Computer Project II
Math 2270-1, April 2002

In this second part of the project you will experiment with a discrete dynamical system.
One can model the level of glucose and insulin in the blood using a system of the form

$$G(n+1) = aG(n) + bH(n)$$
$$H(n+1) = cG(n) + dH(n) .$$

Here G(n) represents the level of glucose in the blood at time n and H(n) represents the level of insulin in the blood at time n. The coefficients a,b,c and d vary for different body chemistries. This model is supposed to be valid between meals. Below I will do some computations for one choice of coefficients a,b,c and d.

```
> restart: with (linalg): with (plots):
Warning, the protected names norm and trace have been redefined and unprotected

Warning, the name changecoords has been redefined
```

```
> A_1 := matrix(2,2, [.9, -.4, .1, .9]);
```

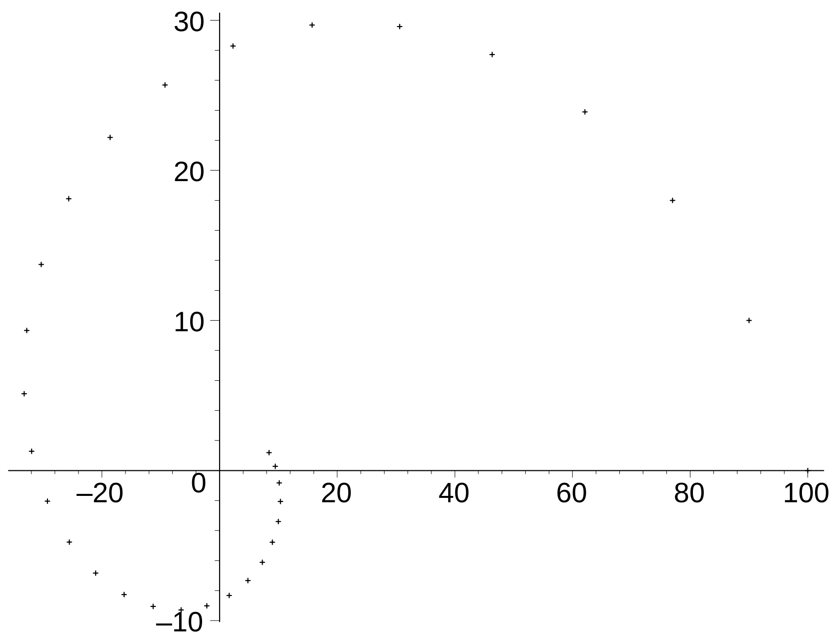
$$A\_1 := \begin{bmatrix} .9 & -.4 \\ .1 & .9 \end{bmatrix}$$

This is one choice of coefficients a,b,c and d. Let's see what happens with the initial conditions G(0)= 100 and H(0) = 0.

```
> v_0 := vector ([100,0]);
```

$$v\_0 := [100, 0]$$

```
> G[0] := 100: H[0] := 0:
  for i from 1 to 30 do
  G[i] := evalm ((A_1)^i&* v_0) [1]:
  H[i] := evalm ((A_1)^i&*v_0) [2];
  #print (i, G[i], H[i]);
  od:
> plot1 := pointplot ({seq([G[i], H[i]], i= 0..30)}):
  display ({plot1});
```



This plot tracks our solution for the first thirty iterations. Notice that the solution seems to spiral inwards and time increases. We can find a formula for the soltution using eigenvalues and eigenvectos as follows. First we find the eigenvecotrs and eigenvalues od A_1.

```
> (v_1,v_2) := eigenvects(A_1);
```

$$v\_1, v\_2 := [.9 + .2000000000\, I, 1, \{[-2.000000000 + 0.\, I, 0. + 1.\, I]\}],$$
$$[.9 - .2000000000\, I, 1, \{[-2.000000000 - 0.\, I, 0. - 1.\, I]\}]$$

```
> v_1; v_2;
>
```

$$[.9 - .2000000000\, I, 1, \{[-2.000000000 - 0.\, I, 0. - 1.\, I]\}]$$
$$[.9 + .2000000000\, I, 1, \{[-2.000000000 + 0.\, I, 0. + 1.\, I]\}]$$

Now we want to write v_0 as a linear combination of the eigenvectors v_1 and v_2. In fact, if we work with the eigenvectors v_1 = [-2, i] and v_2 = [2,i] then we see

v_0 = -50 v_1 + 50 v_2.

Then, because v_1 has an eigenvalue of .9 + .2 i and v_2 has an eigenvalue of .9 - .2 i, we see that (don't worry about the imaginary parts; they cancel)

A_1^k (v_0) = -50 A_1^k (v_1) + 50 A_1^k (v_2) =

-50(.9+.2i)^k v_1 + 50(.9-.2i)^k v_2.

The important thing to ntoice is that the real part of each eignevalue is .9, which is a positive number less than one. Thus we can see that the solutiongets closer and closer to the origin with each iteration (by a factor of .9).

Your part of this project is the following:

Find a 2x2 matrix A_2 such that for some initial data the solution collapses towards the origin and for other initial data the solution gets arbitrarily large. Demonstrate this behavior by plotting some data for 30 iterations. (Hint: think about what the eigenvalues of A_2 have to be.)