Differential Equation Model.

Let $x(t)$ be a fish population size at time t.

$$x(t + \Delta t) - x(t) \approx \Delta t\,(b - d)\,x(t)$$

The simplest model is for $b - d$ being a constant.

A second model is the logistic model where

$b - d = c(M - x)$ with c, M constant and M is the maximum population size.

A third model may involve harvesting, $-h(t)$, or stocking, $+s(t)$, at a given rate.

The differential equation model is attainted by dividing by dt and letting dt go to zero. When b – d is a constant,

$$x' = (b - d)x \text{ whose solution is}$$

$$x(t) = x(0)\, e^{(b - d)t}.$$

A variation of the logistic model that includes harvesting is

$$x' = c(M - x)x - h(t).$$

The exact solution of this is more difficult to find, and therefore, we will need numerical methods.

Method of Solution.

We will use a variation of the Taylor method, which avoids the computation of the partial derivatives of the right side function f, f_t, f_u, f_{tt}, f_{tu}, f_{uu}, ….

These are known as the Runge-Kutta methods, and one important example is the fourth order Runge-Kutta scheme.

This scheme can be viewed as a

combination of the fourth order Taylor polynomial method and the Simpson's rule for numerical integration.

The equivalent integral equation

$$u(t+h) = u(t) + \int_{t}^{t+h} f(z, u(z))dz.$$

The integral can be approximated by

$(h/6)(k_1 + 4(k_2/2 + k_3/2) + k_4)$ where

$k_1 = f(t,u(t))$,

$k_2 = f(t+h/2, u(t) + (h/2)k_1)$,

$k_3 = f(t+h/2, u(t) + (h/2)k_2)$ and

$k_4 = f(t+h, u(t) + hk_3)$.

The **fourth order Runge-Kutta method,** where t and u(t) are replaced by t^k and u^k, is

$$u^{k+1} = u^k + (h/6)(k_1 + 2k_2 + 2k_3 + k_4).$$

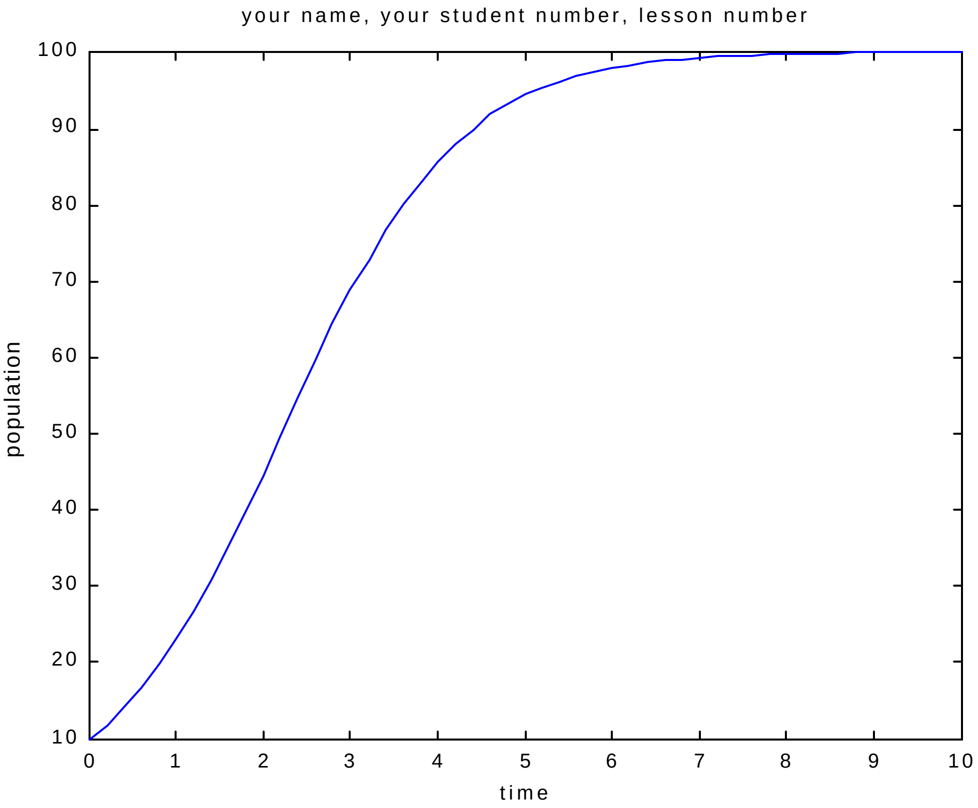
Matlab Implementation.

Use the function file fpop.m and the m-file rk4.m.

```
function fpop = fpop(t,x)
  fpop = .01*(100 - x)*x;

%your name, your student number,
lesson number
clear;
x(1) = 10.;
T = 10;
KK = 50
h = T/KK;
t(1)= 0.;
for k = 1:KK
    k1 = h*fpop(t(k),x(k));
    k2 =  h*fpop(t(k)+.5*h,
          x(k)+.5*k1);
    k3 = h*fpop(t(k)+.5*h,
          x(k)+.5*k2);
    k4 = h*fpop(t(k)+h,x(k)+.5*k3);
    x(k+1) = x(k) + (k1 + 2*k2
               + 2*k3 +k4)/6;
    t(k+1) = t(k) + h;
end
```

```
plot(t,x)
title('your name, your student
number, lesson number')
xlabel('time')
ylabel('population')
```



This illustrates stability of the largest steady state Solution, x = 100. rk4 was run with a number of different initial populations.