MCS 320 Project Two : modeling the outbreak of war
due Monday 31 March 2003, at 2PM.

This goal of this project is to study coupled nonlinear difference equations. Such equations occur frequently, for example, in modeling an arms race, or price determination by competing firms. This document is made from a Maple worksheet. A good way to start this project is to download the worksheet from the course web site.

0. A model for an arms race.

In this project we study the behavior of coupled nonlinear difference equations with Maple. The equations model the devotion of two nations (we call them `X` and `Y`) to arms spending in their annual budget cycle. The variables `x[n]` and `y[n]` are defined as follows:

```
[> x[n] = `arms expenditure of nation X`/`gross national product of X`@`year n`;
[> y[n] = `arms expenditure of nation Y`/`gross national product of Y`@`year n`;
```

Obviously: `0 < x[n] < 1 and 0 < y[n] < 1.` To stay competitive, the spending of `X` in year `n+1` is proportional to what `Y` spent in year `n`. Likewise, `y[n+1]` is proportional to `x[n]`.

The equation proposed in the literature depends on a parameter `p`:

```
[> `F[p](x,y)` := x[n+1] = 4*p*y[n]*(1-y[n]);
```

and is symmetric in `x` and `y`. Assuming the spending of `X` is regulated by the parameter a and the spending of `Y` by `b`, we have the following equations:

```
[> F[a] := subs(p=a,`F[p](x,y)`);
[> F[b] := subs({p=b,x=y,y=x},`F[p](x,y)`);
```

These two equations form a system of two coupled difference equations. The goal is to understand the evolution of `x[n]` and `y[n]` in function of the parameters `a` and `b`.

Note: when thinking about an arms race depresses you, coupled difference equations occur also as models for price setting mechanisms, when two competing comparable firms determine their price for the same commodity.

1. Defining functions for the model

So far, so good, but we need numbers to start playing with the model. Let us turn the equations into functions:

```
[> Fa := unapply(rhs(F[a]),y[n]);
[> Fb := unapply(rhs(F[b]),x[n]);
```

We are a bit confused by the output, as we no longer recognize the `y[n]` and `x[n]` as formal parameters. From a mechanical point of view, everything is fine, as we can use any name for a formal parameter in a procedure, so Maple can choose anything as formal parameter. Logically, we must not forget the meaning of the functions: `x[n+1] = Fa(y[n])` and `y[n+1] = Fb(x[n])`. Notice: `y[n] = Fb(x[n-1])`, so we may write `x[n+1] = Fa(Fb(x[n-1]))`, and likewise: `y[n+1] = Fb(Fa(y[n-1]))`. So the composition of functions comes in handy:

```
[> Fab := Fa@Fb; Fba := Fb@Fa;
```

After these compositions, we seem to have decoupled the system, but let us again form an equation to reveal the meaning:

```
[> x[2] = Fab(x[0]); y[2] = Fba(y[0]);
```

The solutions to the equations above will turn out to be very interesting. But before we get into this, we leave the composite functions for now.

Let us choose values for `a` and `b` and evaluate.

```
[> fa := subs(a=0.8,eval(Fa)); fb := subs(b=0.4,eval(Fb));
```

Now we have two "real" functions we can evaluate.

2. The plots

In order to see something about the model, we must make a plot. To make a plot, we sample data from the functions, for `N` budget cycles:

```
[> N := 20:
[> x[0] := 0.01: y[0] := 0.05:
[> points := []:
[> for n from 0 to N-1 do
>     x[n+1] := fa(y[n]):
>     y[n+1] := fb(x[n]):
>     points := [op(points),[n+1,x[n+1],y[n+1]]]:
> end do:
```

We define a sequence of plot parameters:

```
[> plp := axes=boxed,style=line,color=red,tickmarks=[3,4,4],labels=["t","x","y"]:
```

for the command pointplot3d:

```
[> plots[pointplot3d](points,plp);
```

This 3-dimensional plot may be oriented manually to study the relationships between the variables. With some extra parameters to the plot command, we choose special orientations to view `y` as function of time, `x` as function of time, and finally `y` as function of `x`. The three respective plots which show projections on the coordinate planes `yt`, `xt`, and `xy` are as follows:

```
[> plots[pointplot3d](points,plp,view=[1..N,0..1,0..1],orientation=[-90,90]);
[> plots[pointplot3d](points,plp,view=[1..N,0..1,0..1],orientation=[-90,180]);
[> plots[pointplot3d](points,plp,view=[1..N,0..1,0..1],orientation=[0,90]);
```

So we see that in time, the ratio of arms expenditures over the gross national product converges to an equilibrium.

3. Computing Fixed Points

From the plots we just discovered the existence of an equilibrium. With sampling, we can just compute this equilibrium:

```
[> x[N],y[N];
```

but this method raises several questions. For instance, how do we know that these values are independent of `x[0]` and `y[0]`? Is there just one equilibrium, or can we have many, or even, can we have none?

Remember the composite functions? Well, here we can use them, as they define fixed-point equations:

```
[> e1 := x = Fab(x); e2 := y = Fba(y);
```

Recall our special values for a and b:

```
[> e1ab := subs(a=0.8,b=0.4,e1); e2ab := subs(a=0.8,b=0.4,e2);
[> fsolve(e1ab,x); fsolve(e2ab,y);
```

With `fsolve` we confirmed the end points of the sampling and found (besides the trivial zero solutions) more accurate values for the equilibrium. But wait, look at what we just solved: a polynomial equation of degree four. Couldn't it be that there are four solutions? Indeed, check this out:

```
[> e1ab1 := subs(a=0.8,b=0.8,e1);
[> fsolve(e1ab1,x);
```

What does this mean? Well, we should make a plot...

4. Chaos triggers war

In a situation where multiple fixed points are possible, the natural question is: are there a finite number of fixed points? In looking for fixed points, we might as well at multiple iterations of `Fab`:

```
[> x = Fab(Fab(x));
```

As we apply `Fab` repeatedly, the degree of the polynomial equation can get arbitrarily high, with as many real solutions as we like. In the case of an infinity number of equilibria, the solutions jump around without convergence to a stable fixed point. This situation occurs for high values of `a` and `b`, when arms expenditures run at very high levels.

4. Assignments

The solution to this project consists of a well documented worksheet, organized in three subsections, containing the answers to the three questions below.

4.1 Write F[p](z) as indexed procedure

As seen from above, there is actually only one formula for the function, i.e. `F[p](z) = 4*p*z*(1-z)`. If we wish to study the sensitivity of the model in function of the parameters `a` and `b` (and thus change `a` and `b` often), we better leave them as parameters.

Write a Maple procedure `F` which takes the parameter `p` as an index and the variable `z` as regular argument. Test whether your implementation of `F` satisfies the following equalities:

```
[> F[0.8](0.2) = fa(0.2); F[0.4](0.2) = fb(0.2);
```

If the index is omitted, then the procedure call should return an error message, like this:

```
[> F(0.2);
Error, (in F) provide parameter as index
```

Use “error” return from a procedure to produce this error message (see the help pages of Maple, typing ?error).

4.2 A procedure to generate the samples

To plot the result of a simulation, we first need to generate samples (section 3 above). Write a Maple procedure with prototype

```
[> sample
:= proc(F::procedure,N::nonnegint,a::float,b::float,x0::float,y0::float)
```

This procedures takes on input the procedure `F`, created in 4.1, `N` the number of sample points, the parameters `a` and `b`, and the respective initial values `x0` and `y0` for `x[0]` and `y[0]`. On return is a list of samples, ready to be used for a plot. For example,

```
[> s := sample(F,20,0.8,0.4,0.01,0.05);
[> plots[pointplot3d](s,plp);
```

should show the same plot as the one produced in section 3.

Now that we can make plots more easily, we can try to visualize what happens if one nation increases steadily its parameter `p`. In particular, suppose nation `Y` increases its parameter `b` from 0.4 to 0.95 in increments of 0.01. Just as we have built a list of points in the procedure `sample`, we can build a list of plots. For this example, we need $(0.95 - 0.4)/0.01 = 55$ plots, made with `sample` and `plots[pointplot3d]` as above. Suppose we store the list of plots in `pp`, then the command

```
[> plots[display](pp,insequence=true);
```

produces an animation of 55 frames, illustrating the transition from a converging simulation into a chaotic situation. For a more dramatic effect, we may use more than 20 budget cycles, e.g., `N` = 40 offers a nice view. Without the `insequence=true`, Maple plots the surface traced by the trajectories.

Give the Maple commands to create the list of plots, the animation, and the surface traced by the trajectories. Investigate two cases: when one nation increases its spending (like above), and when the two nations are simultaneously increasing their parameters.

While the animation is nice to watch, it is hard to catch on paper. The paper you hand in should contain a plot of the two surfaces, in the default 3-d view, plus the projections of the surfaces on the coordinate planes `xt`, `yt`, and `xy`.

4.3 Characterization of the critical values

From the plots made when nation Y is increasing its parameter `b`, we can see three cases:

1. only one equilibrium;
2. multiple equilibria, with oscillating trajectories;
3. chaos, as the trajectories follow no pattern.

Select a viewpoint in the 3-d plot which shows these three cases and put this plot on paper.

Try to find the two critical values for `b` which mark the transition between cases 1 and 2, and cases 2 and 3. One good method to attempt this is to experiment with the range of the plots, i.e.:

```
[> plots[display](pp[1..30]);
```

only shows the first 30 values for `b`. From this experiment, we can approximate the two critical values which mark the transitions between the three cases.

In addition to this experiment, compute the fixed points using the approximations of the critical values of `b` with the fsolve command.

4.4 The deadline is Monday 31 March 2003, at 2PM.

Bring to the lecture the print out of your worksheet. In addition also e-mail me your worksheet.

Besides a clear structure of the worksheet, it is very important that your solution describes the model *in your own words*. Imagine you are writing a report on this arms race model for a friend who is not taking this class. Your worksheet and the paper produced from it should thus be well documented and self contained.

If you have questions, comments, or difficulties, feel free to come to my office for help.