The Hill-Keller Model

Kurt Bryan and SIMIODE

A Maple worksheet to fit the Hill-Keller model v'(t) = P - k*v(t) to the data from Usain Bolt's 2008 Beijing gold medal Olympic race. This is a minor extension of the worksheet "hill_keller_model1.mw". Here we fit both P and k.

In this example we adjust both P and k to fit the data, in a least-squares sense.

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

The Data: Here is the data for Usain Bolt's 2008 Beijing race. The data consists of pairs [t, d], where d is 0, 10, 20, ..., 100 meters and t is the corresponding split time.

```
> bolttimes := [0.165, 1.85, 2.87, 3.78, 4.65, 5.50, 6.32, 7.14, 7.96, 8.79, 9.69] : #Raw times
  bolt := [seq([bolttimes[k], 10*(k-1)], k=1..11)]; #Arranged into (time,position) pairs.
```

A quick plot

```
> plt1 := pointplot(bolt, color=red, symbol=solidcircle, symbolsize=15, labels
        = ["Time (seconds)", "Distance (meters)"], labeldirections=[horizontal, vertical]);
```

Solving The ODE: Solve the Hill-Keller ODE, treating both parameters P and k as unknown. Also use initial condition v(0.165) = 0.

```
> de := v'(t) = P - k·v(t);
  t0 := 0.165 : #Start of race
  vsol := rhs(dsolve({de, v(t0) = 0}, v(t)))
```

Compute position by integrating

```
> vsol2 := subs(t=tau, vsol) :
  xpos := simplify(int(vsol2, tau=t0..t)) #Expression for his position in terms of t, and k.
```

The position of Bolt on the track is given by defining the function X(t):

```
> X := unapply(xpos, t);
```

Let's guess a value of P and k and plot X(t) with the data.

```
> plt2 := plot(subs(k=1, P=11, X(t)), t=t0..9.69, color=blue) :
> display(plt1, plt2, title="Hill-Keller Model Versus Data")
```

The Optimal Choice for P and k: Not bad, but we can do better by forming a sum of squares SS and minimizing

```
> SS := add((X(bolt[j, 1]) - bolt[j, 2])^2, j=1..11)
```

We already know that k somewhere around 1 and P around 11 yields a reasonable fit. Plot SS as a function of P and k to get a better
estimate of where SS is minimized. Although plotting ln(SS) is even better:

```
> plot3d(ln(SS), k=0.7..1.1, P=8..12)
```

Rotating the graph around shows k around 0.85 and P around 10.3 looks promising. So set d(SS)/dk = 0 and d(SS)/dP = 0 and use Maple's fsolve command to find a good solution

```
> dSSdk := diff(SS, k) :
  dSSdP := diff(SS, P) :
> kPbest := fsolve({dSSdk=0, dSSdP=0}, {k, P}, k=0.8..0.9, P=10..11)
```

The residual is

> $evalf(subs(kPbest, SS))$

Use this value in X(t) to plot and compare to the data

> $plt2 := plot(subs(kPbest, X(t)), t = t0 ..9.69, color = blue)$:

> $display(plt1, plt2, title =$ "Hill-Keller Model Versus Data")

Alternatively, we can minimize SS with respect to k by using Maple's built-in optimization routines.
First load the ***Optimization*** package:

> $with(Optimization)$:

> $minsol := Minimize(SS, initialpoint = \{k = 0.9, P = 10\})$

The residual is 0.775, minimizing value for k is 0.865, for P it is 10.38.

>