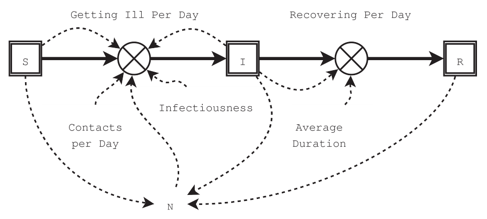
System Dynamics with MCSim

A Facilitated Systems Quick Reference

An Example: the SIR Model

This simple model of a disease epidemic has three state variables, S, I, and R, representing the number of susceptible, infectious, and recovered (and thus immune) people in the population.



Susceptible people become ill and thus infectious by contacting an infectious person. The number getting ill per day is the product of the total number of susceptible people, the number of contacts they have per day with infectious people, and the probability that a contact results in disease. The probability of a contact being with an infectious person is the ratio of the number of infectious people to the total population. There is no incubation period.

The disease has a fixed average duration. The number of people recovering per day is the number of infectious people divided by the average duration.

A stock and flow diagram can be used to help design a system dynamics model and to communicate its structure to others. In the diagram, rectangles represent state variables (stocks), pipes with stylized valves represent flows between state variables, and dotted lines represent information flows. Clouds at the start or end of flows indicate system boundaries.

Model Listing: `sir.model`

```
States = {
  S,  # No of susceptibles
  I,  # No of infectious
  R   # No of recovered
};

Outputs = {
  Getting_Ill_PD,
  Recovering_PD,
  Reproduction_Rate,
  N
};

Initial_S =     9999.0;
Initial_I =        1.0;
Initial_R =        0.0;
Contacts_PD =      4.0;
Infectiousness =   0.25;
Average_Duration = 4.0;

Initialize{
  S = Initial_S;
  I = Initial_I;
  R = Initial_R;
}

Dynamics{
  N = S + I + R;
  Getting_Ill_PD =
    Contacts_PD * (I / N)
    * Infectiousness * S;
  Recovering_PD =
    I / Average_Duration;
  dt(S) = -Getting_Ill_PD;
  dt(I) = Getting_Ill_PD
    - Recovering_PD;
  dt(R) = Recovering_PD;
}

CalcOutputs{
  Reproduction_Rate =
    Contacts_PD *
    Infectiousness *
    Average_Duration * (S/N);
}
```

Simulation File: `sir.test01.in`

```
Integrate(Lsodes,1.0e-6,1.0e-6,0);
OutputFile("sir.test01.out");
StartTime(0.0);
Initial_S = 9999.0;
Infectiousness = 0.25;

Simulation { # Base
Contacts_PD = 3.0;
Average_Duration = 4.0;
PrintStep(I,0.0,200.0,1.0);
PrintStep(Getting_Ill_PD,0.0,200.0,
  1.0);
}

Simulation { # Safe
Contacts_PD = 1.0;
Average_Duration = 1.0;
PrintStep(I,0.0,200.0,1.0);
PrintStep(Getting_Ill_PD,0.0,200.0,
  1.0);
}

Simulation { # Scary
Contacts_PD = 5.0;
Average_Duration = 7.0;
       .
       .
       .
```

Simulation File Commands

```
Integrate(Lsodes, <rtol>, <atol>,
  <method>);
```

<rtol>: rel. error tolerance at each step
<atol>: abs. error tolerance at each step
<method>: 0 for non-stiff, 1 for stiff systems

```
Integrate(Euler, <time-step>, 0,
  0);
Print(<identifier1>,
  <identifier2>, ..., <time1>,
  <time2>, ...);
PrintStep(<identifier>, <start-
  time>, <end-time>, <time-step>);
```

Model Execution

Cygwin:

```
$ makemcsims sir.model
$ ./mcsimstd.sir sir.test01.in
```

Other Environments:

```
$ makemcsim sir.model
$ ./mcsimstd.sir sir.test01.in
```

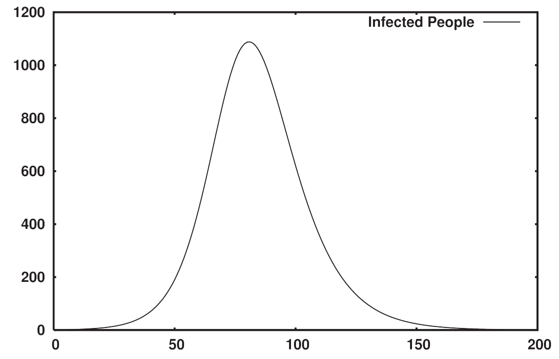
Model Output

```
$ less sir.test01.out
Results of Simulation 1

Time  I            Getting_Ill_PD
0     1            0.749925
1     1.64854      1.23616
2     2.71743      2.03735
3     4.47868      3.35692
         .
         .
         .
```

Gnuplot

```
gnuplot> plot 'sir.test01.out'
  index 4 using 1:2 title
'Infected People' with lines
```



Facilitated Systems

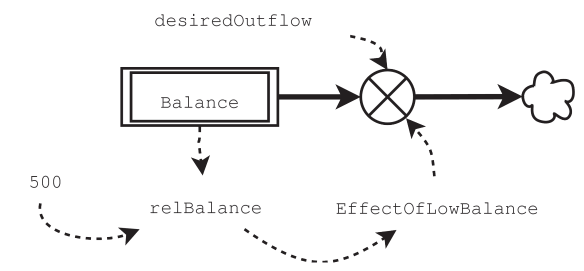
making sense of tough problems together

`http://facilitatedsystems.com/`

`bill_harris@facilitatedsystems.com`

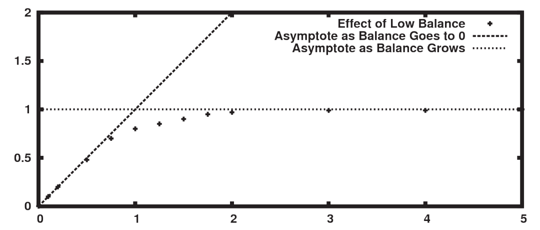
+1 425 374-1845

Nonlinearities



Copy the italicized Inline code, substituting your own expressions for the bold text.

```
States = {Balance};
Initialize {Balance = 1000.0;}
desiredOutflow = 100.0;
Dynamics {
Inline(
#include <gsl/gsl_errno.h>
#include <gsl/gsl_spline.h>
gsl_interp_accel *accel_ptr;
gsl_spline *spline_ptr;
int npts = 13;
double x_array[13] = {0.0, 0.1,
 0.2, 0.5, 0.75, 1.0, 1.25, 1.5,
 1.75, 2.0, 3.0, 4.0, 5.0};
double y_array[13] = {0.0, 0.1,
 0.2, 0.48, 0.7, 0.8, 0.85, 0.9,
 0.95, 0.97, 0.99, 0.99, 1.0};
accel_ptr =
 gsl_interp_accel_alloc ();
spline_ptr = gsl_spline_alloc
 (gsl_interp_akima, npts);
gsl_spline_init (spline_ptr,
 x_array, y_array, npts);
);
```



```
EffectOfLowBalance = 0.0;
relBalance = Balance / 500.0;
Inline(
EffectOfLowBalance =
 gsl_spline_eval (spline_ptr,
 relBalance, accel_ptr);
);
dt(Balance) = -desiredOutflow *
  EffectOfLowBalance;}
```

Inline Variable Names

To use the output variable **foo** in an inline section, call it **rgModelVars[ID_foo]**. For an input, use **vrgInputs[ID_foo]**. For the calculated derivative of a state variable, use **rgDerivs**[**ID_foo**]. For a dynamic variable or a global parameter, use **foo**.

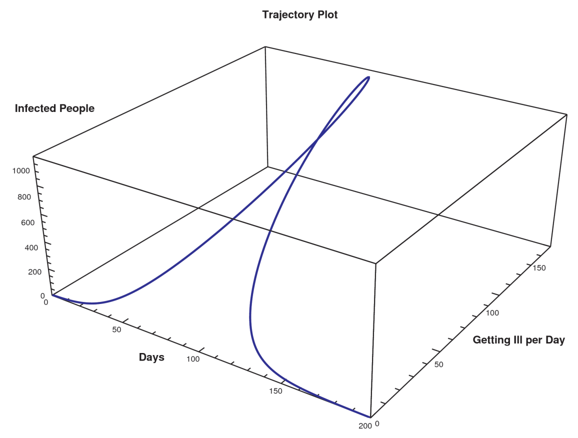
J

The following script will load an MCSim output file containing multiple experiments if each experiment has the same outputs. Note that J commands can only be one line long, so you must type what are shown as indented continuation lines all on one line.

```
require 'convert csv misc plot'
require 'statfns strings'
data =: 0 & ". @: detab"1
simcount =: +/@:(,@:-.)@:(-:"0
  1&0)@:('Simulation'&(I.@E.)"1@:>)
vc0 =: 2 & {
vc1 =: TAB & chop @: >
varcount =: <: @: # @: vc1 @: vc0
datalength =: (<:^:5) @: (# %
  simcount)
rs =: 3 : 0
d =: readcsv y
sc =: simcount d
vc =: varcount d
dl =: datalength d
dselect =. (sc * dl + 5) $ (3#0 ),
  (dl # 1), 2 # 0
D1=. data > dselect # d
(sc, dl, vc+1)& $ @: ,D1
)
gd =: 4 : 0
'i j' =. x
(i & {"1) @: (j & {) y
)
vn =: 3 : 0
baseindex =. +/\ sc # dl + 5
offsetindex =. (dl + 3) -~
  baseindex
(chop ": i. >: vc), ,"2 ;: >
   offsetindex { d
)
```

Run the script, and then use the following commands to analyze the data:

```
D=:rs fselect'' NB. data => D
vn''            NB. variable names
(1 2) gd D  NB. get data for the
  1st variable in the 2nd experiment
plot ((0 0)&gd;(1 2)&gd) D NB.
  plot that data vs. time
plot ((0 0)&gd;(2 4)&gd;(1 4)&gd)D
  NB. trajectory plot: 2nd and 1st
  variables, 4th experiment vs .time
cpd=: 3 1 5 5 1 [ ad=: 4 1 7 1 7
NB. The left-most 1 in 1&{"1 below
  selects the column for the
  variable to plot, starting with 0
  for the time column.
'type stick;title "Max Infected";
  pensize 7' plot cpd;ad;(max"1)
  1&{"1 D NB. peak infection across
  all five experiments
'type stick;title "Peak Time";
  pensize 7' plot cpd;ad;({."1
  (<<1){d){~(](i."1 0)(max"1))
  1&{"1 d=.D NB. time of peak
  infection
```



Software

Programs:

MCSim (5.1.beta or later)
bash, gcc, make, GNU Scientific Library
man, texinfo
Gnuplot
J
Emacs or Vim
RCS, CVS, or Subversion
Dia; OpenOffice.org, LaTeX, or DocBook

Windows Sources:

http://cygwin.com/
http://gnu.org/software/emacs/windows/ntemacs.html
http://www.componentsoftware.com/csrcs/
http://miktex.org/
http://e-novative.info/software/ede.php

Mac OS X Sources:

http://fink.sourceforge.net/

Linux Sources:

```
$ apt-get ...
```

Common Sources:

http://fredomatic.free.fr/
http://gnuplot.info/
http://jsoftware.com/
http://gnu.org/software/emacs/
http://vim.org/
http://www.gnome.org/projects/dia/
http://openoffice.org/
http://ctan.org/
http://wiki.docbook.org/

References

Bois, Frédéric Y. and Don R. Maszle. 2004. *MCSim: A Monte Carlo Simulation Program.* User's Manual, software version 5.0.0.

Sterman, John D. 2000. *Business Dynamics: Systems Thinking and Modeling for a Complex World.* Boston: McGraw-Hill.

Making Sense With Facilitated Systems
http://facilitatedsystems.com/weblog/