Sensitivity Analysis Computations in MLAB

Gary D. Knott, Ph.D.
Civilized Software Inc.
12109 Heritage Park Circle
Silver Spring, MD 20906 USA
Tel: (301)962-3711
Email: csi@civilized.com

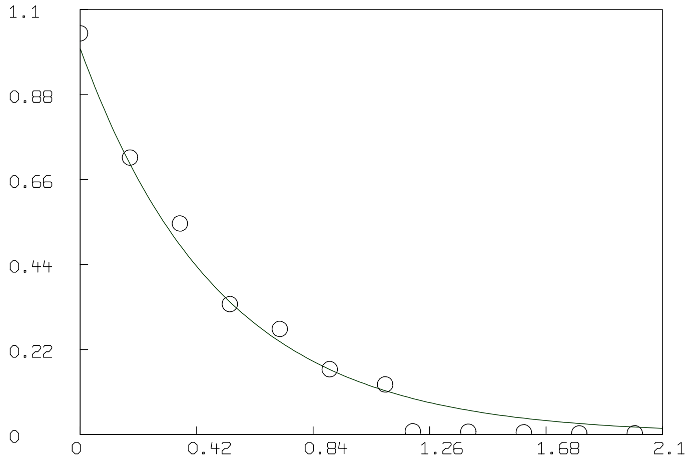
It may be interesting to look at the rate of change of a model function with respect to a parameter. This derivative function is a kind of sensitivity indicator that shows locally in a neighborhood of a fixed value of the parameter how sensitive that model is to a change in the specified parameter. This shows, among other things, where a new observation would be most valuable for determining the paramter of interest. There are other kinds of sensitivity measures, but we shall initially focus on this rate-of-change sensitivity.

If our model function is explicitly known as an algebraic formula, then it is a simple matter to use MLAB to compute the desired derivative function symbolically and evaluate and graph it over the desired range.

For example, suppose our model function is a simple one-component decaying exponential with the parameter k that has been determined to best fit the data points in the matrix d as shown below.

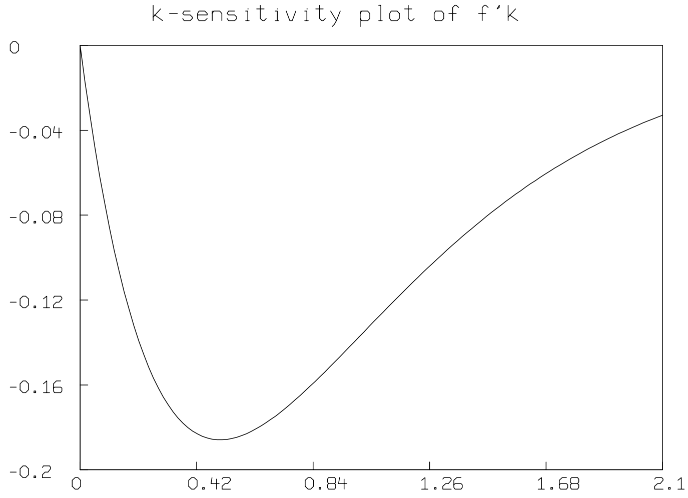
```
* function f(t)=exp(-k*t)
* constraints q={k>0}
* k=1;
* d = list(0,  1.0386,\
: .18, .717,  \
: .36, .5463, \
: .54, .33764,\
: .72, .27316,\
: .9,  .1692, \
: 1.1, .12954,\
: 1.2, .00815,\
: 1.4, .00707,\
: 1.6, .00555,\
: 1.8, .00291,\
: 2,   .003)
* d=shape(12,2,d)
* fit(k),f to d, constraints q
final parameter values
value             error               dependency          parameter
1.979230909       0.09787757296       -1.110223025e-16    K
4 iterations
CONVERGED
```

```
best weighted sum of squares = 1.878653e-02
weighted root mean square error = 4.132634e-02
weighted deviation fraction = 4.569453e-02
R squared = 9.849903e-01
no active constraints
* draw d pt circle lt none
* draw points(f,0:2.1!120) color red
* view
```



Then the k-sensitivity derivative curve can be computed and graphed as follows. Note this curve is a slice of a k-sensitivity surface defined by t and k.

```
* del w
* draw points(f’k,0:2.1!120)
* top title " k-sensitivity plot of f’’k"
* view
```



If our model function is defined by a system of one or more differential equations, then computing the sensitivity derivative curves requires solving an augmented system of differential equations.

For example, if we have the simple kinetics differential equation model

$$\frac{dc(t)}{dt} = k_1(a_0 - c(t)) \cdot (b_0 - c(t)) - k_2 c(t) \quad \text{with } c(0) = 0,$$

with the parameters k_1 and k_2, then the function $\partial c(t)/\partial k_1$ can be computed numerically for a range of t-values by defining $y_1(t) = \partial c(t)/\partial k_1$, and introducing a differential equation for y_1, namely

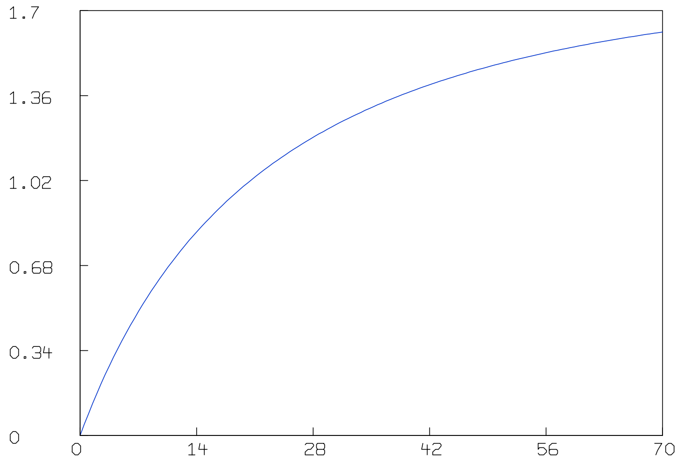
$$\frac{dy_1(t)}{dt} = \frac{\partial^2 c(t)}{\partial t \partial k_1}.$$

The function $\partial c(t)/\partial k_1$ can be similarly obtained by defining $y_2(t) = \partial c(t)/\partial k_2$, and introducing the differential equation

$$\frac{dy_2(t)}{dt} = \frac{\partial^2 c(t)}{\partial t \partial k_2}.$$

We show below how this example is handled in MLAB.

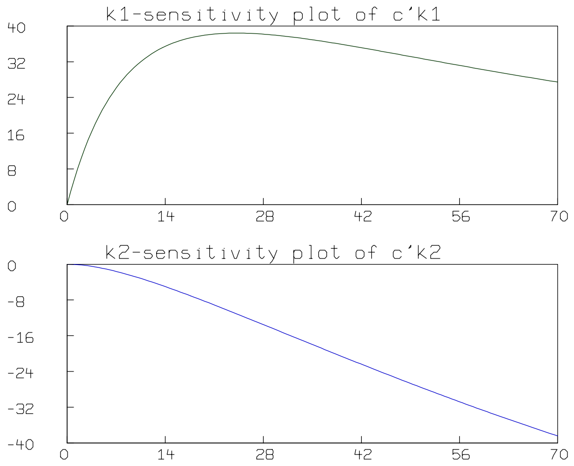
```
fct c't(t) = k1*(a0-c(t))*(b0-c(t))-k2*c(t)
initial c(0) = 0
k1 = .015;k2 = .002
a0 = 2;b0 = 3
draw points(c,0:70!140) color yellow
view
```



```
fct c'k1(t) = y1(t)
fct y1't(t)=c't'k1(t)
```

```
initial y1(0) = 0
fct c’k2(t) = y2(t)
fct y2’t(t) = c’t’k2(t)
initial y2(0) = 0
m=integrate(y1’t,y2’t,c’t,0:70!140)
del w
draw m col (1,2) color red
frame 0 to 1, .5 to 1
top title "k1-sensitivity plot of c’’k1"
w1=w

draw m col (1,4) color green
frame 0 to 1, 0 to .5
top title "k2-sensitivity plot of c’’k2"
view
```



This picture is the same result that we obtain when we compute $\partial c/\partial k_1$ and $\partial c/\partial k_2$ by a centered finite difference approximation.

File: /General/MLAB-Text/Paper/sensitivity/sens.tex