

Solving and Fitting France's Rumen Model with MLAB

Gary Knott, Ph.D. and Daniel Kerner, Ph.D.

Civilized Software Inc.
12109 Heritage Park Circle
Silver Spring MD 20906, USA
Tel: (301) 962-3711
Email: csi@civilized.com
URL: <http://www.civilized.com>

Ruminants (cattle, sheep, etc.) “ferment” their ingested food in their “first stomach”, called the *rumen*. The food is there partially digested by mastication and microbial action. As a result of this “pre-processing”, the animal is able to extract more nutrient substances from its food than would otherwise be the case.

A mathematical description for the action of the rumen was given in “A Mathematical Model of the Rumen” by J. France, J. H. M. Thornley, and D. E. Beaver, published in the Cambridge Journal of Agricultural Science, Vol. 99, pp. 343-353, 1982.

This model treats the rumen as a fermentation tank into which water and feed consisting of carbohydrates and proteins “bound” within plant cells flow. A microbial mass in the rumen then enzymatically “extracts” these inputs into various amounts of unencapsulated proteins, sugars, and other carbohydrates which are thereby made available for absorption and utilization by the animal. The microbial mass also utilizes some of the materials to support growth and enzymatic action. The model consists of 9 differential equations that describes the input, processing and output of the various materials of interest. These differential equations involve 12 parameters, and various other input and output rate functions, that must be somehow assigned values in order to establish a definite model.

France et al. originally solved their model differential equations using the CSMP (Continuous System Modeling Program) software employing the first-order Euler method. By manually solving the equations many times with

different parameter values, they essentially curve-fit their model to data by choosing parameter values by trial and error.

We will show the use of the MLAB Mathematical and Statistical Modeling System in both solving and fitting France's model.

France et al. modeled a sheep eating either chopped or pelleted Italian ryegrass, or timothy. They also studied continuous input (obviously an approximation to reality), and "pulsed" input where feeding occurs intermittently. Here we will study only chopped Italian ryegrass.

The inputs consist of:

1. water from drinking and from saliva
2. water-soluble carbohydrates (sugars, and short-chain polymers)
3. α -hexose (pectin and α -linked polymers)
4. rumen-degradable β -hexose (cellulose and lignin)
5. non-rumen-degradable β -hexose
6. rumen-degradable protein (includes nucleic acids)
7. non-rumen-degradable protein
8. non-protein nitrogen (amino acids, urea, ammonia, etc.)
9. microbial mass

Inputs 2-8 are contained in the Italian ryegrass feed.

The outputs consist of these same materials as extracted from the feed, minus the amounts utilized and/or synthesized by the microbial content of the rumen.

The fermentation tank rumen model requires a differential equation that defines the volume of water in the rumen (measured in *meters*³ (1 *meter*³ = 10³ *liters*). This volume changes as water is taken in and out mixed with various nutrients. The nutrient materials themselves are measured in concentration units of *kilograms/meters*³, and these concentration values change, both as the amount of materials change *and* the amount of water present changes. There is also a differential equation defining the growth and destruction of the microbial population. Thus, altogether, each material listed above has an associated mass-balance differential equation that define the following functions of time. The derivation of these equations is well-presented in France et al.

The mathematical functions of the model are:

$V(t)$ = volume of water from drinking and from saliva (m^3)

$Bn(t)$ = non-rumen-degradable β -hexose (kg/m^3)

$Br(t)$ = rumen-degradable β -hexose (kg/m^3)
 $A(t)$ = α -hexose (kg/m^3)
 $C(t)$ = water-soluble carbohydrates (kg/m^3)
 $N(t)$ = non-protein nitrogen (kg/m^3)
 $Pr(t)$ = rumen-degradable protein (kg/m^3)
 $Pn(t)$ = non-rumen-degradable protein (kg/m^3)
 $M(t)$ = microbial population concentration (kg/m^3)

The MLAB do-file that defines France's rumen model, solves the differential equations, and graphs them for comparison to France's results and to the data given in France's paper, is presented below.

```

/* file: rumen.do                                3/26/00 */
/* This file sets up and solves the compartmental model of the
   rumen described in J.France, et.al, J.Agric.Sci.Camb. 99
   (1982) 343-353.
  
```

The state functions are:

```

V(t) = rumen water volume (m^3) (10^6 cm^3 = 1 m^3 = 10^3 liters)
Bn(t) = non-rumen degradable beta hexose (kg/m^3)
Br(t) = rumen degradable beta hexose (kg/m^3)
A(t) = alpha hexose (kg/m^3)
C(t) = water soluble carbohydrates (kg/m^3)
N(t) = non-protein nitrogen (kg/m^3)
Pr(t) = rumen degradable protein (kg/m^3)
Pn(t) = non-rumen degradable protein (kg/m^3)
M(t) = microbial population concentration (kg/m^3)
  
```

The values of the constants and input/output rates are taken from France et al. for sheep on a continuous diet of chopped Italian ryegrass.

```

*/
reset
echodo = 3

/* define first order differential equations for state functions */
fct A't(t) = (DA(t)-A(t)*Dv(t)-A(t)*Sv(t))/V(t)-kam*A(t)*M(t)
fct Br't(t) = (DBr(t)-Br(t)*Dv(t)-Br(t)*Sv(t))/V(t)-kbrm*Br(t)*M(t)
fct Bn't(t) = (DBn(t)-Bn(t)*Dv(t)-Bn(t)*Sv(t))/V(t)
fct Pn't(t) = (DPn(t)-Pn(t)*Dv(t)-Pn(t)*Sv(t))/V(t)
  
```

```

fct Pr't(t) = (DPr(t)+SPr(t)-Pr(t)*Dv(t)-Pr(t)*Sv(t))/V(t)-kprm*Pr(t)*M(t)
fct C't(t) = (DC(t)-C(t)*Dv(t)-C(t)*Sv(t))/V(t)+(L(t)-mu(t)/YM)*fC*M(t)+ \
      kam*A(t)*M(t)+kbrm*Br(t)*M(t)
fct N't(t) = (DN(t)+SN(t)-N(t)*Dv(t)-N(t)*Sv(t))/V(t)+(L(t)-mu(t))*fN*M(t)+ \
      +kprm*Pr(t)*M(t)

/* ODE for the volume of the rumen */
fct V't(t) = Dv(t)+Sv(t)-vr(t)

/* ODE for the microbial mass */
fct M't(t) = mu(t)*M(t)-L(t)*M(t)-M(t)*(Dv(t)+Sv(t))/V(t)

/* model for the specific daily growth rate of microbes */
fct mu(t) = mum/(1+(kc/C(t))+(kn/N(t))+(kcn/(C(t)*N(t))))

/* model for the daily catabolism(destruction) rate for microbes */
fct L(t) = Lm/(1+kmu*mu(t))

/* initial conditions, all in units of kg/m^3, except V is in m^3 units. */
init V(0) = .005
init Bn(0) = 36
init Pn(0) = 4
init A(0) = 5
init Br(0) = 30
init Pr(0) = 1.5
init C(0) = 7
init N(0) = 2
init M(0) = 26.5

/* constants */
kam = 1.5 /* m^3/kg M per day */
kbrm = 2.0 /* m^3/kg M per day */
kprm = 2.0 /* m^3/kg M per day */
mum = 5.0 /* per day */
kn = .2
kc = .2
kcn = 0.
Lm = .2 /* per day */
kmu = 5.0 /* per day */
fC = .3

```

```

fN = .5
YM = .1

/* input functions for continuous diet model */
fct DBn(t) = .0673 /* kg/day */
fct DPn(t) = .0243
fct DA(t) = .0266
fct DBr(t) = .4937
fct DPr(t) = .0729
fct DC(t) = .215
fct DN(t) = .0422
fct SN(t) = .0061
fct SPr(t) = .0121

/* input/output flow rate functions. all constant, in m3/day units.
   Note the input and output balances. */
fct vr(t) = .0163 /* total water output from rumen */
fct Dv(t) = .0042 /* water input from drinking */
fct Sv(t) = .0121 /* water input from saliva */

/* Integrate from 0 to tmax days, reporting at np points in the
   interval. 3 days is sufficient time to reach equilibrium (within
   1 percent or so of the stationary point. */
tmax=3; np=400
t = 0:tmax!np
Res = points(V,Bn,Pn,A,Br,Pr,C,N,M,t);
/* Res col 1 = t, Res col 2:10 = [V,Bn,Pn,A,Br,Pr,C,N,M] values */

F = (Res row np)' /* F = steady-state values to be used later.*/

/* Multiply concentrations by outflow rate value vr(0) to obtain output
   per day amounts in kilograms */
Res col (3:10) = (Res col (3:10)) * vr(0)

/* data from J.France paper for observed outputs from chopped
   Italian ryegrass in kg/m3 per day units. To be used for
   graphics output. */
ddb row 1 = list(tmax-.6,.1121)'
dda row 1 = list(tmax-.6,.0027)'
ddc row 1 = list(tmax-.6,.005)'

```

```

ddp row 1 = list(tmax-.6,.0365)'
ddm row 1 = list(tmax-.6,.1756)'
ddn row 1 = list(tmax-.6,.0495)'

draw Res col (1,5) lt dashed      /* A = Alpha hexose */
title "A" at (3.1,Res[np,5]) world
draw dda pt circle ptsize .025 ffract
draw dda pt "A" ptsize .017 ffract

draw Res col (1,8) lt ldash      /* C = carbohydrate */
title "C" at (3.1,Res[np,8]) world, place (left,top)
draw ddc pt circle ptsize .025 ffract
draw ddc pt "C" ptsize .017 ffract

/* Res col 7 = digestable protein + non-digestable protein */
Res col 7 = (Res col 7)+(Res col 4)
draw Res col (1,7) lt solid      /* Pr+Pn = total protein */
title "P" at (3.1,Res[np,7]) world
draw ddp pt circle ptsize .025 ffract
draw ddp pt "P" ptsize .017 ffract

draw Res col (1,9) lt ddash      /* N = non-protein nitrogen */
title "N" at (3.1,Res[np,9]) world, place (left,bottom)
draw ddn pt circle ptsize .025 ffract
draw ddn pt "N" ptsize .017 ffract

window 0 to tmax, 0 to .05
bottom title "days"
left title "kg/day"
frame 0 to 1, .5 to 1
w1 = w

draw Res col (1,10) lt solid     /* M = microbial concentration */
title "M" at (3.1,Res[np,10]) world
draw ddm pt circle ptsize .025 ffract
draw ddm pt "M" ptsize .017 ffract

/* Res col 6 = digestable and non-digestable beta hexose */
Res col 6 = (Res col 6)+(Res col 3)
draw Res col (1,6) lt dashed     /* Br+Bn = total Beta hexose */

```

```

title "B" at (3.1,Res[np,6]) world
draw ddb pt circle ptsize .025 ffract
draw ddb pt "B" ptsize .017 ffract

```

```

bottom title "days"
left title "kg/day"
frame 0 to 1, 0 to .5
window 0 to tmax, 0 to 1.1

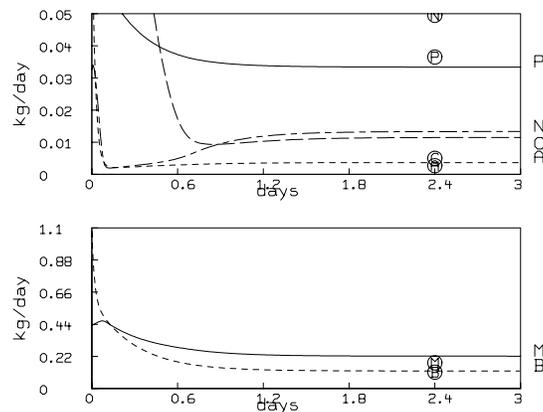
```

```

view
unview
delete w,w1

```

The graphs produced above via MLAB are as follows. Note the observed steady-state output values reported in France et al. are drawn as single labeled points, with the labels enclosed in circles. Except for non-protein nitrogen, N, France's parameter values cause the model to fit the data reasonably well.



Now we may introduce various step-functions for the inputs in order to model intermittent feeding and drinking following the regimes described by France et al.

The MLAB statements shown below redefine the model and solve it for such "pulsed" inputs. They may be provided as a continuation of the do-file given above.

```

/* use the steady state values computed earlier above for the initial
conditions */

/* initial conditions, all in units of kg/m^3 */
init V(0) = F[2]
init Bn(0) = F[3]
init Pn(0) = F[4]
init A(0) = F[5]
init Br(0) = F[6]
init Pr(0) = F[7]
init C(0) = F[8]
init N(0) = F[9]
init M(0) = F[10]

/* h(t) is a smoothly transitioning sigmoidal curve representing
a smoothed step function. Using h in preference to a pure
step-function avoids integration warnings due to discontinuities.*/
fct h(t,b) = .5*(1+sign(t)*(1-exp(-b*abs(t))))

/* pf(t) is the feeding pulse function */
fct pf(t) = sum(i,1,nf,h(mod(t,1)-ts[i],100)-h(mod(t,1)-ts[i]-df,100))/(nf*df)
nf = 2
df = .1
ts[1] = .4; ts[2] = .7

/* ps(t) is the salivary flow pulse function */
fct ps(t) = (1+sum(i,1,nf,h(mod(t,1)-ts[i],100)-h(mod(t,1)-ts[i]-2*df,100)))\
/(1+2*nf*df)

/* pd(t) is the drinking pulse function */
fct pd(t) = sum(i,1,nd,h(mod(t,1)-tr[i],200)-h(mod(t,1)-tr[i]-dd,200))/(nd*dd)
nd = 4
dd = .01
tr[1] = .01; tr[2] = .31; tr[3] = .51; tr[4] = .81

/* redefine the input functions to incorporate pulse variation */
fct DBn(t) = .0673*pf(t)
fct DPn(t) = .0243*pf(t)
fct DA(t) = .0266*pf(t)
fct DBr(t) = .4937*pf(t)

```

```

fct DPr(t) = .0729*pf(t)
fct DC(t) = .215*pf(t)
fct DN(t) = .0422*pf(t)

fct SN(t) = .0061*ps(t)
fct SPr(t) = .0121*ps(t)

/* redefine fluid input flow rate functions */
fct Dv(t) = .0042*pd(t)
fct Sv(t) = .0121*ps(t)

/* integrate from 0 to tmax days reporting at np points in the interval */
jacs w = 0
method = gear2
disasters w = -2
Res = points(V,Bn,Pn,A,Br,Pr,C,N,M,t);
/* Res col 1 = t, Res col 2:10 = [V,Bn,Pn,A,Br,Pr,C,N,M] values */

/* multiply concentrations by outflow rate value to obtain output
per day amounts in kilograms */
Res col (3:10) = (Res col (3:10)) * vr(0)

draw Res col (1,5) lt dashed /* A = Alpha hexose */
title "A" at (3.1,Res[np,5]) world

draw Res col (1,8) lt ldash /* C = carbohydrate */
title "C" at (3.1,Res[np,8]) world, place (left,top)

/* Res col 7 = digestable protein + non-digestable protein */
Res col 7 = (Res col 7)+(Res col 4)
draw Res col (1,7) lt solid /* Pr+Pn = total protein */
title "P" at (3.1,Res[np,7]) world

draw Res col (1,9) lt ddash /* N = non-protein nitrogen */
title "N" at (3.1,Res[np,9]) world, place (left,bottom)

bottom title "days"
left title "kg/day"
frame 0 to 1, .5 to 1
w1 = w

```

```

draw Res col (1,10) lt solid      /* M = microbial concentration */
title "M" at (3.1,Res[np,10]) world

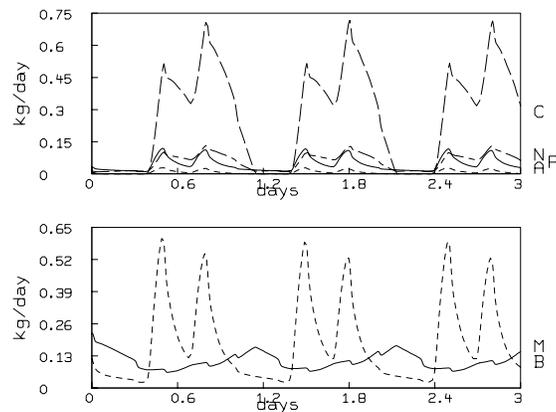
/* Res col 6 = digestable and non-digestable beta hexose */
Res col 6 = (Res col 6)+(Res col 3)
draw Res col (1,6) lt dashed     /* Br+Bn = total Beta hexose */
title "B" at (3.1,Res[np,6]) world

bottom title "days"
left title "kg/day"
frame 0 to 1, 0 to .5

view
unview

```

The results for pulsed feeding and drinking are shown below.



Finally, we may use the MLAB system to try to adjust some of the parameters in France's model to more closely fit the observed steady-state data reported in France et al. This fitting was done with a do-file that defined the model, adjusted the parameters and graphed the resulting fit. The output log-file from this computation is shown below; it incorporates the numerical fitting results.

MLAB Mathematical Modeling System, Revision: January 12, 1999

Copyright: Civilized Software, Inc. (301)962-3711, email: csi@civilized.com
Web-site: WWW.CIVILIZED.COM

Thu Apr 13 18:00:10 2000

'* ' is the command prompt

```
* casesw = 1
*
* /* define first order differential equations for state variables */
* fct A't(t) = (DA(t)-A(t)*Dv(t)-A(t)*Sv(t))/V(t)-kam*A(t)*M(t)
* fct Br't(t) = (DBr(t)-Br(t)*Dv(t)-Br(t)*Sv(t))/V(t)-kbrm*Br(t)*M(t)
* fct Bn't(t) = (DBn(t)-Bn(t)*Dv(t)-Bn(t)*Sv(t))/V(t)
* fct Pn't(t) = (DPn(t)-Pn(t)*Dv(t)-Pn(t)*Sv(t))/V(t)
* fct Pr't(t) = (DPr(t)+SPr(t)-Pr(t)*Dv(t)-Pr(t)*Sv(t))/V(t)-kprm*Pr(t)*M(t)
* fct C't(t) = (DC(t)-C(t)*Dv(t)-C(t)*Sv(t))/V(t)+(L(t)-mu(t)/YM)*fC*M(t)+ \
: kam*A(t)*M(t)+kbrm*Br(t)*M(t)
* fct N't(t) = (DN(t)+SN(t)-N(t)*Dv(t)-N(t)*Sv(t))/V(t)+(L(t)-mu(t))*fN*M(t)+ \
: +kprm*Pr(t)*M(t)
*
* /* model for the volume of the rumen */
* fct V't(t) = Dv(t)+Sv(t)-vr(t)
*
* /* model for the microbial mass */
* fct M't(t) = mu(t)*M(t)-L(t)*M(t)-M(t)*(Dv(t)+Sv(t))/V(t)
*
* /* model for the specific growth rate of microbes per day */
* fct mu(t) = mum/(1+(kc/C(t))+(kn/N(t))+(kcn/(C(t)*N(t))))
*
* /* model for catabolism rate of microbes per day */
* fct L(t) = Lm/(1+kmu*mu(t))
*
* /* initial conditions, all in units of kg/m^3 */
* init V(0) = .005
* init Bn(0) = 36
* init Pn(0) = 4
* init A(0) = 5
* init Br(0) = 30
* init Pr(0) = 1.5
* init C(0) = 7
* init N(0) = 2
```

```

* init M(0) = 26.5
*
* /* constants */
* kam = 1.5 /* m^3/kg M per day */
* kbrm = 2.0 /* m^3/kg M per day */
* kprm = 2.0 /* m^3/kg M per day */
* mum = 5.0 /* per day */
* kn = .2
* kc = .2
* kcn = 0.
* Lm = .2 /* per day */
* kmu = 5.0 /* per day */
* fC = .3
* fN = .5
* YM = .1
*
* /* input functions for continuous diet model */
* fct DBn(t) = .0673 /* kg/day */
* fct DPn(t) = .0243
* fct DA(t) = .0266
* fct DBr(t) = .4937
* fct DPr(t) = .0729
* fct DC(t) = .215
* fct DN(t) = .0422
* fct SN(t) = .0061
* fct SPr(t) = .0121
*
* /* output flow rate function */
* fct vr(t) = .0163
* fct Dv(t) = .0042 /* m^3/day */
* fct Sv(t) = .0121
*
* /* try fitting the functions tb, tp, M, N, C, and A to improve
   the match to the experimental data */
* fct tb(t) = Bn(t)+Br(t)
* fct tp(t) = Pr(t)+Pn(t)
*
* /* data from J.France paper for chopped Italian ryegrass (converted
   to concentration units kg/m^3 for day 5). */
* ddb row 1 = list(5,.1121/vr(5))'

```

```

* dda row 1 = list(5,.0027/vr(5))'
* ddc row 1 = list(5,.005/vr(5))'
* ddp row 1 = list(5,.0365/vr(5))'
* ddm row 1 = list(5,.1756/vr(5))'
* ddn row 1 = list(5,.0495/vr(5))'
*
* jacsw = 0
* lsqrpt = 15
* constraints q = {0 < kam, kam < 5,\
: 0 < kbrm, kbrm < 5,\
: 0 < kprm, kprm < 5,\
: 0 < mum, mum < 10,\
: 0.05 < kn, kn < 1,\
: 0 < kc, kc < 1,\
: 0.002 < kcn, kcn < .5,\
: 0 < Lm, Lm < 1,\
: 0 < kmu, kmu < 10,\
: 0 < YM, YM < .5}
*
* fit (kam,kbrm,kprm,mum,kn,kc,kcn,Lm,km,YM), tb to ddb, tp to ddp, \
: M to ddm, N to ddn, C to ddc, A to dda, constraints q
final parameter values
      value          error      dependency  parameter
2.7347008215      12.1729267511      0.0887696057  kam
3.0956270498       1.2347624716      0.6364731937  kbrm
  4.943072645       8.3999729901      0.0837336262  kprm
8.4375657657     186030.7204326921      0.9999999999  mum
0.7277384723     113630.94347594      0.9999999999  kn
0.0364426711     339101.834692397           1  kc
0.4693825109     877671.267524227           1  kcn
          1         2406.0068217221      0.9999999899  Lm
    0.003719028     1407.8269877652      0.9999999983  kmu
    0.0961969358     157.4025002796      0.9999999999  YM
3 iterations
CONVERGED
best weighted sum of squares = 4.390074e-01
weighted root mean square error = 6.625764e-01
weighted deviation fraction = 2.651032e-02
lagrange multiplier[16] = 6.158843348e-07
*

```

```

* t = 0:6!160; np=160
* Res = points(V,Bn,Pn,A,Br,Pr,C,N,M,t);
*
* /* multiply concentrations by outflow rate value to obtain output
* per day amounts in kilograms */
* Res col (3:10) = (Res col (3:10)) * vr(0)

* /* data from J.France paper for observed outputs from chopped
* Italian ryegrass in kg/m^3 per day units. Used to fit above.
* Also to be used for graphics output. */
* ddb row 1 = list(5,.1121)'
* dda row 1 = list(5,.0027)'
* ddc row 1 = list(5,.005)'
* ddp row 1 = list(5,.0365)'
* ddm row 1 = list(5,.1756)'
* ddn row 1 = list(5,.0495)'

* draw Res col (1,5) lt dashed      /* A = Alpha hexose */
* title "A" at (6.2,Res[np,5]) world
* draw dda pt circle ptsize .025 ffract
* draw dda pt "A" ptsize .017 ffract

* draw Res col (1,8) lt ldash      /* C = carbohydrate */
* title "C" at (6.2,Res[np,8]) world, place (left,top)
* draw ddc pt circle ptsize .025 ffract
* draw ddc pt "C" ptsize .017 ffract

* /* Res col 7 = digestable protein + non-digestable protein */
* Res col 7 = (Res col 7)+(Res col 4)
* draw Res col (1,7) lt solid      /* Pr+Pn = total protein */
* title "P" at (6.2,Res[np,7]) world
* draw ddp pt circle ptsize .025 ffract
* draw ddp pt "P" ptsize .017 ffract

* draw Res col (1,9) lt ddash      /* N = non-protein nitrogen */
* title "N" at (6.2,Res[np,9]) world, place (left,bottom)
* draw ddn pt circle ptsize .025 ffract
* draw ddn pt "N" ptsize .017 ffract

* window 0 to 6, 0 to .05

```

```

* bottom title "days"
* left title "kg/day"
* window 0 to 6, 0 to .1
* frame 0 to 1, .5 to 1
* w1 = w

* draw Res col (1,10) lt solid          /* M = microbial concentration */
* title "M" at (6.2,Res[np,10]) world
* draw ddm pt circle ptsize .025 ffract
* draw ddm pt "M" ptsize .017 ffract

* /* Res col 6 = digestable and non-digestable beta hexose */
* Res col 6 = (Res col 6)+(Res col 3)
* draw Res col (1,6) lt dashed          /* Br+Bn = total Beta hexose */
* title "B" at (6.2,Res[np,6]) world
* draw ddb pt circle ptsize .025 ffract
* draw ddb pt "B" ptsize .017 ffract

* bottom title "days"
* left title "kg/day"
* frame 0 to 1, 0 to .5
* window 0 to 6, 0 to .5
* view
* unview
*
* /* end of file */
*
* exit
end of MLAB.LOG

```

Here is the graphical output showing the curve-fit results. All the steady-state data-points given in France et al. are fit reasonably well, and these parameters may be preferred over those proposed in France et al. By adjusting other parameters, it is possible that we could improve this fit even more.

