Some setup: commands for plotting the dynamical system (ignore this.)

In[1]:=
```
iterates[x10_, x20_, L_, S_] :=
 Partition[Flatten[(x[0] := {{x10}, {x20}};
    x[n_] := S.x[n - 1]; Table[x[n], {n, 0, L}]) ], 2]
```

In[2]:=
```
plotiter[x10_, x20_, L_, S_] := ListPlot[iterates[x10, x20, L, S],
  PlotStyle → PointSize[Large], AspectRatio → 1]
```

Constructing a matrix B as composition of a rotation and a rescaling. (In lecture this matrix was called C, but *Mathematica* reserves the name "C" for a constant.)

In[3]:=
```
θ = 0.22312; r = 1.1;
```

In[4]:=
```
B = {{r, 0}, {0, r}}.{{Cos[θ], -Sin[θ]}, {Sin[θ], Cos[θ]}} ;
```

In[5]:=
```
B // MatrixForm
```

Out[5]//MatrixForm=

$$\begin{pmatrix} 1.07273 & -0.243401 \\ 0.243401 & 1.07273 \end{pmatrix}$$

A matrix A which is similar to B; the two are related by the change-of-basis matrix P.

In[6]:=
```
P = {{4, 1}, {-2, 1}};
```

In[7]:=
```
A = P.B.Inverse[P];
```

In[8]:=
```
A // MatrixForm
```

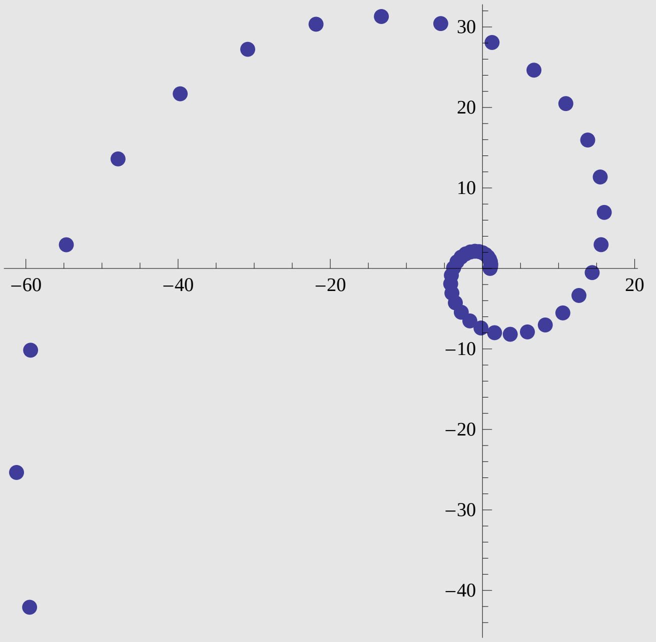
Out[8]//MatrixForm=

$$\begin{pmatrix} 0.788766 & -0.689635 \\ 0.202834 & 1.3567 \end{pmatrix}$$

Iterating the dynamical systems defined by B (first) and A (second) for 50 time steps, starting at the point (1,0). The two pictures are related to one another by the linear transformation P.

In[9]:=
```
plotiter[1, 0, 50, B]
```

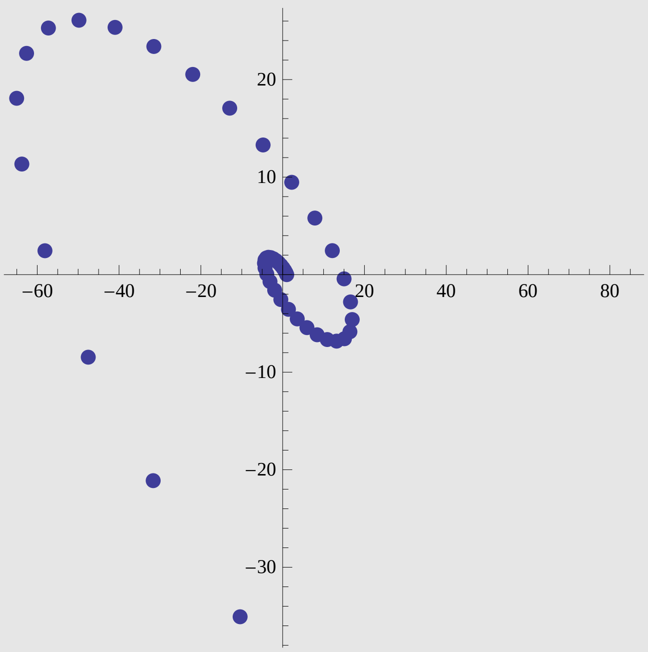
Out[9]=



In[10]:=

```
plotiter[1, 0, 50, A]
```

Out[10]=



The eigenvalues of A. Note they are the same as the eigenvalues of B, as they must be since the two matrices are similar.

In[11]:=

```
Eigenvalues[A]
```

Out[11]=

{1.07273 + 0.243401 i, 1.07273 − 0.243401 i}

In[12]:=
```
Eigenvalues[B]
```
Out[12]=
```
{1.07273 + 0.243401 ⅈ, 1.07273 − 0.243401 ⅈ}
```

The absolute values of the eigenvalues. These predict whether the dynamical system has an attracting, repelling or saddle point.

In[13]:=
```
Abs[Eigenvalues[A]]
```
Out[13]=
```
{1.1, 1.1}
```