feel for this flow, let *Mathematica* calculate the eigenvectors of A:

```
Clear[eigenvector];
{eigenvector[1], eigenvector[2]} = Eigenvectors[A]
```

```
{{-0.36945, 0.929251}, {0.929251, 0.36945}}
```

Plot these special vectors and their negatives together with the field:

```
scalefactor = 3.0;
headsize = 1;
eigenplot = Show[flowplot, Arrow[eigenvector[1], Tail → {0, 0},
    VectorColor → Red, ScaleFactor → scalefactor, HeadSize → headsize],
  Arrow[-eigenvector[1], Tail → {0, 0}, VectorColor → Red,
   ScaleFactor → scalefactor, HeadSize → headsize],
  Arrow[eigenvector[2], Tail → {0, 0}, VectorColor → Red,
   ScaleFactor → scalefactor, HeadSize → headsize],
  Arrow[-eigenvector[2], Tail → {0, 0}, VectorColor → Red,
   ScaleFactor → scalefactor, HeadSize → headsize], Axes → True,
  AxesLabel → {"x", "y"}];
```

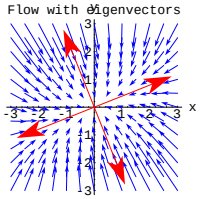


Describe what you see.

□ **Answer:**

Take another look:

```
Show[eigenplot, PlotLabel → "Flow with eigenvectors"];
```



Both eigenvectors work together to suck the flow toward {0, 0}.

That's why DiffEq&*Mathematica* folks call
both of these eigenvectors by the name "suckers."

The sucking influence of the sucker eigenvectors on the upper left and lower right is a bit stronger than the sucking influence of the sucker eigenvectors on the lower left and upper right.

□B.2.a.iv) Eigenvectors: Propellers and no suckers

Try it again with this new linear system:

x'[t] = 0.1 x[t] + 0.5 y[t]
y'[t] = −0.5 x[t] + 1.5 y[t].

The coefficient matrix is:

```
A = {{0.1, 0.5}, {-0.5, 1.5}};
MatrixForm[A]
```

$$\begin{pmatrix} 0.1 & 0.5 \\ -0.5 & 1.5 \end{pmatrix}$$

Compare with the given linear system

x'[t] = 0.1 x[t] + 0.5 y[t]
y'[t] = −0.5 x[t] + 1.5 y[t]:

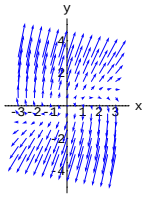
```
Clear[x, y, t]
linearsystem = ({x'[t], y'[t]} == A . {x[t], y[t]});
ColumnForm[Thread[linearsystem]]
```

```
x'[t] == 0.1 x[t] + 0.5 y[t]
y'[t] == -0.5 x[t] + 1.5 y[t]
```

That's the right matrix.

Look at the flow:

```
Clear[Field]
Field[x_, y_] = A . {x, y};
flowplot = Table[Arrow[Field[x, y], Tail → {x, y},
   VectorColor → Blue, ScaleFactor → 0.35, HeadSize → 0.3],
  {x, -3, 3, 6/12}, {y, -3, 3, 6/12}];
Show[flowplot, Axes → True, AxesLabel → {"x", "y"}];
```



Look at this flow carefully.

It's a little tougher to see what's happening with this one.

After you've got a good feel for this flow, let *Mathematica* calculate the eigenvectors of A:

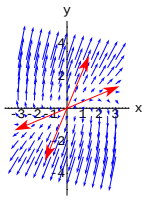
```
Clear[eigenvector];
{eigenvector[1], eigenvector[2]} = Eigenvectors[A]
```

```
{{-0.387392, -0.921915}, {-0.921915, -0.387392}}
```

Plot scaled versions these eigenvectors and their negatives together with the field:

```
scalefactor = 3.5;
headsize = 1;
```

```
eigenplot = Show[flowplot, Arrow[eigenvector[1], Tail → {0, 0},
    VectorColor → Red, ScaleFactor → scalefactor, HeadSize → headsize],
  Arrow[-eigenvector[1], Tail → {0, 0}, VectorColor → Red,
   ScaleFactor → scalefactor, HeadSize → headsize],
  Arrow[eigenvector[2], Tail → {0, 0}, VectorColor → Red,
   ScaleFactor → scalefactor, HeadSize → headsize],
  Arrow[-eigenvector[2], Tail → {0, 0}, VectorColor → Red,
   ScaleFactor → scalefactor, HeadSize → headsize], Axes → True,
  AxesLabel → {"x", "y"}];
```



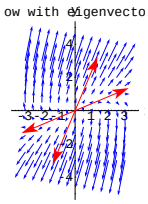
Now it's not so hard to see what's happening.

Describe what you see.

□ **Answer:**

Take another look:

```
plot1 = Show[eigenplot, PlotLabel → "Flow with eigenvectors"];
```



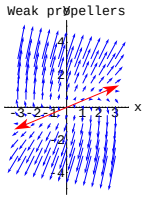
Both eigenvectors work together to propell the flow away from {0, 0}.

That's why DiffEq&*Mathematica* folks call these eigenvectors "propellers."

The influence of the following two propelling eigenvectors is not very strong:

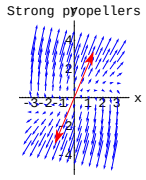
```
Show[flowplot, Arrow[eigenvector[2], Tail → {0, 0},
  VectorColor → Red, ScaleFactor → scalefactor, HeadSize → headsize],
```

```
Arrow[-eigenvector[2], Tail → {0, 0}, VectorColor → Red,
 ScaleFactor → scalefactor, HeadSize → headsize], Axes → True,
AxesLabel → {"x", "y"}, PlotLabel → "Weak propellers"];
```



The dominant propelling influence comes from the other eigenvectors:

```
Show[flowplot, Arrow[eigenvector[1], Tail → {0, 0},
  VectorColor → Red, ScaleFactor → scalefactor, HeadSize → headsize],
 Arrow[-eigenvector[1], Tail → {0, 0}, VectorColor → Red,
  ScaleFactor → scalefactor, HeadSize → headsize], Axes → True,
 AxesLabel → {"x", "y"}, PlotLabel → "Strong propellers"];
```



Grab the last two plots and animate slowly.

The flow is greatly influenced by these two dominant eigenvectors.

□B.2.b.i) Eigenvalues tell you whether an eigenvector is a propeller or a sucker

How do you use *Mathematica* to tell in advance whether an eigenvector is a sucker or a propeller?

□Answer:

It's not very hard.

For instance when you come across a linear system like this:

$$x'[t] = 1.75\,x[t] + 1.8\,y[t]$$
$$y'[t] = 2.17\,x[t] - 0.75\,y[t].$$

The coefficient matrix is:

```
A = {{1.75, 1.8}, {2.17, -0.75}};
MatrixForm[A]

Clear[x, y, t]
linearsystem = ({x'[t], y'[t]} == A.{x[t], y[t]});
ColumnForm[Thread[linearsystem]]
```

$$\begin{pmatrix} 1.75 & 1.8 \\ 2.17 & -0.75 \end{pmatrix}$$

$$x'[t] == 1.75\,x[t] + 1.8\,y[t]$$
$$y'[t] == 2.17\,x[t] - 0.75\,y[t]$$

You ask *Mathematica* for the eigenvectors of A:

```
Clear[eigenvector];
{eigenvector[1], eigenvector[2]} = Eigenvectors[A]
```

{{0.855709, 0.517458}, {-0.448361, 0.893853}}

Now you ask for the eigenvalues of A:

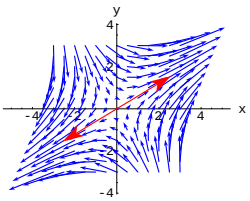
```
Clear[eigenvalue];
{eigenvalue[1], eigenvalue[2]} = Eigenvalues[A]
```

{2.83848, -1.83848}

There is one eigenvalue corresponding to each eigenvector.

Eigenvalue[1] is positive. This tells you that eigenvector[1] is a propeller. Check it out:

```
Clear[Field]
Field[x_, y_] = A.{x, y};
flowplot = Table[Arrow[Field[x, y], Tail → {x, y}, VectorColor → Blue,
   ScaleFactor → 0.2, HeadSize → 0.3], {x, -3, 3, 6/12}, {y, -3, 3, 6/12}];
Show[flowplot, Arrow[eigenvector[1], Tail → {0, 0}, VectorColor → Red,
  ScaleFactor → 3, HeadSize → 1], Arrow[-eigenvector[1],
  Tail → {0, 0}, VectorColor → Red, ScaleFactor → 3, HeadSize → 1],
 Axes → True, AxesLabel → {"x", "y"}];
```



Yes, eigenvector[1] is a propeller.

You knew this in advance because you knew that eigenvalue[1] is positive:

```
eigenvalue[1]
```

2.83848

Now look at eigenvalue[2]:

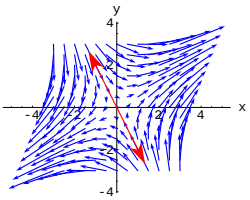
```
eigenvalue[2]
```

-1.83848

Negative. This tells you that eigenvector[2] is a sucker.

Check it out:

```
Show[flowplot, Arrow[eigenvector[2], Tail → {0, 0}, VectorColor → Red,
  ScaleFactor → 3, HeadSize → 1], Arrow[-eigenvector[2],
  Tail → {0, 0}, VectorColor → Red, ScaleFactor → 3, HeadSize → 1],
 Axes → True, AxesLabel → {"x", "y"}];
```



Yes, eigenvector[2] is a sucker.

You knew this in advance because you knew that eigenvalue[2] is negative:

```
eigenvalue[2]
```

-1.83848

□B.2.b.ii) Eigenvalues measure the relative strengths of propellers

How do you use *Mathematica* to measure the relative strengths of two propellers?

□Answer:

When you come across a linear system like this:

$$x'[t] = 0.36\,x[t] + 0.81\,y[t]$$
$$y'[t] = -0.21\,x[t] + 1.53\,y[t].$$

The coefficient matrix is:

```
A = {{0.36, 0.81}, {-0.21, 1.53}};
MatrixForm[A]
```

$$\begin{pmatrix} 0.36 & 0.81 \\ -0.21 & 1.53 \end{pmatrix}$$

Check:

```
Clear[x, y, t]
linearsystem = ({x'[t], y'[t]} == A.{x[t], y[t]});
ColumnForm[Thread[linearsystem]]
```

$$x'[t] == 0.36\,x[t] + 0.81\,y[t]$$
$$y'[t] == -0.21\,x[t] + 1.53\,y[t]$$

You ask *Mathematica* for the eigenvectors of A:

```
Clear[eigenvector];
{eigenvector[1], eigenvector[2]} = Eigenvectors[A]
```

{{-0.629468, -0.777027}, {-0.978649, -0.205541}}

Now you ask for the eigenvalues of A:

```
Clear[eigenvalue];
{eigenvalue[1], eigenvalue[2]} = Eigenvalues[A]
```

{1.35988, 0.53012}

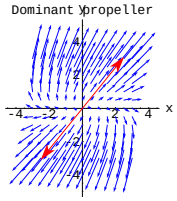
Both eigenvalue[1] and eigenvalue[2] are positive. This signals that both eigenvector[1] and eigenvector[2] are propellers.

But eigenvalue[1]is more than twice the size of eigenvalue[2].

This tells you that the dominant propeller is eigenvector[1].

Check it out:

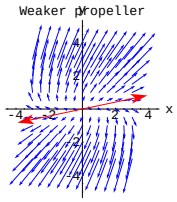
```
Clear[Field]
Field[x_, y_] = A . {x, y};
scalefactor = 0.35;
flowplot = Table[Arrow[Field[x, y], Tail → {x, y}, VectorColor → Blue,
   ScaleFactor → 0.4, HeadSize → 0.3], {x, -3, 3, 6/12}, {y, -3, 3, 6/12}];
Show[flowplot,
 Arrow[eigenvector[1], Tail → {0, 0}, VectorColor → Red,
  ScaleFactor → 4.0, HeadSize → 1], Arrow[-eigenvector[1],
  Tail → {0, 0}, VectorColor → Red, ScaleFactor → 4.0, HeadSize → 1],
 Axes → True, AxesLabel → {"x", "y"},
 PlotLabel → "Dominant propeller"];
```



Look at the field vectors. Yes, eigenvector[1] is the dominant propeller.

Check out eigenvector[2]:

```
Show[flowplot,
 Arrow[eigenvector[2], Tail → {0, 0}, VectorColor → Red,
  ScaleFactor → 4.0, HeadSize → 1], Arrow[-eigenvector[2],
  Tail → {0, 0}, VectorColor → Red, ScaleFactor → 4.0, HeadSize → 1],
 Axes → True, AxesLabel → {"x", "y"}, PlotLabel → "Weaker propeller"];
```



Grab both plots, align and animate slowly.

Look at the field vectors. A much weaker propeller.

You knew this in advance because you knew that

eigenvalue[1] > eigenvalue[2] > 0.

```
{eigenvalue[1], eigenvalue[2]}
```

```
{1.35988, 0.53012}
```

□B.2.b.iii) Eigenvalues measure the relative strengths of suckers

How can you use *Mathematica* to measure the relative strengths of two suckers?

□Answer:

When you come across a linear system like this:

$$x'[t] = -0.88\,x[t] + 0.16\,y[t]$$

$$y'[t] = 0.54\,x[t] - 0.92\,y[t].$$

The matrix you want is:

```
A = {{-0.88, 0.16}, {0.54, -0.92}};
MatrixForm[A]

Clear[x, y, t]
linearsystem = ({x'[t], y'[t]} == A . {x[t], y[t]});
ColumnForm[Thread[linearsystem]]
```

$$\begin{pmatrix} -0.88 & 0.16 \\ 0.54 & -0.92 \end{pmatrix}$$

```
x'[t] == -0.88 x[t] + 0.16 y[t]
y'[t] == 0.54 x[t] - 0.92 y[t]
```

You ask *Mathematica* for the eigenvectors of A:

```
Clear[eigenvector];
{eigenvector[1], eigenvector[2]} = Eigenvectors[A]
```

```
{{-0.453302, 0.891357}, {0.503415, 0.864045}}
```

Now you ask for the eigenvalues of A:

```
Clear[eigenvalue];
{eigenvalue[1], eigenvalue[2]} = Eigenvalues[A]
```

```
{-1.19462, -0.605382}
```

Both eigenvalue[1] and eigenvalue[2] are negative.

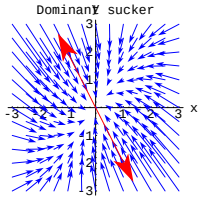
This tells you that both eigenvector[1] and eigenvector[2] are suckers.

But eigenvalue[1] is about twice as negative as eigenvalue[2].

This tells you that the dominant sucker is eigenvector[1].

Check it out:

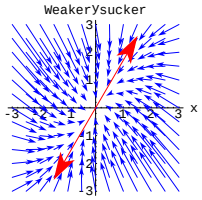
```
Clear[Field]
Field[x_, y_] = A . {x, y};
flowplot = Table[Arrow[Field[x, y], Tail → {x, y},
    VectorColor → Blue, ScaleFactor → 0.35, HeadSize → 0.3],
  {x, -3, 3, 6/12}, {y, -3, 3, 6/12}];
Show[flowplot,
 Arrow[eigenvector[1], Tail → {0, 0},
  VectorColor → Red, ScaleFactor → 3.0, HeadSize → 1],
 Arrow[-eigenvector[1], Tail → {0, 0},
  VectorColor → Red, ScaleFactor → 3.0, HeadSize → 1], Axes → True,
 AxesLabel → {"x", "y"}, PlotLabel → "Dominant sucker"];
```



Look at the field vectors. Yes, eigenvector[1] is the dominant sucker.

Check out eigenvector[2]:

```
Show[flowplot, Arrow[eigenvector[2], Tail → {0, 0},
  VectorColor → Red, ScaleFactor → 3.0, HeadSize → 1],
 Arrow[-eigenvector[2], Tail → {0, 0},
  VectorColor → Red, ScaleFactor → 3.0, HeadSize → 1], Axes → True,
 AxesLabel → {"x", "y"}, PlotLabel → "Weaker sucker"];
```



Grab both plots and animate slowly.

Look at the field vectors. A much weaker sucker.

You knew this in advance because you knew that

eigenvalue[1] < eigenvalue[2] < 0.

B.3) How you can use eigenvectors and eigenvalues to understand where exact formulas for solutions of linear systems come from

This assumes familiarity with B.2) above.

Most everyone seems to want quick code to copy, paste and edit to produce exact formulas for the solutions of given linear systems with given starter data. If you want it, then here it is.

Here's a random coefficient matrix for a random linear system:

```
A = {{Random[Real, {-2, 2}], Random[Real, {-2, 2}]},
   {Random[Real, {-2, 2}], Random[Real, {-2, 2}]}};
Clear[x, y, t]
linearsystem =
  ({x'[t], y'[t]} == A . {x[t], y[t]});
ColumnForm[Thread[linearsystem]]
```

```
x'[t] == 0.989698 x[t] + 1.71716 y[t]
y'[t] == -0.0461689 x[t] - 1.63099 y[t]
```

With random starter data on {x[0], y[0]}:

```
{xstarter, ystarter} = {Random[Real, {-2, 2}], Random[Real, {-2, 2}]}
```
```
{0.0402287, 1.54997}
```

Formulas for the solution pair {x[t], y[t]} of this linear system with this starter data are:

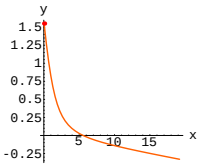
```
Clear[eigenvector, eigenvalue];
{eigenvalue[1], eigenvalue[2]} = Chop[Eigenvalues[A]];
{eigenvector[1], eigenvector[2]} = Chop[Eigenvectors[A]];

Clear[x, y, x1, y1, x2, y2, t, C1, C2]
{x1[t_], y1[t_]} = Chop[eigenvector[1] Exp[eigenvalue[1] t]];
{x2[t_], y2[t_]} = Chop[eigenvector[2] Exp[eigenvalue[2] t]];
{x[t_], y[t_]} = C1 {x1[t], y1[t]} + C2 {x2[t], y2[t]};

starter = {xstarter, ystarter};
starterequation = {x[0], y[0]} == starter;
Csols = Solve[starterequation];

{x[t_], y[t_]} = Chop[ComplexExpand[{x[t], y[t]} /. Csols[[1]]]]
```
```
{-1.04037 E^-1.60038 t + 1.0806 E^0.959089 t,
 1.56923 E^-1.60038 t - 0.019262 E^0.959089 t}
```

Check:

```
Expand[{x'[t], y'[t]}] == Expand[A . {x[t], y[t]}]
{x[0], y[0]} == {xstarter, ystarter}
```
```
True
True
```

See part of the corresponding trajectory:

```
endtime = 3; ParametricPlot[{x[t], y[t]},
 {t, 0, endtime}, PlotStyle → {{Thickness[0.01], CadmiumOrange}},
 AxesLabel → {"x", "y"}, AspectRatio → 1,
 Epilog → {PointSize[0.03], Red, Point[{xstarter, ystarter}]}];
```



Note how the eigenvalues of the coefficient matrix show up in the formula for the solution:

```
{x[t], y[t]}
Chop[Eigenvalues[A]]
```
```
{-1.04037 E^-1.60038 t + 1.0806 E^0.959089 t,
 1.56923 E^-1.60038 t - 0.019262 E^0.959089 t}
{-1.60038, 0.959089}
```

Give it another whirl with a different random linear system with a different random starting point for {x[0], y[0]}:

```
A = {{Random[Real, {-2, 2}], Random[Real, {-2, 2}]},
   {Random[Real, {-2, 2}], Random[Real, {-2, 2}]}};
Clear[x, y, t]
linearsystem =
  ({x'[t], y'[t]} == A . {x[t], y[t]});
ColumnForm[Thread[linearsystem]]
{xstarter, ystarter} = {Random[Real, {-2, 2}], Random[Real, {-2, 2}]};
({x[0], y[0]} == starter)

Clear[eigenvector, eigenvalue];
{eigenvalue[1], eigenvalue[2]} = Chop[Eigenvalues[A]];
{eigenvector[1], eigenvector[2]} = Chop[Eigenvectors[A]];

Clear[x, y, x1, y1, x2, y2, t, C1, C2]
{x1[t_], y1[t_]} = Chop[eigenvector[1] Exp[eigenvalue[1] t]];
{x2[t_], y2[t_]} =  Chop[eigenvector[2] Exp[eigenvalue[2] t]];

{x[t_], y[t_]} = C1 {x1[t], y1[t]} + C2 {x2[t], y2[t]};

starter = {xstarter, ystarter};
starterequation = ({x[0], y[0]} == starter);
Csols = Solve[starterequation];

{x[t_], y[t_]} = Chop[ComplexExpand[{x[t], y[t]} /. Csols[[1]]]]
```
```
x'[t] == -1.73251 x[t] - 0.705918 y[t]
y'[t] == -0.103493 x[t] - 1.36263 y[t]
{x[0], y[0]} == {0.0402287, 1.54997}
{0.268969 E^-1.87508 t - 2.05981 E^-1.22006 t,
 0.0543203 E^-1.87508 t + 1.49529 E^-1.22006 t}
```

Note how the formulas are related to the eigenvalues of the coefficient matrix:

```
{x[t], y[t]}
Eigenvalues[A]
```
```
{0.268969 E^-1.87508 t - 2.05981 E^-1.22006 t,
 0.0543203 E^-1.87508 t + 1.49529 E^-1.22006 t}
{-1.87508, -1.22006}
```
```
Rerun the last two active cells a couple of times.
```

If you want to see why these formulas came out the way they did, then jump into B.3) and follow it up with a healthy romp in B.4)

□B.3.a.i) Straight line trajectories along eigenvectors.

Here's a linear system of differential equations:

```
Clear[x, y, t]
linearsystem =
({x'[t], y'[t]} == {0.84 x[t] + 0.90 y[t], 0.40 x[t] - 0.31 y[t]});

ColumnForm[Thread[linearsystem]]
```
```
x'[t] == 0.84 x[t] + 0.9 y[t]
y'[t] == 0.4 x[t] - 0.31 y[t]
```

You read off the coefficient matrix A:

```
A = {{0.84, 0.90}, {0.40, -0.31}};
MatrixForm[A]
```
```
(0.84   0.9
 0.4   -0.31)
```

Check:

```
ColumnForm[Thread[{x'[t], y'[t]} == A . {x[t], y[t]}]]
```
```
x'[t] == 0.84 x[t] + 0.9 y[t]
y'[t] == 0.4 x[t] - 0.31 y[t]
```

Good.

You ask for the eigenvectors:

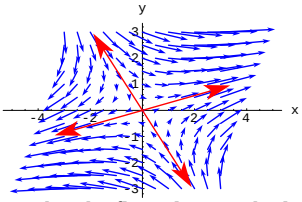
```
Clear[eigenvector];
{eigenvector[1], eigenvector[2]} = Eigenvectors[A]
```
```
{{0.961835, 0.27363}, {-0.539111, 0.842235}}
```

The eigenvalues:

```
Clear[eigenvalue];
{eigenvalue[1], eigenvalue[2]} = Eigenvalues[A]
```
```
{1.09604, -0.566039}
```

The flow with eigenvectors:

```
Clear[Field]
Field[x_, y_] = A . {x, y};
flowplot = Table[Arrow[Field[x, y], Tail → {x, y}, VectorColor → Blue,
   ScaleFactor → 0.4, HeadSize → 0.3], {x, -3, 3, 6/12}, {y, -3, 3, 6/12}];
eigenplot =
 Show[flowplot, Arrow[eigenvector[1], Tail → {0, 0}, VectorColor → Red,
   ScaleFactor → 3.5, HeadSize → 1], Arrow[-eigenvector[1],
   Tail → {0, 0}, VectorColor → Red, ScaleFactor → 3.5, HeadSize → 1],
  Arrow[eigenvector[2], Tail → {0, 0}, VectorColor → Red,
   ScaleFactor → 3.5, HeadSize → 1], Arrow[-eigenvector[2],
   Tail → {0, 0}, VectorColor → Red, ScaleFactor → 3.5, HeadSize → 1],
  Axes → True, AxesLabel → {"x", "y"}];
```



Note that the flow along each plotted eigenvector is a straight line determined by the eigenvector you are looking at.

Now look at these calculations:

```
Clear[s]
A . (s eigenvector[1]) == eigenvalue[1] (s eigenvector[1])
```
```
True
```

And:

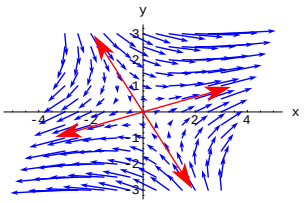
```
A . (s eigenvector[2]) == eigenvalue[2] (s eigenvector[2])
```
```
True
```

How do you use these calculations to explain why the trajectories along the eigenvectors are straight lines?

□**Answer:**

Look again:

```
Show[eigenplot];
```



And look at the calculations again.

```
Clear[s]
A . (s eigenvector[1]) == eigenvalue[1] (s eigenvector[1])
```
```
True
```

And:

```
A . (s eigenvector[2]) == eigenvalue[2] (s eigenvector[2])
```
```
True
```

These properties define eigenvalues and eigenvectors.

Remembering that the linear system is

{x'[t], y'[t]} = A . {x[t], y[t]},

this tells you that if {x[t], y[t]} is a multiple of one of the eigenvectors, then

{x'[t], y'[t]}

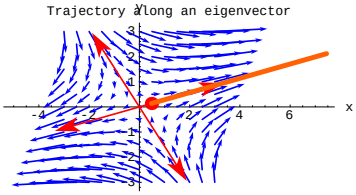
is forced to point in the same (or opposite) direction as the eigenvector points.

The upshot:

If a trajectory hits the line determined by one of the eigenvectors, it has no choice but to run along the line determined by that eigenvector.

Here's a plot of a solution that starts right on the middle of eigenvector[1]:

```
multiplier = 0.5;
{a, b} = multiplier eigenvector[1];
starterpoint = {a, b};
Clear[x, y, t, m, n]
{m[x_, y_], n[x_, y_]} = A . {x, y};
xequation = x'[t] == m[x[t], y[t]];
yequation = y'[t] == n[x[t], y[t]];
xstarter = x[0] == a;
ystarter = y[0] == b;
endtime = 2.5;
approxsolutions = NDSolve[{xequation, yequation, xstarter, ystarter},
  {x[t], y[t]}, {t, 0, endtime}];
Clear[trajectory]
trajectory[t_] =
 {x[t] /. approxsolutions[[1]], y[t] /. approxsolutions[[1]]};
trajectoryplot =
 ParametricPlot[trajectory[t], {t, 0, endtime}, PlotStyle →
   {{CadmiumOrange, Thickness[0.015]}}, DisplayFunction → Identity];
starterplot = Graphics[{Red, PointSize[0.04], Point[starterpoint]}];
Show[
 eigenplot, starterplot, trajectoryplot, PlotRange → All, Axes → True,
 AxesLabel → {"x", "y"}, DisplayFunction → $DisplayFunction,
 PlotLabel → "Trajectory along an eigenvector"];
```

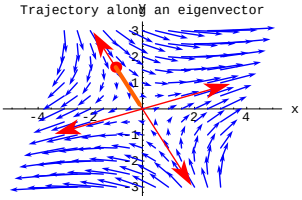


Propelled away from {0, 0} on the straight line trajectory determined by eigenvector[1].

Check out what happens when you start a trajectory at

1.9 eigenvector[2]:

```
multiplier = 1.9;
{a, b} = multiplier eigenvector[2];
starterpoint = {a, b};
Clear[x, y, t, m, n]
{m[x_, y_], n[x_, y_]} = A . {x, y};
xequation = x'[t] == m[x[t], y[t]];
yequation = y'[t] == n[x[t], y[t]];
xstarter = x[0] == a;
ystarter = y[0] == b;
endtime = 4;
approxsolutions = NDSolve[{xequation, yequation, xstarter, ystarter},
  {x[t], y[t]}, {t, 0, endtime}];
Clear[trajectory]
trajectory[t_] =
 {x[t] /. approxsolutions[[1]], y[t] /. approxsolutions[[1]]};
trajectoryplot =
 ParametricPlot[trajectory[t], {t, 0, endtime}, PlotStyle →
   {{CadmiumOrange, Thickness[0.015]}}, DisplayFunction → Identity];
starterplot = Graphics[{Red, PointSize[0.04], Point[starterpoint]}];
Show[
 eigenplot, starterplot, trajectoryplot, PlotRange → All, Axes → True,
 AxesLabel → {"x", "y"}, DisplayFunction → $DisplayFunction,
 PlotLabel → "Trajectory along an eigenvector"];
```



Sucked towards {0, 0} on the straight line trajectory determined by eigenvector[2].

Play with different starter points on the eigenvectors and different endtimes.

□B.3.a.ii) Formulas for the straight line trajectories

Stay with the same linear system as above. Formulas for the trajectories that start on the tips of the eigenvector[1] and eigenvector[2] are:

```
Clear[x1, y1, x2, y2, t]
{x1[t_], y1[t_]} = eigenvector[1] E^(eigenvalue[1] t)
{x2[t_], y2[t_]} = eigenvector[2] E^(eigenvalue[2] t)
```
```
{0.961835 E^(1.09604 t), 0.27363 E^(1.09604 t)}
{-0.539111 E^(-0.566039 t), 0.842235 E^(-0.566039 t)}
```

Explain where these formulas come from.

□Answer:

The key idea comes from these calculations:

```
Clear[s]
A . (s eigenvector[1]) == eigenvalue[1] (s eigenvector[1])
```
```
True
```

And:

```
A . (s eigenvector[2]) == eigenvalue[2] (s eigenvector[2])
```
```
True
```

The linear system is

{x'[t], y'[t]} = A . {x[t], y[t]}.

If {x1[t], y1[t]} is on a multiple of eigenvector[1], then

{x1'[t], y1'[t]} = A . {x1[t], y1[t]} = eigenvalue[1] {x1[t], y1[t]}.

This gives

x1'[t] = eigenvalue[1] x1[t]

y1'[t] = eigenvalue[1] y1[t].

These are both exponential differential equations, and you know that these give you

x1[t] = x1[0] Exp[eigenvalue[1] t]

and

y1[t] = y1[0] Exp[eigenvalue[1] t].

This gives

$$\{x1[t], y1[t]\} = \{x1[0], y1[0]\}\,\text{Exp}[\text{eigenvalue}[1]\,t]\,.$$

where {x1[0], y1[0]} is the starter point at the tip of eigenvector[1].

Check this out:

```
Clear[x1, y1, t]
{x1[t_], y1[t_]} = eigenvector[1] Exp[eigenvalue[1] t]
```

```
{0.961835 E^1.09604 t, 0.27363 E^1.09604 t}
```

```
{x1'[t], y1'[t]} == A . {x1[t], y1[t]}
```

```
True
```

```
{x1[0], y1[0]} == eigenvector[1]
```

```
True
```

This tells you that

$$\{x1[t], y1[t]\}$$

solves the linear system

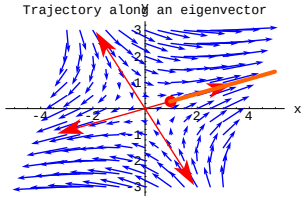
$$\{x1'[t], y1'[t]\} = A\,.\,\{x1[t], y1[t]\}.$$

with

$$\{x1[0], y1[0]\} = \text{eigenvector}[1].$$

See these formulas for x1[t] and y1[t] plot out this straight line trajectory:

```
endtime = 1.5;
eigentrajectoryplot =
 ParametricPlot[{x1[t], y1[t]}, {t, 0, endtime}, PlotStyle →
   {{CadmiumOrange, Thickness[0.015]}}, DisplayFunction → Identity];
starterplot = Graphics[{Red, PointSize[0.04], Point[{x1[0], y1[0]}]}];
Show[eigenplot, starterplot,
 eigentrajectoryplot, PlotRange → All, Axes → True,
 AxesLabel → {"x", "y"}, DisplayFunction → $DisplayFunction,
 PlotLabel → "Trajectory along an eigenvector"];
```



Similarly, the formulas for solutions coming from the straight line trajectory coming from eigenvector[2] are:

```
Clear[x2, y2, t]
{x2[t_], y2[t_]} = eigenvector[2] Exp[eigenvalue[2] t]
```

```
{-0.539111 E^-0.566039 t, 0.842235 E^-0.566039 t}
```

Check it out:

```
{x2'[t], y2'[t]} == A . {x2[t], y2[t]}
```

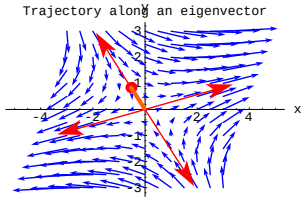
```
True
```

```
{x2[0], y2[0]} == eigenvector[2]
```

```
True
```

See it now:

```
endtime = 5.5;
eigentrajectoryplot =
 ParametricPlot[{x2[t], y2[t]}, {t, 0, endtime}, PlotStyle →
   {{CadmiumOrange, Thickness[0.015]}}, DisplayFunction → Identity];
starterplot = Graphics[{Red, PointSize[0.04], Point[{x2[0], y2[0]}]}];
Show[eigenplot, starterplot,
 eigentrajectoryplot, PlotRange → All, Axes → True,
 AxesLabel → {"x", "y"}, DisplayFunction → $DisplayFunction,
 PlotLabel → "Trajectory along an eigenvector"];
```



□B.3.a.iii) Formulas for other trajectories

Stay with the same linear system and matrix A.
Here are straight line solution pairs starting at the tips of each of the eigenvectors:

```
Clear[x1, y1, x2, y2, t]
{x1[t_], y1[t_]} = eigenvector[1] E^eigenvalue[1] t
{x2[t_], y2[t_]} = eigenvector[2] E^eigenvalue[2] t
```

```
{0.961835 E^1.09604 t, 0.27363 E^1.09604 t}
```

```
{-0.539111 E^-0.566039 t, 0.842235 E^-0.566039 t}
```

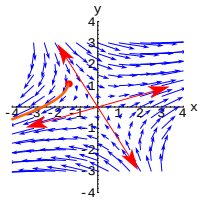
Now see how this random combination

$$C1\,\{x1[t], y1[t]\} + C2\,\{x2[t], y2[t]$$

of the straight line solution pairs plots out:

Here C1 and C2 are random constants.

```
C1 = Random[Real, {-2, 2}];
C2 = Random[Real, {-2, 2}];
Clear[x, y, t]
{x[t_], y[t_]} = C1 {x1[t], y1[t]} + C2 {x2[t], y2[t]};
endtime = 5;
curveplot = ParametricPlot[{x[t], y[t]}, {t, 0, endtime}, PlotStyle →
   {{Thickness[0.015], CadmiumOrange}}, DisplayFunction → Identity];
starterplot = Graphics[{PointSize[0.04], Red, Point[{x[0], y[0]}]}];
Show[eigenplot, starterplot, curveplot,
 PlotRange → {{-4, 4}, {-4, 4}}, Axes → True, AxesLabel → {"x", "y"},
 DisplayFunction → $DisplayFunction, PlotLabel → None];
{x[t], y[t]}
Eigenvalues[A]
```




```
{-0.798559 E^-0.566039 t - 0.574321 E^1.09604 t,
 1.24756 E^-0.566039 t - 0.163387 E^1.09604 t}
```

```
{1.09604, -0.566039}
```

The formula at the top is the parameterization of the plotted trajectory.

Below the formulas, the eigenvalues of the coefficient matrix are listed.

Rerun many times.

Damned if those curves don't look like trajectories!
Check out

$$\{x[t], y[t]\} = C1\,\{x1[t], y1[t]\} + C2\,\{x2[t], y2[t]$$

for cleared constants C1 and C2.

```
Clear[x, y, t, C1, C2]
{x[t_], y[t_]} = C1 {x1[t], y1[t]} + C2 {x2[t], y2[t]};
{x'[t], y'[t]} == Expand[A . {x[t], y[t]}]
```

```
True
```

What's the message?

□Answer:

The message is:

When you go with straight line trajectory solutions

$$\{x1[t],\ y1[t]\} \text{ and } \{x2[t],\ y2[t]\}$$

coming from the tips of eigenvector[1] and eigenvector[2],

and you go with and two constants C1 and C2 you like, then any combination {x[t], y[t]} like this:

```
Clear[x, y, t, C1, C2]
{x[t_], y[t_]} = C1 {x1[t], y1[t]} + C2 {x2[t], y2[t]}
```

```
{-0.539111 C2 E^-0.566039 t + 0.961835 C1 E^1.09604 t,
 0.842235 C2 E^-0.566039 t + 0.27363 C1 E^1.09604 t}
```

gives you a formula for a solution pair {x[t], y[t]} (a trajectory) of the linear system

$$\{x'[t], y'[t]\} = A\,.\,\{x[t], y[t]\}.$$

Check:

```
{x'[t], y'[t]} == Expand[A . {x[t], y[t]}]
```

```
True
```

You can explain why this happens this way:

$$A\,.\,\{x[t], y[t]\} = A\,.\,(C1\,\{x1[t], y1[t]\} + C2\,\{x2[t], y2[t]\})$$

$$= C1\,A.\{x1[t], y1[t]\} + C2\,A.\{x2[t], y2[t]\})$$
$$= C1\,\{x1'[t], y1'[t]\} + C2\,\{x2'[t], y2'[t]\}$$
$$= \{x'[t], y'[t]\}.$$

□B.3.a.iv) The formula for any trajectory

Stay with the same linear system and matrix A.

```
A = {{0.84, 0.90}, {0.40, -0.31}};
MatrixForm[A]

Clear[x, y, t]
ColumnForm[Thread[{x'[t], y'[t]} == A.{x[t], y[t]}]]
```

$$\begin{pmatrix} 0.84 & 0.9 \\ 0.4 & -0.31 \end{pmatrix}$$

$$x'[t] == 0.84\,x[t] + 0.9\,y[t]$$
$$y'[t] == 0.4\,x[t] - 0.31\,y[t]$$

Given starter data

$$\{x[0], y[0]\} = \{-0.8, 1.5\},$$

come up with formulas for the functions x[t] and y[t] that solve this linear system.

□Answer:

Calculate the eigenvalues and the eigenvectors:

```
Clear[eigenvector, eigenvalue]
{eigenvector[1], eigenvector[2]} = Eigenvectors[A]
{eigenvalue[1], eigenvalue[2]} = Eigenvalues[A]
```

{{0.961835, 0.27363}, {-0.539111, 0.842235}}

{1.09604, -0.566039}

One sucker and one propeller.

Set up formulas for the straight line trajectories starting at the tips of the two eigenvectors:

```
Clear[x1, y1, x2, y2, t]
{x1[t_], y1[t_]} = eigenvector[1] Exp[eigenvalue[1] t]
```

$$\{0.961835\,E^{1.09604\,t},\ 0.27363\,E^{1.09604\,t}\}$$

```
{x2[t_], y2[t_]} = eigenvector[2] Exp[eigenvalue[2] t]
```

$$\{-0.539111\,E^{-0.566039\,t},\ 0.842235\,E^{-0.566039\,t}\}$$

Make a combination of the two straight line formulas but use undetermined cconstant coefficients C1 and C2 this way:

```
Clear[C1, C2]
{x[t_], y[t_]} = C1 {x1[t], y1[t]} + C2 {x2[t], y2[t]}
```

$$\{-0.539111\,C2\,E^{-0.566039\,t} + 0.961835\,C1\,E^{1.09604\,t},\ 0.842235\,C2\,E^{-0.566039\,t} + 0.27363\,C1\,E^{1.09604\,t}\}$$

This trajectory starts at:

```
{x[0], y[0]}
```

{0.961835 C1 - 0.539111 C2, 0.27363 C1 + 0.842235 C2}

The given starter data are

$$\{x[0], y[0]\} = \{-0.8, 1.5\}.$$

```
starter = {-0.8, 1.5}
```

{-0.8, 1.5}

Determine the coefficients C1 and C2 so that

$$\{x[0], y[0]\} = \{-0.8, 1.5\}:$$

```
starterequation = {x[0], y[0]} == starter
```

{0.961835 C1 - 0.539111 C2, 0.27363 C1 + 0.842235 C2} == {-0.8, 1.5}

Solve for C1 and C2:

```
Csols = Solve[starterequation]
```

{{C1 → 0.14085, C2 → 1.73522}}

The formulas you are after are:

```
{x[t_], y[t_]} = {x[t], y[t]} /. Csols[[1]]
```

$$\{-0.935475\,E^{-0.566039\,t} + 0.135475\,E^{1.09604\,t},\ 1.46146\,E^{-0.566039\,t} + 0.0385408\,E^{1.09604\,t}\}$$

Check it out:

```
Expand[{x'[t], y'[t]}] == Expand[A.{x[t], y[t]}]
```

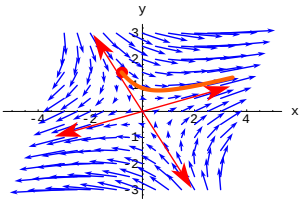
True

```
{x[0], y[0]} == starter
```

True

See it:

```
endtime = 3;
trajectoryplot =
 ParametricPlot[{x[t], y[t]}, {t, 0, endtime}, PlotStyle →
   {{CadmiumOrange, Thickness[0.015]}}, DisplayFunction → Identity];
starterplot = Graphics[{Red, PointSize[0.04], Point[{x[0], y[0]}]}];
Show[
 eigenplot, starterplot, trajectoryplot, PlotRange → All, Axes → True,
 AxesLabel → {"x", "y"}, DisplayFunction → $DisplayFunction];
```



Perfecto.

Play with other starting points.

B.4) Eigenvectors involving the imaginary number $I = \sqrt{-1}$ do not stand in your way; they are swirlers

Just when you thought you were getting good at this, someone flips you this linear system:

```
Clear[x, y, t]
linearsystem =
  {{x'[t], y'[t]} == {0.3 x[t] + 1.5 y[t], -1.7 x[t] + 0.1 y[t]});
ColumnForm[Thread[linearsystem]]
```

$$x'[t] == 0.3\,x[t] + 1.5\,y[t]$$
$$y'[t] == -1.7\,x[t] + 0.1\,y[t]$$

And asks you to come up with formulas for the functions x[t] and y[t] that solve this linear system with starter data

$$\{x[0], y[0]\} = \{-0.2,\ 0.3\}:$$

You read off the coefficient matrix A:

```
A = {{0.3, 1.5}, {-1.7, 0.1}};
MatrixForm[A]
```

$$\begin{pmatrix} 0.3 & 1.5 \\ -1.7 & 0.1 \end{pmatrix}$$

Intent on looking for the straight line trajectory formulas, you calculate the eigenvalues of A,

```
Clear[eigenvalue];
{eigenvalue[1], eigenvalue[2]} = Chop[Eigenvalues[A]]
```

{0.2 + 1.59374 I, 0.2 - 1.59374 I}

and the eigenvectors of A:

```
Clear[eigenvector];
{eigenvector[1], eigenvector[2]} = Chop[Eigenvectors[A]]
```

{{-0.0428746 - 0.683309 I, 0.728869}, {-0.0428746 + 0.683309 I, 0.728869}}

You inspect this carefully and note that the dreaded number

$$I = \sqrt{-1}$$

is sitting conspicuously in the output.

But when you use the code at the beginning of B.3) above to go after formulas for the solution pair {x[t], y[t]}, you get:

```
{xstarter, ystarter} = {-0.2, 0.3};
Clear[eigenvector, eigenvalue];
{eigenvalue[1], eigenvalue[2]} = Chop[Eigenvalues[A]];
{eigenvector[1], eigenvector[2]} = Chop[Eigenvectors[A]];

Clear[x, y, x1, y1, x2, y2, t, C1, C2]
{x1[t_], y1[t_]} = Chop[eigenvector[1] Exp[eigenvalue[1] t]];
{x2[t_], y2[t_]} = Chop[eigenvector[2] Exp[eigenvalue[2] t]];
{x[t_], y[t_]} = C1 {x1[t], y1[t]} + C2 {x2[t], y2[t]};

starter = {xstarter, ystarter};
starterequation = {x[0], y[0]} == starter;
Csols = Solve[starterequation];

{x[t_], y[t_]} = Chop[ComplexExpand[{x[t], y[t]} /. Csols[[1]]]]
```

$$\{-0.2\,E^{0.2\,t}\,\text{Cos}[1.59374\,t] + 0.269806\,E^{0.2\,t}\,\text{Sin}[1.59374\,t],\ 0.3\,E^{0.2\,t}\,\text{Cos}[1.59374\,t] + 0.194511\,E^{0.2\,t}\,\text{Sin}[1.59374\,t]\}$$

Exponentials combined with sines and cosines.

Compare the formula with the eigenvalues of the coefficient matrix:

```
Eigenvalues[A]
```

{0.2 + 1.59374 I, 0.2 - 1.59374 I}

If you want to see why these formulas came out the way they did, then jump into what follows.

□B.4.a.i) Formulas for trajectories

Stay with the same linear system

$\{x'[t], y'[t]\} = A.\{x[t].y[t]\}$:

```
Clear[x, y, t]
linearsystem =
  ({x'[t], y'[t]} == {0.3 x[t] + 1.5 y[t], -1.7 x[t] + 0.1 y[t]});
ColumnForm[Thread[linearsystem]]
```

```
x'[t] == 0.3 x[t] + 1.5 y[t]
y'[t] == -1.7 x[t] + 0.1 y[t]
```

Look again at the eigenvectors of the coefficient matrix A:

```
Clear[eigenvector];
{eigenvector[1], eigenvector[2]} = Chop[Eigenvectors[A]]
```

```
{{-0.0428746 - 0.683309 I, 0.728869}, {-0.0428746 + 0.683309 I, 0.728869}}
```

and eigenvalues of A:

```
Clear[eigenvalue];
{eigenvalue[1], eigenvalue[2]} = Chop[Eigenvalues[A]]
```

```
{0.2 + 1.59374 I, 0.2 - 1.59374 I}
```

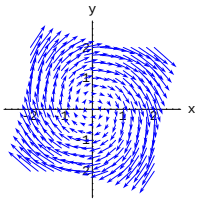
How do you interpret the presence of the complex numbers involving $I = \sqrt{-1}$?

□Answer:

The presence of the complex numbers involving $I = \sqrt{-1}$ tells you that the straight line trajectories along the eigenvectors are strictly imaginary. In the real world, there are no straight line trajectories.

Check this out:

```
Clear[Field]
Field[x_, y_] = A.{x, y};
scalefactor = 0.2;
flowplot = Table[Arrow[scalefactor Field[x, y],
   Tail → {x, y}, VectorColor → Blue, HeadSize -> 0.2],
  {x, -2, 2, 0.25}, {y, -2, 2, 0.25}];
Show[flowplot, Axes → True, AxesLabel → {"x", "y"}];
```



All the trajectories swirling like all get-out.

No straight line trajectories here.

And this was predicted by the presence of $I = \sqrt{-1}$ in the output from:

```
{eigenvalue[1], eigenvalue[2]}
```

```
{0.2 + 1.59374 I, 0.2 - 1.59374 I}
```

The upshot:

The presence of $I = \sqrt{-1}$ in eigenvalues indicates that the eigenvectors are what DiffEq&*Mathematica* folks call "swirlers."

□B.4.a.iii) Propelling swirlers

Stay with the same linear system.

```
ColumnForm[Thread[{x'[t], y'[t]} == A.{x[t], y[t]}]]
```

```
x'[t] == 0.3 x[t] + 1.5 y[t]
y'[t] == -1.7 x[t] + 0.1 y[t]
```

In spite of the imaginary numbers in the eigenvectors and eigenvalues, come up with formulas for functions x[t] and y[t] that solve this linear system with the starter data

$\{x[0], y[0]\} = \{-0.2, 0.3\}$

given above.

□Answer:

This is just a copy, paste and edit job on B.3.

Don't worry about the imaginary numbers and go with them as if they were ordinary numbers:

```
Clear[eigenvector];
{eigenvector[1], eigenvector[2]} = Chop[Eigenvectors[A]]
```

```
{{-0.0428746 - 0.683309 I, 0.728869}, {-0.0428746 + 0.683309 I, 0.728869}}
```

```
Clear[eigenvalue];
{eigenvalue[1], eigenvalue[2]} = Chop[Eigenvalues[A]]
```

```
{0.2 + 1.59374 I, 0.2 - 1.59374 I}
```

Set up formulas for the (imaginary) straight line trajectories starting at the tips of the two eigenvectors:

```
Clear[x1, y1, x2, y2, t]
{x1[t_], y1[t_]} = Chop[eigenvector[1] E^(eigenvalue[1] t)]
```

```
{(-0.0428746 - 0.683309 I) E^((0.2+1.59374 I) t), 0.728869 E^((0.2+1.59374 I) t)}
```

```
If this output bothers you, take the Tutorial on the complex exponential.
```

```
{x2[t_], y2[t_]} = Chop[eigenvector[2] E^(eigenvalue[2] t)]
```

```
{(-0.0428746 + 0.683309 I) E^((0.2-1.59374 I) t), 0.728869 E^((0.2-1.59374 I) t)}
```

```
If this output bothers you, take the Tutorial on the complex exponential.
```

Make a combination of the two (imaginary) straight line formulas but use undetermined coefficients C1 and C2 this way:

```
Clear[x, y, t, C1, C2]
{x[t_], y[t_]} = C1 {x1[t], y1[t]} + C2 {x2[t], y2[t]}
```

```
{(-0.0428746 + 0.683309 I) C2 E^((0.2-1.59374 I) t) -
   (0.0428746 + 0.683309 I) C1 E^((0.2+1.59374 I) t),
  0.728869 C2 E^((0.2-1.59374 I) t) + 0.728869 C1 E^((0.2+1.59374 I) t)}
```

This (yet undetermined) trajectory starts at:

```
{x[0], y[0]}
```

```
{(-0.0428746 - 0.683309 I) C1 - (0.0428746 - 0.683309 I) C2,
 0.728869 C1 + 0.728869 C2}
```

The given starter data are

$\{x[0], y[0]\} = \{-0.2, 0.3\}$.

```
starter = {-0.2, 0.3}
```

```
{-0.2, 0.3}
```

Determine the coefficients C1 and C2 so that

$\{x0], y[0]\} =$ starter:

```
starterequation = ({x[0], y[0]} == starter)
Csols = Solve[starterequation]
```

```
{(-0.0428746 - 0.683309 I) C1 - (0.0428746 - 0.683309 I) C2,
  0.728869 C1 + 0.728869 C2} ==
 {-0.2, 0.3}
```

```
{{C1 → 0.205798 - 0.133434 I, C2 → 0.205798 + 0.133434 I}}
```

Use these values of C1 and C2 to nail down the solution pair {x[t], y[t]} with {x[0], y[0]} = starter:

```
{x[t_], y[t_]} = {x[t], y[t]} /. Csols[[1]]
```

```
{(-0.1 + 0.134903 I) E^((0.2-1.59374 I) t) - (0.1 + 0.134903 I) E^((0.2+1.59374 I) t),
 (0.15 + 0.0972556 I) E^((0.2-1.59374 I) t) + (0.15 - 0.0972556 I) E^((0.2+1.59374 I) t)}
```

Clean this up by having *Mathematica* apply the fundamental identity

$E^{(a+Ib)t} = E^{at}(\text{Cos}[b] + i\,\text{Sin}[b])$:

```
Clear[cleanx, cleany]
{cleanx[t_], cleany[t_]} = Chop[ComplexExpand[{x[t], y[t]}]]
```

```
{-0.2 E^(0.2 t) Cos[1.59374 t] + 0.269806 E^(0.2 t) Sin[1.59374 t],
 0.3 E^(0.2 t) Cos[1.59374 t] + 0.194511 E^(0.2 t) Sin[1.59374 t]}
```

Check these formulas out:

```
Expand[{cleanx'[t], cleany'[t]}] == Expand[A.{cleanx[t], cleany[t]}]
```

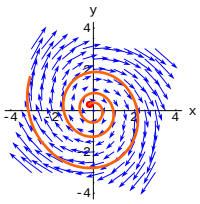
```
True
```

```
{cleanx[0], cleany[0]} == starter
```

```
True
```

See the flow plot and the corresponding trajectory:

```
flowplot = Table[Arrow[Field[x, y], Tail → {x, y},
    VectorColor → Blue, ScaleFactor → 0.2, HeadSize → 0.35],
   {x, -3, 3, 6/12}, {y, -3, 3, 6/12}];
endtime = 11.5;
trajectoryplot =
 ParametricPlot[{x[t], y[t]}, {t, 0, endtime}, PlotStyle →
   {{CadmiumOrange, Thickness[0.015]}}, DisplayFunction → Identity];
starterplot = Graphics[{Red, PointSize[0.04], Point[{x[0], y[0]}]}];
Show[
 flowplot, starterplot, trajectoryplot, PlotRange → All, Axes → True,
 AxesLabel → {"x", "y"}, DisplayFunction → $DisplayFunction];
```



Life in the fast lane.

Look at that spiral!

Take another look at the formula for {x[t], y[t]}:

```
{cleanx[t], cleany[t]}
```

$\{-0.2\,E^{0.2\,t}\,\text{Cos}[1.59374\,t] + 0.269806\,E^{0.2\,t}\,\text{Sin}[1.59374\,t],\ 0.3\,E^{0.2\,t}\,\text{Cos}[1.59374\,t] + 0.194511\,E^{0.2\,t}\,\text{Sin}[1.59374\,t]\}$

Look at the eigenvalues of the coefficient matrix:

```
Eigenvalues[A]
```

$\{0.2 + 1.59374\,I,\ 0.2 - 1.59374\,I\}$

The positive exponent on the exponential explains why the trajectory is propelled away from {0, 0}.

Do you see that number anywhere in the eigenvalues?

The sine and cosine terms explain why the trajectory oscillates around {0, 0}.

Do you see the numbers inside the cosine and sine terms anywhere in the eigenvalues?

The spiral results from the combined effects of the exponential terms and the sine and cosine terms.

This is why DiffEq&*Mathematica* folks call the eigenvectors propelling swirlers.

B.5) Eigenvalue-trajectory analysis summary:

Propellers, suckers, pure swirlers, sucking swirlers, and propelling swirlers

□B.5.a.i) Two Propellers:

Two positive eigenvalues

If you find that both eigenvalues of the coefficient matrix A are positive, then you already know that both eigenvectors are propellers and that all the trajectories are propelled away from {0, 0}.
How do the formulas back this up?

□Answer:

The formulas back this up.

For instance if

eigenvalue[1] of A is 0.9

and

eigenvalue[2] of A is 3.0,

then every trajectory is given by the formula

$$\{x[t],\ y[t]\} = C1\ \text{eigenvector[1]}\ E^{0.9\,t} + C2\ \text{eigenvector[2]}\ E^{3.0\,t}$$

The exponential terms

$E^{0.9\,t}$ and $E^{3.0\,t}$

propel the trajectories away from {0, 0}.

The result: All solutions go to infinity as t becomes very large.

See two propellers in action:

```
Clear[x, y, t]
linearsystem =
 ({x'[t], y'[t]} == {1.3 x[t] + 0.5 y[t], -0.3 x[t] + 1.2 y[t]});
ColumnForm[Thread[linearsystem]]
```

$x'[t] == 1.3\,x[t] + 0.5\,y[t]$
$y'[t] == -0.3\,x[t] + 1.2\,y[t]$

```
A = {{1.3, 0.5}, {0.3, 1.2}};
MatrixForm[A]
```

$\begin{pmatrix} 1.3 & 0.5 \\ 0.3 & 1.2 \end{pmatrix}$

Check the eigenvalues of the coefficient matrix A:

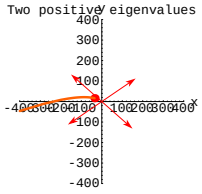
```
Clear[eigenvalue]
{eigenvalue[1], eigenvalue[2]} = Eigenvalues[A]
```

{1.64051, 0.859488}

Both eigenvalues are positive. This means that both eigenvectors are propellers. See some random trajectories with scaled eigenvectors:

```
{xstarter, ystarter} =
 {Random[Real, {-40, 40}], Random[Real, {-40, 40}]};
endtime = 7;
starterpoint = {xstarter, ystarter};
Clear[x, y, t]
{x[t_], y[t_]} =
 Chop[ComplexExpand[Expand[MatrixExp[A t] . {xstarter, ystarter}]]]
trajectoryplot =
 ParametricPlot[{x[t], y[t]}, {t, 0, endtime}, PlotStyle →
   {{CadmiumOrange, Thickness[0.015]}}, DisplayFunction → Identity];
starterplot = Graphics[{Red, PointSize[0.05], Point[starterpoint]}];

Clear[eigenvector]
{eigenvector[1], eigenvector[2]} = Eigenvectors[A];
scalefactor = 200;
headsize = 50;
eigenplot = {Arrow[eigenvector[1], Tail → {0, 0}, VectorColor → Red,
   ScaleFactor → scalefactor, HeadSize → headsize],
  Arrow[-eigenvector[1], Tail → {0, 0}, VectorColor → Red,
   ScaleFactor → scalefactor, HeadSize → headsize],
  Arrow[eigenvector[2], Tail → {0, 0}, VectorColor → Red,
   ScaleFactor → scalefactor, HeadSize → headsize],
  Arrow[-eigenvector[2], Tail → {0, 0}, VectorColor → Red,
   ScaleFactor → scalefactor, HeadSize → headsize]};

Show[starterplot, trajectoryplot,
 eigenplot, PlotRange → {{-2 scalefactor, 2 scalefactor},
   {-2 scalefactor, 2 scalefactor}}, Axes → True,
 AxesLabel → {"x", "y"}, PlotLabel → "Two positive eigenvalues",
 DisplayFunction → $DisplayFunction];
```

$\{-25.8525\,E^{0.859488\,t} - 9.43947\,E^{1.64051\,t},\ 22.7767\,E^{0.859488\,t} - 6.42852\,E^{1.64051\,t}\}$



Rerun many times.

Off we go into the wild orange yonder.

After you run this several times, you will be able to detect which eigenvector corresponds to the larger eigenvalue.

□B.5.a.ii) Two suckers:

Two negative eigenvalues guarantee {x[t], y[t]} is sucked to {0, 0}

no matter what the starting data on {x[0], y[0]} are.

If you find that both eigenvalues of A are negative, then you already know that both eigenvectors are suckers and all the trajectories are sucked toward {0, 0}.
This means that if x[t] and y[t] solve the linear system, then you are guaranteed that as t gets large

$\{x[t],\ y[t]\} \to \{0,\ 0\}$

no matter what the starting data on {x[0], y[0]} are.
How do the formulas back this up?

□Answer:

The formulas back this up.

For instance if

eigenvalue[1] of A is −0.9

and

eigenvalue[2] of A is −3.0,

then every trajectory is given by the formula

$\{x[t],\ y[t]\} = C1\ \text{eigenvector[1]}\ E^{-0.9\,t} + C2\ \text{eigenvector[2]}\ E^{-3.0\,t}$

The exponential terms

$E^{-0.9\,t}$ and $E^{-3.0\,t}$

suck the trajectories toward the drain at {0, 0}.

This forces

$\{x[t], y[t]\} \to \{0, 0\}$ as $t \to \infty$

no matter what the starting data on {x[0], y[0]} are.

The result: All solutions end at {0, 0} when t is sufficiently large.

See two suckers in action:

```
Clear[x, y, t]
linearsystem =
 ({x'[t], y'[t]} == {-1.3 x[t] + 0.5 y[t], 0.3 x[t] - 1.2 y[t]});
ColumnForm[Thread[linearsystem]]
```

$x'[t] == -1.3\,x[t] + 0.5\,y[t]$
$y'[t] == 0.3\,x[t] - 1.2\,y[t]$

Read off the coefficient matrix:

```
A = {{-1.3, 0.5}, {0.3, -1.2}};
MatrixForm[A]
```

$\begin{pmatrix} -1.3 & 0.5 \\ 0.3 & -1.2 \end{pmatrix}$

Check the eigenvalues of the coefficient matrix A:

```
Clear[eigenvalue]
{eigenvalue[1], eigenvalue[2]} = Eigenvalues[A]
```

{-1.64051, -0.859488}

Down the drain at {0, 0}.

After you run this several times, you will be able to spot the eigenvector that corresponds to the most negative eigenvalue.

□B.5.a.iii) A propeller and a sucker:

One positive and one negative eigenvalue

If you find that one of the eigenvalues of A is positive and the other eigenvalue of A is negative, then you already know that one eigenvectors is a propeller, the other eigenvector is a sucker and that almost all trajectories end up being propelled away from {0, 0}, but some of the trajectories are sucked toward {0, 0} before they begin their journeys away from {0, 0}.
How do the formulas back this up?

□Answer:

The formulas back this up.

For instance if

eigenvalue[1] of A is −4.8

and

eigenvalue[2] of A is 2.7,

then every trajectory is given by the formula

$\{x[t],\ y[t]\} = C1\ \text{eigenvector[1]}\ E^{-4.8\,t} + C2\ \text{eigenvector[2]}\ E^{2.7\,t}$

As t continues to advance from 0, the exponential term

$E^{-4.8\,t}$

eventually decays to 0 but the the exponential term

$E^{2.7\,t}$

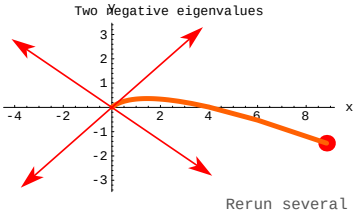
blasts the trajectories into outerspace in the direction of eigenvector[2] (unless C2 = 0).

Both eigenvalues are negative. This means that both eigenvectors are suckers.

See some random trajectories with scaled eigenvectors:

```
ranger = 10;
{xstarter, ystarter} =
 {Random[Real, {-ranger, ranger}], Random[Real, {-ranger, ranger}]};
Clear[x, y, t]
{x[t_], y[t_]} =
 Chop[ComplexExpand[Expand[MatrixExp[A t] . {xstarter, ystarter}]]]
trajectoryplot =
 ParametricPlot[{x[t], y[t]}, {t, 0, endtime}, PlotStyle →
   {{CadmiumOrange, Thickness[0.015]}}, DisplayFunction → Identity];
starterpoint = {xstarter, ystarter};
starterplot = Graphics[{Red, PointSize[0.05], Point[starterpoint]}];
scaler = ranger/3;
Clear[eigenvector]
{eigenvector[1], eigenvector[2]} = Eigenvectors[A];
scalefactor = 5;
headsize = 1;
eigenplot = {Arrow[eigenvector[1], Tail → {0, 0}, VectorColor → Red,
   ScaleFactor → scalefactor, HeadSize → headsize],
  Arrow[-eigenvector[1], Tail → {0, 0}, VectorColor → Red,
   ScaleFactor → scalefactor, HeadSize → headsize],
  Arrow[eigenvector[2], Tail → {0, 0}, VectorColor → Red,
   ScaleFactor → scalefactor, HeadSize → headsize],
  Arrow[-eigenvector[2], Tail → {0, 0}, VectorColor → Red,
   ScaleFactor → scalefactor, HeadSize → headsize]};
Show[
 starterplot, trajectoryplot, eigenplot, PlotRange → All, Axes → True,
 AxesLabel → {"x", "y"}, PlotLabel → "Two negative eigenvalues",
 DisplayFunction → $DisplayFunction];
```

$\{5.93\,E^{-1.64051\,t} + 2.91135\,E^{-0.859488\,t},\ -4.03848\,E^{-1.64051\,t} + 2.56497\,E^{-0.859488\,t}\}$



Rerun several times.

The result: You can expect the trajectories to go to infinity as t becomes very large, but some may move towards {0, 0} first, under the influence of the sucking eigenvector.

See a propeller and a sucker in combined action:

```
Clear[x, y, t]
linearsystem =
 ({x'[t], y'[t]} == {0.8 x[t] + 0.2 y[t], 0.4 x[t] - 1.2 y[t]});
ColumnForm[Thread[linearsystem]]

A = {{0.8, 0.2}, {0.4, -1.2}};
MatrixForm[A]
```

$x'[t] == 0.8\,x[t] + 0.2\,y[t]$
$y'[t] == 0.4\,x[t] - 1.2\,y[t]$

$\begin{pmatrix} 0.8 & 0.2 \\ 0.4 & -1.2 \end{pmatrix}$

Check the eigenvalues of the coefficient matrix A:

```
Clear[eigenvalue]
{eigenvalue[1], eigenvalue[2]} = Eigenvalues[A]
```

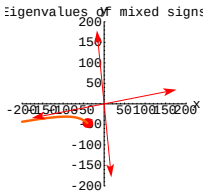
{-1.23923, 0.83923}

Mixed signs. This means one eigenvector is a propeller and the other is a sucker. See some random trajectoies shown with the eigenvectors:

```
ranger = 200;
{xstarter, ystarter} = {Random[Real, {-ranger / 2, ranger / 2}],
          Random[Real, {-ranger / 2, ranger / 2}]};
endtime = 9;

Clear[x, y, t]
{x[t_], y[t_]} =
 Chop[ComplexExpand[Expand[MatrixExp[A t] . {xstarter, ystarter}]]]

trajectoryplot = ParametricPlot[{x[t], y[t]}, {t, 0, endtime},
   PlotStyle -> {{CadmiumOrange, Thickness[0.015]}},
       DisplayFunction -> Identity];

starterpoint = {xstarter, ystarter};
   starterplot = Graphics[{Red, PointSize[0.06],
         Point[starterpoint]}];
```

```
scaler = 0.9 ranger;
Clear[eigenvector]
{eigenvector[1], eigenvector[2]} = Eigenvectors[A];
eigenplot =
 {Arrow[scaler eigenvector[1], Tail -> {0, 0}, VectorColor -> Red],
   Arrow[-scaler eigenvector[1], Tail -> {0, 0}, VectorColor -> Red],
   Arrow[scaler eigenvector[2], Tail -> {0, 0}, VectorColor -> Red],
   Arrow[-scaler eigenvector[2], Tail -> {0, 0}, VectorColor -> Red]};
Show[starterplot, trajectoryplot, eigenplot,
  PlotRange -> {{-ranger, ranger}, {-ranger, ranger}},
 Axes -> True, AxesLabel -> {"x", "y"},
  DisplayFunction -> $DisplayFunction,
 PlotLabel -> "Eigenvalues of mixed signs"];
```

$\{3.72675\, E^{-1.23923\, t} - 44.7886\, E^{0.83923\, t},\ -37.9985\, E^{-1.23923\, t} - 8.78539\, E^{0.83923\, t}\}$



Rerun several times.

After you run this several times, you will be able to spot the propelling eigenvectors that correspond to the the positive eigenvalue but occasionally you will see the initial influence of the sucker.

□B.5.a.iv) Two pure swirlers:

Eigenvalues p + I q and p − I q with p = 0 and q not 0

If you find that the eigenvalues of A are of the form:

p + I q and p − I q with p = 0 and q not 0,

then you know that both eigenvectors are pure swirlers and that all the trajectories will oscillate on ellipses centered on {0, 0}.

How do the formulas back this up?

□**Answer:**

The formulas back this up.

For instance if

eigenvalue[1] of A is 0 + 2.3 I

and

eigenvalue[2] of A is 0 − 2.3 I,

then every trajectory is given by the formula

{x[t], y[t]}

= C1 eigenvector[1] $E^{2.3\, I\, t}$ + C2 eigenvector[2] $E^{-2.3\, I\, t}$

The exponents involving $I = \sqrt{-1}$ tell you that the trajectories are purely oscillatory. To see why look at:

```
Clear[t];
ComplexExpand[E^(2.3 I t)]
```

Cos[2.3 t] + I Sin[2.3 t]

```
ComplexExpand[E^(-2.3 I t)]
```

Cos[2.3 t] - I Sin[2.3 t]

This tells you that the ingredients in the formula are really combinations of Sin[2.3 t] and Cos[2.3 t].

This explains why the trajectories must oscillate along ellipses centered at {0, 0} and why individual solutions are genuine sine and cosine waves.

This is why the corresponding eigenvectors are called swirlers.

The result: All trajectories oscillate around {0, 0} in the same track, and all solutions oscillate with constant amplitude.

See pure swirlers in action:

```
Clear[x, y, t]
linearsystem =
 ({x'[t], y'[t]} == {-1.2 x[t] + 1.5 y[t], -2.3 x[t] + 1.2 y[t]});
ColumnForm[Thread[linearsystem]]
A = {{-1.2, 1.5}, {-2.3, 1.2}};
MatrixForm[A]
```

$x'[t] == -1.2\, x[t] + 1.5\, y[t]$

$y'[t] == -2.3\, x[t] + 1.2\, y[t]$

$\begin{pmatrix} -1.2 & 1.5 \\ -2.3 & 1.2 \end{pmatrix}$

Check the eigenvalues of A:

```
Clear[eigenvalue]
{eigenvalue[1], eigenvalue[2]} = Eigenvalues[A]
```

{0. + 1.41774 I, 0. - 1.41774 I}

The form is p + I q with

p = 0 and q not 0.

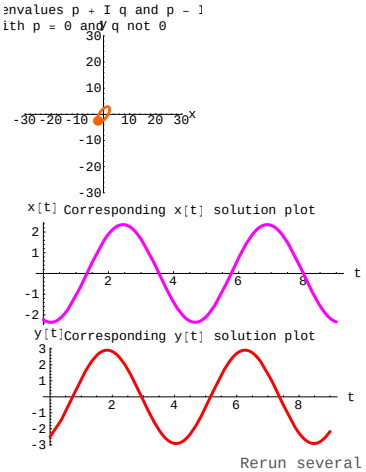
This means that the eigenvectors are pure whirlers (no suck and no propell).

The same result; the trajectories oscillate around{0, 0} on ellipses. And the solution plots are sine waves.

See some trajectories followed by the corresponding individual solution plots:

```
ranger = 30;
{xstarter, ystarter} = {Random[Real, {-ranger/2, ranger/2}],
  Random[Real, {-ranger/2, ranger/2}]};
Clear[x, y, t]
Clear[x, y, t]
{x[t_], y[t_]} =
 Chop[ComplexExpand[Expand[MatrixExp[A t] . {xstarter, ystarter}]]]
trajectoryplot =
 ParametricPlot[{x[t], y[t]}, {t, 0, endtime}, PlotStyle ->
   {{CadmiumOrange, Thickness[0.015]}}, DisplayFunction -> Identity];
starterpoint = {xstarter, ystarter};
starterplot =
 Graphics[{CadmiumOrange, PointSize[0.06], Point[starterpoint]}];
Show[starterplot, trajectoryplot,
 PlotRange -> {{-ranger, ranger}, {-ranger, ranger}}, Axes -> True,
 AxesLabel -> {"x", "y"}, DisplayFunction -> $DisplayFunction,
 PlotLabel ->
  "Eigenvalues p + I q and p - I q \n  with p=0 and q not 0"];
xsolutionplot =
 Plot[x[t], {t, 0, endtime}, PlotStyle -> {{Magenta, Thickness[0.01]}},
  AspectRatio -> 1/3, AxesLabel -> {"t", "x[t]"},
  PlotLabel -> "Corresponding x[t] solution plot"];
ysolutionplot =
 Plot[y[t], {t, 0, endtime}, PlotStyle -> {{Red, Thickness[0.01]}},
  AspectRatio -> 1/3, AxesLabel -> {"t", "y[t]"},
  PlotLabel -> "Corresponding y[t] solution plot"];
```

{-2.23211 Cos[1.41774 t] - 0.758084 Sin[1.41774 t],
 -2.5022 Cos[1.41774 t] + 1.50324 Sin[1.41774 t]}



Rerun several times.

Elliptical trajectories and corresponding sine wave solutions everytime.

□B.5.a.v) Two sucking swirlers:

Eigenvalues p + I q and p − I q with p < 0 and q not 0 guarantee {x[t], y[t]} is sucked to {0, 0} no matter what the starting data on {x[0], y[0]} are.

If you find that the eigenvalues of A are of the form:

p + I q and p − I q with p < 0 and q not 0

then you know that both eigenvectors are sucking swirlers and that all the trajectories will spiral toward {0, 0}, so that all solutions are sucked to 0 no matter what the starting data are.
How do the formulas back this up?

□Answer:

The formulas back this up.

For instance if

eigenvalue[1] of A is −2.7 + 4.2 I

and

eigenvalue[1] of A is −2.7 − 4.2 I,

then every trajectory is given by the formula

{x[t], y[t]} = C1 eigenvector[1] $E^{(-2.7+4.2\,I)\,t}$ + eigenvector[2] $E^{(-2.7$

The exponents involving

−2.7 + 4.2 I and −2.7 − 4.2 I

tell you that the trajectories are sucked to {0, 0} and are compelled to spiral as they are sucked. To see why look at:

```
Clear[t]
ComplexExpand[E^((-2.7+4.2 I) t)]
```

$E^{-2.7\,t}\,\text{Cos}[4.2\,t] + I\,E^{-2.7\,t}\,\text{Sin}[4.2\,t]$

```
Clear[t]
ComplexExpand[E^((-2.7-4.2 I) t)]
```

$E^{-2.7\,t}\,\text{Cos}[4.2\,t] - I\,E^{-2.7\,t}\,\text{Sin}[4.2\,t]$

This tells you that the terms in the formula

{x[t], y[t]} = C1 eigenvector[1] $E^{(-2.7+4.2\,I)\,t}$ + eigenvector[2] $E^{(-2}$

are sucked to {0, 0} by the

$E^{-2.7\,t}$

factors and they are compelled to oscillate (spiral) along the way by the

Sin[4.2 t] and Cos[4.2 t]

factors.

The result: The trajectories spiral into {0, 0}. And the solution plots are damped sine waves.

See sucking swirlers in action:

```
Clear[x, y, t]
linearsystem =
 ({x'[t], y'[t]} == {-1.2 x[t] + 1.5 y[t], -2.3 x[t] + 0.9 y[t]});
ColumnForm[Thread[linearsystem]]

A = {{-1.2, 1.5}, {-2.3, 0.9}};
MatrixForm[A]
```

x′[t] == −1.2 x[t] + 1.5 y[t]
y′[t] == −2.3 x[t] + 0.9 y[t]

$\begin{pmatrix} -1.2 & 1.5 \\ -2.3 & 0.9 \end{pmatrix}$

Check the eigenvalues of the coefficient matrix A:

```
Clear[eigenvalue]
{eigenvalue[1], eigenvalue[2]} = Eigenvalues[A]
```

{−0.15 + 1.53216 I, −0.15 − 1.53216 I}

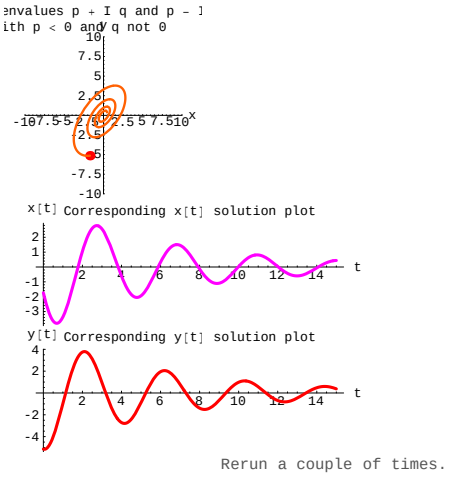
The form is

p + I q with p < 0 and q not 0.

This means that the eigenvectors are sucking swirlers. See some random trajectories followed by individual corrersponding solution plots:

```
ranger = 10;
{xstarter, ystarter} = {Random[Real, {-0.8 ranger, 0.8 ranger}],
          Random[Real, {-0.8 ranger, 0.8 ranger}]};
endtime = 15;

Clear[x, y, t]
{x[t_], y[t_]} =
  Chop[ComplexExpand[Expand[MatrixExp[A t] . {xstarter, ystarter}]]]

trajectoryplot = ParametricPlot[{x[t], y[t]}, {t, 0, endtime},
   PlotStyle -> {{CadmiumOrange, Thickness[0.015]}},
        DisplayFunction -> Identity];

starterpoint = {xstarter, ystarter};
   starterplot = Graphics[{Red, PointSize[0.06],
         Point[starterpoint]}];

Show[starterplot, trajectoryplot,
  PlotRange -> {{-ranger, ranger}, {-ranger, ranger}},
 Axes -> True, AxesLabel -> {"x", "y"},
  DisplayFunction -> $DisplayFunction,
  PlotLabel ->
  "Eigenvalues p + I q and p - I q \n  with p < 0 and q not 0"];

xsolutionplot =
 Plot[x[t], {t, 0, endtime}, PlotStyle -> {{Magenta, Thickness[0.01]}},
     AspectRatio -> 1 / 3, AxesLabel -> {"t", "x[t]"},
     PlotLabel -> "Corresponding x[t] solution plot"];

ysolutionplot =
 Plot[y[t], {t, 0, endtime}, PlotStyle -> {{Red, Thickness[0.01]}},
   AspectRatio -> 1 / 3, AxesLabel -> {"t", "y[t]"},
    PlotLabel -> "Corresponding y[t] solution plot"];
```

{−1.73504 $E^{-0.15\,t}$ Cos[1.53216 t] − 3.84513 $E^{-0.15\,t}$ Sin[1.53216 t],
−5.14209 $E^{-0.15\,t}$ Cos[1.53216 t] − 0.919355 $E^{-0.15\,t}$ Sin[1.53216 t]}



Rerun a couple of times.

Inward spiralling trajectories and corresponding damped sine wave solutions everytime.

As you can see, no matter what trajectory {x[t], y[t]} you ride, {x[t], y[t]} is eventually sucked to {0, 0}.

□B.5.a.vi) Two propelling swirlers:

Eigenvalues p + I q and p − I q with p > 0 and q not 0

If you find that the eigenvalues of A are

p + I q and p − I q with p > 0 and q not 0

then you know that both eigenvectors are propelling swirlers and that all the trajectories will spiral away from {0, 0}.
How do the formulas back this up?

□Answer:

The formulas back this up.

For instance if

eigenvalue[1] of A is 2.1 + 3.5 I

and

eigenvalue[1] of A is 2.1 + 3.5 I,

then every trajectory is given by the formula:

$$\{x[t],\ y[t]\} = C1\ \text{eigenvector[1]}\ E^{(2.1+3.5\,I)\,t} + \text{eigenvector[2]}\ E^{(2.1-3.5\,I)\,t}$$

The exponents involving 2.1 + 3.5 I and 2.1 − 3.5 I tell you that the trajectories are propelled away from {0, 0} and are forced to spiral. To see why look at:

```
Clear[t];
ComplexExpand[E^((2.1+3.5 I) t)]
```

$E^{2.1\,t}\,\text{Cos}[3.5\,t] + I\,E^{2.1\,t}\,\text{Sin}[3.5\,t]$

```
Clear[t];
ComplexExpand[E^((2.1-3.5 I) t)]
```

$E^{2.1\,t}\,\text{Cos}[3.5\,t] - I\,E^{2.1\,t}\,\text{Sin}[3.5\,t]$

This tells you that the terms in the formula

$$\{x[t],\ y[t]\} = C1\ \text{eigenvector[1]}\ E^{(2.1+3.5\,I)\,t} + \text{eigenvector[2]}\ E^{(2.1-3.5\,I)\,t}$$

are propelled away from {0, 0} by the

$E^{2.1\,t}$

factors and they are compelled to oscillate (spiral) by the

Sin[3.5 t] and Cos[3.5 t]

factors.

The result: The trajectories spiral away from {0, 0}.

See propelling swirlers in action:

```
Clear[x, y, t]
linearsystem =
 ({x'[t], y'[t]} == {-0.9 x[t] + 1.5 y[t], -2.3 x[t] + 1.2 y[t]});
ColumnForm[Thread[linearsystem]]
A = {{-0.9, 1.5}, {-2.3, 1.2}};
MatrixForm[A]
```

$x'[t] == -0.9\,x[t] + 1.5\,y[t]$
$y'[t] == -2.3\,x[t] + 1.2\,y[t]$

$$\begin{pmatrix} -0.9 & 1.5 \\ -2.3 & 1.2 \end{pmatrix}$$

Check the eigenvalues of A:

```
Clear[eigenvalue]
{eigenvalue[1], eigenvalue[2]} = Eigenvalues[A]
```

{0.15 + 1.53216 I, 0.15 − 1.53216 I}

The form is

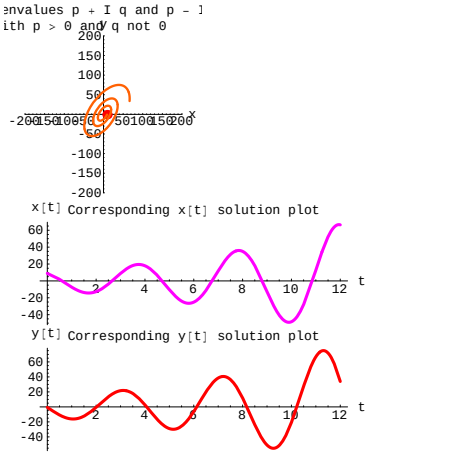
p + I q with p > 0 and q not 0.

This means that the eigenvectors are propelling swirlers. See some random trajectories followed by corresponding individual solution plots:

```
ranger = 200;
{xstarter, ystarter} = {Random[Real, {-ranger / 8, ranger / 8}],
          Random[Real, {-ranger / 8, ranger / 8}]};
endtime = 12;
Clear[x, y, t]
{x[t_], y[t_]} =
  Chop[ComplexExpand[Expand[MatrixExp[A t] . {xstarter, ystarter}]]]


trajectoryplot = ParametricPlot[{x[t], y[t]}, {t, 0, endtime},
   PlotStyle -> {{CadmiumOrange, Thickness[0.015]}},
       DisplayFunction -> Identity];

starterpoint = {xstarter, ystarter};
   starterplot = Graphics[{Red, PointSize[0.06],
         Point[starterpoint]}];
```

```
Show[starterplot, trajectoryplot,
  PlotRange -> {{-ranger, ranger}, {-ranger, ranger}},
 Axes -> True, AxesLabel -> {"x", "y"},
  DisplayFunction -> $DisplayFunction,
  PlotLabel ->
  "Eigenvalues p + I q and p - I q \n  with p > 0 and q not 0"];

xsolutionplot =
 Plot[x[t], {t, 0, endtime}, PlotStyle -> {{Magenta, Thickness[0.01]}},
     AspectRatio -> 1 / 3, AxesLabel -> {"t", "x[t]"},
     PlotLabel -> "Corresponding x[t] solution plot"];

ysolutionplot =
 Plot[y[t], {t, 0, endtime}, PlotStyle -> {{Red, Thickness[0.01]}},
   AspectRatio -> 1 / 3, AxesLabel -> {"t", "y[t]"},
    PlotLabel -> "Corresponding y[t] solution plot"];
```

{8.90859 $E^{0.15\,t}$ Cos[1.53216 t] − 6.72665 $E^{0.15\,t}$ Sin[1.53216 t],
 −0.634838 $E^{0.15\,t}$ Cos[1.53216 t] − 13.8082 $E^{0.15\,t}$ Sin[1.53216 t]}



Rerun a couple of times.

Outward spiraling trajectories and corresponding increasingly excited sine wave solutions everytime.

DE.07 Eigenvectors and Eigenvalues for Linear Systems Tutorials

T.1) The matrix exponential $E^{A\,t}$:

A short cut to coming up with formulas

Here's a random coefficient matrix for a random linear system:

```
A = {{Random[Real, {-2, 2}], Random[Real, {-2, 2}]},
   {Random[Real, {-2, 2}], Random[Real, {-2, 2}]}};
Clear[x, y, t]
linearsystem =
  ({x'[t], y'[t]} == A . {x[t], y[t]});
ColumnForm[Thread[linearsystem]]
```

$x'[t] == 1.61241\,x[t] - 1.65454\,y[t]$
$y'[t] == -1.32754\,x[t] + 0.39924\,y[t]$

With random starter data on {x[0], y[0]}:

```
{xstarter, ystarter} = {Random[Real, {-2, 2}], Random[Real, {-2, 2}]}
```

{1.34492, 1.05138}

Formulas for the solution pair {x[t], y[t]} of this linear system with this starter data are:

```
Clear[eigenvector, eigenvalue];
{eigenvalue[1], eigenvalue[2]} = Chop[Eigenvalues[A]];
{eigenvector[1], eigenvector[2]} = Chop[Eigenvectors[A]];
Clear[x, y, x1, y1, x2, y2, t, C1, C2]
{x1[t_], y1[t_]} = Chop[eigenvector[1] Exp[eigenvalue[1] t]];
{x2[t_], y2[t_]} = Chop[eigenvector[2] Exp[eigenvalue[2] t]];
{x[t_], y[t_]} = C1 {x1[t], y1[t]} + C2 {x2[t], y2[t]};
starter = {xstarter, ystarter};
starterequation = {x[0], y[0]} == starter;
Csols = Solve[starterequation];
{x[t_], y[t_]} = Chop[ComplexExpand[{x[t], y[t]} /. Csols[[1]]]]
```

{0.960882 $E^{-0.595553\,t}$ + 0.384039 $E^{2.6072\,t}$,
 1.28229 $E^{-0.595553\,t}$ − 0.230904 $E^{2.6072\,t}$}

That's a lot of algebra.
Mathematica has an instruction involving a new idea called the matrix exponential.
This instruction will reproduce this formula for you in one line of typing.
Here you go:

```
Clear[x, y, t]
 {x[t_], y[t_]} =
 Chop[ComplexExpand[Expand[MatrixExp[A t] . {xstarter, ystarter}]]]
```

$$\{0.960882\, E^{-0.595553\, t} + 0.384039\, E^{2.6072\, t},\ 1.28229\, E^{-0.595553\, t} - 0.230904\, E^{2.6072\, t}\}$$

Use it and love it.

T.2) Quick eigenvalue analysis

□T.2.a) Eigenvalues indicate sucking swirlers

The given linear system is:

```
Clear[x, y, t]
linearsystem =
 ({x'[t], y'[t]} ==
  {0.5 x[t] - 1.7 y[t], 2.8 x[t] - 2.3 y[t]});
ColumnForm[Thread[linearsystem]]
```

$$x'[t] == 0.5\, x[t] - 1.7\, y[t]$$
$$y'[t] == 2.8\, x[t] - 2.3\, y[t]$$

You read off the coefficient matrix A:

```
A = {{0.5, -1.7}, {2.8, -2.3}};
MatrixForm[A]
```

$$\begin{pmatrix} 0.5 & -1.7 \\ 2.8 & -2.3 \end{pmatrix}$$

You ask *Mathematica* for the eigenvalues of A:

```
Clear[eigenvalue];
{eigenvalue[1], eigenvalue[2]} = Chop[Eigenvalues[A]]
```

$$\{-0.9 + 1.67332\, I,\ -0.9 - 1.67332\, I\}$$

What information about the trajectories do you get from this calculation?

□**Answer:**

Almost everything you want.

Take another look:

```
Clear[eigenvalue];
{eigenvalue[1], eigenvalue[2]} = Chop[Eigenvalues[A]]
```

$$\{-0.9 + 1.67332\, I,\ -0.9 - 1.67332\, I\}$$

The presence of $I = \sqrt{-1}$ tells you both eigenvectors are swirlers. The presence of the negative number -0.9 tells you that both eigenvectors are also sucking swirlers.

The upshot:

The trajectories in this system spiral toward {0, 0}. No matter what the starter data on solutions {x[t], y[t]} of this system, you are guaranteed that

$$x[t] \to 0 \text{ and } y[t] \to 0 \text{ as } t \to \infty$$

but they wiggle a lot on their way.

In fact, you can know the essence of what are used to make up solution pairs {x[t], y[t]} just by looking at the eigenvalues.

For instance here's a formula for {x[t], y[t]} starting at a random point.

```
{xstarter, ystarter} = {Random[Real, {-3, 3}], Random[Real, {-3, 3}]};
Clear[x, y, t]
{x[t_], y[t_]} =
 Chop[ComplexExpand[Expand[MatrixExp[A t] . {xstarter, ystarter}]]]
```

$$\{1.16393\, E^{-0.9\, t} \text{Cos}[1.67332\, t] + 1.33671\, E^{-0.9\, t} \text{Sin}[1.67332\, t],\ -0.357201\, E^{-0.9\, t} \text{Cos}[1.67332\, t] + 2.24648\, E^{-0.9\, t} \text{Sin}[1.67332\, t]\}$$

When you look at these formulas, you can tell almost everything you need to know simply by looking at the eigenvalues of the coefficient matrix A:

```
Eigenvalues[A]
```

$$\{-0.9 + 1.67332\, I,\ -0.9 - 1.67332\, I\}$$

□T.2.b) Eigenvalues that indicate two suckers

The given linear system is:

```
Clear[x, y, t]
linearsystem =
 ({x'[t], y'[t]} ==
  {-2.3 x[t] + 0.7 y[t], 1.8 x[t] - 4.3 y[t]});
ColumnForm[Thread[linearsystem]]
```

$$x'[t] == -2.3\, x[t] + 0.7\, y[t]$$
$$y'[t] == 1.8\, x[t] - 4.3\, y[t]$$

You read off the coefficient matrix A:

```
A = {{-2.3, 0.7}, {1.8, -4.3}};
MatrixForm[A]
```

$$\begin{pmatrix} -2.3 & 0.7 \\ 1.8 & -4.3 \end{pmatrix}$$

You ask *Mathematica* for the eigenvalues of A:

```
Clear[eigenvalue];
{eigenvalue[1], eigenvalue[2]} = Chop[Eigenvalues[A]]
```

$$\{-4.80333, -1.79667\}$$

What information about the trajectories do you get from this calculation?

□**Answer:**

Almost everything you want.

Take another look:

```
Clear[eigenvalue];
{eigenvalue[1], eigenvalue[2]} = Chop[Eigenvalues[A]]
```

$$\{-4.80333, -1.79667\}$$

Both negative, with no imaginary component. The eigenvectors are both suckers.

The upshot:

The trajectories in this system hustle their buns toward {0, 0} and they don't spiral.

No matter what the starter data on solutions {x[t], y[t]} of this system, you are guaranteed that

$$x[t] \to 0 \text{ and } y[t] \to 0 \text{ as } t \to \infty$$

but they don't wiggle much on their way.

In fact, you can know the essence of what are used to make up solution pairs {x[t], y[t]} just by looking at the eigenvalues.

For instance, here's a formula for {x[t], y[t]} starting at a random point.

```
{xstarter, ystarter} =
 {Random[Real, {-3, 3}], Random[Real, {-3, 3}]}; Clear[x, y, t]
{x[t_], y[t_]} =
 Chop[ComplexExpand[Expand[MatrixExp[A t] . {xstarter, ystarter}]]]
```

$$\{-0.239257\, E^{-4.80333\, t} + 1.9429\, E^{-1.79667\, t},\ 0.855626\, E^{-4.80333\, t} + 1.39703\, E^{-1.79667\, t}\}$$

When you look at these formulas, you can tell almost everything you need to know simply by looking at the eigenvalues of the coefficient matrix A:

```
Eigenvalues[A]
```

$$\{-4.80333, -1.79667\}$$

□T.2.c) Eigenvalues that indicate a propeller and a sucker

The given linear system is:

```
Clear[x, y, t]
linearsystem =
 ({x'[t], y'[t]} == {-0.5 x[t] + 1.7 y[t], 2.8 x[t] - 2.3 y[t]});
ColumnForm[Thread[linearsystem]]
```

$$x'[t] == -0.5\, x[t] + 1.7\, y[t]$$
$$y'[t] == 2.8\, x[t] - 2.3\, y[t]$$

You read off the coefficient matrix A:

```
A = {{-0.5, 1.7}, {2.8, -2.3}};
MatrixForm[A]
```

$$\begin{pmatrix} -0.5 & 1.7 \\ 2.8 & -2.3 \end{pmatrix}$$

You ask *Mathematica* for the eigenvalues of A:

```
Clear[eigenvalue];
{eigenvalue[1], eigenvalue[2]} = Chop[Eigenvalues[A]]
```

```
{-3.76008, 0.960085}
```

What information about the trajectories do you get from this calculation?

□ Answer:

Almost everything you want.

Take another look:

```
Clear[eigenvalue];
{eigenvalue[1], eigenvalue[2]} = Chop[Eigenvalues[A]]
```

```
{-3.76008, 0.960085}
```

Mixed signs. One eigenvector is a sucker and the other is a propeller.

The upshot:

The trajectories in this system hustle their buns toward the line determined by the propelling eigenvector which shows up as eigenvector[2] here:

```
Clear[eigenvector];
{eigenvector[1], eigenector[2]} = Chop[Eigenvectors[A]]
```

```
{{-0.462371, 0.886687}, {0.758608, 0.651548}}
```

And they don't spiral.

You are guaranteed that all trajectories eventually try to merge with the line through {0, 0} determined by the propelling eigenvector: eigenvector[2].

There is an exception.
Any trajectory that starts on the line through {0, 0}

determined by the sucking eigenvector stays on that line
and eventually stalls at {0, 0}.

In fact, you can know the essence of what are used to make up solution pairs {x[t], y[t]} just by looking at the eigenvalues.

For instance, here's a formula for {x[t], y[t]} starting at a random point.

```
{xstarter, ystarter} = {Random[Real, {-3, 3}], Random[Real, {-3, 3}]};
Clear[x, y, t]
{x[t_], y[t_]} =
 Chop[ComplexExpand[Expand[MatrixExp[A t] . {xstarter, ystarter}]]]
{xstarter, ystarter} = {Random[Real, {-3, 3}], Random[Real, {-3, 3}]};
```

```
{-0.442493 E^(-3.76008 t) - 0.497915 E^(0.960085 t),
 0.848567 E^(-3.76008 t) - 0.427646 E^(0.960085 t)}
```

When you look at these formulas, you can tell almost everything you need to know simply by looking at the eigenvalues of the coefficient matrix A:

```
Eigenvalues[A]
```

```
{-3.76008, 0.960085}
```

□T.2.d) Eigenvalues indicate propelling swirlers

The given linear system is:

```
Clear[x, y, t]
linearsystem =
  ({x'[t], y'[t]} == {0.5 x[t] + 6.7 y[t], -2.8 x[t] + 2.3 y[t]});
ColumnForm[Thread[linearsystem]]
```

```
x'[t] == 0.5 x[t] + 6.7 y[t]
y'[t] == -2.8 x[t] + 2.3 y[t]
```

You read off the coefficient matrix A:

```
A = {{0.5, 6.7}, {-2.8, 2.3}};
MatrixForm[A]
```

$$\begin{pmatrix} 0.5 & 6.7 \\ -2.8 & 2.3 \end{pmatrix}$$

You ask *Mathematica* for the eigenvalues of A:

```
Clear[eigenvalue];
{eigenvalue[1], eigenvalue[2]} = Chop[Eigenvalues[A]]
```

```
{1.4 + 4.23674 I, 1.4 - 4.23674 I}
```

What information about the trajectories do you get from this calculation?

□ Answer:

Almost everything you want.

Take another look:

```
Clear[eigenvalue];
{eigenvalue[1], eigenvalue[2]} = Chop[Eigenvalues[A]]
```

```
{1.4 + 4.23674 I, 1.4 - 4.23674 I}
```

The presence of $I = \sqrt{-1}$ tells you both eigenvectors are swirlers. The presence of the positive number 1.4
tells you that both eigenvectors are also propelling swirlers.

The upshot:

The trajectories in this system spiral away from {0, 0}. No matter what the starter data on solutions {x[t], y[t]} of this system, you are guaranteed that eventually {x[t], y[t]} will try to run off the screen with increasingly wild oscillation.

In fact, you can know the essence of what are used to make up solution pairs {x[t], y[t]} just by looking at the eigenvalues.

For instance, here's a formula for {x[t], y[t]} starting at a random starting point.

```
{xstarter, ystarter} = {Random[Real, {-3, 3}], Random[Real, {-3, 3}]};
Clear[x, y, t]
{x[t_], y[t_]} =
 Chop[ComplexExpand[Expand[MatrixExp[A t] . {xstarter, ystarter}]]]
```

```
{-0.592813 E^(1.4 t) Cos[4.23674 t] - 3.25357 E^(1.4 t) Sin[4.23674 t],
 -2.13703 E^(1.4 t) Cos[4.23674 t] - 0.0621814 E^(1.4 t) Sin[4.23674 t]}
```

When you look at these formulas, you can tell almost everything you need to know simply by looking at the eigenvalues of the coefficient matrix A:

```
Eigenvalues[A]
```

```
{1.4 + 4.23674 I, 1.4 - 4.23674 I}
```

□T.2.e) Eigenvalues that indicate pure swirlers

The given linear system is:

```
Clear[x, y, t]
linearsystem =
  ({x'[t], y'[t]} == {0.5 x[t] - 1.7 y[t], 3.6 x[t] - 0.5 y[t]});
ColumnForm[Thread[linearsystem]]
```

```
x'[t] == 0.5 x[t] - 1.7 y[t]
y'[t] == 3.6 x[t] - 0.5 y[t]
```

You read off the matrix A:

```
A = {{0.5, -1.7}, {3.6, -0.5}};
MatrixForm[A]
```

$$\begin{pmatrix} 0.5 & -1.7 \\ 3.6 & -0.5 \end{pmatrix}$$

You ask *Mathematica* for the eigenvalues of A:

```
Clear[eigenvalue];
{eigenvalue[1], eigenvalue[2]} = Chop[Eigenvalues[A]]
```

```
{2.42281 I, -2.42281 I}
```

What information about the trajectories do you get from this calculation?

□ Answer:

Almost everything you want.

Take another look:

```
Clear[eigenvalue];
{eigenvalue[1], eigenvalue[2]} = Chop[Eigenvalues[A]]
```

```
{2.42281 I, -2.42281 I}
```

The presence the lone $I = \sqrt{-1}$ term tells you both eigenvectors are pure swirlers.

The upshot:

The trajectories in this system oscillate on ellipses centered at {0, 0}.

You should a little careful with this system because small errors in the coefficients in the linear system might send you into eigenvalues such as

$0.1 + 2.43\,I$ and $0.1 - 2.43$ or

$-0.1 + 2.42\,I$ and $-0.1 - 2.42\,I$,

both of which signal entirely different trajectory behavior.

In fact, you can know the essence of what are used to make up solution pairs {x[t], y[t]} just by looking at the eigenvalues.

For instance, here's a formula for {x[t], y[t]} starting at a random point.

```
{xstarter, ystarter} = {Random[Real, {-3, 3}], Random[Real, {-3, 3}]};
Clear[x, y, t]
{x[t_], y[t_]} =
 Chop[ComplexExpand[Expand[MatrixExp[A t] . {xstarter, ystarter}]]]
```

```
{-0.417604 Cos[2.42281 t] - 1.79127 Sin[2.42281 t],
 2.43006 Cos[2.42281 t] - 1.12201 Sin[2.42281 t]}
```

When you look at these formulas, you can tell almost everything you need to know simply by looking at the eigenvalues of the coefficient matrix A:

```
Eigenvalues[A]
```

```
{0. + 2.42281 I, 0. - 2.42281 I}
```

□T.2.f) Eigenvalues that indicate two propellers

The given linear system is:

```
Clear[x, y, t]
linearsystem =
  ({x'[t], y'[t]} == {3.5 x[t] - 0.4 y[t], 0.6 x[t] + 2.3 y[t]});
ColumnForm[Thread[linearsystem]]
```

$x'[t] == 3.5\,x[t] - 0.4\,y[t]$

$y'[t] == 0.6\,x[t] + 2.3\,y[t]$

You read off the matrix A:

```
A = {{3.5, -0.4}, {0.6, 2.3}};
MatrixForm[A]
```

$$\begin{pmatrix} 3.5 & -0.4 \\ 0.6 & 2.3 \end{pmatrix}$$

You ask *Mathematica* for the eigenvalues of A:

```
Clear[eigenvalue];
{eigenvalue[1], eigenvalue[2]} = Chop[Eigenvalues[A]]
```

```
{3.24641, 2.55359}
```

What information about the trajectories do you get from this calculation?

□**Answer:**

Almost everything you want.

Take another look:

```
Clear[eigenvalue];
{eigenvalue[1], eigenvalue[2]} = Chop[Eigenvalues[A]]
```

```
{3.24641, 2.55359}
```

Both positive. The eigenvectors are both propellers.

The upshot:

The trajectories in this system hustle their buns off the screen and they don't spiral.

Eventually all the trajectories will try to merge with the line through {0, 0} determined by the dominant propeller which shows up as eigenvector[1] in:

```
{eigenvector[1], eigenvector[2]} = Eigenvectors[A]
```

```
{{0.844574, 0.535439}, {0.389306, 0.921108}}
```

In fact, you can know the essence of what are used to make up solution pairs {x[t], y[t]} just by looking at the eigenvalues.

For instance, here's a formula for {x[t], y[t]} starting at a random point.

```
{xstarter, ystarter} = {Random[Real, {-3, 3}], Random[Real, {-3, 3}]};
Clear[x, y, t]
{x[t_], y[t_]} =
 Chop[ComplexExpand[Expand[MatrixExp[A t] . {xstarter, ystarter}]]]
```

```
{-0.257326 E^(2.55359 t) + 3.11094 E^(3.24641 t),
 -0.608839 E^(2.55359 t) + 1.97225 E^(3.24641 t)}
```

When you look at these formulas, you can tell almost everything you need to know simply by looking at the eigenvalues of the coefficient matrix A:

```
Eigenvalues[A]
```

```
{3.24641, 2.55359}
```

In fact, you can know the essence of what are used to make up x[t] and y[t] just by looking at the eigenvalues.

T.3) Traces, determinants and the cheat sheet chart - print it and put it in your wallet

□T.3.a.i)

Given a linear system with given numbers a, b, c and d:

```
Clear[x, y, t, a, b, c, d]
linearsystem =
  ({x'[t], y'[t]} == {a x[t] + b y[t], c x[t] + d y[t]});
ColumnForm[Thread[linearsystem]]
```

$x'[t] == a\,x[t] + b\,y[t]$

$y'[t] == c\,x[t] + d\,y[t]$

Your response is to write down the coefficient matrix A for this system:

```
A = {{a, b}, {c, d}};
MatrixForm[A]
```

$$\begin{pmatrix} a & b \\ c & d \end{pmatrix}$$

Now calculate the eigenvalues of A:

```
Eigenvalues[A]
```

$$\left\{\frac{1}{2}\left(a + d - \sqrt{a^2 + 4\,b\,c - 2\,a\,d + d^2}\right),\ \frac{1}{2}\left(a + d + \sqrt{a^2 + 4\,b\,c - 2\,a\,d + d^2}\right)\right\}$$

Two quantities stand out:

$(a + d)$ and $a\,d - b\,c$

Folks have special names for these quantities. They say

$\text{trace of A} = a + d$

and

$\text{determinant of A} = a\,d - b\,c.$

```
{trace = a + d, det = Det[A]}
```

```
{a + d, -b c + a d}
```

You can easily express the eigenvalues of A in terms of the trace of A and the determinant of A:

```
ExpandAll[Eigenvalues[A]] ==
  ExpandAll[{1/2 (trace - Sqrt[trace^2 - 4 det]), 1/2 (trace + Sqrt[trace^2 - 4 det])}]
```

```
True
```

So the two eigenvalues of A are given by the formulas:

$\frac{1}{2}\left(\text{trace} + \sqrt{\text{trace}^2 - 4\,\text{det}}\right)$

and

$\frac{1}{2}\left(\text{trace} - \sqrt{\text{trace}^2 - 4\,\text{det}}\right).$

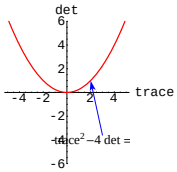
These formulas make it possible for you to calculate the eigenvalues of a coefficient matrix with pencil and paper or with a cheap pocket calculator.

Lots of reference books have charts, of the type you will see below, that help you to see what kind of linear system you are dealing with once you know the trace and the determinant.
In this problem, you will participate in building one of these charts.
To start building the chart, you go with trace and determinant axes and then you plot the curve

$\text{trace}^2 - 4\,\text{determinant} = 0$:

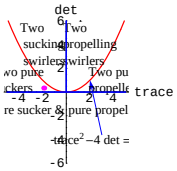
```
cutoff = Plot[trace^2/4, {trace, -5, 5},
  PlotStyle -> {{Red, Thickness[0.01]}}, AxesLabel -> {"trace", "det"},
  PlotRange -> {-6, 6}, DisplayFunction -> Identity];
cutofflabel = Graphics[
  Text[FontForm["trace^2-4 det == 0", {"Times", 10}], {3.0, -4.0}]];
```

```
pointer = Arrow[{2, 1} - {3.0, -3.6}, Tail → {3.0, -3.6}];
Show[
 cutoff, cutofflabel, pointer, DisplayFunction → $DisplayFunction];
```



And then you annote this cheat sheet plot as follows:

```
twosuckercutoff =
 Graphics[{Blue, Thickness[0.01], Line[{{-5, 0}, {0, 0}}]}];
twosuckerlabel = Graphics[
  Text[FontForm[" Two pure \n suckers", {"Times", 10}], {-4, 1}]];
twopropcutoff =
 Graphics[{Blue, Thickness[0.01], Line[{{0, 0}, {5, 0}}]}];
twoproplabel = Graphics[
  Text[FontForm[" Two pure \n propellers", {"Times", 10}], {4, 1}]];
propswirlcutoff =
 Graphics[{Blue, Thickness[0.01], Line[{{0, 0}, {0, 6}}]}];
propswirllabel = Graphics[Text[FontForm[
    " Two\n propelling \n swirlers", {"Times", 10}], {2, 4}]];
pureswirlerlabel =
 Graphics[Text[FontForm[" p\n u\n r ", {"Times", 12}], {0, 7}]];
suckswirlcutoff =
 Graphics[{Blue, Thickness[0.01], Line[{{0, 0}, {0, 6}}]}];
suckswirllabel = Graphics[Text[
   FontForm["Two\n sucking\n swirlers", {"Times", 10}], {-2, 4}]];
negdetcutoff =
 Graphics[{Blue, Thickness[0.01], Line[{{-5, 0}, {5, 0}}]}];
negdetlabel = Graphics[Text[FontForm[
    "Pure sucker & pure propeller", {"Times", 10}], {0, -1.5}]];
pureswirlerlabel = Graphics[
  Text[FontForm["Two pure swirlers on the positive vertical axis",
    {"Times", 10}], {0, 6.5}]];
chart = Show[cutoff, cutofflabel, pointer, twosuckercutoff,
  twosuckerlabel, twopropcutoff, twoproplabel, propswirlcutoff,
  propswirllabel, suckswirlcutoff, suckswirllabel, negdetcutoff,
  negdetlabel, pureswirlerlabel, DisplayFunction → $DisplayFunction];
```

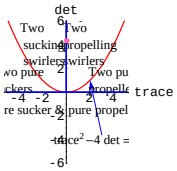


This tells you that the given linear system has two pure suckers. All trajectories hustle towards {0, 0} as time goes on.

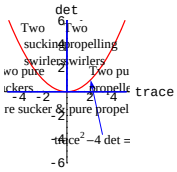
Another:

```
a = -1.4;
b = -3.0;
c = 2.1;
d = 1.4;
A = {{a, b}, {c, d}};
Clear[x, y, t]
ColumnForm[Thread[{x'[t], y'[t]} == A . {x[t], y[t]}]]
trace = a + d;
det = a d - b c;
chartpointpoint =
 Graphics[{HotPink, PointSize[0.04], Point[{trace, det}]}];
Show[chart, chartpointpoint];
```

$x'[t] == -1.4\,x[t] - 3.\,y[t]$
$y'[t] == 2.1\,x[t] + 1.4\,y[t]$



This landed on the vertical axis. And this tells you that this linear system has two pure swirlers. Solutions are sine waves.



The chart is not to be memorized.
It is to be called up and used when you decide you want it.

Some folks even carry a copy of this chart in their wallets.
How do folks use this chart?

□ **Answer:**

Here it is for a sample linear system:

```
a = -1.4;
b = 0.9;
c = 0.4;
d = -0.5;
A = {{a, b}, {c, d}};
Clear[x, y, t]
ColumnForm[Thread[{x'[t], y'[t]} == A . {x[t], y[t]}]]
```

$x'[t] == -1.4\,x[t] + 0.9\,y[t]$
$y'[t] == 0.4\,x[t] - 0.5\,y[t]$

You calculate

trace = a + d

and

determinant = a d − b c

by hand. And then you plot the resulting point

{trace, determinant}

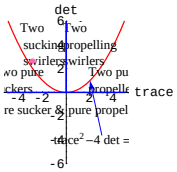
on the chart:

```
trace = a + d;
det = a d - b c;
chartpointpoint =
 Graphics[{Magenta, PointSize[0.04], Point[{trace, det}]}];
Show[chart, chartpointpoint];
```

Here is the same thing for some random linear systems:

```
a = Random[Real, {-2, 2}];
b = Random[Real, {-2, 2}];
c = Random[Real, {-2, 2}];
d = Random[Real, {-2, 2}];
A = {{a, b}, {c, d}};
Clear[x, y, t]
ColumnForm[Thread[{x'[t], y'[t]} == A . {x[t], y[t]}]]
trace = a + d;
det = a d - b c;
chartpointpoint =
 Graphics[{HotPink, PointSize[0.04], Point[{trace, det}]}];
Show[chart, chartpointpoint];
```

$x'[t] == -1.84464\,x[t] + 0.905562\,y[t]$
$y'[t] == -0.810281\,x[t] - 1.04027\,y[t]$



Rerun a few times.

Neat cheat sheet.

Print one up and put it in your wallet.

□ T.3.a.ii)

Explain why the chart is correct.

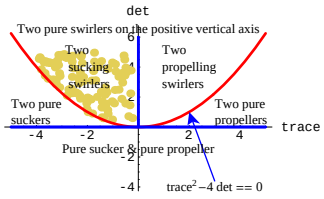
□ **Answer:**

Select the ones you want.
Reading them all gets really tiresome.

□Two sucking swirlers explanation:

Here are some sample points {trace, det} that plot out in the two swirling sucker region of the chart:

```
Clear[trace, k]
trace[k_] := trace[k] = Random[Real, {-4, -0.1}];
samplepoints = Table[{trace[k],
   trace[k]^2/4 + Random[Real, {0, 5 - trace[k]^2/4}]}, {k, 1, 100}];
samplepointplot = ListPlot[samplepoints,
  PlotStyle → {Banana, PointSize[0.03]}, DisplayFunction → Identity];
Show[samplepointplot, chart, PlotRange → All,
 AxesLabel → {"trace", "det"}, DisplayFunction → $DisplayFunction];
```



The two sucking whirlers region corresponds to the case that

trace of A < 0,

determinant of A > 0, and

(trace of A)^2 < 4 determinant of A.

Explanation:

The eigenvalues of A are

$$p + I\,q = \frac{1}{2}\left(\text{trace} + \sqrt{\text{trace}^2 - 4\,\det}\right)$$

and

$$p - I\,q = \frac{1}{2}\left(\text{trace} - \sqrt{\text{trace}^2 - 4\,\det}\right).$$

Because

$\det > 0$

and

$\text{trace}^2 - 4\,\det < 0$,

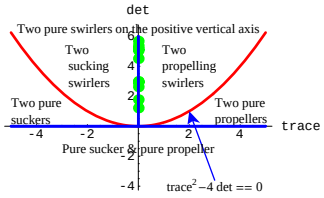
you see q is not 0. This gives you the whirl.

Because trace < 0, you see that p < 0. This gives you the suck.

□Two pure swirlers explanation:

Here are some sample points {trace, det} that plot out in the two pure swirler region of the chart:

```
samplepoints = Table[{0, Random[Real, {0, 6}]}, {k, 1, 10}];
samplepointplot = ListPlot[samplepoints,
  PlotStyle → {Green, PointSize[0.04]}, DisplayFunction → Identity];
Show[samplepointplot, chart, PlotRange → All,
 AxesLabel → {"trace", "det"}, DisplayFunction → $DisplayFunction];
```



The two pure swirler region corresponds to the case that

trace of A = 0,

and

determinant of A > 0.

Explanation:

The eigenvalues of A are

$$p + I\,q = \frac{1}{2}\left(\text{trace} + \sqrt{\text{trace}^2 - 4\,\det}\right)$$

and

$$p - I\,q = \frac{1}{2}\left(\text{trace} - \sqrt{\text{trace}^2 - 4\,\det}\right).$$

Because

trace = 0,

$$p + I\,q = \frac{1}{2}\left(0 + \sqrt{0 - 4\,\det}\right)$$

and

$$p - I\,q = \frac{1}{2}\left(0 - \sqrt{0 - 4\,\det}\right).$$

And because det > 0,

$$p + I\,q = \frac{1}{2}\left(0 + 2\,I\,\sqrt{\text{Abs}[\det]}\right)$$

and

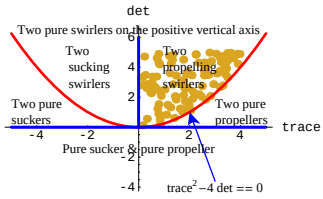
$$p - I\,q = \frac{1}{2}\left(0 - 2\,I\,\sqrt{\text{Abs}[\det]}\right)$$

This gives you the swirl. Because p = 0, there is no suck or propel.

□Two propelling swirlers explanation:

Here are some sample points {trace, det} that plot out in the two propelling swirlers region of the chart:

```
Clear[trace, k]; trace[k_] := trace[k] = Random[Real, {0.1, 4}];
samplepoints = Table[{trace[k],
   trace[k]^2/4 + Random[Real, {0, 5 - trace[k]^2/4}]}, {k, 1, 100}];
samplepointplot = ListPlot[samplepoints, PlotStyle →
   {Goldenrod, PointSize[0.03]}, DisplayFunction → Identity];
Show[samplepointplot, chart, PlotRange → All,
 AxesLabel → {"trace", "det"}, DisplayFunction → $DisplayFunction];
```



The two propelling whirlers region corresponds to the case that

trace of A > 0,

determinant of A > 0, and

(trace of A)^2 < 4 determinant of A.

Explanation:

The eigenvalues of A are

$$p + I\,q = \frac{1}{2}\left(\text{trace} + \sqrt{\text{trace}^2 - 4\,\det}\right)$$

and

$$p - I\,q = \frac{1}{2}\left(\text{trace} - \sqrt{\text{trace}^2 - 4\,\det}\right).$$

Because

$\text{trace}^2 - 4\,\det < 0$,

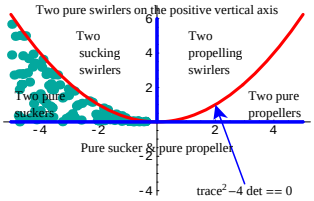
you see q is not 0. This gives you the whirl.

Because trace > 0, you see that p > 0. This gives you the propulsion.

□Two pure suckers explanation:

Here are some sample points {trace, det} that plot out in the two pure sucker region of the chart:

```
Clear[trace, k]
trace[k_] := trace[k] = Random[Real, {-5, -0.3}]; samplepoints =
  Table[{trace[k], Random[Real, {0, trace[k]^2/4}]}, {k, 1, 100}];
samplepointplot = ListPlot[samplepoints, PlotStyle →
   {ManganeseBlue, PointSize[0.03]}, DisplayFunction → Identity];
Show[samplepointplot, chart, PlotRange → All,
 DisplayFunction → $DisplayFunction];
```



The two pure suckers region corresponds to the case that

trace of A < 0,

determinant of A > 0, and

(trace of A)^2 > 4 determinant of A.

Explanation:

The eigenvalues of A are

$\frac{1}{2}\left(\text{trace} - \sqrt{\text{trace}^2 - 4\,\text{det}}\right)$

and

$\frac{1}{2}\left(\text{trace} + \sqrt{\text{trace}^2 - 4\,\text{det}}\right)$.

Because

$\text{trace}^2 - 4\,\text{det} > 0$,

you see that neither eigenvalue involves I = $\sqrt{-1}$. So there is no whirl.

Because det > 0, you are guaranteed that

$0 < \sqrt{\text{trace}^2 - 4\,\text{det}} < \sqrt{\text{trace}^2} = -\text{trace}$.

Remember trace < 0 in this case.

This guarantees that both eigenvalues

$\frac{1}{2}\left(\text{trace} - \sqrt{\text{trace}^2 - 4\,\text{det}}\right) < 0$

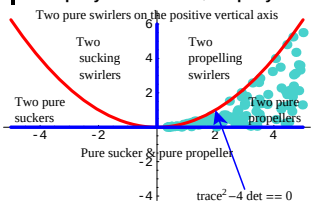
$\frac{1}{2}\left(\text{trace} + \sqrt{\text{trace}^2 - 4\,\text{det}}\right) < 0$.

The upshot: Both eigenvalues are negative. And so the linear system has two pure suckers.

□Two pure propellers explanation:

Here are some sample points {trace, det} that plot out in the two pure propeller region of the chart:

```
Clear[trace, k]
trace[k_] := trace[k] = Random[Real, {0.3, 5}]; samplepoints =
 Table[{trace[k], Random[Real, {0, trace[k]^2/4}]}, {k, 1, 100}];
samplepointplot = ListPlot[samplepoints, PlotStyle →
   {MediumTurquoise, PointSize[0.03]}, DisplayFunction → Identity];
Show[samplepointplot, chart, PlotRange → All,
 DisplayFunction → $DisplayFunction];
```



The two pure propellers region corresponds to case that

trace of A > 0,

determinant of A > 0, and

(trace of A)^2 > 4 determinant of A.

Explanation:

The eigenvalues of A are

$\frac{1}{2}\left(\text{trace} - \sqrt{\text{trace}^2 - 4\,\text{det}}\right)$

and

$\frac{1}{2}\left(\text{trace} + \sqrt{\text{trace}^2 - 4\,\text{det}}\right)$

Because

$\text{trace}^2 - 4\,\text{det} > 0$,

you see that neither eigenvalue involves I = $\sqrt{-1}$. So there is no whirl.

Because det > 0, you are guaranteed that

$0 < \sqrt{\text{trace}^2 - 4\,\text{det}} < \sqrt{\text{trace}^2} = \text{trace}$.

Remember trace > 0 in this case.

This guarantees that both eigenvalues

$\frac{1}{2}\left(\text{trace} - \sqrt{\text{trace}^2 - 4\,\text{det}}\right) > 0$

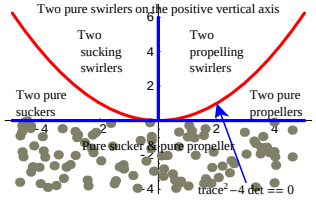
$\frac{1}{2}\left(\text{trace} + \sqrt{\text{trace}^2 - 4\,\text{det}}\right) > 0$.

The upshot: Both eigenvalues are positive. And so this linear system has two pure propellers.

□One pure propeller and one pure sucker explanation:

Here are some sample points {trace, det} that plot out in the the one pure propeller and one pure sucker region of the chart:

```
Clear[trace, k]
trace[k_] := trace[k] = Random[Real, {-5, 5}];
samplepoints = Table[{trace[k], Random[Real, {-4, 0}]}, {k, 1, 100}];
samplepointplot = ListPlot[samplepoints, PlotStyle →
   {WarmGray, PointSize[0.03]}, DisplayFunction → Identity];
Show[samplepointplot, chart, PlotRange → All,
 DisplayFunction → $DisplayFunction];
```



The one pure propeller and one pure sucker region corresponds to the case that

determinant of A < 0.

Explanation:

The eigenvalues of A are

$\frac{1}{2}\left(\text{trace} - \sqrt{\text{trace}^2 - 4\,\text{det}}\right)$

and

$\frac{1}{2}\left(\text{trace} + \sqrt{\text{trace}^2 - 4\,\text{det}}\right)$.

Because det < 0, you are gauanteed that

$\text{trace}^2 - 4\,\text{det} > 0$,

so that neither eigenvalue involves I = $\sqrt{-1}$. So there is no whirl.

Also because det < 0, you are guaranteed that

$\sqrt{\text{trace}^2 - 4\,\text{det}} > \sqrt{\text{trace}^2}$ = Abs[trace].

This tells you that A has one negative eigenvalue - namely

$\frac{1}{2}\left(\text{trace} - \sqrt{\text{trace}^2 - 4\,\text{det}}\right)$.

This guarantees one pure sucker.

And because the other eigenvalue

$\frac{1}{2}\left(\text{trace} + \sqrt{\text{trace}^2 - 4\det}\right)$

is automatically positive, this guarantees one pure propeller.

T.4) How do you pronounce the words "eigenvector" and "eigenvalue"?

What about the algebra behind calculating them?

□T.4.a)

How do you pronounce the words "eigenvector" and "eigenvalue"?

□Answer:

The words "eigenvector" and "eigenvalue" are bastardized words. "Eigen" comes from German, "vector" and "value" come from English. You pronounce "eigen" as if it were German and you pronouce "vector" and "value" as usual.

"Eigen" is pronounced I - gen with a hard g as in "garden" or "geezer," not the soft g as in "general" or "genetics."

The German word "eigen" corresponds to the English word "own."

□T.4.b.i) Calculating eigenvalues and eigenvectors.

Here is a matrix:

```
A = {{1.7, 0.6}, {-1.3, -3.1}};
MatrixForm[A]
```

$$\begin{pmatrix} 1.7 & 0.6 \\ -1.3 & -3.1 \end{pmatrix}$$

Here's *Mathematica*'s calculation of the eigenvalues of A:

```
Eigenvalues[A]
```

```
{-2.93159, 1.53159}
```

Here's *Mathematica*'s calculation of the eigenvectors of A:

```
Eigenvectors[A]
```

```
{{-0.128472, 0.991713}, {0.962794, -0.270238}}
```

What's the algebra behind calculating eigenvectors and eigenvalues?

□Answer:

Designate the unknown eigenvector by

{xeigenvect, yeigenvect}

and designate the eigenvalue attached to this unknown eigenvector by the name eigenval.

The algebraic relationship is

A . {xeigenvect, yeigenvect} = eigenval {xeigenvect, yeigenvect}:

This is the algebra definition of eigenvector and eigenvalue.

```
Clear[xeigvect, yeigvect, eigenval];
eigenequation =
 A . {xeigvect, yeigvect} == eigenval {xeigvect, yeigvect};
ColumnForm[Thread[eigenequation]]
```

```
1.7 xeigvect + 0.6 yeigvect == eigenval xeigvect
-1.3 xeigvect - 3.1 yeigvect == eigenval yeigvect
```

There are three unknowns here, but only two equations.

To nail down all three unknowns, you need one more equation. This extra equation will try to make {xeigenvect, yeigenvect} into a unit vector:

```
extraequation = xeigvect^2 + yeigvect^2 == 1
```

```
xeigvect^2 + yeigvect^2 == 1
```

Now solve the three equations:

```
Solve[{eigenequation, extraequation}]
```

```
{{eigenval → -2.93159, xeigvect → -0.128472, yeigvect → 0.991713},
 {eigenval → -2.93159, xeigvect → 0.128472, yeigvect → -0.991713},
 {eigenval → 1.53159, xeigvect → -0.962794, yeigvect → 0.270238},
 {eigenval → 1.53159, xeigvect → 0.962794, yeigvect → -0.270238}}
```

Compare:

```
Eigenvalues[A]
```

```
{-2.93159, 1.53159}
```

```
Eigenvectors[A]
```

```
{{-0.128472, 0.991713}, {0.962794, -0.270238}}
```

Perfect.

An algebraic mess by hand, but easy by machine.

The output seems to confirm that when you go with decimal entries in A, then *Mathematica* spits out unit eigenvectors.

Try another:

```
A = {{5.22, 0.64}, {0.71, 9.59}};
MatrixForm[A]
```

$$\begin{pmatrix} 5.22 & 0.64 \\ 0.71 & 9.59 \end{pmatrix}$$

```
Clear[xeigvect, yeigvect, eigenval];
eigenequation =
 A . {xeigvect, yeigvect} == eigenval {xeigvect, yeigvect};
ColumnForm[Thread[eigenequation]]
```

```
5.22 xeigvect + 0.64 yeigvect == eigenval xeigvect
0.71 xeigvect + 9.59 yeigvect == eigenval yeigvect
```

```
extraequation = xeigvect^2 + yeigvect^2 == 1
```

```
xeigvect^2 + yeigvect^2 == 1
```

```
Solve[{eigenequation, extraequation}]
```

```
{{eigenval → 5.11838, xeigvect → -0.987628, yeigvect → 0.156815},
 {eigenval → 5.11838, xeigvect → 0.987628, yeigvect → -0.156815},
 {eigenval → 9.69162, xeigvect → -0.141681, yeigvect → -0.989912},
 {eigenval → 9.69162, xeigvect → 0.141681, yeigvect → 0.989912}}
```

Compare:

```
Eigenvalues[A]
```

```
{9.69162, 5.11838}
```

```
Eigenvectors[A]
```

```
{{-0.141681, -0.989912}, {-0.987628, 0.156815}}
```

Pity those poor devils in outdated courses who spend a lot of their time solving for eigenvectors and eigenvalues by pencil and paper.

□T.4.b.ii)

Look at this calculation of the eigenvectors and eigenvalues of this matrix:

```
A = {{3.24, 2.64}, {-2.84, 4.29}};
MatrixForm[A]
Clear[xeigvect, yeigvect, eigenval];
eigenequation =
 A . {xeigvect, yeigvect} == eigenval {xeigvect, yeigvect};
ColumnForm[Thread[eigenequation]]
```

$$\begin{pmatrix} 3.24 & 2.64 \\ -2.84 & 4.29 \end{pmatrix}$$

```
3.24 xeigvect + 2.64 yeigvect == eigenval xeigvect
-2.84 xeigvect + 4.29 yeigvect == eigenval yeigvect
```

```
extraequation = xeigvect^2 + yeigvect^2 == 1
```

```
xeigvect^2 + yeigvect^2 == 1
```

```
Solve[{eigenequation, extraequation}]
```

```
{{eigenval → 3.765 - 2.68737 I, xeigvect → -1.11585 - 1.10669 I,
  yeigvect → -1.34845 + 0.91579 I}, {eigenval → 3.765 - 2.68737 I,
  xeigvect → 1.11585 + 1.10669 I, yeigvect → 1.34845 - 0.91579 I},
 {eigenval → 3.765 + 2.68737 I, xeigvect → -1.11585 + 1.10669 I,
  yeigvect → -1.34845 - 0.91579 I}, {eigenval → 3.765 + 2.68737 I,
  xeigvect → 1.11585 - 1.10669 I, yeigvect → 1.34845 + 0.91579 I}}
```

Compare:

```
Eigenvalues[A]
```

```
{3.765 + 2.68737 I, 3.765 - 2.68737 I}
```

The eigenvalues check.

```
Eigenvectors[A]
```

```
{{0.133079 - 0.681206 I, 0.719895 + 0. I},
 {0.133079 + 0.681206 I, 0.719895 + 0. I}}
```

The eigenvectors don't check.
What gives?

□Answer:

Mathematica has done the algebra slightly differently than shown above. This will happen anytime the number

$$I = \sqrt{-1}$$

pops up. Both calculations are correct and equally reliable.

□T.4.b.iii)

Should you worry about the calculation of eigenvectors and eigenvalues by hand?

□Answer:

Only if a professor requires you to.

Or if you enjoy self-inflicted pain.

The cheat sheet problem above gives as easy a hand path as there is.

T.5) When you get only one eigenvector, go with approximate formulas

This is in the never-never land of math pathology.

□T.5.a.i)

When you come across this linear system:

```
Clear[x, y, t]
originalsystem = ({x'[t], y'[t]} == {1.2 x[t], 1.2 x[t] + 1.2 y[t]});
ColumnForm[Thread[originalsystem]]
```

$$x'[t] == 1.2\,x[t]$$
$$y'[t] == 1.2\,x[t] + 1.2\,y[t]$$

you immediately write down the coefficient matrix:

```
A = {{1.2, 0}, {1.2, 1.2}};
MatrixForm[A]
```

$$\begin{pmatrix} 1.2 & 0 \\ 1.2 & 1.2 \end{pmatrix}$$

You ask *Mathematica* for the eigenvectors and eigenvalues of A:

```
Clear[eigenvector];
{eigenvector[1], eigenvector[2]} = Chop[Eigenvectors[A]]
```

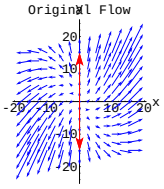
```
{{0, -1.}, {0, 1.}}
```

```
Clear[eigenvalue];
{eigenvalue[1], eigenvalue[2]} = Chop[Eigenvalues[A]]
```

```
{1.2, 1.2}
```

This is a problem because the two calculated eigenvectors are in fact just one eigenvector.
Reason: They point in opposite directions.
Take a look:

```
Clear[Field, x, y]
Field[x_, y_] = A . {x, y};
{xlow, xhigh} = {-15, 15};
{ylow, yhigh} = {-15, 15};
jump = (xhigh - xlow)/12;
flowplot = Table[Arrow[Field[x, y], Tail → {x, y},
    VectorColor → Blue, ScaleFactor → 0.25, HeadSize → 1.5],
  {x, xlow, xhigh, jump}, {y, ylow, yhigh, jump}];
scaler = 15;
sizer = 4;
eigenplot = {Arrow[eigenvector[1], Tail → {0, 0},
    VectorColor → Red, ScaleFactor → scaler, HeadSize → sizer],
  Arrow[-eigenvector[1], Tail → {0, 0},
    VectorColor → Red, ScaleFactor → scaler, HeadSize → sizer],
  Arrow[eigenvector[2], Tail → {0, 0},
    VectorColor → Red, ScaleFactor → scaler, HeadSize → sizer],
  Arrow[-eigenvector[2], Tail → {0, 0},
    VectorColor → Red, ScaleFactor → scaler, HeadSize → sizer]};
originalflow =
 Show[flowplot, eigenplot, Axes → True, AxesLabel → {"x", "y"},
  PlotRange → All, PlotLabel → "Original Flow"];
```

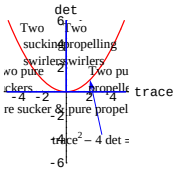


Just one eigenvector.
But there seems to be a phantom swirler at work here.

Use the cheat sheet chart and the idea of approximation to get an idea about why this happened.

□Answer:

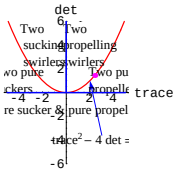
```
cutoff = Plot[trace^2/4, {trace, -5, 5},
  PlotStyle → {{Red, Thickness[0.01]}}, AxesLabel → {"trace", "det"},
  PlotRange → {-6, 6}, DisplayFunction → Identity];
cutofflabel = Graphics[
  Text[FontForm["trace^2 - 4 det == 0", {"Times", 10}], {3.0, -4.0}]];
pointer = Arrow[{2, 1} - {3.0, -3.6}, Tail → {3.0, -3.6}];
twosuckercutoff =
 Graphics[{Blue, Thickness[0.01], Line[{{-5, 0}, {0, 0}}]}];
twosuckerlabel = Graphics[
  Text[FontForm[" Two pure \n suckers", {"Times", 10}], {-4, 1}]];
twopropcutoff =
 Graphics[{Blue, Thickness[0.01], Line[{{0, 0}, {5, 0}}]}];
twoproplabel = Graphics[
  Text[FontForm[" Two pure \n propellers", {"Times", 10}], {4, 1}]];
propswirlcutoff =
 Graphics[{Blue, Thickness[0.01], Line[{{0, 0}, {0, 6}}]}];
propswirllabel = Graphics[Text[FontForm[
    " Two\n propelling \n swirlers", {"Times", 10}], {2, 4}]];
pureswirlerlabel =
 Graphics[Text[FontForm[" p\n u\n r ", {"Times", 12}], {0, 7}]];
suckswirlcutoff =
 Graphics[{Blue, Thickness[0.01], Line[{{0, 0}, {0, 6}}]}];
suckswirllabel = Graphics[Text[
   FontForm["Two\n sucking\n swirlers", {"Times", 10}], {-2, 4}]];
negdetcutoff =
 Graphics[{Blue, Thickness[0.01], Line[{{-5, 0}, {5, 0}}]}];
negdetlabel = Graphics[Text[FontForm[
    "Pure sucker & pure propeller", {"Times", 10}], {0, -1.5}]];
pureswirlerlabel = Graphics[
  Text[FontForm["Two pure swirlers on the positive vertical axis",
    {"Times", 10}], {0, 6.5}]];
chart = Show[cutoff, cutofflabel, pointer, twosuckercutoff,
  twosuckerlabel, twopropcutoff, twoproplabel, propswirlcutoff,
  propswirllabel, suckswirlcutoff, suckswirllabel, negdetcutoff,
  negdetlabel, pureswirlerlabel, DisplayFunction → $DisplayFunction];
```



Calculate the trace and the determinant of the given coefficient matrix A and plot {trace, det} on the chart:

```
{{a, b}, {c, d}} = A;
Clear[x, y, t]
ColumnForm[Thread[{x'[t], y'[t]} == A . {x[t], y[t]}]]
trace = a + d;
det = a d - b c;
chartpointpoint =
 Graphics[{Magenta, PointSize[0.04], Point[{trace, det}]}];
Show[chart, chartpointpoint];
```

$$x'[t] == 1.2\,x[t]$$
$$y'[t] == 1.2\,x[t] + 1.2\,y[t]$$



Now you can see why the the flow plot came out the way it did. The given linear system is on the border between having two swirling propellers and two pure propellers. The swirl you see in the flow comes from the influence of the propelling swirlers from nearby linear systems.

You can tweak the given linear system slightly to get a nearby linear

system with two propelling swirlers.

Here's how:

```
tweaker = {{0, -Random[Real, {0, 0.01}]}, {0, 0}};
tweakedA = A + tweaker;
MatrixForm[tweakedA]
```

$$\begin{pmatrix} 1.2 & -0.00682929 \\ 1.2 & 1.2 \end{pmatrix}$$

Compare with the original coefficient matrix A:

```
MatrixForm[A]
```

$$\begin{pmatrix} 1.2 & 0 \\ 1.2 & 1.2 \end{pmatrix}$$

A and tweakedA are almost the same.

Compare the original system with the system resulting from tweakedA:

```
ColumnForm[Thread[{x'[t], y'[t]} == A . {x[t], y[t]}]]
ColumnForm[Thread[{x'[t], y'[t]} == tweakedA . {x[t], y[t]}]]
```

$$x'[t] == 1.2\,x[t] + 1.2\,y[t]$$
$$y'[t] == 1.2\,x[t] + 1.2\,y[t]$$
$$x'[t] == 1.2\,x[t] - 0.00682929\,y[t]$$
$$y'[t] == 1.2\,x[t] + 1.2\,y[t]$$

The two linear systems are almost the same.

Check the eigenvalues of tweakedA:

```
Eigenvalues[tweakedA]
```

$$\{1.2 + 0.0905271\,I,\ 1.2 - 0.0905271\,I\}$$

Propelling swirlers. That gives you a pretty good idea of why you saw the swirl in the original flow.

□T.5.a.ii)

Stay with the same set-up as in part i).
You are out of luck if you want to come up with formulas for trajectories for the original linear system:

```
Clear[x, y, t]
ColumnForm[Thread[originalsystem]]
```

$$x'[t] == 1.2\,x[t] + 1.2\,y[t]$$
$$y'[t] == 1.2\,x[t] + 1.2\,y[t]$$

Reason: The coefficient matrix for this system doesn't have two genuinely different eigenvectors.
But you can do the next best thing, you can come up with high quality approximate formulas for the trajectories.
How?

□Answer:

This answer should come as no surprise.

Take the tweaked linear system:

```
ColumnForm[Thread[{x'[t], y'[t]} == tweakedA . {x[t], y[t]}]]
```

$$x'[t] == 1.2\,x[t] - 0.00682929\,y[t]$$
$$y'[t] == 1.2\,x[t] + 1.2\,y[t]$$

This nearby system has everything you need to calculate solution formulas:

```
Eigenvalues[tweakedA]
Chop[Eigenvectors[tweakedA]]
```

$$\{1.2 + 0.0905271\,I,\ 1.2 - 0.0905271\,I\}$$
$$\{\{0.0752255\,I,\ 0.997167\},\ \{-0.0752255\,I,\ 0.997167\}\}$$

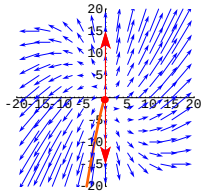
If you want a high quality approximate formula for a trajectory in the original linear system starting at the sample starting point

$$\{x[0], y[0]\} = \{-0.2,\ 0.9\}$$

you go after the formula for the solution of the tweaked system with the same starting point and plot:

```
{xstarter, ystarter} = {Random[Real, {-2, 2}], Random[Real, {-2, 2}]};
Clear[eigenvector, eigenvalue];
{eigenvalue[1], eigenvalue[2]} = Chop[Eigenvalues[tweakedA]];
{eigenvector[1], eigenvector[2]} = Chop[Eigenvectors[tweakedA]];
Clear[x, y, approxx, approxy, x1, y1, x2, y2, t, C1, C2]
{x1[t_], y1[t_]} = Chop[eigenvector[1] Exp[eigenvalue[1] t]];
{x2[t_], y2[t_]} = Chop[eigenvector[2] Exp[eigenvalue[2] t]];
{x[t_], y[t_]} = C1 {x1[t], y1[t]} + C2 {x2[t], y2[t]};
starter = {xstarter, ystarter};
starterequation = {x[0], y[0]} == starter;
Csols = Solve[starterequation];
{approxx[t_], approxy[t_]} =
 Chop[ComplexExpand[{x[t], y[t]} /. Csols[[1]]]]
endtime = 2.5;
approxtrajectoryplot =
 ParametricPlot[{approxx[t], approxy[t]}, {t, 0, endtime}, PlotStyle →
   {{Thickness[0.015], CadmiumOrange}}, DisplayFunction → Identity];
starterplot =
 Graphics[{Red, PointSize[0.04], Point[{xstarter, ystarter}]}];
Show[approxtrajectoryplot, originalflow, starterplot,
 PlotRange → {{xlow - 5, xhigh + 5}, {ylow - 5, yhigh + 5}},
 DisplayFunction → $DisplayFunction];
```

$$\{-0.264077\,E^{1.2\,t}\,\text{Cos}[0.0905271\,t] + 0.0340089\,E^{1.2\,t}\,\text{Sin}[0.0905271\,t],$$
$$-0.450811\,E^{1.2\,t}\,\text{Cos}[0.0905271\,t] - 3.50052\,E^{1.2\,t}\,\text{Sin}[0.0905271\,t]\}$$



Rerun a couple of times.

These are all approximations and very good ones at that!

Reason: They go with the flow very well.

□T.5.a.iii)

Is it humanly possible to come up with exact formulas for trajectories for the original linear system above?

□Answer:

Yes, with great algebraic agony.

If you're really driven to see how to do it, you can consult Boyce and DePrima's print book "Elementary Differential Equations and Boundary Value Problems," 5th Edition, Wiley, New York, 1992.

Look in section 7.7.

□T.5.a.iv)

But isn't an exact formula always better than an approximation?

□Answer:

Not necessarily.

Often approximate formulas are very much as legitimate as exact formulas.

Reason:

The constants in linear systems usually come from experimental measurements which have built-in errors.

Look at the linear system studied above:

```
Clear[x, y, t]
ColumnForm[Thread[{x'[t], y'[t]} == A . {x[t], y[t]}]]
```

$$x'[t] == 1.2\,x[t] + 1.2\,y[t]$$
$$y'[t] == 1.2\,x[t] + 1.2\,y[t]$$

When you approximated this linear system with:

```
ColumnForm[Thread[{x'[t], y'[t]} == tweakedA . {x[t], y[t]}]]
```

$$x'[t] == 1.2\,x[t] - 0.00682929\,y[t]$$
$$y'[t] == 1.2\,x[t] + 1.2\,y[t]$$

you went with a linear system that could have resulted from more precise measurements of the constants.

T.6) The complex exponential and Euler's formula

$$E^{(p+Iq)t} = E^{pt}(\text{Cos}[qt] + I\,\text{Sin}[qt])$$

Some of this may be review for some of you.

□T.6.a.i)

Look at these partial expansions of
Cos[x], Sin[x] and E^x
in powers of x:

```
cosexpansion = Normal[Series[Cos[x], {x, 0, 8}]]
```

$$1 - \frac{x^2}{2} + \frac{x^4}{24} - \frac{x^6}{720} + \frac{x^8}{40320}$$

```
sinexpansion = Normal[Series[Sin[x], {x, 0, 8}]]
```

```
x - x^3/6 + x^5/120 - x^7/5040
```

```
eexpansion = Normal[Series[E^x, {x, 0, 8}]]
```

```
1 + x + x^2/2 + x^3/6 + x^4/24 + x^5/120 + x^6/720 + x^7/5040 + x^8/40320
```

Notice that the terms in the expansion of

E^x

seem to have been lifted from the expansions of

Cos[x] and Sin[x].

Use complex numbers to explain this phenomenom.

□Answer:

Activate the complex (imaginary) number $I = \sqrt{-1}$:

```
I
```

```
I
```

```
I^2
```

```
-1
```

Now replace x by (I x) in the expansion of E^x:

```
Normal[Series[E^x, {x, 0, 10}]] /. x → I x
```

```
1 + I x - x^2/2 - I x^3/6 + x^4/24 + I x^5/120 - x^6/720 - I x^7/5040 + x^8/40320 + I x^9/362880 - x^10/3628800
```

Embedded in this is the expansion of Cos[x]:

```
Normal[Series[Cos[x], {x, 0, 10}]]
```

```
1 - x^2/2 + x^4/24 - x^6/720 + x^8/40320 - x^10/3628800
```

You can also see I times the expansion of Sin[x]:

```
Expand[I Normal[Series[Sin[x], {x, 0, 10}]]]
```

```
I x - I x^3/6 + I x^5/120 - I x^7/5040 + I x^9/362880
```

In fact:

```
k = Random[Integer, {15, 20}];
Normal[Series[E^x, {x, 0, k}]] /. x → I x
```

```
1 + I x - x^2/2 - I x^3/6 + x^4/24 + I x^5/120 - x^6/720 - I x^7/5040 + x^8/40320 + I x^9/362880 - x^10/3628800 - I x^11/39916800 + x^12/479001600 + I x^13/6227020800 - x^14/87178291200 - I x^15/1307674368000 + x^16/20922789888000 + I x^17/355687428096000 - x^18/6402373705728000 - I x^19/121645100408832000 + x^20/2432902008176640000
```

```
Normal[Series[Cos[x], {x, 0, k}]] +
 Expand[I Normal[Series[Sin[x], {x, 0, k}]]]
```

```
1 + I x - x^2/2 - I x^3/6 + x^4/24 + I x^5/120 - x^6/720 - I x^7/5040 + x^8/40320 + I x^9/362880 - x^10/3628800 - I x^11/39916800 + x^12/479001600 + I x^13/6227020800 - x^14/87178291200 - I x^15/1307674368000 + x^16/20922789888000 + I x^17/355687428096000 - x^18/6402373705728000 - I x^19/121645100408832000 + x^20/2432902008176640000
```

□T.6.a.ii)

Euler's formula

$E^{I x} = \text{Cos}[x] + I\,\text{Sin}[x]$

is so basic that it is programmed into *Mathematica*:

```
Clear[x]
ComplexExpand[E^(I x)]
```

```
Cos[x] + I Sin[x]
```

What's the story behind this basic formula?

□Answer:

You have already seen the whole story.

When you take the expansion of

E^x

in powers of x and change x to (I x), you get the expansion of

Cos[x]

plus I times the expansion of

Sin[x]

in powers of x.

That's why the whole world agrees that

$E^{I x} = \text{Cos}[x] + I\,\text{Sin}[x]$.

□T.6.a.iii)

For a real number x, you are pretty confident about calculating E^x. But if z = x + I y is a complex number, then how can you make sense of

$E^z = E^{x+I y}$

□Answer:

Just write

$z = x + I\,y$

and use normal laws of exponents:

$E^z = E^{x+I y} = E^x E^{I y} = E^x (\text{Cos}[y] + I\,\text{Sin}[y])$

So

$E^{x+I y} = E^x (\text{Cos}[y] + I\,\text{Sin}[y])$

Not much to it.

And *Mathematica* agrees:

```
Clear[x, y]
ComplexExpand[E^(x+I y)]
```

```
E^x Cos[y] + I E^x Sin[y]
```

□T.6.a.iv)

Here's a linear system:

```
A = {{-0.2, -1.3}, {0.9, 0.4}};
Clear[x, y, t]
linearsystem =
  ({x'[t], y'[t]} == A.{x[t], y[t]});
ColumnForm[Thread[linearsystem]]
```

```
x'[t] == -0.2 x[t] - 1.3 y[t]
y'[t] == 0.9 x[t] + 0.4 y[t]
```

With random starter data on {x[0], y[0]}:

```
{xstarter, ystarter} = {Random[Real, {-2, 2}], Random[Real, {-2, 2}]}
```

```
{-1.0692, 1.14798}
```

Here's its solution formula in complex exponential notation:

```
Clear[eigenvector, eigenvalue];
{eigenvalue[1], eigenvalue[2]} = Chop[Eigenvalues[A]];
{eigenvector[1], eigenvector[2]} = Chop[Eigenvectors[A]];

Clear[x, y, x1, y1, x2, y2, t, C1, C2]
{x1[t_], y1[t_]} = Chop[eigenvector[1] Exp[eigenvalue[1] t]];
{x2[t_], y2[t_]} = Chop[eigenvector[2] Exp[eigenvalue[2] t]];
{x[t_], y[t_]} = C1 {x1[t], y1[t]} + C2 {x2[t], y2[t]};

starter = {xstarter, ystarter};
starterequation = {x[0], y[0]} == starter;
Csols = Solve[starterequation];

{x[t_], y[t_]} = {x[t], y[t]} /. Csols[[1]]
```

```
{(-0.5346 - 0.563693 I) E^((0.1-1.03923 I) t) -
   (0.5346 - 0.563693 I) E^((0.1+1.03923 I) t),
  (0.57399 - 0.297281 I) E^((0.1-1.03923 I) t) +
   (0.57399 + 0.297281 I) E^((0.1+1.03923 I) t)}
```

How do you put this in real form showing off the sines and cosines which are embedded in this complex exponential?

□Answer:

Take another look at {x[t], y[t]}:

```
{x[t], y[t]}
```

```
{(-0.5346 - 0.563693 I) E^((0.1-1.03923 I) t) -
   (0.5346 - 0.563693 I) E^((0.1+1.03923 I) t),
  (0.57399 - 0.297281 I) E^((0.1-1.03923 I) t) +
   (0.57399 + 0.297281 I) E^((0.1+1.03923 I) t)}
```

Let *Mathematica* hit this complex formula with Euler's formula

$E^{a+I b} = E^a \text{Cos}[b] + I\,E^a \text{Sin}[b]$:

```
Chop[ComplexExpand[{x[t], y[t]}]]
```

```
{-1.0692 E^(0.1 t) Cos[1.03923 t] - 1.12739 E^(0.1 t) Sin[1.03923 t],
 1.14798 E^(0.1 t) Cos[1.03923 t] - 0.594561 E^(0.1 t) Sin[1.03923 t]}
```

There you go.

□T.6.a.v)

Here's the derivative with respect to t of
$f[t] = E^{(p+Iq)t}$:

```
Clear[f, p, q, t]
f[t_] = E^((p+I q) t);
f'[t]
```

$$E^{(p+I\,q)\,t}\,(p+I\,q)$$

Explain where this comes from.

□Answer:

Let *Mathematica* hit the formula for $f[t] = E^{(p+Iq)t}$ with Euler's formula

$$E^{a+Ib} = E^{a}\,\text{Cos}[b] + I\,E^{a}\,\text{Sin}[b]:$$

```
ComplexExpand[f[t]]
```

$$E^{p\,t}\,\text{Cos}[q\,t] + I\,E^{p\,t}\,\text{Sin}[q\,t]$$

Differentiate with respect to t:

```
Expand[∂t (E^(p t) Cos[q t]) + I ∂t (E^(p t) Sin[q t])]
```

$$E^{p\,t}\,p\,\text{Cos}[q\,t] + I\,E^{p\,t}\,q\,\text{Cos}[q\,t] + I\,E^{p\,t}\,p\,\text{Sin}[q\,t] - E^{p\,t}\,q\,\text{Sin}[q\,t]$$

Compare:

```
ComplexExpand[E^((p+I q) t) (p + I q)]
```

$$E^{p\,t}\,p\,\text{Cos}[q\,t] + I\,E^{p\,t}\,q\,\text{Cos}[q\,t] + I\,E^{p\,t}\,p\,\text{Sin}[q\,t] - E^{p\,t}\,q\,\text{Sin}[q\,t]$$

This is the same as:

```
f'[t]
```

$$E^{(p+I\,q)\,t}\,(p+I\,q)$$

DE.07 Eigenvectors and Eigenvalues for Linear Systems
Give It a Try!

G.1) Eigenvalue-trajectory analysis*

□G.1.a) Six linear systems

Below are six linear systems for you to analyze as indicated.

□G.1.a.i) Linear system 1

Here's a linear system:

```
Clear[x, y, t]
linearsystem =
({x'[t], y'[t]} == {-1.2 x[t] + 1.5 y[t], -2.3 x[t] + 0.3 y[t]})
```

$$\{x'[t],\, y'[t]\} == \{-1.2\,x[t] + 1.5\,y[t],\, -2.3\,x[t] + 0.3\,y[t]\}$$

Here it is in slightly better form:

```
ColumnForm[Thread[linearsystem]]
```

$$x'[t] == -1.2\,x[t] + 1.5\,y[t]$$
$$y'[t] == -2.3\,x[t] + 0.3\,y[t]$$

You read off the coefficient matrix A:

```
A = {{?, ?}, {?, ?}};
MatrixForm[A]
```

```
Move right in and replace the ?'s with the correct numbers.
```

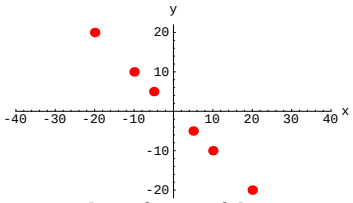
Now you can use this matrix to write the given linear system in this compact way:

```
Clear[x, y, t]
matrixlinearsystem = ({x'[t], y'[t]} == A.{x[t], y[t]});
ColumnForm[Thread[matrixlinearsystem]]
```

$$x'[t] == -0.2\,x[t] - 1.3\,y[t]$$
$$y'[t] == 0.9\,x[t] + 0.4\,y[t]$$

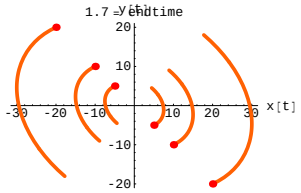
Here are six starter points:

```
Clear[starter];
starter[1] = {20, -20};
starter[2] = {10, -10};
starter[3] = {5, -5};
starter[4] = {-5, 5};
starter[5] = {-10, 10};
starter[6] = {-20, 20};
starterplot = Show[Table[
   Graphics[{Red, PointSize[0.03], Point[starter[j]]}], {j, 1, 6}],
  PlotRange → {{-40, 40}, {-22, 22}}, Axes → True,
  AxesLabel → {"x", "y"}];
```



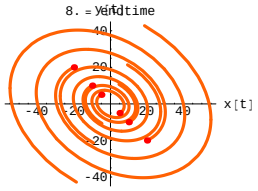
Here are plots of parts of the trajectories in this system starting at the plotted starter points:

```
Clear[trajectory]; trajectory[k_, t_] :=
 Chop[ComplexExpand[Expand[MatrixExp[A t] . starter[k]]]];

Clear[trajectoryplot, trajectoryplots, endtime]
trajectoryplot[k_, endtime_] :=
 ParametricPlot[Evaluate[trajectory[k, t]], {t, 0, endtime},
  PlotRange → All, PlotStyle → {{Thickness[0.015], CadmiumOrange}},
  AxesLabel → {"x[t]", "y[t]"},
  PlotLabel → endtime "= endtime", DisplayFunction → Identity];

trajectoryplots[endtime_] :=
 Table[trajectoryplot[k, endtime], {k, 1, 6}];
(trajectorystory[endtime_] := Show[trajectoryplots[endtime],
   starterplot, DisplayFunction → $DisplayFunction];)
trajectorystory[1.7];
```



See more:

```
trajectorystory[8.0];
```



Here are the calculations of the eigenvalues and eigenvectors of your coefficient matrix A above:

```
Clear[eigenvalue];
{eigenvalue[1], eigenvalue[2]} = Chop[Eigenvalues[A]]
```

$$\{0.1 + 1.03923\,I,\; 0.1 - 1.03923\,I\}$$

```
Clear[eigenvector];
{eigenvector[1], eigenvector[2]} = Chop[Eigenvectors[A]]
```

$$\{\{0.768706,\; -0.177394 - 0.61451\,I\},\; \{0.768706,\; -0.177394 + 0.61451\,I\}\}$$

Write up the information you get from these calculations and discuss how this information meshes with the trajectory plots above.

Identify which of the eigenvectors are propellers, suckers, pure whirlers, propelling whirlers or sucking whirlers.

Describe how the the specific numbers involved in these eigenvalues enable you to predict in advance how all the trajectories look.

Is this system stable in the sense that if {x[t], y[t]} are any two solutions of this linear system, then
$\{x[t], y[t]\} \to \{0, 0\}$ as $t \to \infty$?

□G.1.a.ii) Linear system 2

Here's a linear system:

```
Clear[x, y, t]
linearsystem = ({x'[t], y'[t]} ==
{-0.2 x[t] + 0.1 y[t], 0.2 x[t] - 0.4 y[t]})
```

$$\{x'[t],\, y'[t]\} == \{-0.2\,x[t] + 0.1\,y[t],\, 0.2\,x[t] - 0.4\,y[t]\}$$

Here it is in slightly better form:

```
ColumnForm[Thread[linearsystem]]
```

$$x'[t] == -0.2\,x[t] + 0.1\,y[t]$$
$$y'[t] == 0.2\,x[t] - 0.4\,y[t]$$

You read off the coefficient matrix A:

```
A = {{-0.2, 0.1}, {0.2, -0.4}};
MatrixForm[A]
```

$$\begin{pmatrix} -0.2 & 0.1 \\ 0.2 & -0.4 \end{pmatrix}$$

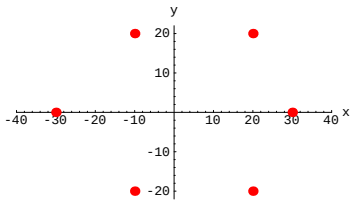
Now you can use this matrix to write the given linear system in this compact way:

```
Clear[x, y, t]
matrixlinearsystem = ({x'[t], y'[t]} == A.{x[t], y[t]});
ColumnForm[Thread[matrixlinearsystem]]
```

$$x'[t] == -0.2\,x[t] + 0.1\,y[t]$$
$$y'[t] == 0.2\,x[t] - 0.4\,y[t]$$

Here are six starter points:

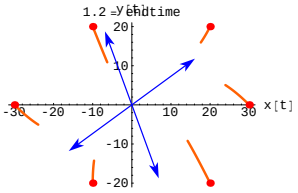
```
Clear[starter];
starter[1] = {20, 20};
starter[2] = {20, -20};
starter[3] = {-10, 20};
starter[4] = {-10, -20};
starter[5] = {30, 0};
starter[6] = {-30, 0};
starterplot = Show[Table[
   Graphics[{Red, PointSize[0.03], Point[starter[j]]}], {j, 1, 6}],
  PlotRange → {{-40, 40}, {-22, 22}}, Axes → True,
  AxesLabel → {"x", "y"}];
```



Here are plots of parts of the trajectories in this system starting at the plotted starter points shown with scaled eigenvectors of A:

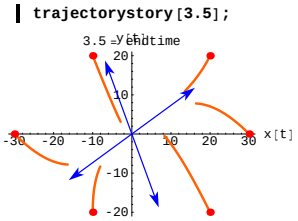
```
Clear[trajectory]; trajectory[k_, t_] :=
 Chop[ComplexExpand[Expand[MatrixExp[A t] . starter[k]]]];
```

```
Clear[trajectoryplot, trajectoryplots, endtime]
trajectoryplot[k_, endtime_] :=
 ParametricPlot[Evaluate[trajectory[k, t]], {t, 0, endtime},
  PlotRange → All, PlotStyle → {{Thickness[0.01], CadmiumOrange}},
  AxesLabel → {"x[t]", "y[t]"},
  PlotLabel → endtime "= endtime", DisplayFunction → Identity];

trajectoryplots[endtime_] :=
 Table[trajectoryplot[k, endtime], {k, 1, 6}]; Clear[eigenvector]
{eigenvector[1], eigenvector[2]} = Eigenvectors[A]; scaler = 20;
sizer = 4; eigenplot = {Arrow[eigenvector[1], Tail → {0, 0},
    VectorColor → Blue, ScaleFactor → scaler, HeadSize → sizer],
  Arrow[-eigenvector[1], Tail → {0, 0},
    VectorColor → Blue, ScaleFactor → scaler, HeadSize → sizer],
  Arrow[eigenvector[2], Tail → {0, 0},
    VectorColor → Blue, ScaleFactor → scaler, HeadSize → sizer],
  Arrow[-eigenvector[2], Tail → {0, 0},
    VectorColor → Blue, ScaleFactor → scaler, HeadSize → sizer]};
trajectorystory[endtime_] := Show[trajectoryplots[endtime],
  starterplot, eigenplot, DisplayFunction → $DisplayFunction];
trajectorystory[1.2];
```



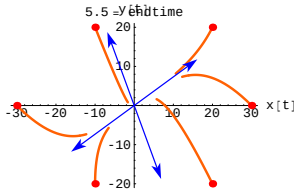
See more:

```
trajectorystory[3.5];
```



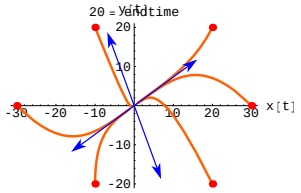
See more:

```
trajectorystory[5.5];
```



And more:

```
trajectorystory[20];
```



Grab the plots and align and animate.

Here are calculations of the eigenvalues and eigenvectors of your coefficient matrix A above:

```
Clear[eigenvalue];
{eigenvalue[1], eigenvalue[2]} = Chop[Eigenvalues[A]]
```

$$\{-0.473205,\ -0.126795\}$$

```
Clear[eigenvector];
{eigenvector[1], eigenvector[2]} = Chop[Eigenvectors[A]]
```

$$\{\{-0.343724,\ 0.939071\},\ \{0.806898,\ 0.59069\}\}$$

Write up the information you get from these calculations and discuss how this information meshes with the trajectory plots above.

Identify which of the eigenvectors are propellers, suckers, pure whirlers, propelling whirlers or sucking whirlers.

Which is dominant?

Describe how the the specific numbers involved in these eigenvalues enable you to predict in advance how all the trajectories look.

Is this system stable in the sense that if {x[t], y[t]} are any two solutions of this linear system, then

$$\{x[t],\ y[t]\} \to \{0,\ 0\} \text{ as } t \to \infty ?$$

□G.1.a.iii) Linear system 3

Here's a linear system:

```
Clear[x, y, t]
linearsystem =
 ({x'[t], y'[t]} == {-5.0 x[t] + 1.5 y[t], -2.3 x[t] + 5.0 y[t]})
```

$$\{x'[t],\ y'[t]\} == \{-5.\,x[t] + 1.5\,y[t],\ -2.3\,x[t] + 5.\,y[t]\}$$

Here it is in slightly better form:

```
ColumnForm[Thread[linearsystem]]
```

$$x'[t] == -5.\,x[t] + 1.5\,y[t]$$
$$y'[t] == -2.3\,x[t] + 5.\,y[t]$$

You read off the coefficient matrix A:

```
A = {{-5, 1.5}, {-2.3, 5.0}};
MatrixForm[A]
```

$$\begin{pmatrix} -5 & 1.5 \\ -2.3 & 5. \end{pmatrix}$$

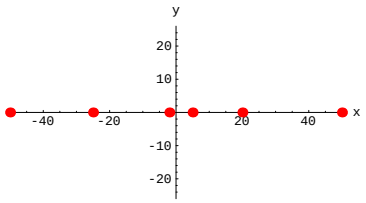
Now you can use this matrix to write the given linear system in this compact way:

```
Clear[x, y, t]
matrixlinearsystem = ({x'[t], y'[t]} == A.{x[t], y[t]});
ColumnForm[Thread[matrixlinearsystem]]
```

$$x'[t] == -5.\,x[t] + 1.5\,y[t]$$
$$y'[t] == -2.3\,x[t] + 5.\,y[t]$$

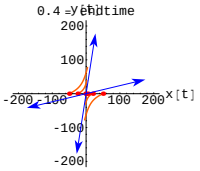
Here are six starter points:

```
Clear[starter];
starter[1] = {-50, 0};
starter[2] = {-25, 0};
starter[3] = {-2, 0};
starter[4] = {5, 0};
starter[5] = {20, 0};
starter[6] = {50, 0};
starterplot = Show[Table[
   Graphics[{Red, PointSize[0.03], Point[starter[j]]}], {j, 1, 6}],
  PlotRange → {{-50, 50}, {-26, 26}}, Axes → True,
  AxesLabel → {"x", "y"}];
```



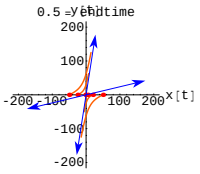
Here are plots of parts of the trajectories in this system starting at the plotted starter points shown with scaled eigenvectors of A:

```
Clear[trajectory]; trajectory[k_, t_] :=
 Chop[ComplexExpand[Expand[MatrixExp[A t] . starter[k]]]];

Clear[trajectoryplot, trajectoryplots, endtime]
trajectoryplot[k_, endtime_] :=
 ParametricPlot[Evaluate[trajectory[k, t]], {t, 0, endtime},
  PlotRange → All, PlotStyle → {{Thickness[0.01], CadmiumOrange}},
  AxesLabel → {"x[t]", "y[t]"},
  PlotLabel → endtime "= endtime", DisplayFunction → Identity];
trajectoryplots[endtime_] :=
 Table[trajectoryplot[k, endtime], {k, 1, 6}];

Clear[eigenvector]
{eigenvector[1], eigenvector[2]} = Eigenvectors[A]; scaler = 180;
sizer = 40; eigenplot = {Arrow[eigenvector[1], Tail → {0, 0},
    VectorColor → Blue, ScaleFactor → scaler, HeadSize → sizer],
  Arrow[-eigenvector[1], Tail → {0, 0},
    VectorColor → Blue, ScaleFactor → scaler, HeadSize → sizer],
  Arrow[eigenvector[2], Tail → {0, 0},
    VectorColor → Blue, ScaleFactor → scaler, HeadSize → sizer],
  Arrow[-eigenvector[2], Tail → {0, 0},
    VectorColor → Blue, ScaleFactor → scaler, HeadSize → sizer]};

trajectorystory[endtime_] :=
 Show[trajectoryplots[endtime], starterplot, eigenplot,
  PlotRange → {{-1.2 scaler, 1.2 scaler}, {-1.2 scaler, 1.2 scaler}},
  DisplayFunction → $DisplayFunction];
trajectorystory[0.4];
```

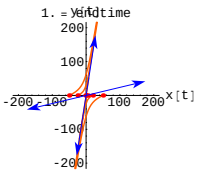


See more:

```
trajectorystory[0.5];
```




```
trajectorystory[1.0];
```



Grab the plots and animate.

Here are calculations of the eigenvalues and eigenvectors of your coefficient matrix A above:

```
Clear[eigenvalue];
{eigenvalue[1], eigenvalue[2]} = Chop[Eigenvalues[A]]
```

{4.6422, -4.6422}

```
Clear[eigenvector];
{eigenvector[1], eigenvector[2]} = Chop[Eigenvectors[A]]
```

{{-0.153717, -0.988115}, {-0.97271, -0.232025}}

Write up the information you get from these calculations and discuss how this information meshes with the trajectory plots above.

Identify which of the eigenvectors are propellers, suckers, pure whirlers, propelling whirlers or sucking whirlers.

Which is dominant?

Describe how the the specific numbers involved in these eigenvalues enable you to predict in advance how all the trajectories look.

For this linear system, you can use the entries in one of the eigenvectors to calculate the limiting ratio

$$\frac{y[t]}{x[t]} \quad \text{as } t \to \infty$$

for any solution pair {x[t], y[t]} (other than one of the straight line trajectories) .
Do it.

□G.1.a.iv) Linear system 4

Here's a linear system:

```
Clear[x, y, t]
linearsystem =
 ({x'[t], y'[t]} == {1.2 x[t] + 1.4 y[t], -3.3 x[t] + 0.5 y[t]})
```

{x′[t], y′[t]} == {1.2 x[t] + 1.4 y[t], -3.3 x[t] + 0.5 y[t]}

Here it is in slightly better form:

```
ColumnForm[Thread[linearsystem]]
```

x′[t] == 1.2 x[t] + 1.4 y[t]
y′[t] == -3.3 x[t] + 0.5 y[t]

You read off the coefficient matrix A:

```
A = {{1.2, 1.4}, {-3.3, 0.5}};
MatrixForm[A]
```

$$\begin{pmatrix} 1.2 & 1.4 \\ -3.3 & 0.5 \end{pmatrix}$$

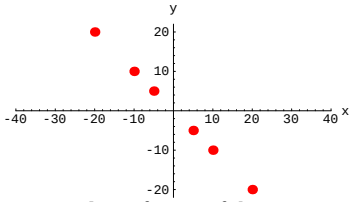
Now you can use this matrix to write the given linear system in this compact way:

```
Clear[x, y, t]
matrixlinearsystem = ({x'[t], y'[t]} == A . {x[t], y[t]});
ColumnForm[Thread[matrixlinearsystem]]
```

x′[t] == 1.2 x[t] + 1.4 y[t]
y′[t] == -3.3 x[t] + 0.5 y[t]

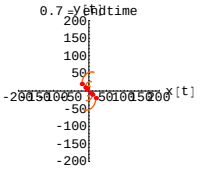
Here are six starter points:

```
Clear[starter];
starter[1] = {20, -20};
starter[2] = {10, -10};
starter[3] = {5, -5};
starter[4] = {-5, 5};
starter[5] = {-10, 10};
starter[6] = {-20, 20};
starterplot = Show[Table[
   Graphics[{Red, PointSize[0.03], Point[starter[j]]}], {j, 1, 6}],
  PlotRange → {{-40, 40}, {-22, 22}}, Axes → True,
  AxesLabel → {"x", "y"}];
```



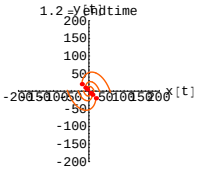
Here are plots of parts of the trajectories in this system starting at the plotted starter points:

```
Clear[trajectory];
trajectory[k_, t_] :=
 Chop[ComplexExpand[Expand[MatrixExp[A t] . starter[k]]]];

Clear[trajectoryplot, trajectoryplots, endtime]
trajectoryplot[k_, endtime_] :=
 ParametricPlot[Evaluate[trajectory[k, t]], {t, 0, endtime},
  PlotRange → All, PlotStyle → {{Thickness[0.01], CadmiumOrange}},
  AxesLabel → {"x[t]", "y[t]"},
  PlotLabel → endtime "= endtime", DisplayFunction → Identity];
trajectoryplots[endtime_] :=
 Table[trajectoryplot[k, endtime], {k, 1, 6}];

trajectorystory[endtime_] := Show[trajectoryplots[endtime],
  starterplot, PlotRange → {{-200, 200}, {-200, 200}},
  DisplayFunction → $DisplayFunction];
trajectorystory[0.7];
```



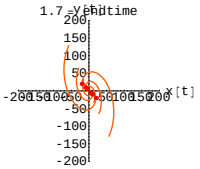
See more:

```
trajectorystory[1.2];
```



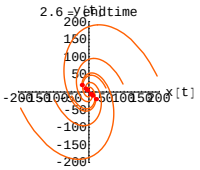
See more:

```
trajectorystory[1.7];
```



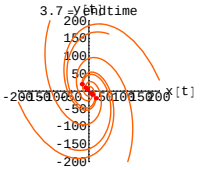
And more:

```
trajectorystory[2.6];
```



And more:

```
trajectorystory[3.7];
```



Align and animate.

Here are calculations of the eigenvalues and eigenvectors of your coefficient matrix A above:

```
Clear[eigenvalue];
{eigenvalue[1], eigenvalue[2]} = Chop[Eigenvalues[A]]
```

{0.85 + 2.12073 I, 0.85 - 2.12073 I}

```
Clear[eigenvector];
{eigenvector[1], eigenvector[2]} = Chop[Eigenvectors[A]]
```

{{-0.0888714 - 0.538493 I, 0.837931}, {-0.0888714 + 0.538493 I, 0.837931}}

Write up the information you get from these calculations and discuss how this information meshes with the trajectory plots above.

Identify which of the eigenvectors are propellers, suckers, pure whirlers, propelling whirlers or sucking whirlers.

Describe how the the specific numbers involved in these eigenvalues enable you to predict in advance how all the trajectories look.

Describe what happens to trajectories {x[t], y[t]} when t gets large.

□G.1..a.v) Linear system 5

Here's a linear system:

```
Clear[x, y, t]
linearsystem =
 ({x'[t], y'[t]} == {0.6 x[t] - 0.2 y[t], -0.3 x[t] + 0.8 y[t]})
```

{x′[t], y′[t]} == {0.6 x[t] - 0.2 y[t], -0.3 x[t] + 0.8 y[t]}

Here it is in slightly better form:

```
ColumnForm[Thread[linearsystem]]
```

x′[t] == 0.6 x[t] - 0.2 y[t]
y′[t] == -0.3 x[t] + 0.8 y[t]

You read off the coefficient matrix A:

```
A = {{0.6, -0.2}, {-0.3, 0.8}};
MatrixForm[A]
```

$$\begin{pmatrix} 0.6 & -0.2 \\ -0.3 & 0.8 \end{pmatrix}$$

Now you can use this matrix to write the given linear system in this compact way:

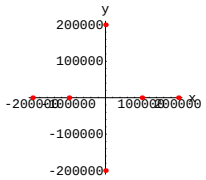
```
Clear[x, y, t]
matrixlinearsystem = ({x'[t], y'[t]} == A . {x[t], y[t]});
ColumnForm[Thread[matrixlinearsystem]]
```

x′[t] == 0.6 x[t] - 0.2 y[t]
y′[t] == -0.3 x[t] + 0.8 y[t]

Here are six starter points:

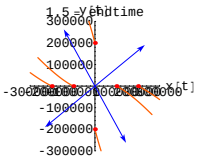
```
starter[1] = {200000, 0};
starter[2] = {100000, 0};
starter[3] = {-100000, 0};
starter[4] = {-200000, 0};
```

```
starter[5] = {0, 200000};
starter[6] = {0, -200000};
starterplot = Show[Table[
   Graphics[{Red, PointSize[0.03], Point[starter[j]]}], {j, 1, 6}],
  PlotRange → All, Axes → True, AxesLabel → {"x", "y"}];
```

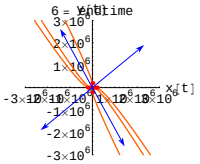


Here are plots of parts of the trajectories in this system starting at the plotted starter points:

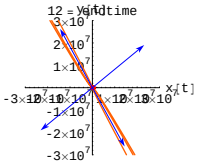
```
Clear[trajectory];
trajectory[k_, t_] :=
 Chop[ComplexExpand[Expand[MatrixExp[A t] . starter[k]]]];
Clear[trajectoryplot, trajectoryplots, endtime]
trajectoryplot[k_, endtime_] :=
 ParametricPlot[Evaluate[trajectory[k, t]], {t, 0, endtime},
  PlotRange → All, PlotStyle → {{Thickness[0.01], CadmiumOrange}},
  AxesLabel → {"x[t]", "y[t]"},
  PlotLabel → endtime "= endtime", DisplayFunction → Identity];
trajectoryplots[endtime_] :=
 Table[trajectoryplot[k, endtime], {k, 1, 6}];
Clear[eigenvector];
{eigenvector[1], eigenvector[2]} = Chop[Eigenvectors[A]];
scaler = 300000;
sizer = 40000;
eigenplot = {Arrow[eigenvector[1], Tail → {0, 0},
   VectorColor → Blue, ScaleFactor → scaler, HeadSize → sizer],
  Arrow[-eigenvector[1], Tail → {0, 0},
   VectorColor → Blue, ScaleFactor → scaler, HeadSize → sizer],
  Arrow[eigenvector[2], Tail → {0, 0},
   VectorColor → Blue, ScaleFactor → scaler, HeadSize → sizer],
  Arrow[-eigenvector[2], Tail → {0, 0},
   VectorColor → Blue, ScaleFactor → scaler, HeadSize → sizer]};
trajectorystory[endtime_] :=
 Show[trajectoryplots[endtime], starterplot,
  eigenplot, PlotRange → {{-scaler, scaler}, {-scaler, scaler}},
  DisplayFunction → $DisplayFunction];
trajectorystory[1.5];
```




```
scaler = 3000000;
sizer = 400000;
eigenplot = {Arrow[eigenvector[1], Tail → {0, 0},
    VectorColor → Blue, ScaleFactor → scaler, HeadSize → sizer],
  Arrow[-eigenvector[1], Tail → {0, 0},
    VectorColor → Blue, ScaleFactor → scaler, HeadSize → sizer],
  Arrow[eigenvector[2], Tail → {0, 0},
    VectorColor → Blue, ScaleFactor → scaler, HeadSize → sizer],
  Arrow[-eigenvector[2], Tail → {0, 0},
    VectorColor → Blue, ScaleFactor → scaler, HeadSize → sizer]};
trajectorystory[6];
```




```
scaler = 30000000;
sizer = 4000000;
eigenplot = {Arrow[eigenvector[1], Tail → {0, 0},
    VectorColor → Blue, ScaleFactor → scaler, HeadSize → sizer],
  Arrow[-eigenvector[1], Tail → {0, 0},
    VectorColor → Blue, ScaleFactor → scaler, HeadSize → sizer],
  Arrow[eigenvector[2], Tail → {0, 0},
    VectorColor → Blue, ScaleFactor → scaler, HeadSize → sizer],
  Arrow[-eigenvector[2], Tail → {0, 0},
    VectorColor → Blue, ScaleFactor → scaler, HeadSize → sizer]};
trajectorystory[12];
```



Here are the eigenvalues and eigenvectors of your coefficient matrix A above:

```
Clear[eigenvalue];
{eigenvalue[1], eigenvalue[2]} = Chop[Eigenvalues[A]]
```

```
{0.964575, 0.435425}
```

```
Clear[eigenvector];
{eigenvector[1], eigenvector[2]} = Chop[Eigenvectors[A]]
```

```
{{0.480965, -0.87674}, {-0.772178, -0.635406}}
```

Write up the information you get from these calculations and discuss how this information meshes with the trajectory plots above.

Identify which of the eigenvectors are propellers, suckers, pure whirlers, propelling whirlers or sucking whirlers.

Which is dominant?

Describe how the the specific numbers involved in these eigenvalues enable you to predict in advance how all the trajectories look.

For this linear system, you can use the entries in one of the eigenvectors to calculate the limiting ratio

$$\frac{y[t]}{x[t]} \quad \text{as } t \to \infty$$

for any solution pair {x[t], y[t]} (other than one of the straight line trajectories) .
Do it.

□G.1.a.vi) Linear system 6

Here's yet another linear system:

```
Clear[x, y, t]
linearsystem =
 ({x'[t], y'[t]} == {-1.4 x[t] - 0.8 y[t], 0.7 x[t] + 0.4 y[t]})
```

```
{x′[t], y′[t]} == {-1.4 x[t] - 0.8 y[t], 0.7 x[t] + 0.4 y[t]}
```

Here it is in slightly better form:

```
ColumnForm[Thread[linearsystem]]
```

```
x′[t] == -1.4 x[t] - 0.8 y[t]
y′[t] == 0.7 x[t] + 0.4 y[t]
```

You read off the coefficient matrix A:

```
A = {{-1.4, -0.8}, {0.7, 0.4}};
MatrixForm[A]
```

$$\begin{pmatrix} -1.4 & -0.8 \\ 0.7 & 0.4 \end{pmatrix}$$

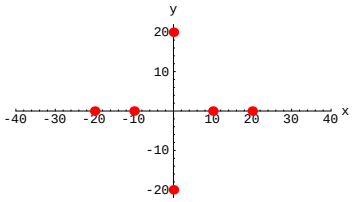
Now you can use this matrix to write the given linear system in this compact way:

```
Clear[x, y, t]
matrixlinearsystem = ({x'[t], y'[t]} == A . {x[t], y[t]});
ColumnForm[Thread[matrixlinearsystem]]
```

```
x′[t] == -1.4 x[t] - 0.8 y[t]
y′[t] == 0.7 x[t] + 0.4 y[t]
```

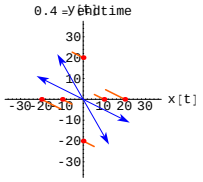
Here are six starter points:

```
Clear[starter];
starter[1] = {20, 0};
starter[2] = {10, 0};
starter[3] = {-10, 0};
starter[4] = {-20, 0};
starter[5] = {0, 20};
starter[6] = {0, -20};
starterplot = Show[Table[
   Graphics[{Red, PointSize[0.03], Point[starter[j]]}], {j, 1, 6}],
  PlotRange → {{-40, 40}, {-22, 22}}, Axes → True,
  AxesLabel → {"x", "y"}];
```



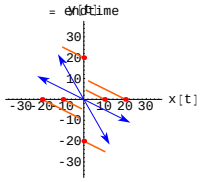
Here are plots of parts of the trajectories in this system starting at the plotted starter points shown with scaled eigenvectors:

```
Clear[trajectory];
trajectory[k_, t_] :=
 Chop[ComplexExpand[Expand[MatrixExp[A t] . starter[k]]]];
Clear[trajectoryplot, trajectoryplots, endtime]
trajectoryplot[k_, endtime_] :=
 ParametricPlot[Evaluate[trajectory[k, t]], {t, 0, endtime},
  PlotRange → All, PlotStyle → {{Thickness[0.01], CadmiumOrange}},
  AxesLabel → {"x[t]", "y[t]"},
  PlotLabel → endtime "= endtime", DisplayFunction → Identity];
trajectoryplots[endtime_] :=
 Table[trajectoryplot[k, endtime], {k, 1, 6}];
Clear[eigenvector]
{eigenvector[1], eigenvector[2]} = Eigenvectors[A];
scaler = 25;
sizer = 6;
eigenplot = {Arrow[eigenvector[1], Tail → {0, 0},
    VectorColor → Blue, ScaleFactor → scaler, HeadSize → sizer],
  Arrow[-eigenvector[1], Tail → {0, 0},
    VectorColor → Blue, ScaleFactor → scaler, HeadSize → sizer],
  Arrow[eigenvector[2], Tail → {0, 0},
    VectorColor → Blue, ScaleFactor → scaler, HeadSize → sizer],
  Arrow[-eigenvector[2], Tail → {0, 0},
    VectorColor → Blue, ScaleFactor → scaler, HeadSize → sizer]};
trajectorystory[endtime_] :=
 Show[trajectoryplots[endtime], starterplot, eigenplot,
  PlotRange → {{-1.5 scaler, 1.5 scaler}, {-1.5 scaler, 1.5 scaler}},
  DisplayFunction → $DisplayFunction];
trajectorystory[0.4];
```



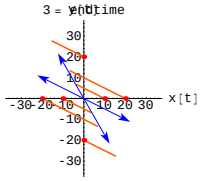
See more:

```
trajectorystory[1];
```

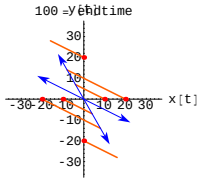


And more:

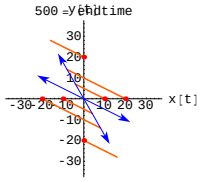
```
trajectorystory[3];
```




```
trajectorystory[100];
```




```
trajectorystory[500];
```



That's no error.
The trajectories do stall right where you see them.

Here are the eigenvalues and eigenvectors of your coefficient matrix A above:

```
Clear[eigenvalue];
{eigenvalue[1], eigenvalue[2]} = Chop[Eigenvalues[A]]
```

```
{-1., 0}
```

```
Clear[eigenvector];
{eigenvector[1], eigenvector[2]} = Chop[Eigenvectors[A]]
```

```
{{-0.894427, 0.447214}, {0.496139, -0.868243}}
```

Write up the information you get from these calculations and discuss how this information meshes with the trajectory plots above.

Identify which of the eigenvectors are propellers, suckers, pure whirlers, propelling whirlers, sucking whirlers or none of these.

Which seems to be dominant?

Describe how the the specific numbers involved in these eigenvalues enable you to predict in advance how all the trajectories look.

For this linear system, you can use the enties in one of the eigenvectors to calculate the limiting ratio

$\frac{y[t]}{x[t]}$ as $t \to \infty$

for any solution pair {x[t], y[t]}.
Do it.

G.2) Matrix nuts and eigen bolts

□G.2.a)

Here's a matrix

```
Clear[A, a, b, c, d, x, y]
A = {{0.5, 3.0}, {-1.0, 0.4}};
MatrixForm[A]
```

$$\begin{pmatrix} 0.5 & 3. \\ -1. & 0.4 \end{pmatrix}$$

Here's a vector.

```
{x, y} = {2.0, -3.0}
```

```
{2., -3.}
```

Here's the matrix (dot) multiplying the vector.

```
A . {x, y}
```

```
{-8., -3.2}
```

Account for the way this turned out.
Would you say that A . {x, y} is a vector or a number?

□G.2.b.i) Visual evidence about eigenvalues and eigenvectors

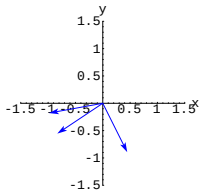
Here's a matrix A:

```
A = {{0.9, -0.3}, {-0.4, 1.3}};
MatrixForm[A]
```

$$\begin{pmatrix} 0.9 & -0.3 \\ -0.4 & 1.3 \end{pmatrix}$$

Here are its unit eigenvectors shown with a random unit vector:

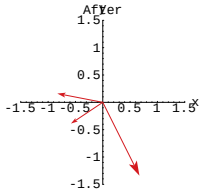
```
Clear[eigenvector]
{eigenvector[1], eigenvector[2]} = Eigenvectors[A];
s = Random[Real, {0, N[2 π]}];
randomvector = {Cos[s], Sin[s]};
ranger = 1.5;
before = Show[Arrow[eigenvector[1], Tail → {0, 0}, VectorColor → Blue],
  Arrow[eigenvector[2], Tail → {0, 0}, VectorColor → Blue],
  Arrow[randomvector, Tail → {0, 0}, VectorColor → Blue],
  Axes → True, PlotRange → {{-ranger, ranger}, {-ranger, ranger}},
  AxesLabel → {"x", "y"}];
```



If you don't see three distinct clear vectors, then rerun.

Here's what you get when you multiply all three plotted vectors by the matrix A:

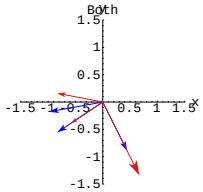
```
A = {{0.9, -0.3}, {-0.4, 1.3}}; after = Show[
  Arrow[A . eigenvector[1], Tail → {0, 0}, VectorColor → VenetianRed],
  Arrow[A . eigenvector[2], Tail → {0, 0}, VectorColor → VenetianRed],
  Arrow[A . randomvector, Tail → {0, 0}, VectorColor → VenetianRed],
  PlotRange → {{-ranger, ranger}, {-ranger, ranger}}, Axes → True,
  AxesLabel → {"x", "y"}, PlotLabel → "After"];
```



Grab both plots, align and animate.

Here are both plots:

```
Show[before, after, PlotLabel → "Both"];
```



Describe how this plot reveals which two of the original vectors are the eigenvectors of A.
Describe how this plot reveals that both eigenvalues of A are positive with one eigenvalue smaller than 1 and the other eigenvalue bigger than 1.

□G.2.b.ii)

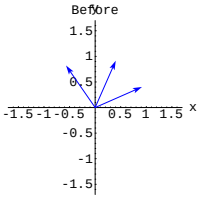
Here's a new matrix A:

```
A = {{1.1, -0.7}, {0.7, -0.8}};
MatrixForm[A]
```

$$\begin{pmatrix} 1.1 & -0.7 \\ 0.7 & -0.8 \end{pmatrix}$$

Here are its unit eigenvectors shown with a random unit vector:

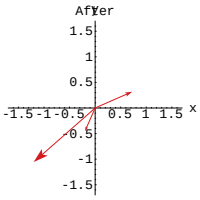
```
Clear[eigenvector]
{eigenvector[1], eigenvector[2]} = Eigenvectors[A];
s = Random[Real, {0, N[2 π]}];
randomvector = {Cos[s], Sin[s]};
ranger = 1.7;
before = Show[Arrow[eigenvector[1], Tail → {0, 0}, VectorColor → Blue],
  Arrow[eigenvector[2], Tail → {0, 0}, VectorColor → Blue],
  Arrow[randomvector, Tail → {0, 0}, VectorColor → Blue],
  Axes → True, PlotRange → {{-ranger, ranger}, {-ranger, ranger}},
  AxesLabel → {"x", "y"}, PlotLabel → "Before"];
```



If you don't see three distinct clear vectors, then rerun.

Here's what you get when you multiply all three plotted vectors by the matrix A:

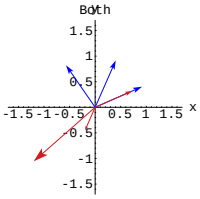
```
A = {{1.1, -0.7}, {0.7, -0.8}}; after = Show[
  Arrow[A . eigenvector[1], Tail → {0, 0}, VectorColor → VenetianRed],
  Arrow[A . eigenvector[2], Tail → {0, 0}, VectorColor → VenetianRed],
  Arrow[A . randomvector, Tail → {0, 0}, VectorColor → VenetianRed],
  PlotRange → {{-ranger, ranger}, {-ranger, ranger}}, Axes → True,
  AxesLabel → {"x", "y"}, PlotLabel → "After"];
```



Grab both plots and animate.

Here are both plots:

```
Show[before, after, PlotLabel → "Both"];
```



Describe how this plot reveals which two of the original vectors are the eigenvectors of A.

Describe how this plot reveals that one of the eigenvalues is positive and that the other is negative.

Describe how this plot reveals that the positive eigenvalue is less than 1.

□G.2.c.i)

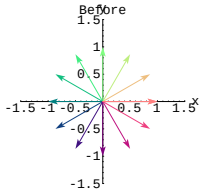
Here's a new matrix A:

```
A = {{1.425, -0.129904}, {-0.129904, 1.275}};
MatrixForm[A]
```

$$\begin{pmatrix} 1.425 & -0.129904 \\ -0.129904 & 1.275 \end{pmatrix}$$

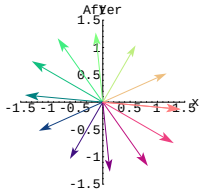
Here is a bunch of color-coded unit vectors:

```
Clear[vectorcolor, t]
vectorcolor[t_] = RGBColor[0.5 (Cos[t] + 1), 0.5 (Sin[t] + 1), 0.5];
unitvector[t_] = {Cos[t], Sin[t]};
before = Show[Table[Arrow[unitvector[t], Tail → {0, 0},
    VectorColor → vectorcolor[t]], {t, 0, 2 π, π/6}], Axes → True,
  AxesLabel → {"x", "y"}, PlotRange → {{-1.5, 1.5}, {-1.5, 1.5}},
  PlotLabel → "Before"];
```



Here are the same vectors after they have been multiplied by A.

```
A = {{1.425, -0.129904}, {-0.129904, 1.275}};
after = Show[Table[Arrow[A . unitvector[t], Tail → {0, 0},
    VectorColor → vectorcolor[t]], {t, 0, 2 π, π/6}], Axes → True,
  AxesLabel → {"x", "y"}, PlotRange → {{-1.5, 1.5}, {-1.5, 1.5}},
  PlotLabel → "After"];
```

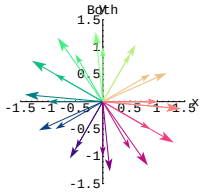


In this plot, A . unitvector[t] is the same color as unitvector[t] in the first plot.

Grab, align and animate slowly.

Here are both plots together:

```
Show[before, after, PlotLabel → "Both"];
```



In this plot, you see enough information to pick out the two eigenvectors of A and their negatives. Say how you made the identification.

□G.2.c.ii)

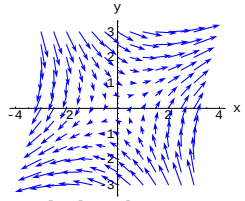
Here's a linear system:

```
Clear[x, y, t]
linearsystem =
 ({x'[t], y'[t]} == {0.5 x[t] + 0.867 y[t], 0.867 x[t] - 0.5 y[t]});
ColumnForm[Thread[linearsystem]]
```

$$x'[t] == 0.5\,x[t] + 0.867\,y[t]$$
$$y'[t] == 0.867\,x[t] - 0.5\,y[t]$$

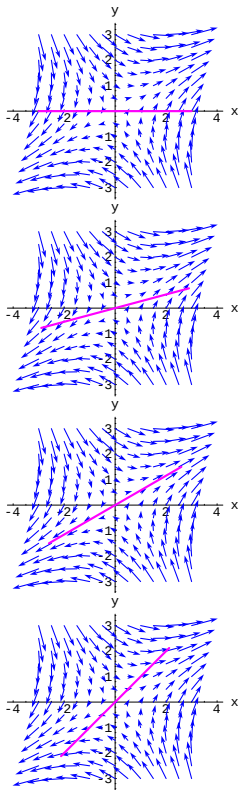
You read off the coefficient matrix A and use it to give a flow plot for this linear system:

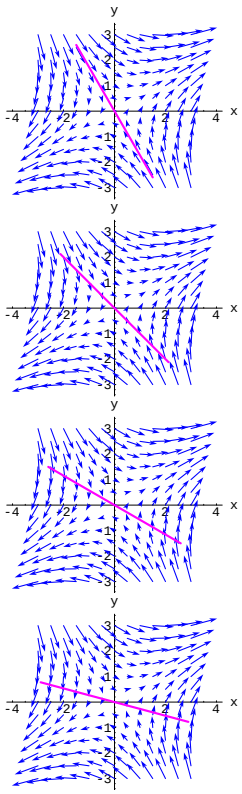
```
A = {{0.5, 0.867}, {0.867, -0.5}};
scalefactor = 0.25;
flowvectors =
 Table[Arrow[scalefactor A . {x, y}, Tail → {x, y}, VectorColor → Blue,
    ScaleFactor → 1, HeadSize → 0.3], {x, -3, 3, 6/12}, {y, -3, 3, 6/12}];
flowplot = Show[flowvectors, Axes → True, AxesLabel → {"x", "y"}];
```



Now look at these rotating lines:

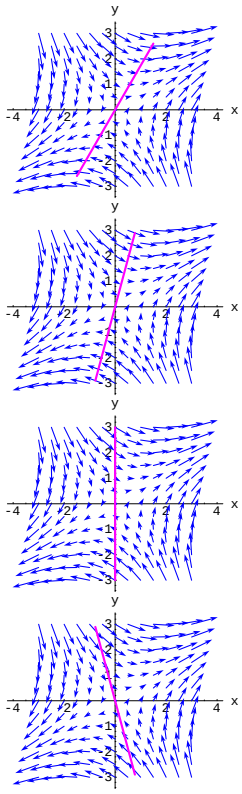
```
Clear[rotor, s]
rotor[s_] = Graphics[{Magenta, Thickness[0.01],
   Line[{-3 {Cos[s], Sin[s]}, 3 {Cos[s], Sin[s]}}]}];
Table[Show[flowplot, rotor[s], Axes → True], {s, 0, 11 π/12, π/12}];
```





Grab and animate.

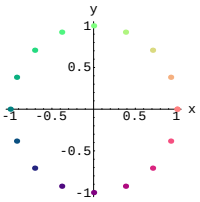
In two of these frames, the rine is running along eigenvectors of the coefficient matrix.
Copy and paste those two frames below and say a word about how you came to your conclusion.



□G.2.c.iii)

Here's a selction of points on the unit circle:

```
{slow, shigh} = {0, 2 π};
jump = π/8;
Clear[point, pointcolor, s]
point[s_] = {Cos[s], Sin[s]};
pointcolor[s_] = RGBColor[0.5 (Cos[s] + 1), 0.5 (Sin[s] + 1), 0.5];
pointplot =
 Table[Graphics[{pointcolor[s], PointSize[0.03], Point[point[s]]}],
  {s, slow, shigh, jump}];
Show[pointplot, Axes → True, AxesLabel → {"x", "y"}];
```



Here's another linear system:

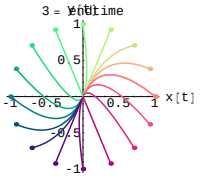
```
Clear[x, y, t]
linearsystem =
({x'[t], y'[t]} ==
   {-1.5 x[t] + 0.207107 y[t], 1.20711 x[t] - 1.5 y[t]});
ColumnForm[Thread[linearsystem]]
```

$$x'[t] == -1.5\,x[t] + 0.207107\,y[t]$$
$$y'[t] == 1.20711\,x[t] - 1.5\,y[t]$$

You read off the coefficient matrix A and plot the trajectories that start at each of the plotted points above:

```
A = {{-1.5, 0.207107}, {1.20711, -1.5}};
endtime = 3;
Clear[x, y, s, t]
Clear[trajectory];
trajectory[s_, t_] :=
 Chop[ComplexExpand[Expand[MatrixExp[A t] . point[s]]]];
Clear[trajectoryplot, trajectoryplots]
trajectoryplot[s_] := ParametricPlot[Evaluate[trajectory[s, t]],
  {t, 0, endtime}, PlotRange → All, PlotStyle →
   {{Thickness[0.01], pointcolor[s]}}, AxesLabel → {"x[t]", "y[t]"},
```

```
  PlotLabel → endtime "= endtime", DisplayFunction → Identity];
trajectoryplots = Table[trajectoryplot[s], {s, slow, shigh, jump}];
Show[trajectoryplots, pointplot, DisplayFunction → $DisplayFunction];
```



When you look at this plot, you can see that four trajectories that started on eigenvectors of A and their negatives. Say how you spotted them.

G.3) Eigenvalues in the formulas

□G.3.a.i)

Here's a new linear system:

```
Clear[x, y, t]
linearsystem =
  ({x'[t], y'[t]} == {-0.1 x[t] + 1.75 y[t], -3.0 x[t] - 0.1 y[t]});
ColumnForm[Thread[linearsystem]]
```

$$x'[t] == -0.1\,x[t] + 1.75\,y[t]$$
$$y'[t] == -3.\,x[t] - 0.1\,y[t]$$

The coefficient matrix for this linear system is:

```
A = {{-0.1, 1.75}, {-3.0, -0.1}};
MatrixForm[A]
```

$$\begin{pmatrix} -0.1 & 1.75 \\ -3. & -0.1 \end{pmatrix}$$

Here is the formula for a trajectory starting at a random point (a, b):

```
A = {{-0.1, 1.75}, {-3.0, -0.1}};
starter = {Random[Real, {-6, 6}], Random[Real, {-6, 6}]};
Clear[x, y, t]
{x[t_], y[t_]} = Chop[ComplexExpand[Expand[MatrixExp[A t] . starter]]]
```

$$\{-3.46\,E^{-0.1\,t}\,\text{Cos}[2.29129\,t] + 0.157637\,E^{-0.1\,t}\,\text{Sin}[2.29129\,t],$$
$$0.206396\,E^{-0.1\,t}\,\text{Cos}[2.29129\,t] + 4.5302\,E^{-0.1\,t}\,\text{Sin}[2.29129\,t]\}$$

You can read the eigenvalues of A directly off this formula.
Do it.

□G.3.a.ii)

Stay with the same linear system as in part i):

```
Clear[x, y, t]
linearsystem =
  ({x'[t], y'[t]} == {-0.1 x[t] + 1.75 y[t], -3.0 x[t] - 0.1 y[t]});
ColumnForm[Thread[linearsystem]]
```

$$x'[t] == -0.1\,x[t] + 1.75\,y[t]$$
$$y'[t] == -3.\,x[t] - 0.1\,y[t]$$

The coefficient matrix for this linear system is:

```
A = {{-0.1, 1.75}, {-3.0, -0.1}};
MatrixForm[A]
```

$$\begin{pmatrix} -0.1 & 1.75 \\ -3. & -0.1 \end{pmatrix}$$

Here are formulas for two trajectories starting at two random points:

```
A = {{-0.1, 1.75}, {-3.0, -0.1}};
starter1 = {Random[Real, {-6, 6}], Random[Real, {-6, 6}]};
starter2 = {Random[Real, {-6, 6}], Random[Real, {-6, 6}]};

 Chop[ComplexExpand[Expand[MatrixExp[A t] . starter1]]]
 Chop[ComplexExpand[Expand[MatrixExp[A t] . starter2]]]
```

$$\{-4.46458\,E^{-0.1\,t}\,\text{Cos}[2.29129\,t] + 3.07426\,E^{-0.1\,t}\,\text{Sin}[2.29129\,t],$$
$$4.02515\,E^{-0.1\,t}\,\text{Cos}[2.29129\,t] + 5.8455\,E^{-0.1\,t}\,\text{Sin}[2.29129\,t]\}$$
$$\{2.38629\,E^{-0.1\,t}\,\text{Cos}[2.29129\,t] - 4.2018\,E^{-0.1\,t}\,\text{Sin}[2.29129\,t],$$
$$-5.50145\,E^{-0.1\,t}\,\text{Cos}[2.29129\,t] - 3.12439\,E^{-0.1\,t}\,\text{Sin}[2.29129\,t]\}$$

Your job is to explain why some (but not all) of the numerical constants are the same in both formulas.
Do it.

□G.3.b.i)

Here's a linear system:

```
Clear[x, y, t]
linearsystem =
  ({x'[t], y'[t]} == {-1.5 x[t] + 0.3 y[t], 0.6 x[t] - 0.5 y[t]});
ColumnForm[Thread[linearsystem]]
```

$$x'[t] == -1.5\,x[t] + 0.3\,y[t]$$
$$y'[t] == 0.6\,x[t] - 0.5\,y[t]$$

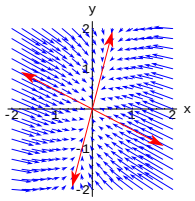
The coefficient matrix for this linear system is:

```
A = {{-1.5, 0.3}, {0.6, -0.5}};
MatrixForm[A]
```

$$\begin{pmatrix} -1.5 & 0.3 \\ 0.6 & -0.5 \end{pmatrix}$$

Intent on seeing the greater scheme of things, you look at the flow with the eigenvectors:

```
Clear[eigenvector];
{eigenvector[1], eigenvector[2]} = Chop[Eigenvectors[A]];
Clear[Field]
Field[x_, y_] = A . {x, y};
scalefactor = 0.2;
flowplot = Table[Arrow[Field[x, y], Tail → {x, y},
    VectorColor → Blue, ScaleFactor → 0.2, HeadSize → 0.15],
  {x, -2, 2, 0.25}, {y, -2, 2, 0.25}];
scaler = 2.0;
sizer = 0.4;
eigenplot = {Arrow[eigenvector[1], Tail → {0, 0},
    VectorColor → Red, ScaleFactor → scaler, HeadSize → sizer],
  Arrow[-eigenvector[1], Tail → {0, 0},
    VectorColor → Red, ScaleFactor → scaler, HeadSize → sizer],
  Arrow[eigenvector[2], Tail → {0, 0},
    VectorColor → Red, ScaleFactor → scaler, HeadSize → sizer],
  Arrow[-eigenvector[2], Tail → {0, 0},
    VectorColor → Red, ScaleFactor → scaler, HeadSize → sizer]};
revealer =
 Show[flowplot, eigenplot, Axes → True, AxesLabel → {"x", "y"}];
```



Take a look at the eigenvalues and eigenvectors of the coefficient matrix A:

```
Clear[eigenvalue];
{eigenvalue[1], eigenvalue[2]} = Chop[Eigenvalues[A]]
```

$$\{-1.65574, -0.344256\}$$

```
Clear[eigenvector];
{eigenvector[1], eigenvector[2]} = Chop[Eigenvectors[A]]
```

$$\{\{-0.887527, 0.460756\}, \{-0.251247, -0.967923\}\}$$

Use what you see to parameterize straight line trajectories starting at the tips of each calculated eigenvector.

□G.3.b.ii)

Stay with the same linear system as in part i) immediately above.
Formulas for the solutions x[t] and y[t] resulting from starter data $x[0] = -1.7$ and $y[0] = -1.8$ are:

```
starter = {-1.7, -1.8};
Clear[x, y, t]
{x[t_], y[t_]} = Chop[ComplexExpand[Expand[MatrixExp[A t] . starter]]]
```

$$\{-1.08637\,E^{-1.65574\,t} - 0.613627\,E^{-0.344256\,t},$$
$$0.563986\,E^{-1.65574\,t} - 2.36399\,E^{-0.344256\,t}\}$$

Reproduce this formula using only the following information:
→ the eigenvectors of A
→ the eigenvalues of A
→ {x[0], y[0]} = {−1.7, −1.8}.

□G.3.c.i)

Here's a new linear system:

```
Clear[x, y, t]
linearsystem =
  ({x'[t], y'[t]} == {-0.5 x[t] + 1.5 y[t], -1.7 x[t] + 0.1 y[t]});
ColumnForm[Thread[linearsystem]]
```

$$x'[t] == -0.5\,x[t] + 1.5\,y[t]$$
$$y'[t] == -1.7\,x[t] + 0.1\,y[t]$$

The coefficient matrix for this linear system is:

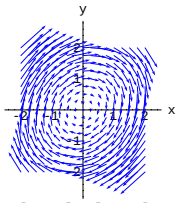
```
A = {{-0.5, 1.5}, {-1.7, 0.1}};
MatrixForm[A]
```

$$\begin{pmatrix} -0.5 & 1.5 \\ -1.7 & 0.1 \end{pmatrix}$$

Intent on seeing the greater scheme of things, you look at the flow.

```
A = {{-0.5, 1.5}, {-1.7, 0.1}};
Clear[Field]
```

```
Field[x_, y_] = A . {x, y};
scalefactor = 0.2;
flowplot = Table[Arrow[Field[x, y], Tail → {x, y},
    VectorColor → Blue, ScaleFactor → 0.2, HeadSize → 0.15],
  {x, -2, 2, 0.25}, {y, -2, 2, 0.25}];
revealer = Show[flowplot, Axes → True, AxesLabel → {"x", "y"}];
```



Take a look at the eigenvalues and eigenvectors of the coefficient matrix A:

```
Clear[eigenvalue];
{eigenvalue[1], eigenvalue[2]} = Chop[Eigenvalues[A]]
```

$\{-0.2 + 1.56844\,I,\ -0.2 - 1.56844\,I\}$

```
Clear[eigenvector];
{eigenvector[1], eigenvector[2]} = Chop[Eigenvectors[A]]
```

$\{\{0.128624 - 0.672463\,I,\ 0.728869\},\ \{0.128624 + 0.672463\,I,\ 0.728869\}\}$

How is the presence of $I = \sqrt{-1}$ in the eigenvalues reflected by the behavior of the flow?

□G.3.c.ii)

Stay with the same linear system as in part i) immediately above. Formulas for the solutions x[t] and y[t] resulting from starter data

x[0] = 1.9 and y[0] = 0

are:

```
starter = {1.9, 0};
Clear[x, y, t]
{x[t_], y[t_]} = Chop[ComplexExpand[Expand[MatrixExp[A t] . starter]]]
```

$\{1.9\,E^{-0.2\,t}\,\text{Cos}[1.56844\,t] - 0.363419\,E^{-0.2\,t}\,\text{Sin}[1.56844\,t],\ -2.05937\,E^{-0.2\,t}\,\text{Sin}[1.56844\,t]\}$

Reproduce this formula using only the following information:
→ the eigenvectors of A
→ the eigenvalues of A

→ {x[0], y[0]} = {1.9, 0}.

□G.3.d) Calculus Cal screws up again

The characters Bright Bridget and Calculus Cal are based on real people.

If you've been around C&M for a while, then you remember the infamous lab pest, Calculus Cal. Poor Cal is always a couple of bricks short of a load in everything. His hair is not quite combed, he sprays you with little bits of saliva when he talks to you, and there is often a hint of a questionable odor about him.

But Cal thinks he is really a slick dude who is good at math and science and likes to show off. Everybody else knows that Cal is just an information sink - what goes in does not come out.

One day Bright Bridget is working away at DiffEq&*Mathematica* and, much to her revulsion, Calculus Cal sits down at the computer next to her. Cal begins to work on the problem of coming up with formulas and using them to plot some sample trajectories for the following linear system:

```
Clear[x, y, t]
linearsystem =
({x'[t], y'[t]} == {0.2 x[t] - 0.4 y[t], -1.2 x[t] + 0.3 y[t]});
ColumnForm[Thread[linearsystem]]
```

$x'[t] == 0.2\,x[t] - 0.4\,y[t]$
$y'[t] == -1.2\,x[t] + 0.3\,y[t]$

Cal correctly reads off the right matrix:

```
A = {{0.2, -0.4}, {-1.2, 0.3}};
MatrixForm[A]
```

$$\begin{pmatrix} 0.2 & -0.4 \\ -1.2 & 0.3 \end{pmatrix}$$

And Cal uses this matrix to write the given linear system in this compact way:

```
Clear[x, y, t]
matrixlinearsystem = ({x'[t], y'[t]} == A . {x[t], y[t]});
ColumnForm[Thread[matrixlinearsystem]]
```

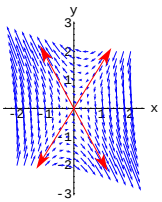
$x'[t] == 0.2\,x[t] - 0.4\,y[t]$
$y'[t] == -1.2\,x[t] + 0.3\,y[t]$

And he plots the flow with eigenvectors:

```
A = {{0.2, -0.4}, {-1.2, 0.3}};
Clear[eigenvector];
{eigenvector[1], eigenvector[2]} = Chop[Eigenvectors[A]]
```

$\{\{0.473228, -0.88094\}, \{-0.527258, -0.849705\}\}$

```
Clear[Field]
Field[x_, y_] = A . {x, y};
scalefactor = 0.35;
flowplot = Table[Arrow[Field[x, y], Tail → {x, y},
    VectorColor → Blue, ScaleFactor → 0.3, HeadSize → 0.15],
  {x, -2, 2, 0.25}, {y, -2, 2, 0.25}];
scaler = 2.5;
sizer = 0.6;
eigenflowplot = Show[flowplot, Arrow[eigenvector[1], Tail → {0, 0},
    VectorColor → Red, ScaleFactor → scaler, HeadSize → sizer],
  Arrow[-eigenvector[1], Tail → {0, 0},
    VectorColor → Red, ScaleFactor → scaler, HeadSize → sizer],
  Arrow[eigenvector[2], Tail → {0, 0},
    VectorColor → Red, ScaleFactor → scaler, HeadSize → sizer],
  Arrow[-eigenvector[2], Tail → {0, 0}, VectorColor → Red,
    ScaleFactor → scaler, HeadSize → sizer], Axes → True,
  AxesLabel → {"x", "y"}];
```



At this point, Bridget is getting a little uneasy because Cal has done everything right so far.
Meanwhile, knowing that the trajectories along the eigenvectors are straight lines, Cal parameterizes them as straight lines and checks:

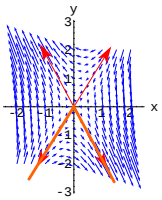
```
Clear[x1, y1, x2, y2, t]
{x1[t_], y1[t_]} = t eigenvector[1]
```

$\{0.473228\,t,\ -0.88094\,t\}$

```
{x2[t_], y2[t_]} = t eigenvector[2]
```

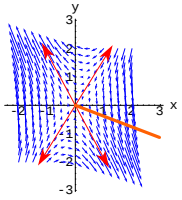
$\{-0.527258\,t,\ -0.849705\,t\}$

```
endtime = 3;
straighttraj1 =
 ParametricPlot[{x1[t], y1[t]}, {t, 0, endtime}, PlotStyle →
   {CadmiumOrange, Thickness[0.02]}}, DisplayFunction → Identity];
straighttraj2 =
 ParametricPlot[{x2[t], y2[t]}, {t, 0, endtime}, PlotStyle →
   {Thickness[0.02], CadmiumOrange}}, DisplayFunction → Identity];
Show[eigenflowplot, straighttraj1, straighttraj2, PlotRange → All,
 DisplayFunction → $DisplayFunction];
```



Cal smiles in satisfaction. He ghinks he has nailed those straight line trajectories!
Looking at Cal's computer, Bridget also smiles, thinking "This is going to be fun because Cal's on the verge of another screw-up."
She saves her file and settles back to enjoy the fun as Cal tries to plot a what he thinks is a sample trajectory:

```
Clear[x, y]
{x[t_], y[t_]} = 1.2 {x1[t], y1[t]} - 0.8 {x2[t], y2[t]};

trajectoryplot =
 ParametricPlot[{x[t], y[t]}, {t, 0, endtime}, PlotStyle →
   {Thickness[0.02], CadmiumOrange}}, DisplayFunction → Identity];
Show[
 eigenflowplot, trajectoryplot, DisplayFunction → $DisplayFunction];
```



Bridget bursts out in laughter.
Cal is bewildered. He looks at Bridget and says, "I did everything right; it says in the Basics that every trajectory {x[t], y[t]} is a combination of the two straight line trajectories like this:
{x[t], y[t]} = C1 {x1[t], y1[t]} + C2 {x2[t], y2[t]}.
But my sample trajectory does not go with the flow. What's wrong?"
Bridget says, "You made a big mistake, you silly dork."

Your job is to step in and take Bridget's role, telling Cal where he screwed up and how to fix it.

G.4) Eigenvalues signal speed of swirl and suck

□G.4.a.i)

Here's a linear system:

```
Clear[x, y, t]
linearsystem1 =
 ({x'[t], y'[t]} == {-0.1 x[t] - 6.75 y[t], 3.0 x[t] - 0.1 y[t]});
ColumnForm[Thread[linearsystem1]]
```

```
x′[t] == -0.1 x[t] - 6.75 y[t]
y′[t] == 3. x[t] - 0.1 y[t]
```

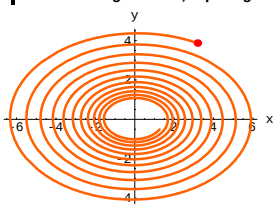
The coefficient matrix for this linear system is:

```
A1 = {{-0.1, -6.75}, {3.0, -0.1}};
MatrixForm[A1]
```

$$\begin{pmatrix} -0.1 & -6.75 \\ 3. & -0.1 \end{pmatrix}$$

Here is the plot of a trajectory of this system starting at a random point:

```
A1 = {{-0.1, -6.75}, {3.0, -0.1}};
endtime = 15;
starter = {Random[Real, {-4, 4}], Random[Real, {-4, 4}]};

Clear[x1, y1, t];
{x1[t_], y1[t_]} =
 Chop[ComplexExpand[Expand[MatrixExp[A1 t] . starter]]];

trajectoryplot1 =
 ParametricPlot[{x1[t], y1[t]}, {t, 0, endtime}, PlotStyle →
   {{CadmiumOrange, Thickness[0.01]}}, AxesLabel → {"x", "y"},
  PlotRange → All, Epilog → {Red, PointSize[0.03], Point[starter]}];
```



Here are the exact formulas for this solution pair:

```
{x1[t], y1[t]}
```

$$\{3.18067\, E^{-0.1\,t} \cos[4.5\,t] - 5.81132\, E^{-0.1\,t} \sin[4.5\,t],\ 3.87422\, E^{-0.1\,t} \cos[4.5\,t] + 2.12045\, E^{-0.1\,t} \sin[4.5\,t]\}$$

Here are the eigenvalues of the coefficient matrix:

```
Eigenvalues[A1]
```

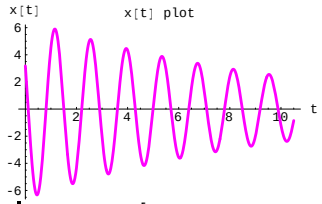
$$\{-0.1 + 4.5\,I,\ -0.1 - 4.5\,I\}$$

Discuss how the eigenvalues of A show up in the formulas for the solutions.

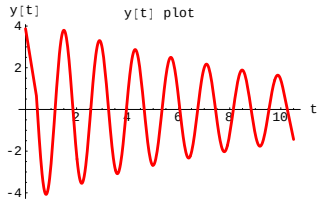
Discuss how the eigenvalues of A signal that ALL trajectories in this linear system spiral in and eventually stall on {0, 0}.

Discuss how this tells you that if x[t] and y[t] solve this linear system, then both x[t] and y[t] oscillate above 0 and below 0 like this:

```
xplot = Plot[x1[t], {t, 0, 0.7 endtime},
  PlotStyle → {{Magenta, Thickness[0.01]}}, AxesLabel → {"t", "x[t]"},
  AspectRatio → 1/GoldenRatio, PlotLabel → "x[t] plot"];
```




```
yplot = Plot[y1[t], {t, 0, 0.7 endtime},
  PlotStyle → {{Red, Thickness[0.01]}}, AxesLabel → {"t", "y[t]"},
  AspectRatio → 1/GoldenRatio, PlotLabel → "y[t] plot"];
```



□G.4.a.ii)

Here is yet another linear system:

```
Clear[x, y, t]
linearsystem2 =
({x'[t], y'[t]} == {-0.5 x[t] - 4.5 y[t], 2.0 x[t] - 0.5 y[t]});
ColumnForm[Thread[linearsystem2]]
```

```
x′[t] == -0.5 x[t] - 4.5 y[t]
y′[t] == 2. x[t] - 0.5 y[t]
```

The coefficient matrix for this linear system is:

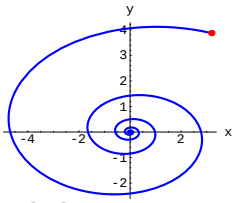
```
A2 = {{-0.5, -4.5}, {2.0, -0.5}};
MatrixForm[A2]
```

$$\begin{pmatrix} -0.5 & -4.5 \\ 2. & -0.5 \end{pmatrix}$$

Here are the formulas for the solution of this system starting at the same point as the trajectory in part i) started and run for the same endtime as the trajectory in part i):

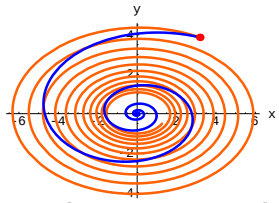
```
A2 = {{-0.5, -4.5}, {2.0, -0.5}};
Clear[x2, y2, t]
```

```
{x2[t_], y2[t_]} =
 Chop[ComplexExpand[Expand[MatrixExp[A2 t] . starter]]];
trajectoryplot2 = ParametricPlot[{x2[t], y2[t]}, {t, 0, endtime},
  PlotStyle → {{Blue, Thickness[0.01]}}, AxesLabel → {"x", "y"},
  PlotRange → All, Epilog → {Red, PointSize[0.03], Point[starter]}];
```



See both trajectories together:

```
Show[trajectoryplot1, trajectoryplot2,
 DisplayFunction → $DisplayFunction];
```



Notice that trajectory2 is sucked to {0, 0} quite a bit more rapidly than trajectory2.
Look at the eigenvalues of the coefficient matrix corresponding to trajectory1:

```
Eigenvalues[A1]
```

$$\{-0.1 + 4.5\,I,\ -0.1 - 4.5\,I\}$$

Look at the eigenvalues of the coefficient matrix corresponding to trajectory2:

```
Eigenvalues[A2]
```

$$\{-0.5 + 3.\,I,\ -0.5 - 3.\,I\}$$

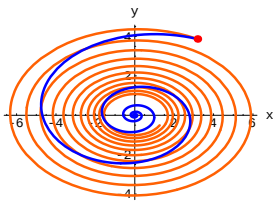
Use the eigenvalue information you have to explain this:
When you start a trajectory in linearsystem1 (with coefficient matrix A1) and start another trajectory in linearsystem2 (with coefficient matrix A2) at the same point, then your trajectory in linearsystem2 is

guaranteed to be sucked towards {0, 0} quite a bit more rapidly than your trajectory in linearsystem1 .

□G.4.a.iii)

Keep everything the same as in part ii) and look at the two trajectory plots - one for each of the two linear systems:

```
Show[trajectoryplot1, trajectoryplot2,
 DisplayFunction → $DisplayFunction];
```



Although trajectory2 is sucked to {0, 0} faster than trajectory1, you can see that trajectory1 oscillates quite a bit more more rapidly than trajectory2 .

Look at the eigenvalues of the coefficient matrix corresponding to trajectory1:

```
Eigenvalues[A1]
```

```
{-0.1 + 4.5 I, -0.1 - 4.5 I}
```

Look at the eigenvalues of the coefficient matrix corresponding to trajectory2:

```
Eigenvalues[A2]
```

```
{-0.5 + 3. I, -0.5 - 3. I}
```

Use what you see to explain why it was guaranteed that trajectory1 oscillates more rapidly than trajectory2 .

□G.4.b)

Here are two linear systems:

```
Clear[x, y, t]
linearsystem1 =
({x'[t], y'[t]} == {0.5 x[t] - 3.0 y[t], 6.8 x[t] - 0.4 y[t]});
ColumnForm[Thread[linearsystem1]]
```

$$x'[t] == 0.5\,x[t] - 3.\,y[t]$$
$$y'[t] == 6.8\,x[t] - 0.4\,y[t]$$

```
Clear[x, y, t]
linearsystem2 =
({x'[t], y'[t]} == {1.4 x[t] - 2.0 y[t], 5.8 x[t] - 0.4 y[t]});
ColumnForm[Thread[linearsystem2]]
```

$$x'[t] == 1.4\,x[t] - 2.\,y[t]$$
$$y'[t] == 5.8\,x[t] - 0.4\,y[t]$$

Write down the coefficient matrices A1 and A2 for these systems and use *Mathematica* to calculate their eigenvalues.

→ Use the eigenvalue information to explain how you know that trajectories in both linear systems are spirals propelled away from {0, 0}.

→ Use the eigenvalue information to answer the next two questions:

1) When you start a trajectory in linearsystem1 and start another trajectory in linearsystem2 at the same point, which of the two trajectories will be propelled away from {0, 0} faster?

2) When you start a trajectory in linearsystem1 and start another trajectory in linearsystem2 at the same point, which of the two trajectories will oscillate the more rapidly?

G.5) Analyzing the effect of resistance in simple parallel electrical circuits*

□G.5.a.i)

```
You don't have to understand the
electrical jargon to be able to do this problem.
```

You are fortunate enough to have your own computer set-up right in your dorm room. One night, Brian, the EE student who lives across the hall comes in and says, "Part of my EE 250 homework for tomorrow is to analyze the effect of varying the size of the resistance on the current in a simple parallel electrical circuit. I don't have a clue about where to start."

At first, you think, "I'm scared because I'm a life science major and I don't know anything about electrical circuits." But you don't let on.

Instead, you say, "Let me see that EE 250 textbook." In the book you see that in a simple parallel electrical circuit, the main measurements are

x[t] = voltage drop across the capacitor and
y[t] = current through the inductor

Reading on, you spot that these measurements are related through the linear system

$$x'[t] = \frac{y[t]}{L}$$

$$y'[t] = -\frac{x[t]}{C} - \frac{y[t]}{R\,C}$$

where L, C, R are given positive numbers with

L = inductance
C = capacitance, and
R = resistance.

You say, "Brian, I don't even know what those words mean, but I think I can help you." You type:

```
Clear[x, y, t, L, R, c];
linearsystem =
 ({x'[t], y'[t]} == {y[t] / L, - x[t] / c - y[t] / (R c)});
ColumnForm[Thread[linearsystem]]
```

$$x'[t] == \frac{y[t]}{L}$$
$$y'[t] == -\frac{x[t]}{c} - \frac{y[t]}{c\,R}$$

You say that the coefficient matrix for this linear system is:

```
Clear[A];
A[L_, R_, c_] = {{0, 1/L}, {-1/c, -1/(R c)}};
MatrixForm[A[L, R, c]]
```

$$\begin{pmatrix} 0 & \frac{1}{L} \\ -\frac{1}{c} & -\frac{1}{c\,R} \end{pmatrix}$$

Now you ask Brian, "Is it OK to go with C = 0.8, L = 5.0, and to go with starter data {x[0], y[0]} = {2.1, 1.0}?"

Eagerly, he says, "Sure!"

You enter C = 0.8, L = 5.0 and then look at the eigenvalues of A[L, R, C] for R running from 0.2 to 8.0 in increments of 0.4:

```
c = 0.8;
L = 5.0;
Clear[R];
ColumnForm[Table[{R, Eigenvalues[A[L, R, c]]}, {R, 0.2, 8.0, 0.4}]]
```

```
{0.2, {-6.20974, -0.0402593}}
{0.6, {-1.95549, -0.127845}}
{1., {-1., -0.25}}
{1.4, {-0.446429 + 0.22517 I, -0.446429 - 0.22517 I}}
{1.8, {-0.347222 + 0.359773 I, -0.347222 - 0.359773 I}}
{2.2, {-0.284091 + 0.411452 I, -0.284091 - 0.411452 I}}
{2.6, {-0.240385 + 0.438424 I, -0.240385 - 0.438424 I}}
{3., {-0.208333 + 0.45453 I, -0.208333 - 0.45453 I}}
{3.4, {-0.183824 + 0.464983 I, -0.183824 - 0.464983 I}}
{3.8, {-0.164474 + 0.472174 I, -0.164474 - 0.472174 I}}
{4.2, {-0.14881 + 0.477342 I, -0.14881 - 0.477342 I}}
{4.6, {-0.13587 + 0.481185 I, -0.13587 - 0.481185 I}}
{5., {-0.125 + 0.484123 I, -0.125 - 0.484123 I}}
{5.4, {-0.115741 + 0.48642 I, -0.115741 - 0.48642 I}}
{5.8, {-0.107759 + 0.48825 I, -0.107759 - 0.48825 I}}
{6.2, {-0.100806 + 0.489733 I, -0.100806 - 0.489733 I}}
{6.6, {-0.094697 + 0.490951 I, -0.094697 - 0.490951 I}}
{7., {-0.0892857 + 0.491963 I, -0.0892857 - 0.491963 I}}
{7.4, {-0.0844595 + 0.492815 I, -0.0844595 - 0.492815 I}}
{7.8, {-0.0801282 + 0.493538 I, -0.0801282 - 0.493538 I}}
```

You say that the current function y[t] will plot out a lot differently for the smaller positive resistance measurements R than for larger resistance measurements but it seems that as t gets large, y[t] → 0, no matter what R you go with.

What is it about the output immediately above that gave you these ideas?

What do you mean?

□G.5.a.ii)

Now you ask Brian whether he is interested in what happens when the resistance measurement R is very big? He replies," I never thought about that. What do you think?"

You say, "Let's see."

```
c = 0.8;
L = 5.0;
Clear[R];
Table[{R, Eigenvalues[A[L, R, c]]}, {R, 10, 1010, 50}]
```

```
{{10, {-0.0625 + 0.496078 I, -0.0625 - 0.496078 I}},
 {60, {-0.0104167 + 0.499891 I, -0.0104167 - 0.499891 I}},
 {110, {-0.00568182 + 0.499968 I, -0.00568182 - 0.499968 I}},
 {160, {-0.00390625 + 0.499985 I, -0.00390625 - 0.499985 I}},
 {210, {-0.00297619 + 0.499991 I, -0.00297619 - 0.499991 I}},
 {260, {-0.00240385 + 0.499994 I, -0.00240385 - 0.499994 I}},
 {310, {-0.00201613 + 0.499996 I, -0.00201613 - 0.499996 I}},
 {360, {-0.00173611 + 0.499997 I, -0.00173611 - 0.499997 I}},
 {410, {-0.00152439 + 0.499998 I, -0.00152439 - 0.499998 I}},
 {460, {-0.0013587 + 0.499998 I, -0.0013587 - 0.499998 I}},
 {510, {-0.00122549 + 0.499998 I, -0.00122549 - 0.499998 I}},
 {560, {-0.00111607 + 0.499999 I, -0.00111607 - 0.499999 I}},
 {610, {-0.00102459 + 0.499999 I, -0.00102459 - 0.499999 I}},
 {660, {-0.00094697 + 0.499999 I, -0.00094697 - 0.499999 I}},
 {710, {-0.000880282 + 0.499999 I, -0.000880282 - 0.499999 I}},
 {760, {-0.000822368 + 0.499999 I, -0.000822368 - 0.499999 I}},
 {810, {-0.000771605 + 0.499999 I, -0.000771605 - 0.499999 I}},
 {860, {-0.000726744 + 0.499999 I, -0.000726744 - 0.499999 I}},
 {910, {-0.000686813 + 0.5 I, -0.000686813 - 0.5 I}},
 {960, {-0.000651042 + 0.5 I, -0.000651042 - 0.5 I}},
 {1010, {-0.000618812 + 0.5 I, -0.000618812 - 0.5 I}}}
```

You say that when you go with very large R's, the plots of the corresponding current functions y[t] don't change much as you change R.
What is it about the output immediately above that gives you this idea?
What do you mean?

G.6) Sensitivity to errors in the starter data

□G.6.a.i) Suckers and propellers

When you come across this linear system

$$x'[t] = 1.6\,x[t] + 2.2\,y[t]$$
$$y'[t] = 0.9\,x[t] - 0.8\,y[t],$$

you immediately write down this matrix:

```
A = {{1.6, 2.2}, {0.9, -0.8}};
MatrixForm[A]
```

$$\begin{pmatrix} 1.6 & 2.2 \\ 0.9 & -0.8 \end{pmatrix}$$

You ask *Mathematica* for the eigenvectors and eigenvalues of A:

```
Clear[eigenvector];
{eigenvector[1], eigenvector[2]} = Eigenvectors[A]
```

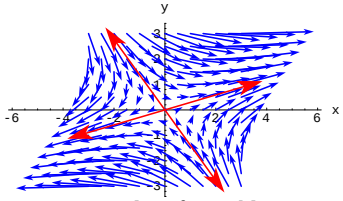
```
{{0.959098, 0.283075}, {-0.58509, 0.810968}}
```

```
Clear[eigenvalue];
{eigenvalue[1], eigenvalue[2]} = Eigenvalues[A]
```

```
{2.24932, -1.44932}
```

And you note that eigenvector[1] is a propeller, but eigenvector[2] is a sucker.
Now you look at the flow with the eigenvectors:

```
Clear[Field]
Field[x_, y_] = A . {x, y};
flowplot = Table[Arrow[Field[x, y], Tail → {x, y},
    VectorColor → Blue, ScaleFactor → 0.25, HeadSize → 0.4],
  {x, -3, 3, 6/12}, {y, -3, 3, 6/12}];
scaler = 4.0;
sizer = 1;
eigenplot = {Arrow[eigenvector[1], Tail → {0, 0},
    VectorColor → Red, ScaleFactor → scaler, HeadSize → sizer],
  Arrow[-eigenvector[1], Tail → {0, 0},
    VectorColor → Red, ScaleFactor → scaler, HeadSize → sizer],
  Arrow[eigenvector[2], Tail → {0, 0},
    VectorColor → Red, ScaleFactor → scaler, HeadSize → sizer],
  Arrow[-eigenvector[2], Tail → {0, 0},
    VectorColor → Red, ScaleFactor → scaler, HeadSize → sizer]};
Show[flowplot, eigenplot, Axes → True, AxesLabel → {"x", "y"}];
```



You can spot a lot of possible starter points {starterx, startery} at which corresponding solutions {x[t], y[t]} to the given linear system

$$x'[t] = 1.6\,x[t] + 2.2\,y[t]$$
$$y'[t] = 0.9\,x[t] - 0.8\,y[t],$$

with x[0] = starterx and y[0] = startery
will change tremendously when you change the starter data
{starterx, startery}
just a wee little bit. Folks say that these are the points {starterx, startery} at which the linear system is very sensitive to errors in the starter data.
Where are these points {starterx, startery}?

□G.6.a.ii)

Do you expect the same thing to happen any time you go with a linear system

$$\{x'[t], y'[t]\} = A\,.\,\{x[t], y[t]\}$$

with one sucker and one propeller?
Explain your reply.

□G.6.b.i) Two suckers

When you come across this linear system

$$x'[t] = -1.3\,x[t] + 0.6\,y[t]$$
$$y'[t] = 0.3\,x[t] - 0.6\,y[t],$$

you immediately write down this matrix:

```
A = {{-1.3, 0.6}, {0.3, -0.6}};
MatrixForm[A]
```

$$\begin{pmatrix} -1.3 & 0.6 \\ 0.3 & -0.6 \end{pmatrix}$$

And you proceed just as above, finding the eigenvectors and their eigenvalues:

```
Clear[eigenvector];
{eigenvector[1], eigenvector[2]} = Eigenvectors[A]
```

```
{{-0.948683, 0.316228}, {-0.5547, -0.83205}}
```

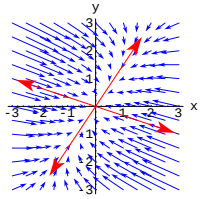
```
Clear[eigenvalue];
{eigenvalue[1], eigenvalue[2]} = Eigenvalues[A]
```

```
{-1.5, -0.4}
```

You note that both eigenvectors are suckers and you look at the flow with the eigenvectors:

```
Clear[Field]
Field[x_, y_] = A . {x, y};
```

```
flowplot = Table[Arrow[Field[x, y], Tail → {x, y},
    VectorColor → Blue, ScaleFactor → 0.25, HeadSize → 0.25],
  {x, -3, 3, 6/12}, {y, -3, 3, 6/12}];
scaler = 3.0;
sizer = 0.7;
eigenplot = {Arrow[eigenvector[1], Tail → {0, 0},
    VectorColor → Red, ScaleFactor → scaler, HeadSize → sizer],
  Arrow[-eigenvector[1], Tail → {0, 0},
    VectorColor → Red, ScaleFactor → scaler, HeadSize → sizer],
  Arrow[eigenvector[2], Tail → {0, 0},
    VectorColor → Red, ScaleFactor → scaler, HeadSize → sizer],
  Arrow[-eigenvector[2], Tail → {0, 0},
    VectorColor → Red, ScaleFactor → scaler, HeadSize → sizer]};
Show[flowplot, eigenplot, Axes → True, AxesLabel → {"x", "y"}];
```



Do you see any starter points {starterx, startery} at which solutions {x[t], y[t]} to the given linear system with x[0] = starterx and y[0] = startery
are very sensitive to small errors in the starter data?
If so, where are these points?

□G.6.b.ii)

Do you expect the same thing to happen any time you go with a linear system

$$\{x'[t], y'[t]\} = A\,.\,\{x[t], y[t]\}$$

with two suckers?
Explain your reply.

□G.6.c.i) Two propellers

When you come across this linear system

$$x'[t] = 0.7\,x[t] + 0.1\,y[t]$$
$$y'[t] = -0.1\,x[t] + 1.2\,y[t],$$

you immediately write down this matrix:

```
A = {{0.7, 0.1}, {-0.1, 1.2}};
MatrixForm[A]
```

$$\begin{pmatrix} 0.7 & 0.1 \\ -0.1 & 1.2 \end{pmatrix}$$

And you proceed just as above, finding the eigenvectors and their eigenvalues:

```
Clear[eigenvector];
{eigenvector[1], eigenvector[2]} = Eigenvectors[A]
```

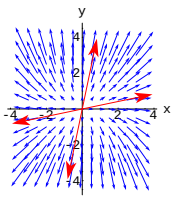
```
{{-0.20431, -0.978906}, {-0.978906, -0.20431}}
```

```
Clear[eigenvalue];
{eigenvalue[1], eigenvalue[2]} = Eigenvalues[A]
```

```
{1.17913, 0.720871}
```

You note that both eigenvectors are propellers and you look at the flow with the eigenvectors:

```
Clear[Field]
Field[x_, y_] = A . {x, y};
flowplot = Table[Arrow[Field[x, y], Tail → {x, y}, VectorColor → Blue,
    ScaleFactor → 0.4, HeadSize → 0.3], {x, -3, 3, 6/12}, {y, -3, 3, 6/12}];
scaler = 4.0;
sizer = 1;
eigenplot = {Arrow[eigenvector[1], Tail → {0, 0},
    VectorColor → Red, ScaleFactor → scaler, HeadSize → sizer],
  Arrow[-eigenvector[1], Tail → {0, 0},
    VectorColor → Red, ScaleFactor → scaler, HeadSize → sizer],
  Arrow[eigenvector[2], Tail → {0, 0},
    VectorColor → Red, ScaleFactor → scaler, HeadSize → sizer],
  Arrow[-eigenvector[2], Tail → {0, 0},
    VectorColor → Red, ScaleFactor → scaler, HeadSize → sizer]};
Show[flowplot, eigenplot, Axes → True, AxesLabel → {"x", "y"}];
```



Do you see any starter points {starterx, startery} at which corresponding solutions {x[t], y[t]} to the given linear system with x[0] = starterx and y[0] = startery
are very sensitive to small errors in the starter data?
If so, where are these points?

□**Tip:**

These points are right in front of your nose.

□G.6.c.ii)

Do you expect the same thing to happen any time you go with a linear system

$$\{x'[t], y'[t]\} = A\,.\,\{x[t], y[t]\}$$

with two propellers?
Explain your reply.

□G.6.d.i) Sucking whirlers

When you come across this linear system

$$x'[t] = 0.4\,x[t] + -2.2\,y[t]$$
$$y'[t] = 3.1\,x[t] - 1.8\,y[t],$$

you immediately write down this matrix:

```
A = {{0.4, -2.2}, {3.1, -1.8}};
MatrixForm[A]
```

$$\begin{pmatrix} 0.4 & -2.2 \\ 3.1 & -1.8 \end{pmatrix}$$

And you proceed just as above, finding the eigenvectors and their eigenvalues:

```
Clear[eigenvector];
{eigenvector[1], eigenvector[2]} = Chop[Eigenvectors[A]]
```

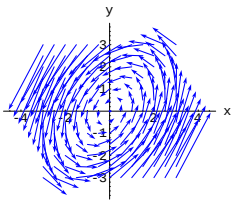
```
{{0.271378 + 0.584336 I, 0.764791}, {0.271378 - 0.584336 I, 0.764791}}
```

```
Clear[eigenvalue];
{eigenvalue[1], eigenvalue[2]} = Eigenvalues[A]
```

```
{-0.7 + 2.36854 I, -0.7 - 2.36854 I}
```

Uh-oh.
You note that −0.7 is negative and say there is a suck, and you note the 2.36854 I term and say there is a swirl. This tells you that the trajectories move on spirals headed toward {0, 0}.
And then you look at the flow:

```
Clear[Field]
Field[x_, y_] = A . {x, y};
flowplot = Table[Arrow[Field[x, y], Tail → {x, y}, VectorColor → Blue,
    ScaleFactor → 0.2, HeadSize → 0.3], {x, -3, 3, 6/12}, {y, -3, 3, 6/12}];
Show[flowplot, Axes → True, AxesLabel → {"x", "y"}];
```



Do you see any starter points {starterx, startery} at which solutions {x[t], y[t]} to the given linear system with x[0] = starterx and y[0] = startery
are very sensitive to small errors in the starter data?
If so, where are these points?

□G.6.d.ii)

Do you expect the same thing to happen any time you go with a linear system

$$\{x'[t], y'[t]\} = A\,.\,\{x[t], y[t]\}$$

and you find that the eigenvalues of A are

$$p + I\,q \text{ and } p - I\,q$$

with $p < 0$ and q not 0?
Explain your reply.

□G.6.d.iii) The damped oscillator

Here is the differential equation of the oscillator:

```
Clear[t, b, c, y]
oscillatordiffeq = y''[t] + b y'[t] + c y[t] == 0
```

```
c y[t] + b y'[t] + y''[t] == 0
```

To rework this second order differential equation into a system of two first order differential equations, you put

$$x[t] = y'[t]$$

and then replace y'[t] by x[t] and replace y"[t] by x'[t]:

```
Clear[x]
ColumnForm[Thread[{new = (y'[t] == x[t]),
oscillatordiffeq /. {y'[t] -> x[t], y''[t] -> x'[t]}}] ]
```

```
y'[t] == x[t]
b x[t] + c y[t] + x'[t] == 0
```

Clean this up:

```
Clear[m, n, x, y, t]
m[x_, y_] = -b x - c y;
n[x_, y_] = x;
linsystem = {x'[t], y'[t]} == {m[x[t], y[t]], n[x[t], y[t]]};
ColumnForm[Thread[linsystem]]
```

```
x'[t] == -b x[t] - c y[t]
y'[t] == x[t]
```

```
ColumnForm[Thread[linsystem]]
```

```
x'[t] == -b x[t] - c y[t]
y'[t] == x[t]
```

The matrix for this system is:

```
A = {{-b, -c}, {1, 0}};
MatrixForm[A]
```

$$\begin{pmatrix} -b & -c \\ 1 & 0 \end{pmatrix}$$

Its eigenvalues are:

```
Clear[eigenvalue];
{eigenvalue[1], eigenvalue[2]} = Eigenvalues[A]
```

$$\left\{\frac{1}{2}\left(-b - \sqrt{b^2 - 4c}\right), \frac{1}{2}\left(-b + \sqrt{b^2 - 4c}\right)\right\}$$

Why do you think folks say that the oscillator is damped if

$$b > 0 \text{ and } b^2 - 4c < 0?$$

When you have a damped oscillator

$y''[t] + b\,y'[t] + c\,y[t] = 0$
(with $b > 0$ and $b^2 - 4c < 0$), you get different solutions y[t] depending on what starter data {y[0], y'[0]} you go with.
Do you expect these solutions to be very sensitive to small errors in the starter data?
Explain yourself.

G.7) Using the cheat sheet chart to see why folks say that linear systems with two pure swirlers are structurally unstable but that other linear systems are not structurally unstable

□G.7.a.i)

Here's a linear system whose trajectories oscillate on ellipses:

```
Clear[x, y, t]
linearsystem =
 ({x'[t], y'[t]} == {-1.2 x[t] + 1.5 y[t], -2.3 x[t] + 1.2 y[t]});
ColumnForm[Thread[linearsystem]]
A = {{-1.2, 1.5}, {-2.3, 1.2}};
MatrixForm[A]
```

$$x'[t] == -1.2\,x[t] + 1.5\,y[t]$$
$$y'[t] == -2.3\,x[t] + 1.2\,y[t]$$
$$\begin{pmatrix} -1.2 & 1.5 \\ -2.3 & 1.2 \end{pmatrix}$$

Check the eigenvalues of the coefficient matrix A:

```
Clear[eigenvalue]
{eigenvalue[1], eigenvalue[2]} = Eigenvalues[A]
```

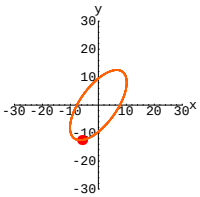
$$\{0. + 1.41774\,I,\ 0. - 1.41774\,I\}$$

The form is $p + I\,q$ with
$p = 0$ and $q \neq 0$.
This means that the eigenvectors are pure whirlers (no suck and no propel).
The result: The trajectories oscillate about {0, 0} on ellipses. And the solution plots are sine waves.

See some trajectories followed by the corresponding individual solution plots:

```
A = {{-1.2, 1.5}, {-2.3, 1.2}};
ranger = 30;
{xstarter, ystarter} = {Random[Real, {-ranger/2, ranger/2}],
  Random[Real, {-ranger/2, ranger/2}]};
Clear[x, y, t]
{x[t_], y[t_]} =
 Chop[ComplexExpand[Expand[MatrixExp[A t] . {xstarter, ystarter}]]]
endtime = 40;
trajectoryplot =
 ParametricPlot[{x[t], y[t]}, {t, 0, endtime}, PlotStyle →
   {CadmiumOrange, Thickness[0.01]}], DisplayFunction → Identity];
starterpoint = {xstarter, ystarter};
starterplot = Graphics[{Red, PointSize[0.06], Point[starterpoint]}];
Show[starterplot, trajectoryplot,
 PlotRange → {{-ranger, ranger}, {-ranger, ranger}}, Axes → True,
 AxesLabel → {"x", "y"}, DisplayFunction → $DisplayFunction];
```

$$\{-5.81826\,\text{Cos}[1.41774\,t] - 8.29199\,\text{Sin}[1.41774\,t],$$
$$-12.4919\,\text{Cos}[1.41774\,t] - 1.13439\,\text{Sin}[1.41774\,t]\}$$



Now take the coefficient matrix A and build in some random noise like this and compare:

```
h = 0.049;
radomnoise = {{Random[Real, {-h, h}], Random[Real, {-h, h}]},
  {Random[Real, {-h, h}], Random[Real, {-h, h}]}};
noisyA = A + radomnoise;
MatrixForm[noisyA]
MatrixForm[A]
```

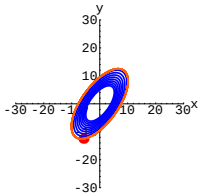
$$\begin{pmatrix} -1.21275 & 1.48826 \\ -2.32605 & 1.15308 \end{pmatrix}$$
$$\begin{pmatrix} -1.2 & 1.5 \\ -2.3 & 1.2 \end{pmatrix}$$

NoisyA is guaranteed to round off nicely to A.
But now look at a trajectory for the linear system
$\{x'[t], y'[t]\} = \text{noisyA} \cdot \{x[t], y[t]\}$
compared to the trajectory for the linear system
$\{x'[t], y'[t]\} = A \cdot \{x[t], y[t]\}$
as plotted above.

Both trajectories have the same starting point.

```
radomnoise = {{Random[Real, {-h, h}], Random[Real, {-h, h}]},
  {Random[Real, {-h, h}], Random[Real, {-h, h}]}};
noisyA = A + radomnoise;
Clear[noisyx, noisyy, t]
{noisyx[t_], noisyy[t_]} = Chop[
  ComplexExpand[Expand[MatrixExp[noisyA t] . {xstarter, ystarter}]]]
noisytrajectoryplot =
 ParametricPlot[{noisyx[t], noisyy[t]}, {t, 0, endtime},
  PlotStyle → {{Blue, Thickness[0.01]}}, DisplayFunction → Identity];
Show[starterplot, noisytrajectoryplot, trajectoryplot,
 PlotRange → {{-ranger, ranger}, {-ranger, ranger}}, Axes → True,
 AxesLabel → {"x", "y"}, DisplayFunction → $DisplayFunction];
```

$$\{-5.81826\,E^{-0.0180251\,t}\,\text{Cos}[1.4494\,t] - 8.36293\,E^{-0.0180251\,t}\,\text{Sin}[1.4494\,t],$$
$$-12.4919\,E^{-0.0180251\,t}\,\text{Cos}[1.4494\,t] - 1.05912\,E^{-0.0180251\,t}\,\text{Sin}[1.4494\,t]\}$$



The original trajectory is orange. The noisy trajectory is blue.

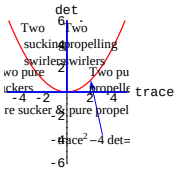
Rerun at least seven times.
Each time you rerun, you get a different noisyA which rounds off to A.

Describe what you see.
Comment on the statement:
→ Even though the eigenvectors of A are pure swirlers and noisyA rounds off to A,
it is quite unlikely that the eigenvectors of noisyA are pure swirlers.

□G.7.a.ii)

One of the goals of mathematics is to explain why things work out the way they do.
To this end, take a look at the cheat sheet chart from one of the Tutorials.

```
cutoff = Plot[trace^2/4, {trace, -5, 5},
  PlotStyle → {{Red, Thickness[0.01]}}, AxesLabel → {"trace", "det"},
  PlotRange → {-6, 6}, DisplayFunction → Identity];
cutofflabel = Graphics[
  Text[FontForm["trace^2-4 det==0", {"Times", 10}], {3.0, -4.0}]];
pointer = Arrow[{2, 1} - {3.0, -3.6}, Tail → {3.0, -3.6}];
twosuckercutoff =
 Graphics[{Blue, Thickness[0.01], Line[{{-5, 0}, {0, 0}}]}];
twosuckerlabel = Graphics[
  Text[FontForm[" Two pure \n suckers", {"Times", 10}], {-4, 1}]];
twopropcutoff =
 Graphics[{Blue, Thickness[0.01], Line[{{0, 0}, {5, 0}}]}];
twoproplabel = Graphics[
  Text[FontForm[" Two pure \n propellers", {"Times", 10}], {4, 1}]];
propswirlcutoff =
 Graphics[{Blue, Thickness[0.01], Line[{{0, 0}, {0, 6}}]}];
propswirllabel = Graphics[Text[FontForm[
    " Two\n propelling \n swirlers", {"Times", 10}], {2, 4}]];
pureswirlerlabel =
 Graphics[Text[FontForm[" p\n u\n r ", {"Times", 12}], {0, 7}]];
suckswirlcutoff =
 Graphics[{Blue, Thickness[0.01], Line[{{0, 0}, {0, 6}}]}];
suckswirllabel = Graphics[Text[
   FontForm["Two\n sucking\n swirlers", {"Times", 10}], {-2, 4}]];
negdetcutoff =
 Graphics[{Blue, Thickness[0.01], Line[{{-5, 0}, {5, 0}}]}];
negdetlabel = Graphics[Text[FontForm[
    "Pure sucker & pure propeller", {"Times", 10}], {0, -1.5}]];
pureswirlerlabel = Graphics[
  Text[FontForm["Two pure swirlers on the positive vertical axis",
    {"Times", 10}], {0, 6.5}]];
chart = Show[cutoff, cutofflabel, pointer, twosuckercutoff,
  twosuckerlabel, twopropcutoff, twoproplabel, propswirlcutoff,
  propswirllabel, suckswirlcutoff, suckswirllabel, negdetcutoff,
  negdetlabel, pureswirlerlabel, DisplayFunction → $DisplayFunction];
```



Given a coefficient matrix:

```
Clear[a, b, c, d]
A = {{a, b}, {c, d}};
MatrixForm[A]
```

$$\begin{pmatrix} a & b \\ c & d \end{pmatrix}$$

The part of the chart corresponding to two pure swirlers is

$$\text{trace} = a + d = 0$$

and

$$\det = a\,d - bc > 0.$$

Try to use this information to explain:
If you start with any coefficient matrix A whose eigenvectors are pure swirlers, and you make the slightest random error in entering A, then the resulting coefficient matrix is all but guaranteed to have eigenvectors that are sucking swirlers or propelling swirlers.

```
Some DiffEq pros say that
linear systems with two pure swirlers are "structurally instable."
```

□G.7.b) Most other linear systems are not structurally unstable

Given a coefficient matrix:

```
Clear[a, b, c, d]
A = {{a, b}, {c, d}};
MatrixForm[A]
```

$$\begin{pmatrix} a & b \\ c & d \end{pmatrix}$$

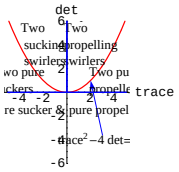
The part of the chart corresponding to two pure swirlers is

$$\text{trace} = a + d = 0$$

and

$$\det = a\,d - bc > 0.$$

```
Show[chart];
```

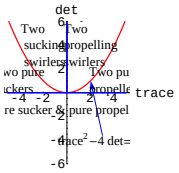


Use the chart to explain:
When you start with a coefficient matrix A and plot {trace, det} on the chart and land comfortably inside any of the generous regions (other than pure swirlers) labeled on the chart, then very slight random errors in entering the matrix are not likely to result in changing the character of the resulting linear system.

□G.7.c)

Take another look at the chart:

```
Show[chart];
```



If you are given a linear system whose coefficient matrix has

$$\text{trace} < 0 \text{ and } \det = 0,$$

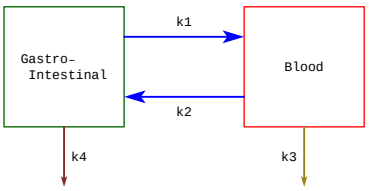
then use the chart to say what wildly different things can happen when very slight random errors are made when you enter the matrix.

G.8) Life Sciences:

Reservoir Models for drug metabolization

Lots of folks deal with the hot issue of models for drug metabolization. The simplest models for these systems are two compartment models such as this one:

```
Clear[box, color]
box[a_, color_] :=
 Graphics[{color, Line[{{a[[1]] - 1, a[[2]] - 1}, {a[[1]] - 1, a[[2]] + 1},
     {a[[1]] + 1, a[[2]] + 1}, {a[[1]] + 1, a[[2]] - 1}, {a[[1]] - 1, a[[2]] - 1}}]}]
generalmodel = Show[box[{0, 0}, GreenDark],
  box[{4, 0}, Red], Arrow[{2, 0}, Tail → {1, 1/2}],
  Arrow[{-2, 0}, Tail → {3, -1/2}], Arrow[{0, -1}, Tail → {4, -1},
   VectorColor → RGBColor[0.6, 0.5, 0.1]], Arrow[{0, -1},
   Tail → {0, -1}, VectorColor → Brown], AspectRatio → Automatic,
  Epilog → {Text["k1", {2, 3/4}], Text["k2", {2, -3/4}],
    Text["k3", {15/4, -3/2}], Text["k4", {1/4, -3/2}],
    Text["Gastro-\n Intestinal", {0, 0}], Text["Blood", {4, 0}]}];
```

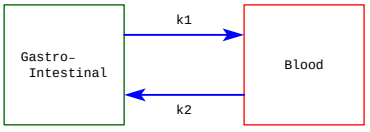



```
The arrows indicate that the amount leaving or entering each reservoir
                (depending on the direction of the arrow)
          is proportional to the amount in the reservoir.
```

Agree that
→ x[t] measures the amount of the substance in the gastro-intestinal tract
→ y[t] measures the amount of the substance in the blood t time units after the substance has been ingested into the gastro-intestinal track.

In the case that k3 = 0 and k4 = 0, you get this simplified model:

```
Show[box[{0, 0}, GreenDark],
  box[{4, 0}, Red], Arrow[{2, 0}, Tail → {1, 1/2}],
  Arrow[{-2, 0}, Tail → {3, -1/2}], AspectRatio → Automatic,
  Epilog → {Text["k1", {2, 3/4}], Text["k2", {2, -3/4}],
    Text["Gastro-\n Intestinal", {0, 0}], Text["Blood", {4, 0}]}];
```



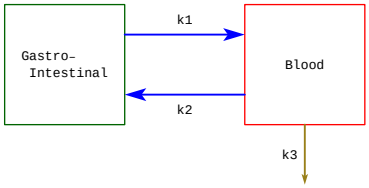
The corresponding linear system is

$$x'[t] = -k1\,x[t] + k2\,y[t]$$
$$y'[t] = k1\,x[t] - k2\,y[t].$$

```
This is the same as the chemical model that
you'll find in the problem on when an eigenvalue is 0.
```

This simplified model isn't realistic because it doesn't allow for elimination through urination or defecation. This following diagram sets up the model for elimination through urination.

```
urinemodel = Show[box[{0, 0}, GreenDark], box[{4, 0}, Red],
  Arrow[{2, 0}, Tail → {1, 1/2}, VectorColor → Blue],
  Arrow[{-2, 0}, Tail → {3, -1/2}, VectorColor → Blue], Arrow[{0, -1},
   Tail → {4, -1}, VectorColor → RGBColor[0.6, 0.5, 0.1]],
  AspectRatio → Automatic, Epilog →
   {Text["k1", {2, 3/4}], Text["k2", {2, -3/4}], Text["k3", {15/4, -3/2}],
    Text["Gastro-\n Intestinal", {0, 0}], Text["Blood", {4, 0}]}];
```



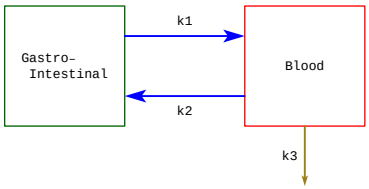
□G.8.a.i)

Write down the linear system resulting from this diagram.

□G.8.a.ii)

Stay with the model:

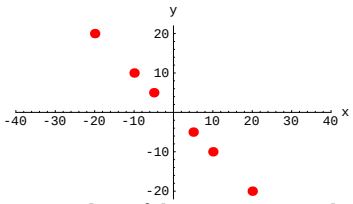
```
Show[urinemodel, AspectRatio → Automatic];
```



Go with specific the specific numbers
k1 = 0.5, k2 = 0.5, and k3 = 0.6

Find formulas for x[t] and y[t] given starting data
x[0] = 5 (milligrams) and y[0] = 0.

Plot x[t] and y[t] and give the limiting values of x[t] and y[t] as t goes to infinity.



Here are plots of the trajectories in this linear system starting at the plotted points for the first 3 time units shown with scaled versions of the eigenvectors of the coefficient matrix:

□G.8.a.iii)

For the same model, change the constants to
k1 = 0.1, k2 = 0.2, and k3 = 0.03.
Find formulas for x[t] and y[t] given starting data
x[0] = 5 (milligrams) and y[0] = 0.
Plot and discuss what you see.

□G.8.a.iv)

For the same model, play with positive constants k1, k2, and k3 of your own choice. Solve, plot and discuss what you see.

G.9) Life Sciences:

Propellers, suckers, population models and the ultimate sex ratio

□G.9.a)

Here's a linear system of differential equations:

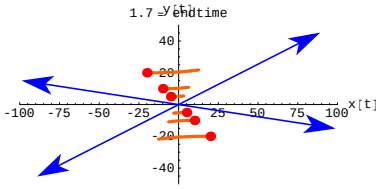
```
Clear[x, y, t]
linearsystem =
({x'[t], y'[t]} == {-0.26 x[t] + 0.90 y[t], 0.07 x[t] + 0.06 y[t]});
ColumnForm[Thread[linearsystem]]
```

$$x'[t] == -0.26\,x[t] + 0.9\,y[t]$$
$$y'[t] == 0.07\,x[t] + 0.06\,y[t]$$

Here are six starter points:

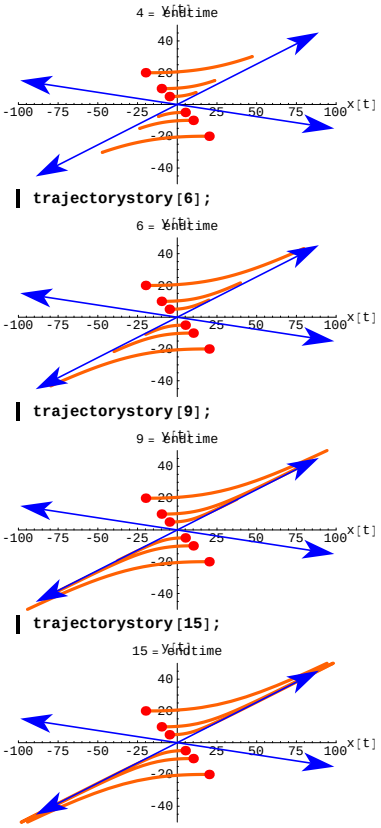
```
Clear[starter]
starter[1] = {20, -20};
starter[2] = {10, -10};
starter[3] = {5, -5};
starter[4] = {-5, 5};
starter[5] = {-10, 10};
starter[6] = {-20, 20};
starterplot = Show[Table[
   Graphics[{Red, PointSize[0.03], Point[starter[j]]}], {j, 1, 6}],
  PlotRange → {{-40, 40}, {-22, 22}}, Axes → True,
  AxesLabel → {"x", "y"}];
```

```
A = {{-0.26, 0.9}, {0.07, 0.06}};
Clear[trajectory];
trajectory[k_, t_] :=
 Chop[ComplexExpand[Expand[MatrixExp[A t] . starter[k]]]];
Clear[trajectoryplot, trajectoryplots, endtime]
trajectoryplot[k_, endtime_] :=
 ParametricPlot[Evaluate[trajectory[k, t]], {t, 0, endtime},
  PlotRange → All, PlotStyle → {{Thickness[0.01], CadmiumOrange}},
  AxesLabel → {"x[t]", "y[t]"},
  PlotLabel → endtime "= endtime", DisplayFunction → Identity];
trajectoryplots[endtime_] :=
 Table[trajectoryplot[k, endtime], {k, 1, 6}];
Clear[eigenvalue]
{eigenvalue[1], eigenvalue[2]} = Eigenvalues[A];
Clear[eigenvector]
{eigenvector[1], eigenvector[2]} = Eigenvectors[A];
scaler = 100;
sizer = 20;
eigenplot = {Arrow[eigenvector[1], Tail → {0, 0},
   VectorColor → Blue, ScaleFactor → scaler, HeadSize → sizer],
  Arrow[-eigenvector[1], Tail → {0, 0},
   VectorColor → Blue, ScaleFactor → scaler, HeadSize → sizer],
  Arrow[eigenvector[2], Tail → {0, 0},
   VectorColor → Blue, ScaleFactor → scaler, HeadSize → sizer],
  Arrow[-eigenvector[2], Tail → {0, 0},
   VectorColor → Blue, ScaleFactor → scaler, HeadSize → sizer]};
trajectorystory[endtime_] :=
 Show[trajectoryplots[endtime], starterplot, eigenplot,
  PlotRange → {{-scaler, scaler}, {-scaler/2, scaler/2}},
  AspectRatio → 1/2, DisplayFunction → $DisplayFunction];
trajectorystory[1.7];
```



See more of the trajectories:

```
trajectorystory[4];
```



Grab and animate the plots slowly.

The linear system is:

```
Clear[x, y, t]
ColumnForm[Thread[linearsystem]]
```

$$x'[t] == -0.26\,x[t] + 0.9\,y[t]$$
$$y'[t] == 0.07\,x[t] + 0.06\,y[t]$$

The coefficient matrix A for this system is:

```
A = {{-0.26, 0.9}, {0.07, 0.06}};
MatrixForm[A]
```

$$\begin{pmatrix} -0.26 & 0.9 \\ 0.07 & 0.06 \end{pmatrix}$$

The eigenvalues of A:

```
Clear[eigenvalue]
{eigenvalue[1], eigenvalue[2]} = Eigenvalues[A]
```

$$\{-0.397658,\ 0.197658\}$$

The eigenvectors of A:

```
Clear[eigenvector]
{eigenvector[1], eigenvector[2]} = Eigenvectors[A]
```

$$\{\{-0.988504,\ 0.151194\},\ \{-0.891373,\ -0.453271\}\}$$

A propeller and a sucker. You have enough information to be able to predict the limiting ratio $\frac{y[t]}{x[t]}$ ast gets large for any trajectory that does not start on the line through {0, 0} determined by the sucking eigenvector.
Do it.

□G.9.b)

Imagine that you are given a linear system with one propeller and one sucker (no swirl). Explain how you can use eigenvalue and eigenvector information to calculate the ultimate limiting ratio
$\frac{y[t]}{x[t]}$ as $t \to \infty$.

□G.9.c.i) Linear population models

This problem was adapted from original research articles by
Leo Goodman (Biometrics 9,1953) and
David G. Kendall (Journal of the Royal Statistical Society B.11, 1949).
Both articles appear in the book Mathematical Demography
(Springer-Verlag Biomathematics Series Volume 6) edited by top-notch
population biologists David Smith and Nathan Keyfitz
(Springer-Verlag, New York, 1977).

Agree that female[t] measures the population of females and that male[t] measures the population of males t years after a given reference year. One linear model for this setup is

$$\text{male}'[t] = -\text{dm male}[t] + \text{bm}(\text{male}[t] + \text{female}[t])$$
$$\text{female}'[t] = -\text{df female}[t] + \text{bf}(\text{male}[t] + \text{female}[t]).$$

In this model, bm, bf, dm, and df are positive numbers.

bm measures the instantaneous percentage birth rate of males,
bf measures the instantaneous percentage birth rate of females,
dm measures the instantaneous percentage death rate of males;
df measures the instantaneous percentage death rate of females.

This model assumes that the instantaneous percentage growth rates of both the male and female populations are proportional to the whole population (male[t] + female[t]).

The matrix for this linear population model is:

```
Clear[dm, bf, dm, df]
A = {{-dm + bm, bm}, {bf, bf - df}};
MatrixForm[A]
```

$$\begin{pmatrix} \text{bm} - \text{dm} & \text{bm} \\ \text{bf} & \text{bf} - \text{df} \end{pmatrix}$$

Check:

```
Clear[male, female, t]
popmodel = {male'[t], female'[t]} == A.{male[t], female[t]};
ColumnForm[Thread[popmodel]]
```

$$\text{male}'[t] == \text{bm female}[t] + (\text{bm} - \text{dm})\,\text{male}[t]$$
$$\text{female}'[t] == (\text{bf} - \text{df})\,\text{female}[t] + \text{bf male}[t]$$

This checks.
When you go with bm = bf and dm = df, so that males and females are equally likely to be born and equally likely to die, then you get

```
ColumnForm[Thread[popmodel /. {bm -> bf, dm -> df}]]
```

$$\text{male}'[t] == \text{bf female}[t] + (\text{bf} - \text{df})\,\text{male}[t]$$
$$\text{female}'[t] == (\text{bf} - \text{df})\,\text{female}[t] + \text{bf male}[t]$$

And:

```
Eigenvalues[A /. {bm → bf, dm → df}]
```

$$\{2\,\text{bf} - \text{df},\ -\text{df}\}$$

```
Eigenvectors[A /. {bm → bf, dm → df}]
```

$$\{\{1, 1\},\ \{-1, 1\}\}$$

When you go with
bm = bf and dm = df,
and you go with

$2\,\text{bf} > \text{df}$,
which eigenvector is the propellor?

□Tip:

Remember that bm, bf, dm, and df are all positive.

□G.9.c.ii)

Explain:
When you go with
bm = bf and dm = df,
and you go with
$2\,\text{bf} > \text{df}$,
then this model predicts that both sexes prosper, growing approximately exponentially, with ultimate sex ratio
$\frac{\text{male}[t]}{\text{female}[t]} \to 1$ as $t \to \infty$.
So that any huge initial excess of males or females disappears over the course of time regardless of starting data.

□G.9.c.iii)

Explain:
When you go with
bm = bf and dm = df,
and you go with
$2\,\text{bf} < \text{df}$,
then this model predicts that both sexes are headed towards extinction in the sense that
$\text{male}[t] \to 0$ and $\text{female}[t] \to 0$ as $t \to \infty$.

□G.9.d.i) When females are population dominant

From an article by evolutionary biologist Jared Diamond of the UCLA Medical School writing in the magazine Natural History (September, 1994):

"Herds of wild horses consist of one stallion and
up to a half dozen mares. . ."

In certain populations, the females are population dominant in the sense that the growth of the populations of both sexes depends only on the number of females. Examples are the wild horse population and

pure-bred dog populations in which very few males are allowed to breed (the top-winning males, sometimes called "matador studs") while females are bred almost indiscriminantly.
Discuss why an appropriate linear population model for a female population dominant society is

male'[t] = − dm male[t] + bm female[t]
female'[t] = − df female[t] + bf female[t].

□G.9.d.ii)

Go with the linear population model for a female population dominant society:

male'[t] = − dm male[t] + bm female[t]
female'[t] = − df female[t] + bf female[t].

Set up the matrix:

```
Clear[dm, bf, dm, df]
A = {{-dm, bm}, {0, bf - df}};
MatrixForm[A]
```

$$\begin{pmatrix} -dm & bm \\ 0 & bf - df \end{pmatrix}$$

Check:

```
Clear[male, female, t]
linpopmodel = {male'[t], female'[t]} == A.{male[t], female[t]};
ColumnForm[Thread[linpopmodel]]
```

male′[t] == bm female[t] − dm male[t]
female′[t] == (bf − df) female[t]

Now do as you are compelled:

```
Clear[eigenvalue]
{eigenvalue[1], eigenvalue[2]} = Eigenvalues[A]
```

{bf − df, −dm}

```
Clear[eigenvector]
{eigenvector[1], eigenvector[2]} = Eigenvectors[A]
```

$$\left\{\left\{\frac{bm}{bf - df + dm}, 1\right\}, \{1, 0\}\right\}$$

If bf > df, then which eigenvector is the propeller?
How do you know that if bf > df, then this model predicts that the both male and female populations eventually grow approximately exponentially and the ultimate sex ratio

$$\frac{male[t]}{female[t]} \to \frac{bm}{(dm + bf - df)} \text{ as } t \to \infty \text{ ?}$$

Does this allow for the possibility that the males population begins to fade out because very few of them are needed to sustain the overall populations?

□Tip:

Remember that bm, bf, dm,and df are all positive.

□G.9.d.iii)

Imagine that you are a big shot in a facist female population dominant society and imagine that you can control the size of dm (percentage death rate of males) chemically or otherwise. How would you set dm in terms of bf, df and bm to guarantee that the ultimate sex ratio

$$\frac{male[t]}{female[t]} \to 1 \text{ as } t \to \infty?$$

□G.9.d.iv)

Look again at the linear population model for a female population dominant society:

male'[t] = − dm male[t] + bm female[t]
female'[t] = − df female[t] + bf female[t].

Set up the matrix:

```
Clear[dm, bf, dm, df]
A = {{-dm, bm}, {0, bf - df}};
MatrixForm[A]
```

$$\begin{pmatrix} -dm & bm \\ 0 & bf - df \end{pmatrix}$$

Check:

```
Clear[male, female, t]
linpopmodel = {male'[t], female'[t]} == A.{male[t], female[t]};
ColumnForm[Thread[linpopmodel]]
```

male′[t] == bm female[t] − dm male[t]
female′[t] == (bf − df) female[t]

Now do as you are compelled:

```
Clear[eigenvalue]
{eigenvalue[1], eigenvalue[2]} = Eigenvalues[A]
```

{bf − df, −dm}

What does the model predict if bf < df?

G.10) Chemistry:

When an eigenvalue is 0: It happens in chemistry!

□G.10.a.i)

When you come across this linear system

x'[t] = −0.75 x[t] + 0.25 y[t]
y'[t] = 0.75 x[t] − 0.25 y[t],

you immediately write down this matrix:

```
A = {{-0.75, 0.25}, {0.75, -0.25}};
MatrixForm[A]
```

$$\begin{pmatrix} -0.75 & 0.25 \\ 0.75 & -0.25 \end{pmatrix}$$

You check it out:

```
Clear[x, y, t]
linearsystem = ({x'[t], y'[t]} == A.{x[t], y[t]});
ColumnForm[Thread[linearsystem]]
```

x′[t] == −0.75 x[t] + 0.25 y[t]
y′[t] == 0.75 x[t] − 0.25 y[t]

You ask *Mathematica* for the eigenvectors and eigenvalues of A:

```
Clear[eigenvector];
{eigenvector[1], eigenvector[2]} = Eigenvectors[A]
```

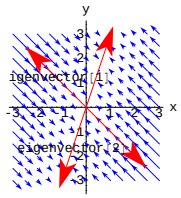
{{−0.707107, 0.707107}, {−0.316228, −0.948683}}

```
Clear[eigenvalue];
{eigenvalue[1], eigenvalue[2]} = Chop[Eigenvalues[A]]
```

{−1., 0}

And you note that eigenvector[1] is a sucker, but eigenvector[2] is neither a sucker nor a propeller. You might not have seen anything like this before, so you look at the flow with the eigenvectors:

```
Clear[Field]
Field[x_, y_] = A.{x, y};
flowplot = Table[Arrow[Field[x, y], Tail → {x, y},
    VectorColor → Blue, ScaleFactor → 0.25, HeadSize → 0.3],
  {x, -3, 3, 6/12}, {y, -3, 3, 6/12}];
scaler = 3.5;
sizer = 1.2;
eigenplot = {Arrow[eigenvector[1], Tail → {0, 0},
   VectorColor → Red, ScaleFactor → scaler, HeadSize → sizer],
  Arrow[-eigenvector[1], Tail → {0, 0},
   VectorColor → Red, ScaleFactor → scaler, HeadSize → sizer],
  Arrow[eigenvector[2], Tail → {0, 0},
   VectorColor → Red, ScaleFactor → scaler, HeadSize → sizer],
  Arrow[-eigenvector[2], Tail → {0, 0},
   VectorColor → Red, ScaleFactor → scaler, HeadSize → sizer]};
labels = {Graphics[Text["eigenvector[1]", 0.5 scaler eigenvector[1]]],
  Graphics[Text["eigenvector[2]", 0.5 scaler eigenvector[2]]]};
Show[flowplot, eigenplot, labels, Axes → True, AxesLabel → {"x", "y"}];
```



Describe the trajectories.
What is the relation of eigenvector[1] to the trajectories?
What is the relation of eigenvector[2] to the trajectories?
What happens when you go with starter data
{starterx, startery} = eigenvector[2]?

□G.10.a.ii) The ultimate ratio $\frac{y[t]}{x[t]}$ for large t

Continue with the same linear system as in part i) above.
When you think about it, you'll realize that no matter what trajectory {x[t], y[t]} you go with, you'll find that the trajectory stalls on a certain line through {0, 0}.
How does this fact allow you to look at the eigenvectors:

```
eigenvector[1]
```

{−0.707107, 0.707107}

```
eigenvector[2]
```

{−0.316228, −0.948683}

And then immediately type the ultimate limiting ratio

$$\frac{y[t]}{x[t]} \text{ as } t \to \infty \text{ ?}$$

G.10.b.i) Chemistry

DiffEq&*Mathematica* thanks chemist Justin Gallivan of California Institute of Technology for suggesting this problem.

This problem is very similar to the problem in part a) above.

Given quantities of two reactants X and Y are mixed together.
Agree that x[t] measures amount of X at time t units after the reaction begins. And agree that y[t] measures the amount of Y at time t units after the reaction begins.
In the reaction under study here,

$$x'[t] = -k1\,x[t] + k2\,y[t]$$
$$y'[t] = k1\,x[t] - k2\,y[t].$$

where k1 and k2 are positive reaction rates.

How does this linear system signal that at all times t,

$$x[t] + y[t] = x[0] + y[0]$$

where x[0] is the amount of X mixed in and y[0] is the amount of Y mixed in at the beginning?

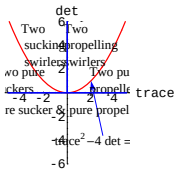
□ **Tip:**

Look at the linear system

$$x'[t] = -k1\,x[t] + k2\,y[t]$$
$$y'[t] = k1\,x[t] - k2\,y[t]$$

and see what happens when you add x'[t] and y'[t].

```
cutoff = Plot[trace^2/4, {trace, -5, 5},
  PlotStyle → {{Red, Thickness[0.01]}}, AxesLabel → {"trace", "det"},
  PlotRange → {-6, 6}, DisplayFunction → Identity];
cutofflabel = Graphics[
  Text[FontForm["trace^2-4 det == 0", {"Times", 10}], {3.0, -4.0}]];
pointer = Arrow[{2, 1} - {3.0, -3.6}, Tail → {3.0, -3.6}];
twosuckercutoff =
 Graphics[{Blue, Thickness[0.01], Line[{{-5, 0}, {0, 0}}]}];
twosuckerlabel = Graphics[
  Text[FontForm[" Two pure \n suckers", {"Times", 10}], {-4, 1}]];
twopropcutoff =
 Graphics[{Blue, Thickness[0.01], Line[{{0, 0}, {5, 0}}]}];
twoproplabel = Graphics[
  Text[FontForm[" Two pure \n propellers", {"Times", 10}], {4, 1}]];
propswirlcutoff =
 Graphics[{Blue, Thickness[0.01], Line[{{0, 0}, {0, 6}}]}];
propswirllabel = Graphics[Text[FontForm[
    " Two\n propelling \n swirlers", {"Times", 10}], {2, 4}]];
pureswirlerlabel =
 Graphics[Text[FontForm[" p\n u\n r ", {"Times", 12}], {0, 7}]];
suckswirlcutoff =
 Graphics[{Blue, Thickness[0.01], Line[{{0, 0}, {0, 6}}]}];
suckswirllabel = Graphics[Text[
   FontForm["Two\n sucking\n swirlers", {"Times", 10}], {-2, 4}]];
negdetcutoff =
 Graphics[{Blue, Thickness[0.01], Line[{{-5, 0}, {5, 0}}]}];
negdetlabel = Graphics[Text[FontForm[
    "Pure sucker & pure propeller", {"Times", 10}], {0, -1.5}]];
pureswirlerlabel = Graphics[
  Text[FontForm["Two pure swirlers on the positive vertical axis",
    {"Times", 10}], {0, 6.5}]];
chart = Show[cutoff, cutofflabel, pointer, twosuckercutoff,
  twosuckerlabel, twopropcutoff, twoproplabel, propswirlcutoff,
  propswirllabel, suckswirlcutoff, suckswirllabel, negdetcutoff,
  negdetlabel, pureswirlerlabel, DisplayFunction → $DisplayFunction];
```



□G.10.b.ii) The ultimate ratio of Y to X

Given quantities of two reactants X and Y are mixed together. Agree that x[t] measures amount of X at time t units after the reaction begins. And agree that y[t]measures the amount of Y at time t units after the reaction begins. In the reaction under study here,

$$x'[t] = -k1\,x[t] + k2\,y[t]$$
$$y'[t] = k1\,x[t] - k2\,y[t].$$

where k1 and k2 are positive reaction rates.

This linear system reflects the fact that X reacts with Y to produce more Y and Y reacts with X to produce more X. Both reactions are reversible.
Look at the eigenvalues and eigenvectors of the cleared cofficient matrix:

```
Clear[k1, k2]
A = {{-k1, k2}, {k1, -k2}};
Eigenvalues[A]
```

$\{0, -k1 - k2\}$

```
Eigenvectors[A]
```

$\{\{\frac{k2}{k1}, 1\}, \{-1, 1\}\}$

The question here is:
What is the ultimate ratio Y to X as t goes to infinity.
Does it depend on the starting values x[0] and y[0]?

G.11) Electrical Engineering:

Using the cheat sheet chart to help to analyze electrical circuits

□G.11.a.i) Using the chart to help to analyze an electrical circuit

You don't have to understand the electrical jargon to be able to handle this problem

Here's that cheat sheet chart from the Tutorials:

In a simple parallel electrical circuit, the main measurements are

$x[t]$ = voltage drop across the capacitor and
$y[t]$ = current through the inductor

These measurements are related through the linear system

$$x'[t] = \frac{y[t]}{L}$$
$$y'[t] = -\frac{x[t]}{cap} - \frac{y[t]}{cap\ res}$$

where L, cap and res are given postive numbers with

L = inductance
cap = capacitance, and
res = resistance.

```
Clear[x, y, t, L, res, cap]
linearsystem =
  ({x'[t], y'[t]} == {y[t] / L, -x[t] / cap - y[t] / (res cap)});
ColumnForm[Thread[linearsystem]]
```

$x'[t] == \frac{y[t]}{L}$

$y'[t] == -\frac{x[t]}{cap} - \frac{y[t]}{cap\,res}$

The right matrix for this linear system is:

```
Clear[a, b, c, d]
{{a, b}, {c, d}} = {{0, 1/L}, {-1/cap, -1/(cap res)}};
A = {{a, b}, {c, d}};
MatrixForm[A]
```

$$\begin{pmatrix} 0 & \frac{1}{L} \\ -\frac{1}{cap} & -\frac{1}{cap\,res} \end{pmatrix}$$

Calculate the trace of A and the determinant of A

```
trace = a + d
```

$-\frac{1}{cap\,res}$

```
determinant = a d - b c
```

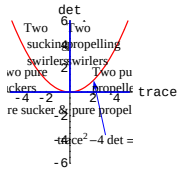
$\frac{1}{cap\,L}$

Because L, cap and res are all positive, this tells you that

trace < 0 and determinant > 0.

Look at the cheat sheet.

```
Show[chart];
```



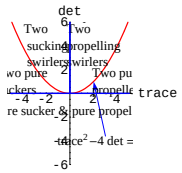
Use it to predict the limiting values as t gets large of
x[t] = voltage drop across the capacitor and
y[t] = current through the inductance
regardless of starter data on {x[0], y[0]}.

□G.11.b)

This is a continuation of part i).

Look at the chart:

```
Show[chart];
```



Take another look at the trace and the determinant calculated in part i):

```
trace = a + d
```

$$-\frac{1}{\text{cap res}}$$

```
determinant = Det[A]
```

$$\frac{1}{\text{cap L}}$$

When you change the resistance, you change the trace but not the determinant.
When you go with fixed L and cap numbers, then in what way are plots of the currentnt y[t] for very big (positive) resistance numbers fundamentally different from plots of the current y[t] for very small (positive) resistance numbers?

What would you do to the resistance number do to try to make the plot of y[t] oscillate above and below 0?
What would you do to the resistance number do to try to make the plot of y[t] decay exponentially to 0?

G.12) 3D and 4D linear systems

Here's a sample 3 D matrix A:

```
A =
 {{-0.18, -0.27, -0.23}, {-0.27, 0.36, -0.32}, {-0.23, -0.32, 0.13}};
MatrixForm[A]
```

$$\begin{pmatrix} -0.18 & -0.27 & -0.23 \\ -0.27 & 0.36 & -0.32 \\ -0.23 & -0.32 & 0.13 \end{pmatrix}$$

And the 3 D linear system based on it:

```
Clear[x, y, z, t]
linearsystem =
({x'[t], y'[t], z'[t]} == A . {x[t], y[t], z[t]});
ColumnForm[Thread[linearsystem]]
```

$$x'[t] == -0.18\,x[t] - 0.27\,y[t] - 0.23\,z[t]$$
$$y'[t] == -0.27\,x[t] + 0.36\,y[t] - 0.32\,z[t]$$
$$z'[t] == -0.23\,x[t] - 0.32\,y[t] + 0.13\,z[t]$$

You can calculate the eigenvectors of the 3 D coefficient matrix A:

```
Clear[eigenvector, eigenvalue];
{eigenvector[1], eigenvector[2], eigenvector[3]} =
 Chop[Eigenvectors[A]]
```

{{0.143821, -0.848382, 0.509475}, {0.751258, 0.428709, 0.501816},
 {0.644148, -0.310576, -0.699011}}

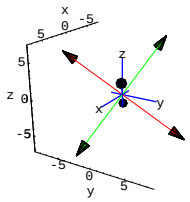
And the eigenvalues:

```
Clear[eigenvector, eigenvalue];
{eigenvalue[1], eigenvalue[2], eigenvalue[3]} = Chop[Eigenvalues[A]]
```

{0.59794, -0.487709, 0.19977}

Eigenvector[1] and eigenvector[3] are pure propellers with eigenvector[1] dominant.
Eigenvector[2] is a pure sucker.
See a scaled plot:

```
scaler = 10;
sizer = 2.0;
Clear[eigenvector];
{eigenvector[1],
  eigenvector[2], eigenvector[3]} = Chop[Eigenvectors[A]];
threeDeigenplot = {Arrow[eigenvector[1], Tail → {0, 0, 0},
    VectorColor → Red, ScaleFactor → scaler, HeadSize → sizer],
  Arrow[-eigenvector[1], Tail → {0, 0, 0},
    VectorColor → Red, ScaleFactor → scaler, HeadSize → sizer],
  Arrow[eigenvector[2], Tail → {0, 0, 0},
    VectorColor → Blue, ScaleFactor → scaler, HeadSize → sizer],
  Arrow[-eigenvector[2], Tail → {0, 0, 0},
    VectorColor → Blue, ScaleFactor → scaler, HeadSize → sizer],
  Arrow[eigenvector[3], Tail → {0, 0, 0},
    VectorColor → Green, ScaleFactor → scaler, HeadSize → sizer],
  Arrow[-eigenvector[3], Tail → {0, 0, 0},
    VectorColor → Green, ScaleFactor → scaler, HeadSize → sizer]};
Show[threeDeigenplot, Axes3D[5], Axes → True,
 AxesLabel → {"x", "y", "z"}, ViewPoint → CMView, Boxed → False];
```



The dominant propeller is plotted in red.
The weak propeller is plotted in green.
The pure sucker is plotted in blue.

You can adapt the 2 D code to 3 D and get formulas for the solution of this 3 D system starting at

$$\{x[0],\ y[0],\ z[0]\} = \{6.1, -10.2, -14.9\}$$

by going with three (instead of two) straight line solutions on the eigenvectors and adapting accordingly:

```
{xstarter, ystarter, zstarter} = {6.1, -10.2, -14.9};
Clear[eigenvector, eigenvalue];
{eigenvalue[1], eigenvalue[2], eigenvalue[3]} = Chop[Eigenvalues[A]];
{eigenvector[1],
  eigenvector[2], eigenvector[3]} = Chop[Eigenvectors[A]];
Clear[x, y, z, x1, y1, z1, x2, y2, z2, x3, y3, z3, t, C1, C2, C3]
{x1[t_], y1[t_], z1[t_]} = Chop[eigenvector[1] E^(eigenvalue[1] t)];
{x2[t_], y2[t_], z2[t_]} = Chop[eigenvector[2] E^(eigenvalue[2] t)];
{x3[t_], y3[t_], z3[t_]} = Chop[eigenvector[3] E^(eigenvalue[3] t)];
{x[t_], y[t_], z[t_]} = C1 {x1[t], y1[t], z1[t]} +
  C2 {x2[t], y2[t], z2[t]} + C3 {x3[t], y3[t], z3[t]};
starter = {xstarter, ystarter, zstarter};
starterequation = {x[0], y[0], z[0]} == starter;
Csols = Solve[starterequation];
{x[t_], y[t_], z[t_]} =
 Chop[ComplexExpand[{x[t], y[t], z[t]} /. Csols[[1]]]]
```

$$\{-5.45956\,E^{-0.487709\,t} + 11.2806\,E^{0.19977\,t} + 0.278958\,E^{0.59794\,t},$$
$$-3.11552\,E^{-0.487709\,t} - 5.43893\,E^{0.19977\,t} - 1.64554\,E^{0.59794\,t},$$
$$-3.64681\,E^{-0.487709\,t} - 12.2414\,E^{0.19977\,t} + 0.98819\,E^{0.59794\,t}\}$$

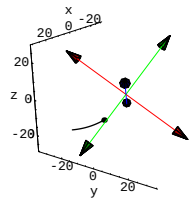
Compare with the eigenvalues of A:

```
Eigenvalues[A]
```

{0.59794, -0.487709, 0.19977}

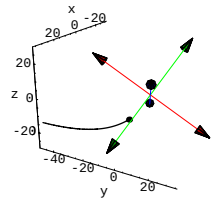
Just the way you expect.
See the trajectory along with the eigenvectors:

```
Clear[trajectoryplot, trajectorystory, endtime]
starterplot = Graphics3D[{PointSize[0.03], Point[starter]}];
scaler = 40;
sizer = 8;
threeDeigenplot = {Arrow[eigenvector[1], Tail → {0, 0, 0},
    VectorColor → Red, ScaleFactor → scaler, HeadSize → sizer],
  Arrow[-eigenvector[1], Tail → {0, 0, 0},
    VectorColor → Red, ScaleFactor → scaler, HeadSize → sizer],
  Arrow[eigenvector[2], Tail → {0, 0, 0},
    VectorColor → Blue, ScaleFactor → scaler, HeadSize → sizer],
  Arrow[-eigenvector[2], Tail → {0, 0, 0},
    VectorColor → Blue, ScaleFactor → scaler, HeadSize → sizer],
  Arrow[eigenvector[3], Tail → {0, 0, 0},
    VectorColor → Green, ScaleFactor → scaler, HeadSize → sizer],
  Arrow[-eigenvector[3], Tail → {0, 0, 0},
    VectorColor → Green, ScaleFactor → scaler, HeadSize → sizer]};
trajectoryplot[endtime_] :=
 ParametricPlot3D[Evaluate[{x[t], y[t], z[t]}],
  {t, 0, endtime}, DisplayFunction → Identity];
trajectorystory[endtime_] :=
 Show[trajectoryplot[endtime], starterplot, threeDeigenplot,
  Axes → True, AxesLabel → {"x", "y", "z"}, ViewPoint → CMView,
  Boxed → False, DisplayFunction → $DisplayFunction];
trajectorystory[3];
```



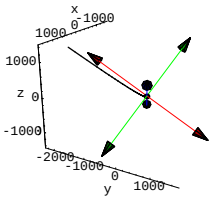
Influenced by that weak propeller (green).
See more:

```
trajectorystory[5];
```



Now the dominant propeller's influence is taking over.
See more:

```
scaler = 2000;
sizer = 400;
threeDeigenplot =
 {Arrow[eigenvector[1], Tail -> {0, 0, 0},
    VectorColor -> Red, ScaleFactor -> scaler, HeadSize -> sizer],
  Arrow[-eigenvector[1], Tail -> {0, 0, 0},
    VectorColor -> Red, ScaleFactor -> scaler, HeadSize -> sizer],
  Arrow[eigenvector[2], Tail -> {0, 0, 0},
    VectorColor -> Blue, ScaleFactor -> scaler, HeadSize -> sizer],
  Arrow[-eigenvector[2], Tail -> {0, 0, 0},
    VectorColor -> Blue, ScaleFactor -> scaler, HeadSize -> sizer],
  Arrow[eigenvector[3], Tail -> {0, 0, 0},
    VectorColor -> Green, ScaleFactor -> scaler, HeadSize -> sizer],
  Arrow[-eigenvector[3], Tail -> {0, 0, 0},
    VectorColor -> Green, ScaleFactor -> scaler, HeadSize -> sizer]};;
trajectorystory[12];
```



See that trajectory try to move over and run with the dominant propeller.

□G.12.a)

Here is a sample 3 D matrix A:

```
A = {{-0.15, -0.65, 0.75}, {0.85, 0.31, 0.43}, {-0.51, 0.70, 0.5}};
MatrixForm[A]
```

$$\begin{pmatrix} -0.15 & -0.65 & 0.75 \\ 0.85 & 0.31 & 0.43 \\ -0.51 & 0.7 & 0.5 \end{pmatrix}$$

And the 3 D linear system based on it:

```
Clear[x, y, z, t]
linearsystem =
({x'[t], y'[t], z'[t]} == A.{x[t], y[t], z[t]});
ColumnForm[Thread[linearsystem]]
```

$$x'[t] == -0.15\,x[t] - 0.65\,y[t] + 0.75\,z[t]$$
$$y'[t] == 0.85\,x[t] + 0.31\,y[t] + 0.43\,z[t]$$
$$z'[t] == -0.51\,x[t] + 0.7\,y[t] + 0.5\,z[t]$$

You can calculate the eigenvectors of this 3 D matrix:

```
Clear[eigenvector, eigenvalue];
{eigenvector[1], eigenvector[2], eigenvector[3]} =
 Chop[Eigenvectors[A]]
```

$$\{\{0.700838, -0.0634825 - 0.53982\,I, -0.0733127 + 0.456088\,I\},$$
$$\{0.700838, -0.0634825 + 0.53982\,I, -0.0733127 - 0.456088\,I\},$$
$$\{0.133858, 0.638164, 0.758174\}\}$$

And the eigenvalues of this 3 D matrix:

```
Clear[eigenvector, eigenvalue];
{eigenvalue[1], eigenvalue[2], eigenvalue[3]} = Chop[Eigenvalues[A]]
```

$$\{-0.169578 + 0.988743\,I, -0.169578 - 0.988743\,I, 0.999156\}$$

Use what you see to describe how trajectories in this linear system plot out.
Show off your description with a good plot.

□G.12.b.i)

Here's a sample 4 D matrix:

```
A = {{0.195, 0.8235, -0.392, -0.308}, {-0.500, -0.864, 0.260, 0.428},
   {0.787, -0.135, -0.675, -0.777}, {-0.120, -0.857, 0.683, -0.545}};
MatrixForm[A]
```

$$\begin{pmatrix} 0.195 & 0.8235 & -0.392 & -0.308 \\ -0.5 & -0.864 & 0.26 & 0.428 \\ 0.787 & -0.135 & -0.675 & -0.777 \\ -0.12 & -0.857 & 0.683 & -0.545 \end{pmatrix}$$

And the 4 D linear system based on it:

```
Clear[x, y, z, w, t]
linearsystem =
({x'[t], y'[t], z'[t], w'[t]} == A.{x[t], y[t], z[t], w[t]});
ColumnForm[Thread[linearsystem]]
```

$$x'[t] == -0.308\,w[t] + 0.195\,x[t] + 0.8235\,y[t] - 0.392\,z[t]$$
$$y'[t] == 0.428\,w[t] - 0.5\,x[t] - 0.864\,y[t] + 0.26\,z[t]$$
$$z'[t] == -0.777\,w[t] + 0.787\,x[t] - 0.135\,y[t] - 0.675\,z[t]$$
$$w'[t] == -0.545\,w[t] - 0.12\,x[t] - 0.857\,y[t] + 0.683\,z[t]$$

You can calculate the eigenvectors of the 4 D coefficient matrix A:

```
Clear[eigenvector, eigenvalue];
{eigenvector[1], eigenvector[2], eigenvector[3], eigenvector[4]} =
 Chop[Eigenvectors[A]]
```

$$\{\{0.214209 + 0.327555\,I, -0.292508 - 0.188313\,I,$$
$$0.626608, 0.127607 - 0.56292\,I\}, \{0.214209 - 0.327555\,I,$$
$$-0.292508 + 0.188313\,I, 0.626608, 0.127607 + 0.56292\,I\},$$
$$\{0.411078, -0.465618, -0.563164, 0.545033\},$$
$$\{-0.73714, 0.0557527, -0.296352, -0.604725\}\}$$

And the eigenvalues:

```
Clear[eigenvector, eigenvalue];
{eigenvalue[1], eigenvalue[2], eigenvalue[3], eigenvalue[4]} =
 Chop[Eigenvalues[A]]
```

$$\{-0.501175 + 1.15\,I, -0.501175 - 1.15\,I, -0.609097, -0.277553\}$$

Does this information lead you to believe that all trajectories in this 4 D linear system

$\{x[t], y[t], z[t], w[t]\} \to \{0, 0, 0, 0\}$ as $t \to \infty$?

If so, why?
If not, why?

□G.12.b.ii)

Stay with the same 4 D linear system as in part i).
Copy, paste and edit the code for 3 D linear systems given above to produce formulas {x[t], y[t], z[t], w[t]} for the trajectory that starts at

$\{x[0], y[0], z[0], w[0]\} = \{3.2,\ 0.5,\ -6.8,\ 4.2\}$.

□G.12.c)

If someone were to hand you a 7 D linear system, do you think you could handle it?
Explain your response.