Minimal Models for Glucose and Insulin Kinetics

Daniel R. Kerner, PhD.
Civilized Software Inc.
12109 Heritage Park Circle
Silver Spring MD 20906
phone:301-962-3711
email:knott@civilized.com
URL:www.civilized.com

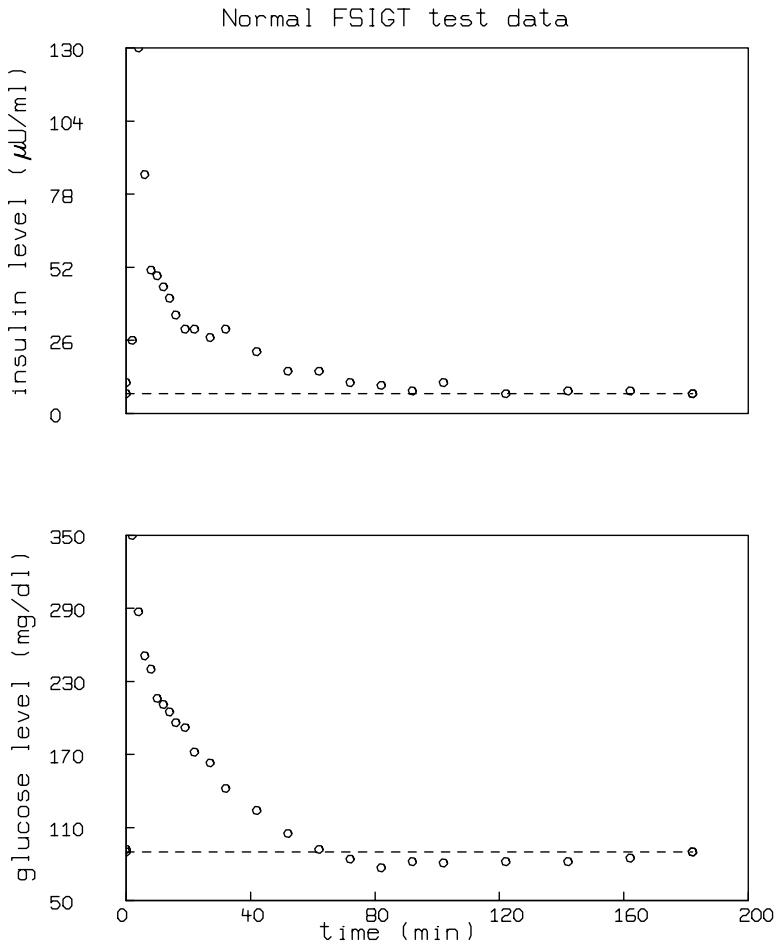
February 6, 2021

0.1 Minimal Models for Insulin-Glucose Interaction

Mathematical models describing glucose and insulin blood levels have been developed and used by Dr. Richard N. Bergman and co-workers since the 1970's. These models are designed to "fit" frequently-sampled intravenous glucose tolerance data (FSIGT) in humans (and other species, e.g. dogs). (See references 1-5.) The basic *blood glucose model* consists of a pair of coupled ordinary differential equations that describe the decay back to "equilibrium" of blood glucose following an injection of glucose. The blood insulin level curve is described either by an empirical curve based on measurements starting at peak blood insulin level and decaying, or by an additional auxillary differential equation describing this decay; in this latter case, we have a combined blood glucose, blood insulin model. The blood insulin differential equation by itself (with a numerically-defined blood glucose function) is called the *blood insulin model*. Thus we have three models: the blood glucose model, the blood insulin model, and the combined blood glucose-insulin model.

These models are called "minimal models" by their authors, since they are very simple "phenomenological" models. None of these minimal models are physiologically realistic; they are purely descriptive. They do not incorporate any of the knowledge about the hormones and biochemical pathways involved in glucose-insulin interaction and the metabolic uptake of glucose thus enabled. Nevertheless, these models can be valuable as a descriptive research tool and also as a diagnostic tool, where the characterization of observations is desired, as opposed to a description of mechanisms.

In a typical FSIGT test, blood samples are taken from a fasting subject at regular intervals of time, following a single intravenous injection of glucose. (Although samples irregularly spaced in time can be accomodated as well.) The blood samples are then analyzed for glucose and insulin content. The figure below shows a typical response from a normal subject.



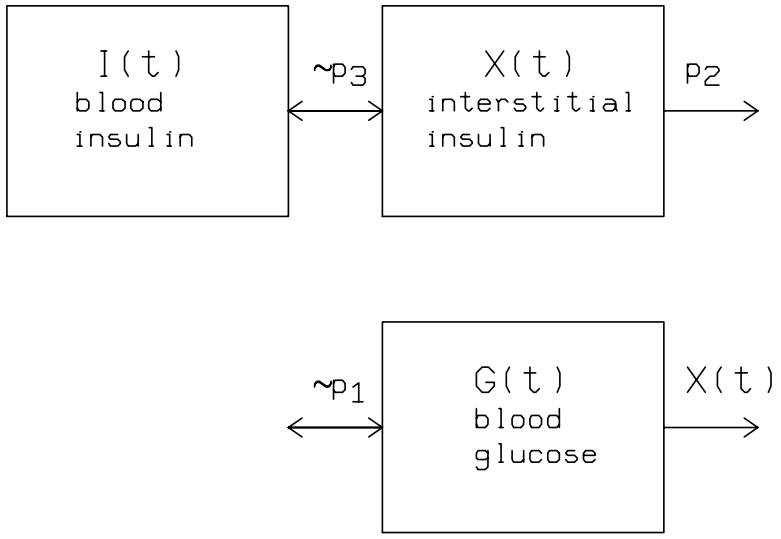
The situation being modeled starts shortly after an injection of glucose. We ignore the fast initial rise of glucose and suppose the glucose level in blood *starts* at an elevated level occuring shortly *after* the injection, and then generally drops to a minimum which is usually below the basal (pre-injection) glucose level; then the glucose level gradually rises back to the basal level as glucose is produced in the liver.

Depending on the state of the subject, there can be wide variations in these responses; for example, the glucose level may not drop below basal level, the first peak in insulin level may have different amplitudes, there may be no secondary peak in insulin level, or there may be more than two peaks in insulin level.

The glucose and insulin minimal models provide a quantitative and parsimonious description of the glucose and insulin concentrations in the blood following the glucose injection. The glucose minimal model involves two physiologic "compartments": a blood compartment and an interstitial tissue compartment; the insulin minimal model involves only the blood compartment. In the blood compartment, functions G and I giving the blood glucose level and blood insulin level in the compartment are defined, and in the interstitial compartment, a function X defining the interstitial insulin level is defined.

Similarly, whether numerically-defined or not, the blood insulin level function is also a decaying function – it does not describe the initial rise in insulin due to the glucose injection.

The following diagram summarizes the minimal model for glucose kinetics:



We write $\sim p_3$ and $\sim p_1$ to indicate that these parameters are multiplied by terms which may "switch sign" to effect "bi-directional flow".

We assume insulin enters the interstitial tissue compartment from the blood compartment, or leaves the interstitial tissue compartment and goes into the blood compartment, at a rate proportional to the difference between the blood insulin level, $I(t)$, and the basal blood insulin level, I_b; if the blood insulin level falls below the basal level, insulin leaves the interstitial tissue compartment, and if the blood insulin level rises above the basal level, insulin enters the interstitial tissue compartment. Insulin also disappears from the interstitial tissue compartment via a second pathway at a rate proportional to the amount of insulin in the interstitial tissue compartment; this represents the insulin taken up in cells and not returned to the interstitial compartment.

Thus with $X(t)$ denoting the interstitial insulin concentration at time t, we have the rate of change of X, $\frac{dX(t)}{dt}$, given by:

$$\frac{dX(t)}{dt} = p_3 \cdot (I(t) - I_b) - p_2 \cdot X(t)$$

Here t is time, $X(t)$ is the interstitial insulin concentration at time t, $I(t)$ is the blood insulin concentration at time t, and I_b is the basal blood insulin concentration. We assume $X(0) = 0$ at the start of the response to the challenge glucose injection.

The non-negative rate parameters p_3 and p_2 are the proportionality constants of the flow of insulin from or to the blood compartment and from the interstitial compartment into cells. Note when $I(t) - I_b$ is negative, the term $p_3 \cdot (I(t) - I_b)$ is negative and insulin flows back into the blood compartment.

Let $G(t)$ denote the blood glucose concentration at time t. We assume glucose leaves or enters the blood compartment at a rate proportional to the *difference* between the blood glucose level, $G(t)$, and the basal blood glucose level, G_b; if the blood glucose level falls below the basal level, glucose *enters* the blood compartment, and if the glucose level rises above the basal level, glucose *leaves* the blood compartment. The source or destination of this glucose is unspecified; there is no attempt in this model to account for the distribution of "mass" among compartments.

Glucose also disappears from the blood compartment via a second pathway at a rate proportional to the amount of insulin in the interstitial tissue. This pathway represents the uptake and utilization of glucose in cells enabled by insulin there.

Thus we have:

$$\frac{dG(t)}{dt} = p_1 \cdot (G_b - G(t)) - X(t) \cdot G(t)$$

Here t is time, $G(t)$ is the blood glucose concentration at time t, and G_b is the basal blood glucose concentration. We assume $G(0) = G_0$, where G_0 is the blood glucose concentration at the peak level obtained after the challenge glucose injection; G_0 is generally determined directly from the measured data, although it can instead be a fitting parameter. As before the function $X(t)$ is the interstitial insulin concentration at time t.

The non-negative rate parameter p_1 is the proportionality constant of the flow of glucose from or to the blood compartment. Note when $G_b - G(t)$ is negative, the term $p_1 \cdot (G_b - G(t))$ is negative and glucose flows back into the blood compartment (presumably from the liver).

Basal blood concentrations of glucose and insulin are typically measured either before, or 180 minutes after, administration of glucose; these measurements determine the values of G_b and I_b. There are three unknown parameters in this model: p_1, p_2, and p_3, or four unknown parameters if

G_0 is not determined from the data, but is treated as an unknown parameter to be estimated, just like p_1, p_2, and p_3.

The blood concentration of insulin in the glucose model as described by the function $I(t)$ is generally given by a data table, *i.e.*, $I(t)$ is empirically-defined and computed with interpolation in a table of observation values. (We can also posit a differential equation model for $I(t)$, *i.e.*, we can employ the minimal insulin model in combination with the glucose model as mentioned before.)

The glucose minimal model thus consists of two ordinary differential equations defining the functions $G(t)$ and $X(t)$, and involving the auxillary function $I(t)$, given either by numerical data, or by a third differential equation.

Note that in the differential equation for G, glucose is utilized at the constant rate p_1, when we neglect *feedback* effects due to the "flow" of interstitial insulin as represented by the term $-X(t) \cdot G(t)$. However, an additional amount of blood insulin will cause the amount of interstitial insulin to change, which in turn, will cause the rate of glucose utilization to change.

Fitting the glucose minimal model to FSIGT test data gives us estimates of the parameters p_1, p_2, and p_3, and we can then compute two metabolic indices:

- S_I = insulin sensitivity: the dependence of fractional glucose disappearance on blood insulin level, (S_I is given in the unit: (milliliter minute / micro-insulin unit)).
- S_G = glucose effectiveness: the fractional ability for blood glucose concentration to decrease independent of increased insulin, (S_G is given in the unit: 1/minutes);

The *insulin sensitivity* is defined as $S_I = p_3/p_2$ and the *glucose effectiveness* is defined as $S_G = p_1$. Glucose effectiveness measures how rapidly glucose leaves the blood, and insulin sensitivity measures how the utilization of glucose increases in response to changes in the blood insulin level.

Fitting the Glucose Minimal Model with MLAB

Now we will demonstrate the use of the mathematical modeling computer program MLAB to fit the glucose minimal model described above to observed insulin and glucose levels in blood resulting from a FSIGT test and use the obtained parameters to determine values of the metabolic indices for the given data.

Bergman, et. al., [3] provide the following FSIGT test data (also shown in the graphs above) from a normal individual:

time (minutes)	glucose level (mg/dl)	insulin level (μU/ml)
0	92	11
2	350	26
4	287	130
6	251	85
8	240	51
10	216	49
12	211	45
14	205	41
16	196	35
19	192	30
22	172	30
27	163	27
32	142	30
42	124	22
52	105	15
62	92	15
72	84	11
82	77	10
92	82	8
102	81	11
122	82	7
142	82	8
162	85	8
182	90	7

Using MLAB, or a spreadsheet program, or a text-editor program, (including the editor invoked by the MLAB "edit file" command), these numbers can be entered and stored in an ASCII text file named `"minmod1.dat"`, and we assume this is the case. (Note the sampling times are not uniformly spaced here.)

We want to fit the glucose model discussed above to this data to estimate the unknown parameters p_1, p_2, p_3, and later to improve our estimates of G_0 and G_b.

We will fit the glucose minimal model to our data, using the given time-course blood insulin data to define the empirical insulin function I with linear interpolation. In this case we will estimate the unknown parameters p_1, p_2, p_3, taking G_0 to be the maximum value seen in the data and taking G_b and I_b to be the basal levels seen in the data; and then we will fit the glucose minimal model to our data again, to estimate the unknown parameters p_1, p_2, p_3, G_0, and G_b, using the maximum value seen in the data as our initial guess for G_0, and the basal glucose level seen in the data as the initial guess for G_b. We shall see that for the given data, there is not much difference in the data-defined values of G_0 and G_b and the values estimated by curve-fitting.

The following MLAB commands compute estimate values for the parameters p_1, p_2, p_3, and G_0 given the time course of blood glucose and insulin. The values of the parameters found minimize, in the least squares sense, the weighted difference between the measured time course of blood

glucose and the parameter-dependent solution to the glucose minimal model differential equations. The blood insulin concentration function $I(t)$ is obtained by linear interpolation of the time-insulin values listed in `"minmod1.dat"`. This is done by employing the MLAB linear-interpolation function `"LOOKUP"` where `LOOKUP(idat, t)` is the value v obtained by linear interpolation in the two-column data matrix `idat` whose first column corresponds to the time value `t` and whose second column contains the corresponding measured blood-insulin level, *i.e.*, the rows of `idat` are data points on the empirical $I(t)$ curve, and "missing" data is obtained by linear interpolation. (Note, text delimited by `/*` and `*/` are explanatory comments and ignored by MLAB.)

MLAB is an interpreter that provides means to read data, define functions and ordinary differential equations with initial conditions, and estimate values of parameters appearing in the functions and derivatives, possibly specified by ordinary differential equations, to minimize the difference between data and the corresponding functions. Thus, we run MLAB, in order to enter MLAB commands, as shown below. (In practice, we would put these commands in a text-file called a "do-file", and command MLAB to execute this do-file; this use of an MLAB "program" allows easy correction of errors and reuse whenever desired.)

We begin by reading the data file `minmod1.dat` consisting of time values in column 1, blood glucose levels in column 2, and blood insulin levels in column 3. The argument '50' specifies that there are at most 50 rows of data.

```
data = read(minmod1,50,3)
```

Then we set the variable `n` to the number of time values (*i.e.*, the number of rows in the data), and we set the two-column array `gdat` to the (time,glucose level) ordered pairs and the two-column array `idat` to the (time,insulin level) ordered pairs.

```
m = nrows(data);
gdat = data col (1,2);
idat = data col (1,3);
```

Next we define the glucose minimal-model involving the following functions, variables, and parameters:

- `t` is time.
- `g(t)` is the blood-glucose level function, and `g(0)=g0` is the initial value of g at time 0.
- `g't(t)` is the derivative (rate-of-change) of the blood glucose level.
- `x(t)` is the interstitial insulin level function, and the initial value of x, `x(0)` is 0.
- `x't(t)` is the derivative (rate-of-change) of the interstitial insulin level.
- `i(t)` is the blood insulin level, empirically-defined by interpolation in the array `idat`.

- `gb` is the basal (180 minute fasting blood glucose level).
- `ib` is the basal (180 minute fasting blood insulin level).
- `g0` is the initial *maximal* blood glucose concentration (just after glucose *bolus* is input and *mixed*).
- `p1` is the rate of glucose transport from or to the blood compartment.
- `p2` is the rate of insulin transport from the interstitial compartment.
- `p3` is the rate of insulin transport between the blood and the interstitial compartment.

The following MLAB commands define the model:

```
fct g't(t) = -p1*(g(t) -gb) - x(t)*g(t)
fct x't(t) = p3*(i(t) -ib)  - p2*x(t)
fct i(t) = lookup(idat,t)  /* define i(t) by interpolation of data */

init g(0) = g0; /* initial condition required to integrate g't */
init x(0) = 0.0 /* initial condition required to integrate x't */

gb = gdat(m,2) /* Here we get the constants gb and ib from the input data. */
ib = idat(m,2)

/* Assign g0 the initial amount of injected glucose.*/
g0 = 287;

/* give initial estimates for parameters p1,p2,p3 */
p1 = .0399; p2 = .02; p3 = .00004;
```

Next, we define weights and constraints, guess parameters and do the fit. The early time glucose data is weighted as 0 because mixing in early time is not complete. We use a reciprocal proportion of each glucose data value. (This is essentially assuming the error in each data value is proportional to the observed value.)

```
/* define weights for glucose level data, censoring data up to time t = 8 */
fct wf(i) = if gdat(i,1) < 8 then 0 else (1/(.015*gdat(i,2)))
ws = wf on 1:m

/* define constraints for p1, p2, p3, and g0 */
constraints q = {p1>0,p2>0,p3>0,g0>0}

/* fit the model to the weighted data with defined constraints */
fit (p1,p2,p3), g to gdat with weight ws constraints q
```

We obtain the following output, with the parameters `p1`, `p2`, and `p3`, *reset* to the indicated values:

```
final parameter values
      value                error                  dependency    parameter
  0.03139866193       0.00247080549          0.8747052697   P1
  0.01877500745      0.007483039124          0.9275506889   P2
 9.491576611e-06      3.029515132e-06           0.9622345353   P3
12 iterations
CONVERGED
best weighted sum of squares = 2.070259e+02
weighted root mean square error = 3.139805e+00
weighted deviation fraction = 2.542746e-02
R squared = 6.760559e-01
no active constraints
```

Now we graph our data and the corresponding estimated or fitted functions that make up the minimal model for the individual with the given blood glucose and insulin levels. To display graphs of (1) the fitted glucose level function G and the blood glucose data versus time, with a horizontal dashed line shown at the basal level, (2) the fitted interstitial insulin function X versus time, and (3) the blood insulin level function I defined by interpolated data versus time,

The following commands display the estimated blood glucose curve that fits the blood glucose data, together with the blood glucose data. Here `tstart:tend!200` is the column vector (list) of 200 numbers starting with `tstart`, and ending with `tend`, with the equally-spaced step-size `(tend-tstart)/200`.

The expression `points(g,tstart:tend!200)` constructs a 2-column matrix (table) of points `(t,g(t))` for `t` ranging through the values in the list (vector) `tstart:tend!200`. Note, constructing this table of points for drawing requires MLAB to solve our system of ordinary differential equations!

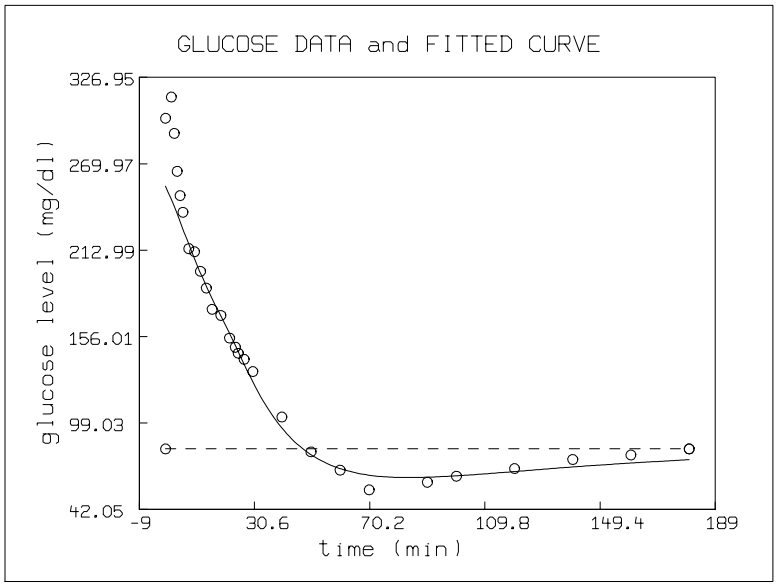
The expression `list(tstart,gb,tend,gb)` is the list (column vector) of the four numbers `tstart`, `gb`, `tend`, `gb`, and the `shape` operation `shape(2,2,list(tstart,gb,tend,gb))` re-forms this 4-element list into the 2-row, 2-column table:

```
 tstart  gb
  tend   gb
```

which represents the two points `(tstart,gb)` and `(tend,gb)`; these two points are drawn with the *pointtype* circle of size .01, and a connecting dashed-line is also drawn.

```
/* Draw the estimated blood glucose curve and the blood glucose data */
tstart  = gdat(1,1); tend = gdat(m,1)
draw points(g, tstart:tend!200)
draw gdat lt none pt circle ptsize .01
draw shape(2,2,list(tstart,gb,tend,gb)) lt dashed pt circle ptsize .01
```

```
left title "glucose level (mg/dl)"
bottom title "time (min)"
top title "GLUCOSE DATA and FITTED CURVE"
window adjust wslack
view
```

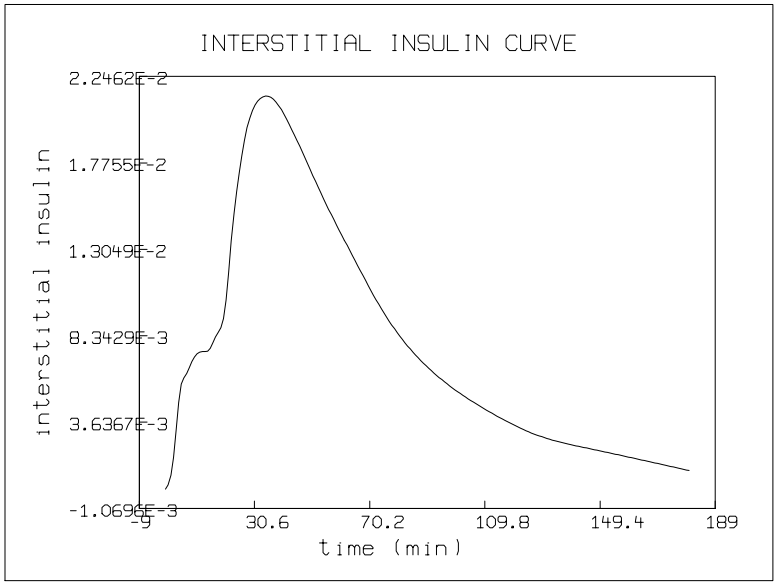


Now, hitting any key will resume MLAB, allowing further commands to be given. We remove the picture from the screen with the command `unview`. Then we discard the default-window `w` (so we can reuse it without "adding" to it) with the command `delete w`. (We can write `del w` for short.)

```
unview
del w
```

Here is the graph of the interstitial insulin curve X:

```
/* Draw the estimated interstitial insulin curve */
draw points(x, tstart:tend!200)
left title "interstitial insulin"
bottom title "time (min)"
top title "INTERSTITIAL INSULIN CURVE"
window adjust wslack
view
```

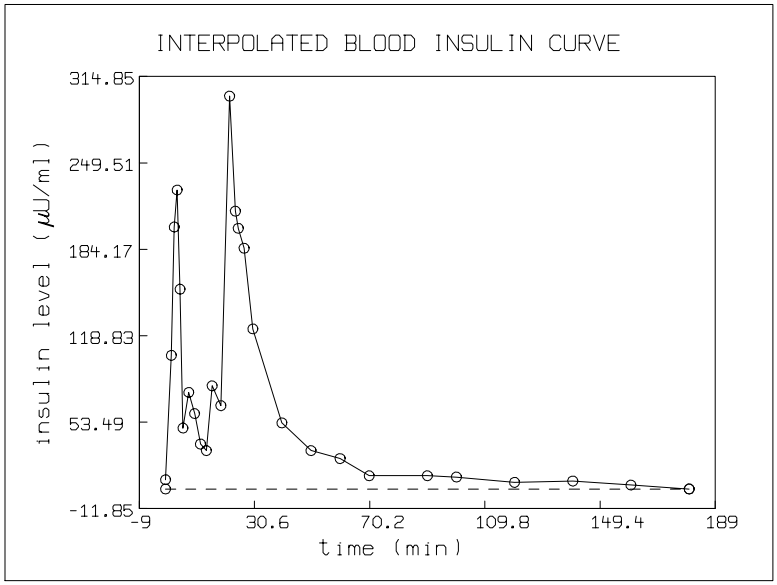


Again, hitting any key will resume MLAB, and we remove the picture from the screen and delete the default-window.

```
unview
del w
```

Here is the blood insulin level function I defined by interpolated data versus time:

```
/* Draw the blood insulin data curve */
draw idat lt none pt circle ptsize .01
draw shape(2,2,list(tstart, ib, tend, ib)) lt dashed pt circle ptsize .01
left title "     insulin level ('15Tm'RU/ml)"
bottom title "time (min)"
top title "INTERPOLATED BLOOD INSULIN CURVE"
window adjust wslack
view
```



Again, hitting any key will resume MLAB. The picture is removed from the screen and the window is deleted with the following commands:

```
unview
del w
```

Now we compute and type-out the derived descriptive measures: glucose effectiveness and insulin sensitivity.

```
type "glucose effectiveness:", p1
```

And we obtain the result:

```
     glucose effectiveness:
  P1 = 3.13986619E-2
```

```
type "insulin sensitivity:", p3/p2
```

And we obtain the result

```
     insulin sensitivity:
    = 5.05543161E-4
```

Note again, all the above MLAB commands can be "encapsulated" in a text-file called an *MLAB do-file* suitable for multiple uses that can be invoked with the MLAB `DO`-command.

Now we will fit our glucose data again using the empirical `LOOKUP`-defined insulin data as above, but adding the two additional parameters `g0` and `gb` corresponding to the initial and basal glucose levels, respectively; previously, these two parameters were "manually" estimated from the data.

```
fit (p1,p2,p3,g0,gb), g to gdat with weight ws constraints q
```

to which MLAB responds:

```
final parameter values
      value               error               dependency    parameter
  0.02119053799       0.01088355415          0.9956186342   P1
  0.01446844628       0.02553591482          0.9996188937   P2
 1.631449439e-05      8.098939899e-06           0.997963305   P3
    260.2693961          6.71727314          0.9006147807   G0
    113.3638794         37.87515246          0.9988722492   GB
13 iterations
CONVERGED
best weighted sum of squares = 1.096825e+02
weighted root mean square error = 2.402658e+00
weighted deviation fraction = 2.137497e-02
R squared = 7.140117e-01
no active constraints
```

Note that this fit using the additional parameters `g0` and `gb` changed our estimates of `p1` from 0.034 to 0.021; `p2` from 0.0188 to 0.0145; and `p3` from 9.5e-6 to 1.6e-5. However, the quality of the fit is not very different.

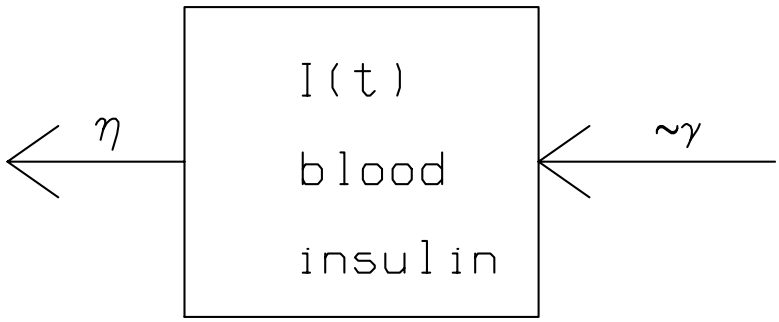
Computing the derived descriptive measures, glucose effectiveness and insulin sensitivity for these parameter estimates yields:

```
type  "glucose effectiveness",p1

     glucose effectiveness:
  P1 = .021190538

type "insulin sensitivity",p2/p3

     insulin sensitivity:
    = 1.12759132E-3
```

These values differ from before by factors of ~ 100 percent, so these measures may not be reliably estimatible based on the results of curve-fitting.

The Minimal Model for Insulin Kinetics:

The following diagram summarizes the minimal model for insulin kinetics; here we define the concentration of insulin in the blood as being increased at a rate proportional to the concentration of glucose in excess of the basal concentration multiplied by the time since the glucose bolus injection, and decreased at constant rate η.



We write $\sim \gamma$ to indicate that γ is not a simple proportionality constant, but is used to provide a modified-flow term which may be bi-directional, as seen below.

The differential equation for insulin kinetics is the following equation; we can employ this equation defining the blood insulin level I in place of using the empirical function obtained by interpolation and replace the differential equation definition of `G` with a `LOOKUP`-defined numerical definition. When we do this, we have the two unknown parameters γ and η, and, if we need to treat the introduced parameters I_0 and/or h as unknown parameters to be estimated, they become additional unknown parameters. These parameters can then be determined by curve-fitting the blood insulin data: this is the insulin minimal model. (When we fit *both* the insulin *and* the glucose data to ODE-defined functions having all 8 parameters, we have the *combined* minimal model.)

We have

$$\frac{dI(t)}{dt} = \begin{cases} -\eta \cdot I(t) + \gamma \cdot (G(t) - h) \cdot t, & \text{if } G(t) > h \\ -\eta \cdot I(t), & \text{otherwise.} \end{cases}$$

with $I(0) = I_0$. Here t is time, $I(t)$ is the blood insulin level at time t, η is the insulin clearance fraction, γ is a measure of the secondary pancreatic response to glucose, and h is roughly the basal blood glucose level; thus it would be appropriate to replace h by the variable G_b introduced earlier, however, we keep these two variables distinct following Pacini, *et.al.*

In the above differential equation, insulin enters the blood compartment at a rate proportional to the product of time and the concentration of glucose above a threshold amount. Here, as before, time is the interval, in minutes, from the glucose injection. If the blood glucose level drops below the

threshold amount, the rate of insulin entering the blood compartment is zero. Insulin is cleared from the blood compartment at a rate proportional to the amount of insulin in the blood compartment.

Note particularly the rate of change of insulin due to the amount of glucose present at time t is not a constant; it is the linear function of t: $-\gamma(G(t)-h)t$. Presumably Bergman *et.al.* found this non-standard formulation descriptively effective.

We may take the differential equation above defining the function I for insulin decay following a glucose injection by itself to be the *insulin minimal model*. In this insulin minimal model, the function $G(t)$ giving the course of blood glucose in time may be defined by interpolation of measured time-glucose values, just as blood insulin was treated in the minimal model for glucose kinetics.

Fitting the insulin minimal model, provides estimates for η, γ, and h and we can then compute two more metabolic indices:

- ϕ_1 = first phase pancreatic responsiveness: a measure of the size of the first peak in blood insulin due to the glucose injection, (ϕ_1 is given in the unit: (micro-insulin unit) minute/milligram).
- ϕ_2 = second phase pancreatic responsiveness: a measure of the size of the second peak of blood insulin which follows the first peak and the refractory period, (ϕ_2 is given in the unit: (micro-insulin unit) / milligram).

The *first phase pancreatic responsiveness* is defined as $\phi_1 = \dfrac{I_{max} - I_b}{\eta \cdot (G_0 - G_b)}$ where I_{max} is the maximum insulin value seen in response to the glucose injection (often, $I_{max} = I_0$). The *second phase pancreatic responsiveness* is defined as $\phi_2 = \gamma \times 10^4$.

Fitting the Insulin Minimal Model with MLAB

Below we show the use of MLAB to fit the insulin minimal model to our data, using the given time-course blood glucose data to define the glucose function G as an empirical function defined with linear interpolation via the MLAB `LOOKUP` function. In this case we will estimate the unknown parameters η and γ, taking $h = G_b$ as estimated before, and taking I_0 to be the maximum value seen in the data.

Thus we *replace* the function `i` (defined above via the `LOOKUP` function) with an initial condition and define the corresponding ODE. Note we use the MLAB variable `cf` as a synonym η.

```
delete i

init i(0) = i0
fct i't(t) = -cf*i + gama*t*(if g(t) < gb then 0 else (g - gb) )
```

Also, we replace the initial condition `g(0)` we defined above with the `LOOKUP`-defined blood level glucose function that computes glucose levels by interpolation. Note the ODE `g't` is still defined as before. Of course it is now of no use, and would be of no value in computing the derivative of

g, but, it is harmless to have g't left in memory, and it may be convenient if we restore g to be an initial conditon again.

```
/* Define g(t) empirically. */
fct g(t) = lookup(gdat, t)

  Redefining G
```

Note MLAB warns us we are redefining the extant symbol G.

The variables, parameters, and functions thus defined are:

- t is time.
- g(t) is blood glucose level (estimated by interpolation).
- i(t) is blood insulin level..
- x(t) is interstitial insulin.
- gb is basal (180 min) blood glucose level h (measured, not estimated).
- g0 is the initial "maximum" blood glucose level after bolus input.
- ib is basal (180 min) blood insulin level.
- p1 is the rate of glucose transport from or to the blood compartment.
- p2 is the rate of insulin transport from the interstitial compartment.
- p3 is the rate of insulin transport between the blood and the interstitial compartment.
- cf is the insulin clearance fraction η.
- gama is a measure of secondary pancreatic response.
- i0 is the initial blood insulin concentration.

The basal blood glucose level gb occurs in the ODE for i(t). We may want a separate parameter here - say gb1, as well as gb, so their values can be different; we would then have 8 parameters. Otherwise, we have the 7 unknown parameters: p1, p2, p3, g0, cf, gama, and i0. However, p1, p2, p3, and g0 have been provisionally determined in our first curve-fit.

Now we will fit the ODE-defined function i to the insulin data to determine the two parameters cf and gama, using a LOOKUP-defined function to give the glucose data.

As usual, we must assign our parameters initial "guess" values (this is already done for p1, p2, p3, and g0), and we must also assign all other parameters appropriate values as well. Also weights for the insulin data and constraints for our fitting paramenters must be defined prior to fitting.

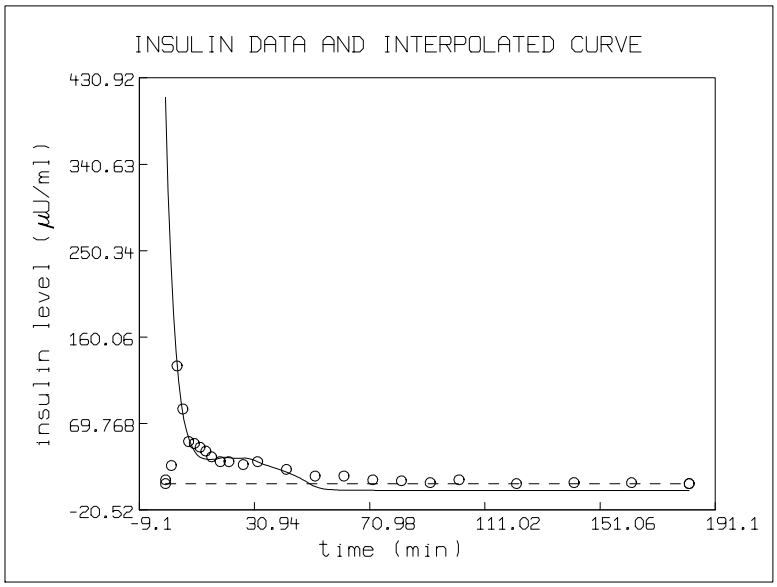
```
/* define weights for blood insulin data, censoring data up to time t = 3 */
fct wfi(i) = if idat(i,1) < 3 then 0 else if idat(i,1) <= 8 then 10 else 1
wsi = wfi on 1:m

/* define constraints for cf, gama, and i0 */
constraints qi = {cf>0, gama>0, i0>0, gb >0}

/* give initial estimates for parameters cf, gama, and i0 */
cf = .3; gama = .003349; i0 = 410.4;   /* gb = 89.5  */

/* fit the model to the weighted data with defined constraints. */
fit (cf, gama), i to idat with weight wsi constraints qi


final parameter values
      value               error                dependency    parameter
   0.3007110142      0.005198555363          0.5025318792   CF
 0.007499870853      0.0008084458361          0.5025318792   GAMA
2 iterations
CONVERGED
best weighted sum of squares = 1.700473e+03
weighted root mean square error = 8.791713e+00
weighted deviation fraction = 3.696547e-02
no active constraints
```

Now we may compute the insulin minimal model phase-1 pancreatic responsiveness and the phase-2 pancreatic insulin responsiveness:

```
type "phase 1 pancreatic responsiveness", \
 (maxv(idat col 2) -ib)/(cf*(g0 -gb))


     phase 1 pancreatic responsiveness
    = 2.78431054


type "phase 2 pancreatic responsitiveness", 10000*gama


     phase 2 pancreatic responsiveness
    = 74.9987085
```

Now we will display the insulin minimal model function i specifying the estimated blood insulin curve and the corresponding data. Also, as above, we will display the interstitial insulin curve and the LOOKUP-defined glucose level function and data versus time. The basal levels are graphed with horizontal dashed lines.

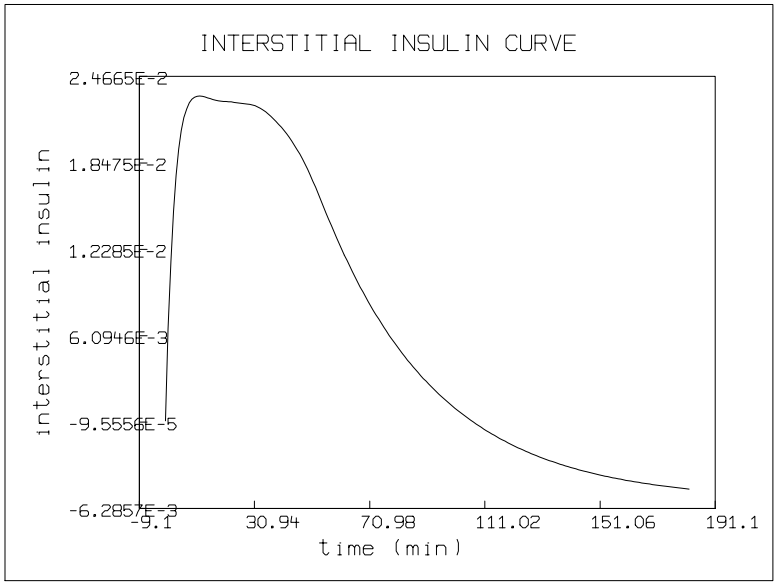
```
/* Draw the blood insulin data curve */
draw idat lt none pt circle ptsize .01
draw points(i, tstart:tend!200)
draw shape(2,2,list(tstart, ib, tend, ib)) lt dashed pt circle ptsize .01
left title "     insulin level ('15Tm'RU/ml)"
bottom title "time (min)"
top title "INSULIN DATA AND INTERPOLATED CURVE"
window adjust wslack
view
```



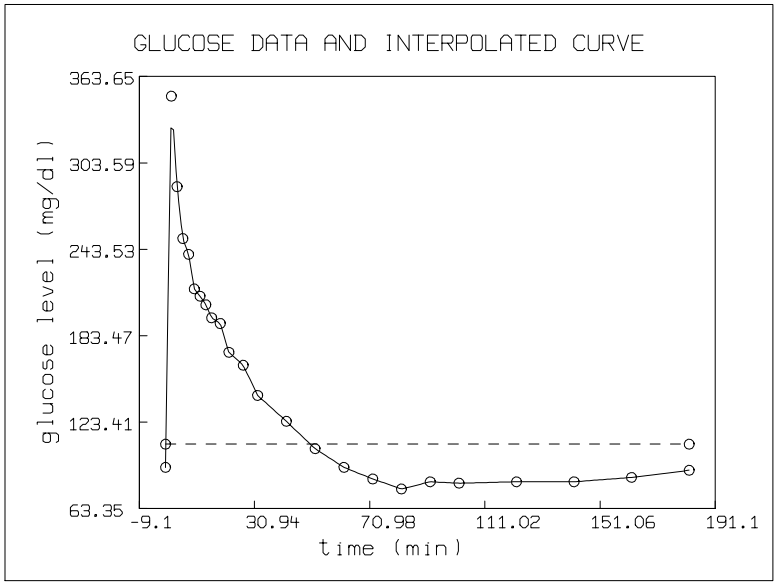
as before, the following commands remove and delete the default-window from the screen:

```
unview
del w
```

```
/* Draw the estimated interstitial insulin curve */
draw points(x, tstart:tend!200)
left title "interstitial insulin"
bottom title "time (min)"
top title "INTERSTITIAL INSULIN CURVE"
window adjust wslack
view
```




```
unview
del w


/* Draw the LOOKUP-estimated blood glucose curve
   based on the glucose data */
draw points(g, tstart:tend!200)
draw gdat lt none pt circle ptsize .01
draw shape(2,2,list(tstart,gb,tend,gb)) lt dashed pt circle ptsize .01
left title "glucose level (mg/dl)"
bottom title "time (min)"
top title "GLUCOSE DATA AND INTERPOLATED CURVE"
window adjust wslack
view
```



Now we will add the parameters `i0` and `gb`, representing the initial insulin level and basal glucose level, respectively, and see if this improves our parameter estimates, *i.e.*, improves our fit. We will note the trade-off that introducing more estimatible parameters results in a better fit, but likewise more uncertain estimates; we must take care not to *over-fit* our data, *i.e.*, introducing so many parameters in a suitably-compliant model that a good-fit is virtually guaranteed, without any descriptive value – this no better than "data-smoothing".

```
fit (cf, gama, gb, i0 ), i to idat with weight wsi constraints qi

final parameter values
      value              error                dependency    parameter
   0.2693951062       0.01660485288          0.9639201396   CF
 0.004294623142      0.0007351212032          0.7511264202   GAMA
    86.20282248         2.824407014          0.2385684391   GB
    366.1383092         27.37151538          0.9492837435   I0
5 iterations
CONVERGED
best weighted sum of squares = 1.284440e+03
weighted root mean square error = 8.013863e+00
weighted deviation fraction = 2.792236e-02
no active constraints
```

We note that in this case, while `CF` has changed by ten percent, `GAMA` has changed by more than forty percent from the values $CF = 0.3007$ and $\gamma = .0075$, obtained earlier from fitting the insulin model with the `LOOKUP`-defined glucose curve.

At this point we have estimates for `cf`, `gama`, `i0`, and `gb`. When these parameters have been estimated for a given data set, as they have here, we can then compute:

- ϕ_1 = phase 1 pancreatic responsiveness defined as $\frac{(I_{max}-I_b)}{\eta\cdot(G_0-G_b)}$ as introduced on page 15. This may be computed in MLAB as `(maxv(idat col 2) -ib)/(cf*(g0 -gb))` (Here `g0` is the maximum glucose level and `gb` is the basal glucose level, and `maxv(idat col 2)` is the maximum blood insulin level and `ib` is the basal blood insulin level. The `maxv(m)` function returns the maximum value in the matrix or vector `m`.)

  [Note rather than computing the maximum value of the insulin data in idat, we could use the maximum value of the i function in the study interval.]

- ϕ_2 = phase 2 pancreatic responsiveness = $10000 \cdot$ gama

Now we may again compute ϕ_1 and ϕ_2 based on these changed parameter values.

```
type "phase 1 pancreatic responsiveness", \
 (maxv(idat col 2) -ib)/(cf*(g0 -gb))


     phase 1 pancreatic responsiveness
    = 2.6230105


type "phase 2 pancreatic responsiveness", 10000*gama


     phase 2 pancreatic responsiveness
    = 42.9462314
```

The Combined Minimal Model

Note we can combine the insulin minimal model and the glucose minimal model to obtain the *combined minimal model* with the parameters η, γ, h, I_0, p_1, p_2, p_3, I_b, G_b, and G_0. Recall `cf` is the synonym for η that is used in our MLAB commands. In this case, we do not use any data-defined functions. The combined model allows us to simultaneously estimate all 8 parameters by fitting *two* ODE-defined functions to corresponding data. We can then characterize the FSIGT test data in terms of all four metabolic parameters: S_I, S_G, ϕ_1, and ϕ_2.

The combined minimal model is:

$$\frac{dG(t)}{dt} = p_1 \cdot (G_b - G(t)) - X(t) \cdot G(t)$$

$$\frac{dX(t)}{dt} = p_3 \cdot (I(t) - I_b) - p_2 \cdot X(t)$$

$$\frac{dI(t)}{dt} = \begin{cases} -\eta \cdot I(t) + \gamma \cdot (G(t) - h) \cdot t, & \text{if } G(t) > h \\ -\eta \cdot I(t), & \text{otherwise.} \end{cases}$$

with the initial conditions $G(0)$ and $I(0)$; the two baseline values G_b and I_b; the four rate parameters p_1, p_2, p_3, and η; and the time coefficient γ, to be assigned values or estimated by fitting. (Note, $X(0)$ is assumed to be 0.)

In order to fit the combined minimal model, we restore the initial condition `g` and the ode `g't` (note it is unneccesary to re-specify `g't`).

```
fct g't(t) = -p1*(g(t) -gb) - x(t)*g(t)
Redefining G DIFF T

init g(0) = g0
```

This results in the following notification by MLAB:

```
Redefining G
all derivatives of G have been deleted
```

Note, each time the `FIT` command is executed, MLAB defines and evaluates derivatives of the objective function with respect to the parameters. Although we have explicitly defined only the derivative of `g` with respect to `t`, MLAB has defined the derivatives of `g't` with respect `cf`, `gama`, `gb`, and `i0`.

A set of constraints limiting each of the parameters to positive values is specified with the `constraints` command:

```
constraints q1 = {cf>0,gama>0,gb>0,p1>0,p2>0,p3>0,i0>0,g0>0}
```

As before, the following MLAB `FIT` command will invoke the MLAB ODE solver as it seeks model parameter values. The ODE solver is a multi-method algorithm that is controlled by several pre-defined system variables. For example, derivatives of the function to be fit with respect to the parameters whose values we seek–*i.e.*, the Jacobian matrix, will be computed symbolically or numerically, based on the value of the system-control variable `jacsw`. Another system-control variable `disastersw` controls to what extent errors during ODE solving–such as truncation error, are reported. The `method` control-variable is used to select the method used; here the stiff-solver implementing a second-order implicit *Gear's* method is appropriate. The following system variable values have been found to be appropriate for this application:

```
jacsw = 1; /* use symbolic derivatives when computing Jacobian matrix elements */
disastersw = -2; /* do not report tolerance errors */
method = gear; /* use Gear's ODE solving method */
```

With these control variables set, we proceed with the FIT command:

```
fit (cf,gama,gb,p1,p2,p3,i0,g0), i to idat with weight wsi, \
   g to gdat with weight ws constraints q1


final parameter values
      value               error                dependency    parameter
   0.2723679397       0.01331368503          0.9765325721   CF
 0.004247946093      0.0005221003783          0.8287921659   GAMA
    80.26707677         2.223166601           0.309390387   GB
  0.01352582347        0.0110378567          0.9961323864   P1
  0.06124157093       0.01727076606          0.9642794685   P2
 2.302478892e-05      1.342490057e-05          0.9954685655   P3
    373.0904137         49.86853519          0.9936606778   I0
    294.8682961          24.3393896          0.9650664307   G0
9 iterations
CONVERGED
best weighted sum of squares = 1.054870e+03
weighted root mean square error = 5.135343e+00
weighted deviation fraction = 2.710210e-02
R squared = 3.740108e-01
no active constraints
```

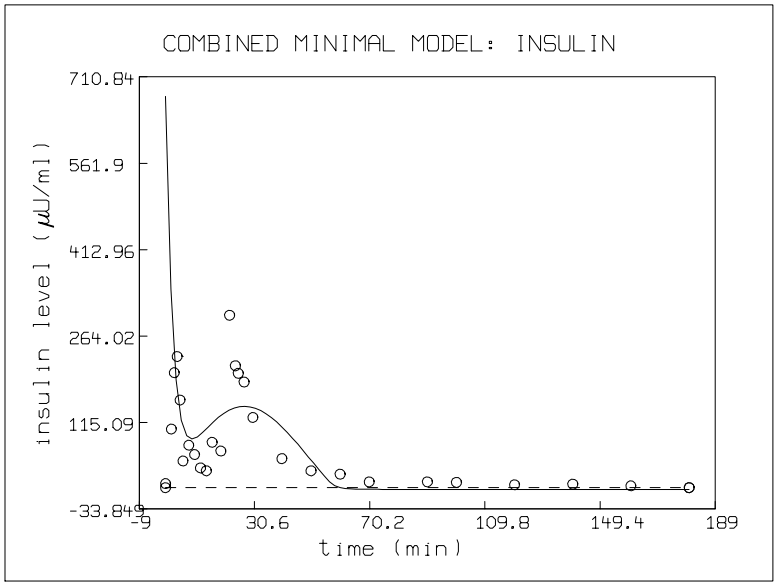
Now we may compute S_I, S_G, ϕ_1 and ϕ_2:

```
type "insulin sensitivity", p3/p2


     insulin sensitivity
    = 37.59667


type "glucose effectiveness", p1


     glucose effectiveness
    = 0.0135258


type "phase 1 pancreatic responsiveness", \
   (maxv(idat col 2) -ib)/(cf*(g0 -gb))


     phase 1 pancreatic responsiveness
    = 2.10434499
```

```
type "phase 2 pancreatic responsiveness", 10000*gama


     phase 2 pancreatic responsiveness
    = 42.4794609
```

Now, we will graph the g function and the corresponding data, the i function and the corresponding data, and the x function resulting from fitting the combined minimal model.

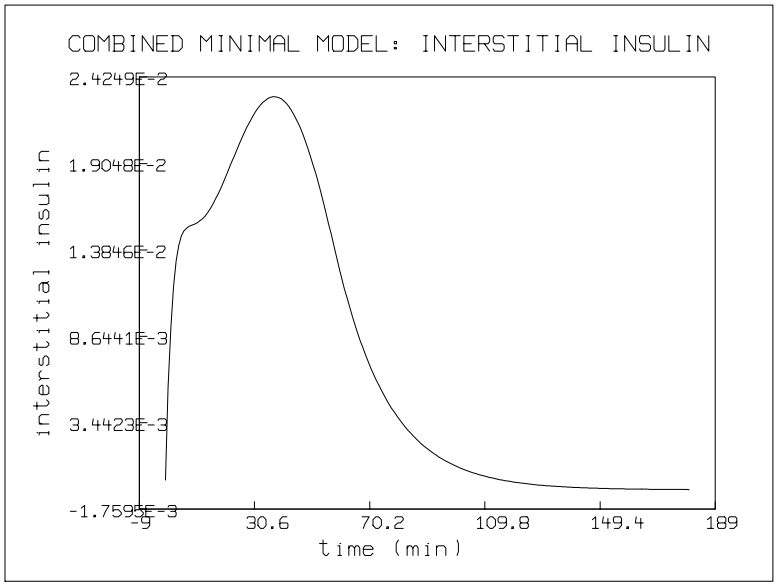
```
/* Draw the blood insulin data curve */
draw points(i, tstart:tend!200)
draw idat lt none pt circle ptsize .01
draw shape(2,2,list(tstart, ib, tend, ib)) lt dashed pt circle ptsize .01
left title "     insulin level ('15Tm'RU/ml)"
bottom title "time (min)"
top title "COMBINED MINIMAL MODEL: INSULIN"
window adjust wslack
view
```



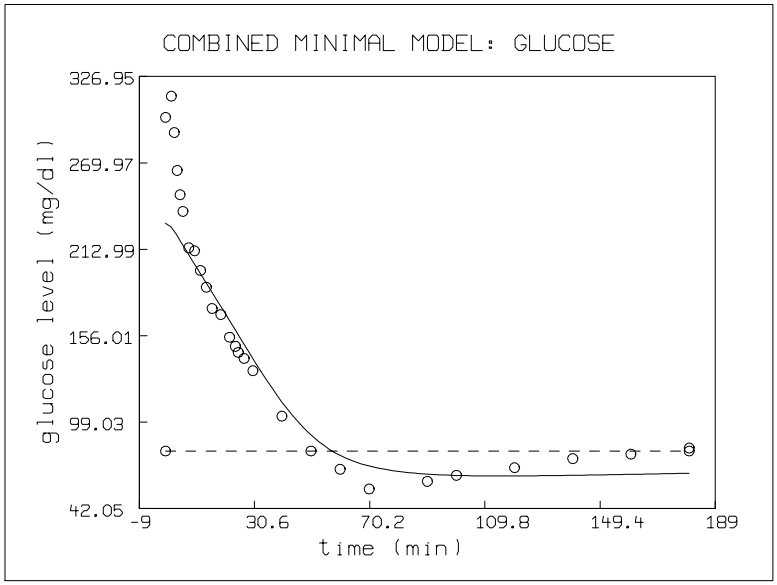

```
unview
del w
```

```
/* Draw the estimated interstitial insulin curve */
draw points(x, tstart:tend!200)
left title "interstitial insulin"
bottom title "time (min)"
```

```
top title "COMBINED MINIMAL MODEL: INTERSTITIAL INSULIN"
window adjust wslack
view
```




```
unview
del w

/* Draw the estimated blood glucose curve and the blood glucose data */
draw points(g, tstart:tend!200)
draw gdat lt none pt circle ptsize .01
draw shape(2,2,list(tstart,gb,tend,gb)) lt dashed pt circle ptsize .01
left title "glucose level (mg/dl)"
bottom title "time (min)"
top title "COMBINED MINIMAL MODEL: GLUCOSE"
window adjust wslack
view
```



These results show that, as is common, there is not a unique set of parameters that characterize a FSIGT test data set. The combined minimal model is seen to generate slightly lower values for glucose effectiveness and insulin sensitivity than the glucose minimal model, and slightly higher values for phase 1 and 2 pancreas responsiveness than the insulin minimal model.

There are various devices that could be explored in order to improve the family of models studied here. First, these models employed the MLAB operator `"LOOKUP"` to linearly interpolate blood glucose and insulin time course data. Alternatively, the MLAB operator `"SMOOTHSPLINE"` could be used to provide blood glucose and insulin time course curves that are not only continuous, but also have continuous first and second derivatives.

Second, several authors have augmented the insulin minimal model to account for blood levels of C-peptide (see references 7-9). It is a straightforward exercise to implement the C-Peptide minimal model using MLAB.

This paper has shown how MLAB can be used to calculate diagnostically important metabolic indices which arise in the glucose and insulin minimal models from frequently-sampled intravenous glucose tolerance test data. The MLAB program is an excellent tool for the study of compartmental models. See `www.civilized.com` for further examples in neurophysiology and pharmacology.

References:

1. Saad, M.F., Anderson, R.L., Laws, A., Watanabe, R.M., Kades, W.W., Chen, Y.-D.I., Sands, R.E., Pei, D., Savage, P.J., Bergman, R.N., *A Comparison Between the Minimal Model and the Glucose Clamp in the Assessment of Insulin Sensitivity Across the Spectrum of Glucose Tolerance* **Diabetes** 43 (1994) pp. 1114-21.

2. Steil, G.M., Volund, A., Kahn, S.E., Bergman, R.N.: *Reduced Sample Number for Calculation of Insulin Sensitivity and Glucose Effectiveness From the Minimal Model*, **Diabetes** 42 (1993) pp. 250-256.

3. Pacini, G., Bergman, R.N.: *MINMOD: A Computer Program to Calculate Insulin Sensitivity and Pancreatic Responsivity from the Frequently Sampled Intravenous Glucose Tolerance Test*, **Computer Methods and Programs in Biomedicine** 23 (1986) pp. 113-22.

4. Toffolo, G., Bergman, R.N., Finegood, D.T., Bowden, C.R., Cobelli, C.: *Quantitative Estimation of Beta Cell Sensitivity to Glucose in the Intact Organism–A Minimal Model of Insulin Kinetics in the Dog*, **Diabetes** 29 (1980) pp. 979-990.

5. Bergman, R.N., Ider, Y.Z., Bowden, C.R., Cobelli, C.: *Quantitative Estimation of Insulin Sensitivity*, **American Journal of Physiology** 236 (1979) pp. E667-77.

6. Walton, C.W., Wynn Department of Metabolic Medicine, Imperial College of Science, Technology, and Medicine, *private communication*, April 1998.

7. Watanabe, R.M., Volund, A., Roy, S., Bergman, R.N.: *Prehepatic B-Cell Secretion During the Intravenous Glucose Tolerance Test in Humans: Application of a Combined Model of Insulin and C-Peptide Kinetics*, **Journal of Clinical Endocrinology and Metabolism** 69 (1988) pp. 790-797.

8. Cobelli, C., Pacini, G.: *Insulin Secretion and Hepatic Extraction in Humans by Minimal Modeling of C-Peptide and Insulin Kinetics*, **Diabetes** 37 (1988) 223-231.

9. Volund, A., Polonsky, K.S., Bergman, R.N.: *Calculated Pattern of Intraportal Insulin Appearance Without Independent Assessment of C-Peptide Kinetics*, **Diabetes** 36 (1987) 1195-1202.