ME 406
The Logistic Map

```
sysid
```

```
Mathematica  6.0.3, DynPac 11.02, 4/22/2009
```

```
plotreset; imsize = 250;
```

```
intreset;
```

1. Introduction

The logistic map is discussed in many references. A very complete and readable discussion is given in Chapter 10 of **Nonlinear Dynamics and Chaos** by Steven Strogatz, Addison-Wesley, 1994. Many of the interesting properties of the map were discovered by the mathematical biologist Robert May ("Simple Mathematical Models with Very Complicated Dynamics," Nature **261**, 459, 1976.) The basic form of the map is

$$x_{n+1} = r x_n (1 - x_n) \ .$$

In many applications, the map is a model for the dynamics of a population, and x_n is the population of the nth generation. As the work of May and others has shown, this map exhibits an astonishing range of behavior as the growth rate r is varied. We will use the range $0 < r \leqslant 4$. For $r = 4$, the interval [0,1] is mapped onto itself; for $0 < r < 4$, it is mapped into itself. ,Many of the basic features of this map can be established analytically, and some of this analyis may be seen in section 10.3 of Strogatz. In this notebook, our approach will be primarily numerical, using the functions built-in to DynPac. We begin by defining the system for DynPac. We start with the command setmap, which tells DynPac that this is a mapping and not a differential equation.

```
setmap;
```

Now we define the state variable, the mapping function, which is assigned to slopevec, and the optional system name.

```
setstate[{x}]; setparm[{r}];
slopevec = {r * x * (1 - x)}; sysname = "Logistic Map";
```

We check our definitions with a sysreport.

```
sysreport
```

```
SYSTEM DEFINITION (11.02)

    System name: sysname = Logistic Map

    State vector: statevec = {x}

    State units: stateunits = {}

    Slope vector: slopevec = {r (1 - x) x}

    Parameter vector: parmvec = {r}

    Parameter values: parmval = {r}

    Parameter units vector: parmunits = {}

    Time unit: timeunit =

    System Type: sysmode = mapping
```

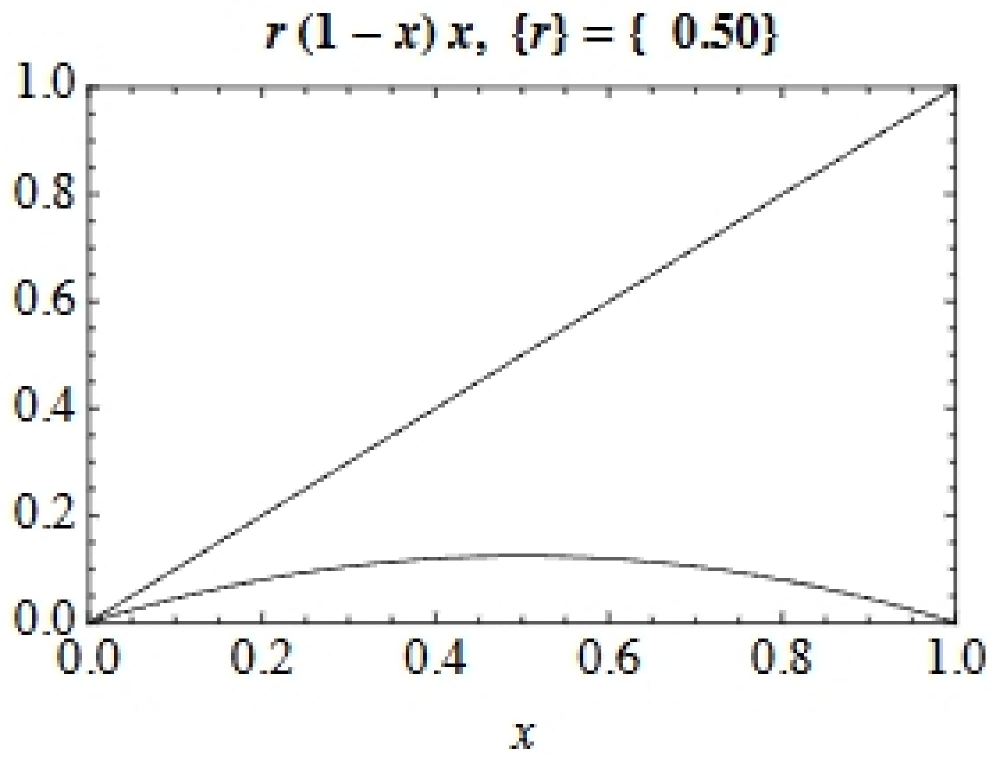
We could have used this same function as the slope for a differential equation, in which case we would use the command setde to tell DynPac that we are working with a differential equation. The primary differencebe tween the mapping and differential equation modes is just the actual stepping algorithm used in constructing solutions -- a Runge-Kutta step for a differential equation, and a map iteration for the mapping. All of the other supporting code is essentially identical in the two cases.

■ 2. Equilibrium Points

We start by viewing the map for several different values of the parameter *r*.

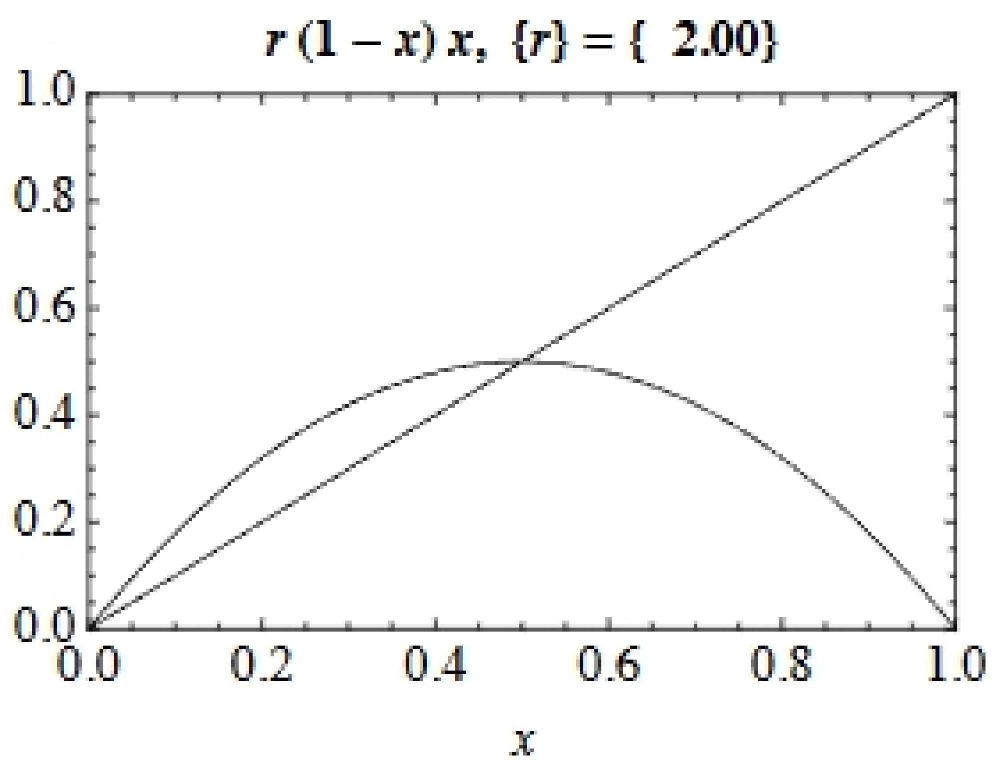
```
parmval = {0.5};
```

```
viewmap[];
```




```
parmval = {2.0};
```

```
viewmap[];
```



For $r = 0.5$, there is a single equilibrium point at $r = 0$. For $r = 2$, there are two equilibrium points -- one at $x = 0$ and the other at $x = 0.5$. For a general value of r, the fixed points are

```
findpolyfix[]
```

$$\left\{\{0\}, \left\{\frac{-1+r}{r}\right\}\right\}$$

Thus there are always two equilibria, but the second one is relevant (in the range [0,1]) only for $1 \leqslant r \leqslant 4$. The stability of the equilibria is determined by the derivative of the mapping. We have

```
D[slopevec, x] /. x → 0
```

```
{r}
```