Fish Harvesting Revisited

Kurt Bryan and SIMIODE

Notebook for fish logistic-growth harvesting project in Section 3.5.2.

```
> restart;
  with(plots) :
```

The Data: The data from Table 3.10, starting with 1978, which we will call year 0.

```
> udata := Array(0..29, [72148, 73793, 74082, 92912, 82323, 59073, 59920, 48789, 70638,
      67462, 68702, 61191, 49599, 46266, 34877, 28827, 21980, 17463, 18057, 22681, 20196,
      25776, 23796, 19240, 16495, 12167, 21104, 18871, 21241, 22962]) :
```

Note that the data is indexed from 0 to 29.

Harvest rates from Table 3.10.

```
> hd := Array(0..29, [0.18847, 0.149741, 0.21921, 0.17678, 0.28203, 0.34528, 0.20655, 0.33819,
      0.14724, 0.19757, 0.23154, 0.20860, 0.33565, 0.29534, 0.33185, 0.35039, 0.28270, 0.19928,
      0.18781, 0.19357, 0.18953, 0.17011, 0.15660, 0.28179, 0.25287, 0.25542, 0.08103,
      0.087397, 0.081952, 0.10518]) :
```

Each list has data points from year t = 0 to year t = n with

```
> n := 29 :
```

Plot of the population data

```
> pdat := [seq([k - 1, udata[k]], k = 0..n)] :
  ppop := plot(pdat, color = red, labels = ["Time (years)", "Biomass (tons)"], labeldirections
      = [horizontal, vertical])
```

Harvest rate by year:

```
> hdat := [seq([k - 1, hd[k]], k = 0..n)] :
  plot(hdat, color = red, labels = ["Time (years)", "Harvest rate"], labeldirections = [horizontal,
      vertical])
```

Fitting the Data to the ODE: Form finite difference approximations (u(k+1) - u(k))/1 from the population data,
for k = 0 to k = 28 (= n-1). This approximates u'(t) when t = k:

```
> udiffdata := Array(0..28, [seq(udata[k + 1] - udata[k], k = 0..n - 1)]) :
```

Next substitute u = udata[k] into the harvested logistic ODE u'(t) = r*u(t)*(1-u(t)/K) - h(t)*u(t) for k = 0 to k = n-1, to approximate the right side of the ODE at time t = k for k = 0 to k = n-1. This expression depends on r and K.

$$> udiffdata2 := Array\left(0..28, \left[seq\left(r \cdot udata[k] \cdot \left(1 - \frac{udata[k]}{K}\right) - hd[k] \cdot udata[k], k = 0..n - 1\right)\right]\right) :$$

Form a sum of squares that depends on r and K

$$> SS := add\left((udiffdata[k] - udiffdata2[k])^2, k = 0..n - 1\right) :$$

Finding the optimal r and K: Start with a plot of SS(r, K), or log(SS).

```
> plot3d(log(SS), K = 40000 ..300000, r = 0.1 ..0.5, labels = ["K", "r", "S(r,K)"])
```

Or a contour plot:

```
> contourplot(log(SS), K = 40000 ..300000, r = 0.1 ..0.5, contours = 100)
```

Something near r = 0.3, K = 200000 might be a good initial guess at a minimizer for SS.

If we use these values for r and K (but you should find the true minimizer) we could compare the predicted cod population to the data by following the suggestion of Modeling Exercise 5.2.3. Let U be an array to hold our numerical estimates of the cod population:

```
> U := Array(0 ..n - 1) :
```

Set U[0] equal to the cod biomass at time t = 0:

```
> U[0] := udata[0]
```

Now use the harvested logistic ODE with our estimates for r and K to march the predicted cod biomass out in time (this is essentially Euler's method):

```
> for k from 0 to n-2 do
```

$$U[k+1] := subs\left(r = 0.3, K = 200000, U[k] + r \cdot U[k] \cdot \left(1 - \frac{U[k]}{K}\right) - hd[k] \cdot U[k]\right) :$$

```
  od:
> ppred := plot([seq([k, U[k]], k = 0 ..n - 1)], color = blue) :
> display(ppop, ppred, legend = ["data", "model"], linestyle = [1, 3], labels = ["year",
      "population"])
>
```