Validation: 2-fluoro-2-deoxy-D-glucose (FDG) Model

Overview

This document includes figures generated by `comkat\validation\validateFDG.m`. which implements the PET FDG model with linear kinetics and two tissue compartments often labeled as C_e and C_m corresponding to ^{18}FDG and ^{18}FDG-6-phosphate trapped within the cell. The model and its application to PET are presented in Phelps ME, Huang SC, Hoffman EJ, Selin C, Sokoloff L, Kuhl DE. Tomographic measurement of local cerebral glucose metabolic rate in humans with (F-18)2-fluoro-2-deoxy-D-glucose: validation of method. *Ann.Neurol.* 6:371-388, 1979.

Model Equations

Let C_e and C_m be the extravascular *molar* concentrations of ^{18}FDG and ^{18}FDG-6-phosphate and let C_p and C_a denote, respectively, the plasma *molar* and whole blood *activity* concentrations. The kinetics are given by

$dC_e/dt = k_1 C_p - k_2 C_e - k_3 C_e + k_4 C_m$

$dC_m/dt = k_3 C_e - k_4 C_m$

with zero initial conditions at time zero.

The model-predicted PET measurement or output for a scan frame beginning at time t^b and ending at time t^e is given as a weighted sum of the compartment concentrations and multiplied by the time-varying specific activity s averaged over the scan frame

$PET(t^b, t^e) = \int_{t^b}^{t^e} \{ PV\, s(C_e + C_m) + FV\, C_a \}\, dt / (t^e - t^b)$

where PV is the partial volume factor and FV is vascular fraction.

Features Tested

- linear kinetics
- reference model solved analytically
- model output
- output sensitivities
- compartment concentrations
- compartment sensitivities

Method of Testing

Use COMKAT to implement model. As a reference for comparison, use an independent implementation in which the model compartment differential equations are solved analytically to obtain an expression for its impulse response (sum of two exponentials). Convolve this impulse response with a specified input function to yield compartment concentration curves. This convolution is implemented according to the definition of the convolution integral which is evaluated via numeric integration using the MATLAB function `quadl`.

Solutions obtained using COMKAT and this independent implementation are compared by **a** plotting on same axes, **b** plotting their difference or error, **c** computing the sum of squared differencs. In point c, this is done when the equations are solved to different levels of precision with the expected behavior that the error decreases as the specified tolerance is decreased (i.e. made more stringent).

Results

Note, `quadl` may generate messages `Warning: Minimum step size reached; singularity possible.`

Interpreting the Results

The figures below compare model solutions obtained using a COMKAT implementation of the model to those obtained using an independent solution. Such results shown in this report are generated on your computer when you ran `comkat\validation\validateFDG.m`. These results should agree if both implementations are correct. To help visualize this, COMKAT solutions are depicted by circles and the independent solution is depicted by lines. If the solutions agree well, the circles will be superimposed on the lines. In order to help visualize small differences, the differences between the COMKAT and independent solutions are shown in the lower panel of each plot.

In addition, corresponding plots generated by the COMKAT developers are also shown for comparison. These results are not generated on-the-fly but are from a prior date. The purpose of this second set of plots is twofold. First, they demonstrate how close the agreement should be between the COMKAT and independent implementation. Second, they provide a reference to document how COMKAT and the independent software performed in a particular configuration; i.e., with a particular version of COMKAT, MATLAB and operating system.

Results shown in the left column are the "Your Computer" that were obtained using your computer. Results shown in the right column are "Developer's Computer" results. Notice there is a time-stamp at the bottom right of each plot.

For the sake of completeness, there are many plots. Perhaps the most important are model output and output sensitivities. As these depend on compartment concentrations and compartment sensitivities, it is unlikely that the output and output sensitivities could be correct unless the compartment and compartment sensitivities are correct.

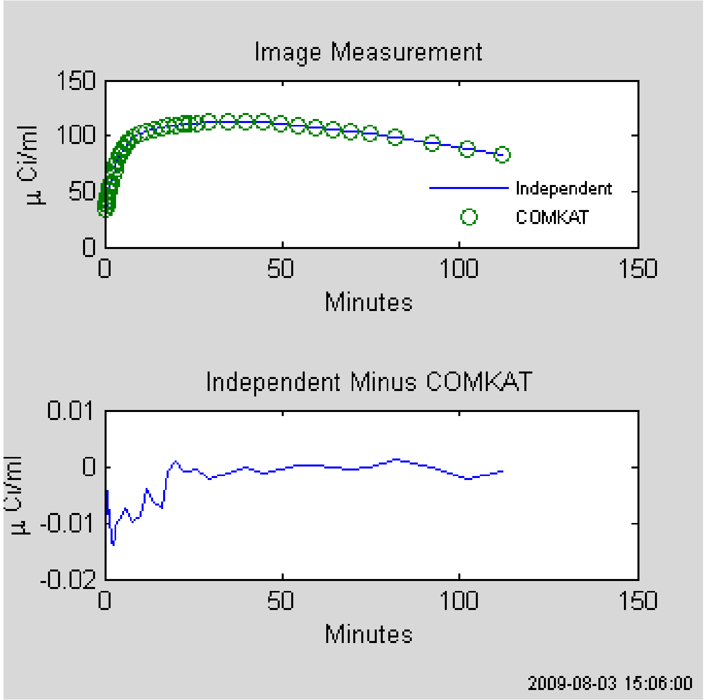
Do not be alarmed if you see messages "Warning: Minimum step size reached; singularity possible."

Model Output

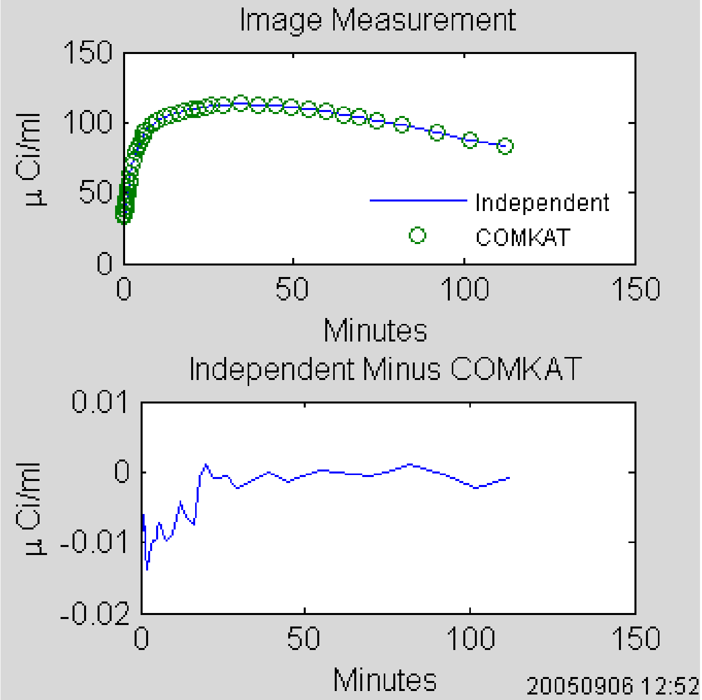
In fitting the values of the parameters are adjusted until the model output "best" matches the measured data.

Your Computer	**Developer's Computer**

Your Computer



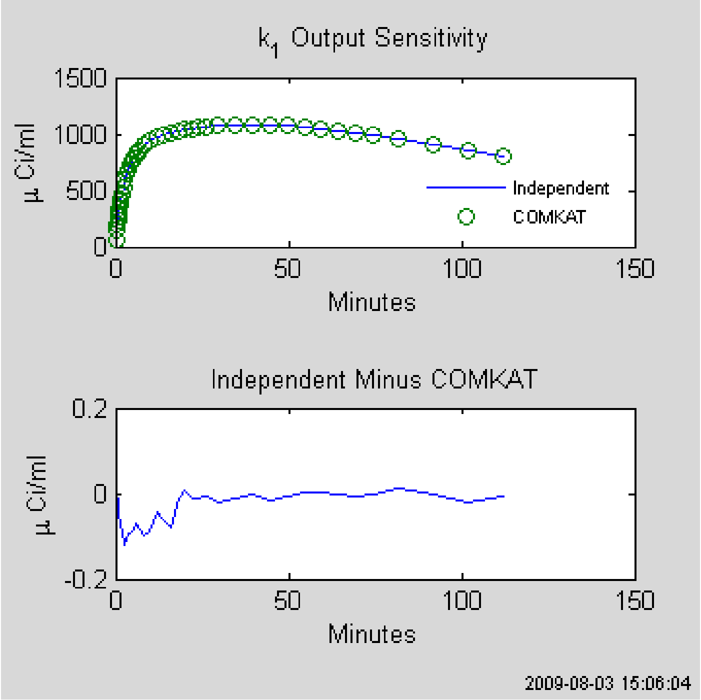
Developer's Computer



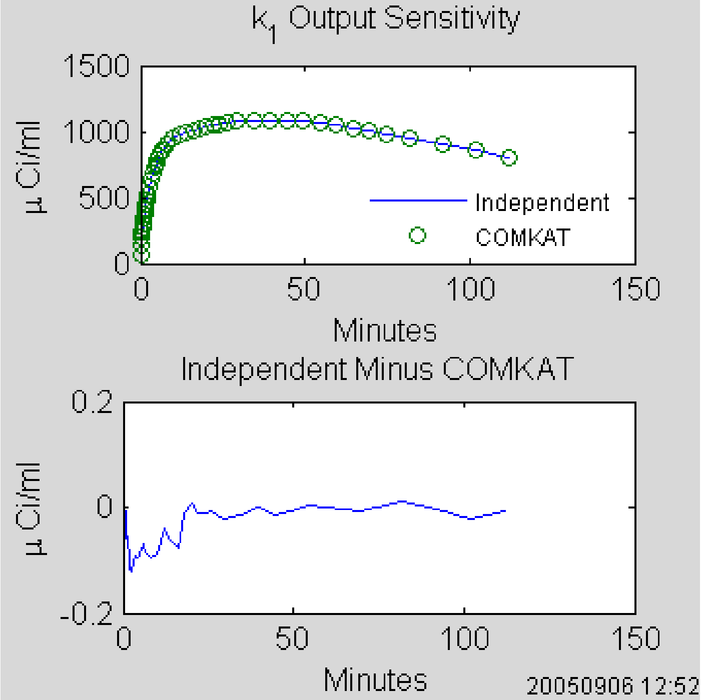
Output Sensitivities

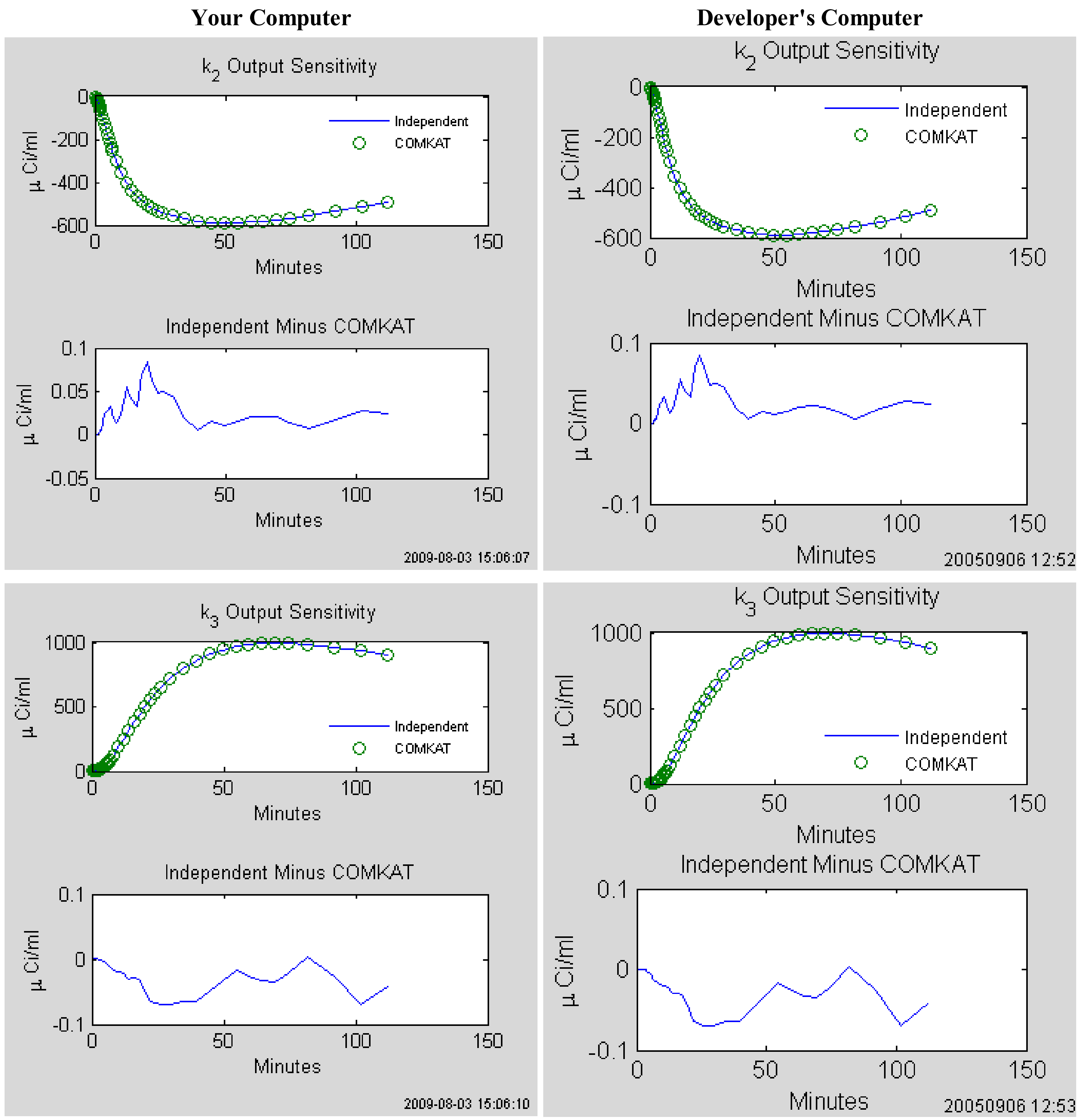
The output sensitivity functions predict how change in values of parameters will alter model output. Sensitivity functions are important in fitting data since they provide the optimizer with information needed to efficiently adjust parameter values so that the model output matches the measured data. COMKAT obtains output and compartment sensitivity functions by setting up extra differential equations and solving them simultaneously with the output and compartment concentration differential equations. This approach is more robust than the finite difference approach which is used in most software.

Your Computer

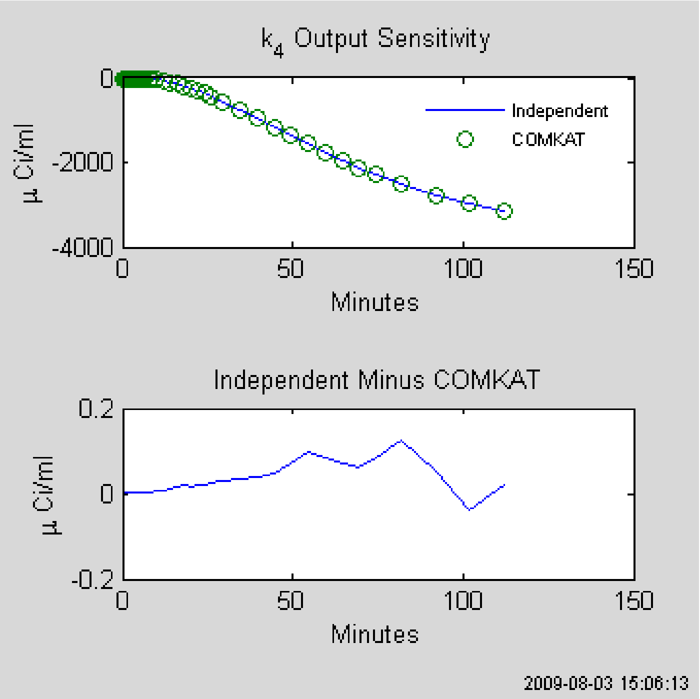


Developer's Computer

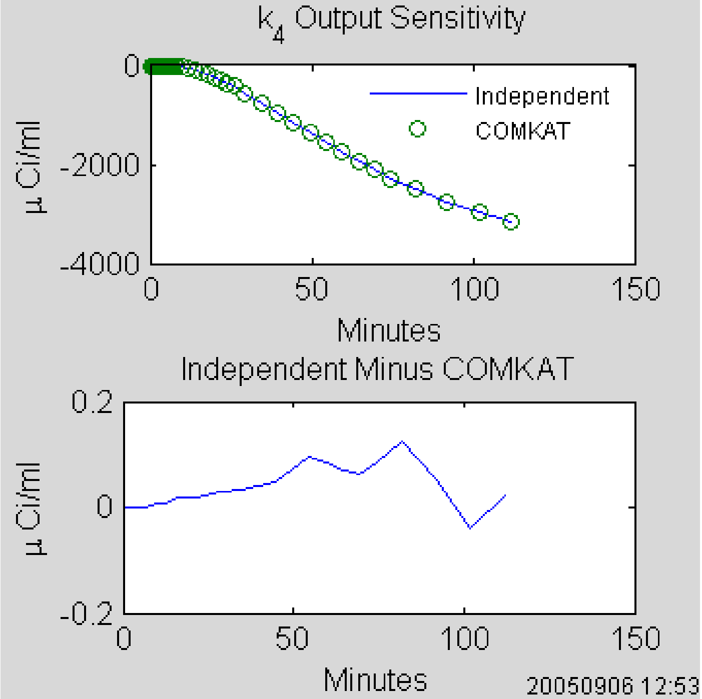



Your Computer
Developer's Computer
k_2 Output Sensitivity
µ Ci/ml
0
-200
-400
-600
Independent
COMKAT
0
50
100
150
Minutes
Independent Minus COMKAT
0.1
0.05
0
-0.05
-0.1
2009-08-03 15:06:07
20050906 12:52
k_3 Output Sensitivity
1000
500
2009-08-03 15:06:10
20050906 12:53

Your Computer



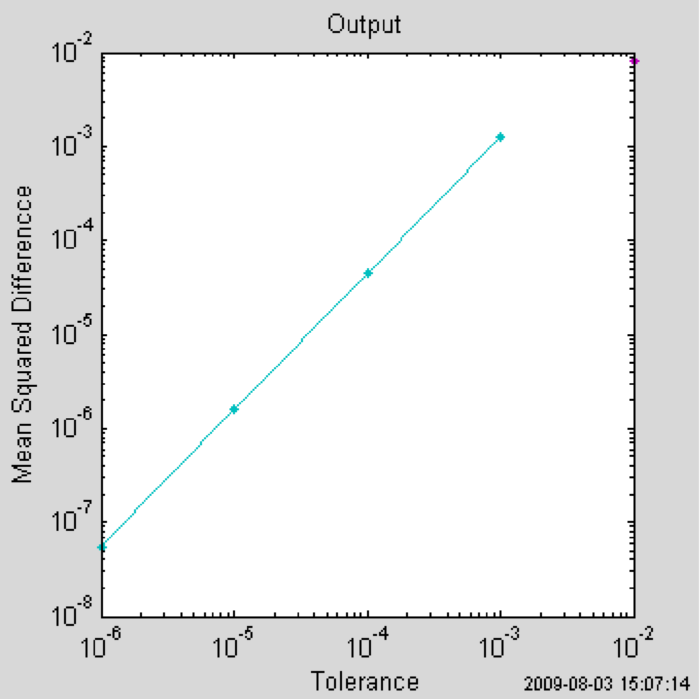
Developer's Computer



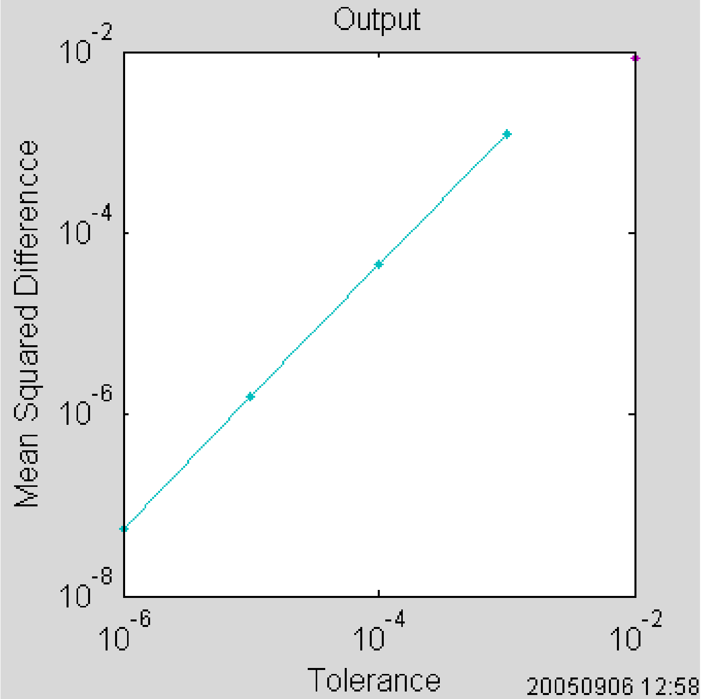
Convergence

The differential equations are solved numerically so differences in solutions are expected due to roundoff and other issues. It is expected that by specifying better precision, such differences will be reduced. If software is functioning correctly, the plots will show the mean squared differences will be reduced as the tolerances are made more stringent.

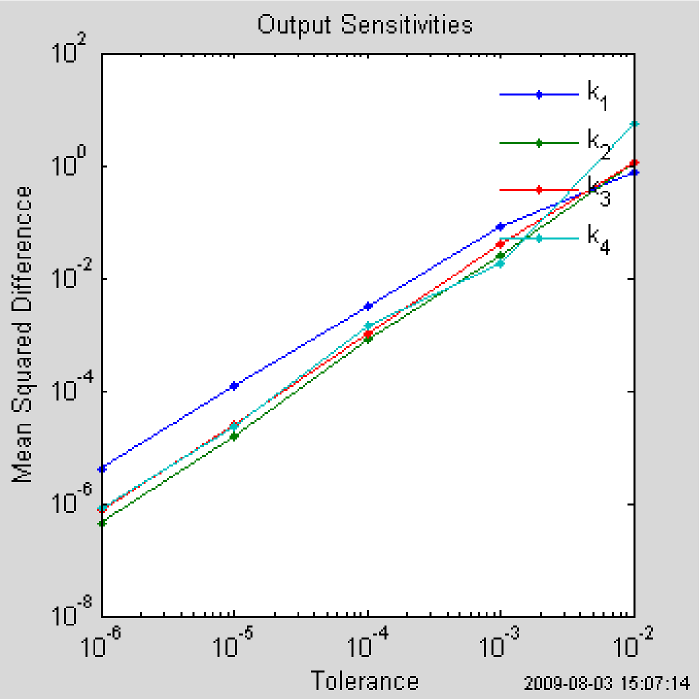
Your Computer



Developer's Computer

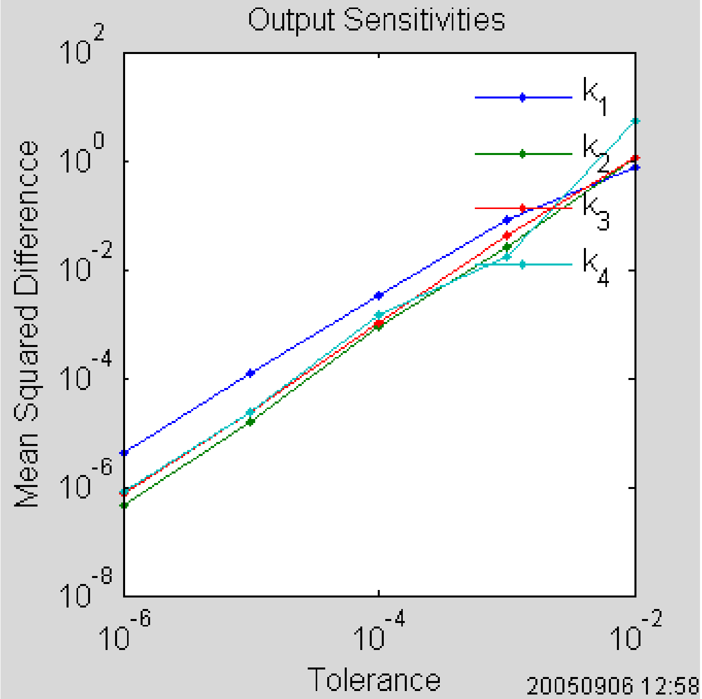


Your Computer



2009-08-03 15:07:14

Developer's Computer

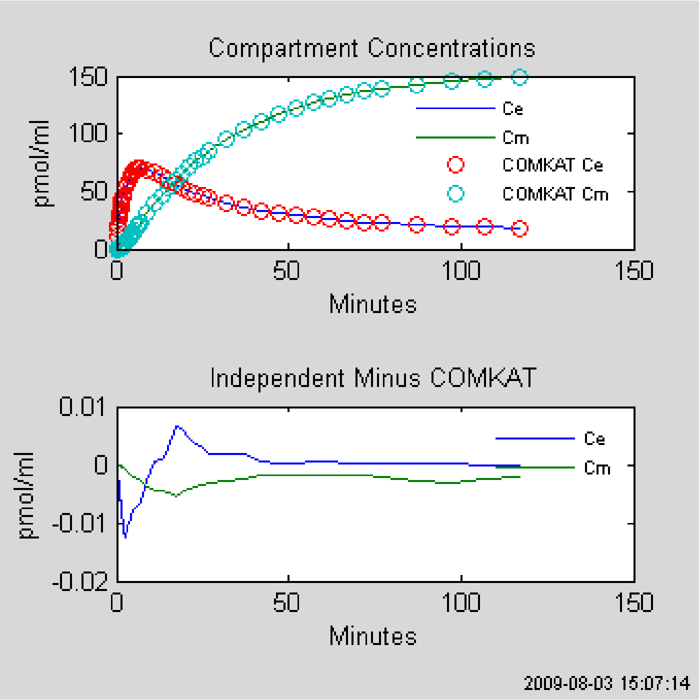


20050906 12:58

Compartment Concentrations

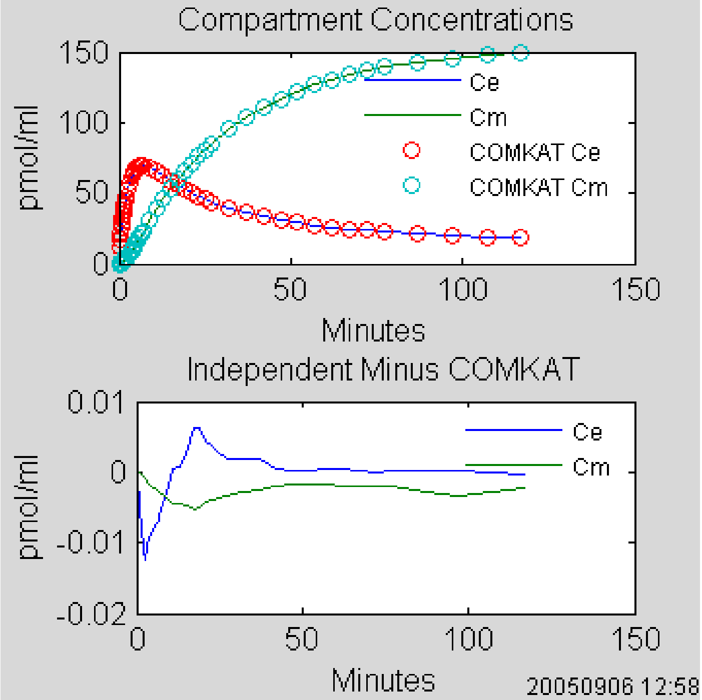
Compartment concentrations determine the pharmacokinetics and are used to calculate the model output.

Your Computer



2009-08-03 15:07:14

Developer's Computer

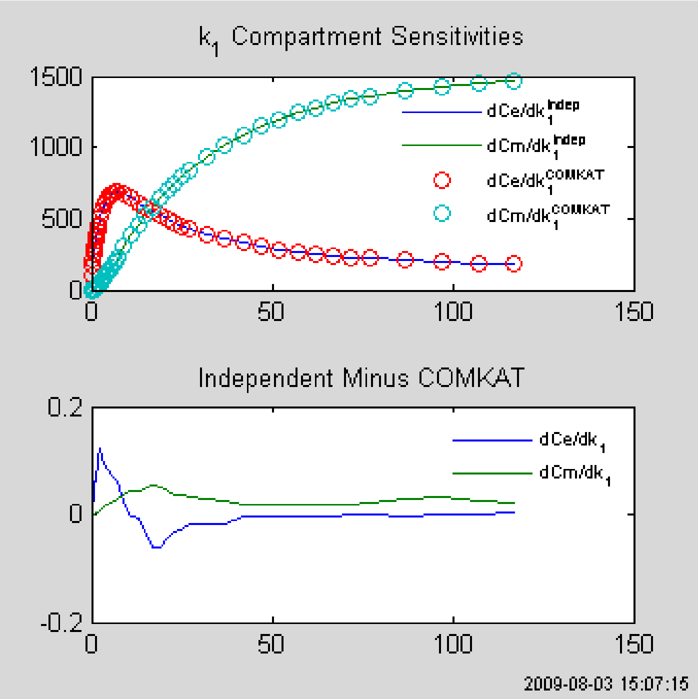


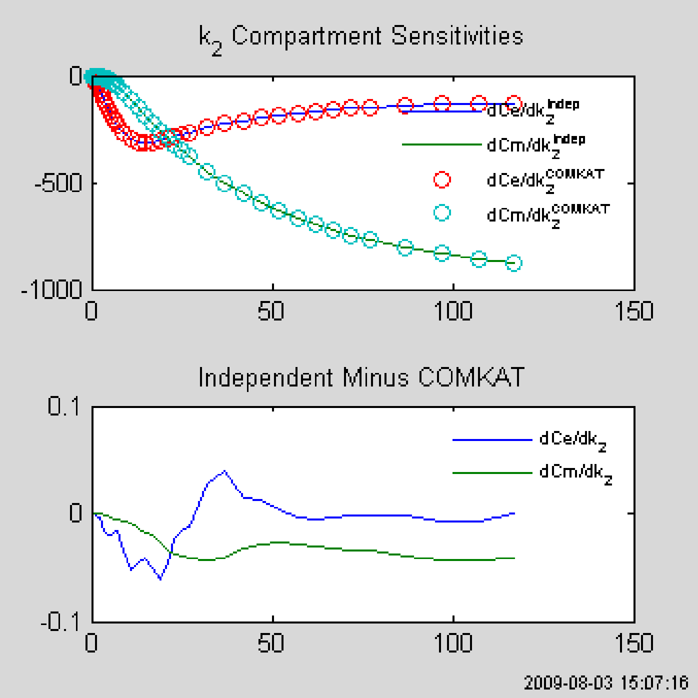
20050906 12:58

Compartment Sensitivities

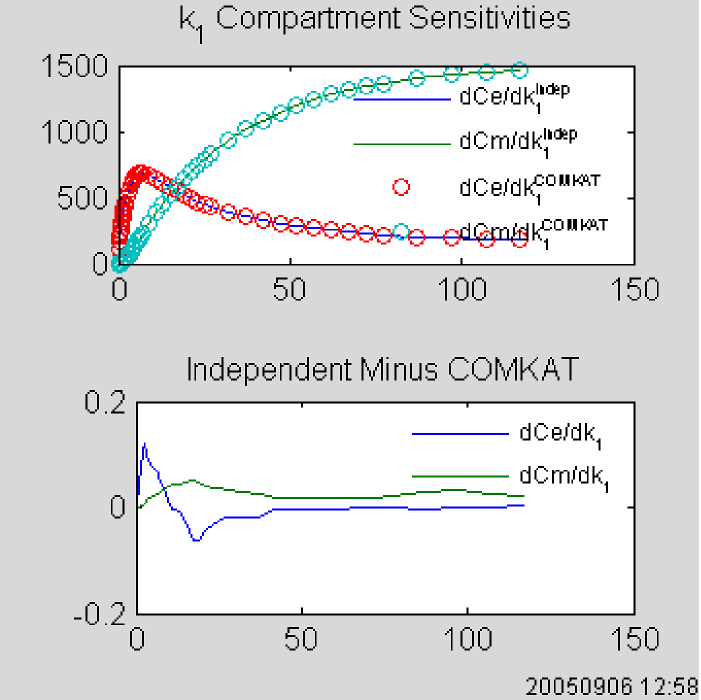
Analogous to output sensitivities, compartment sensitivities are the derivatives of the compartment concentrations with respect to the parameters. Compartment sensitivities are used to calculate the output sensitivities.

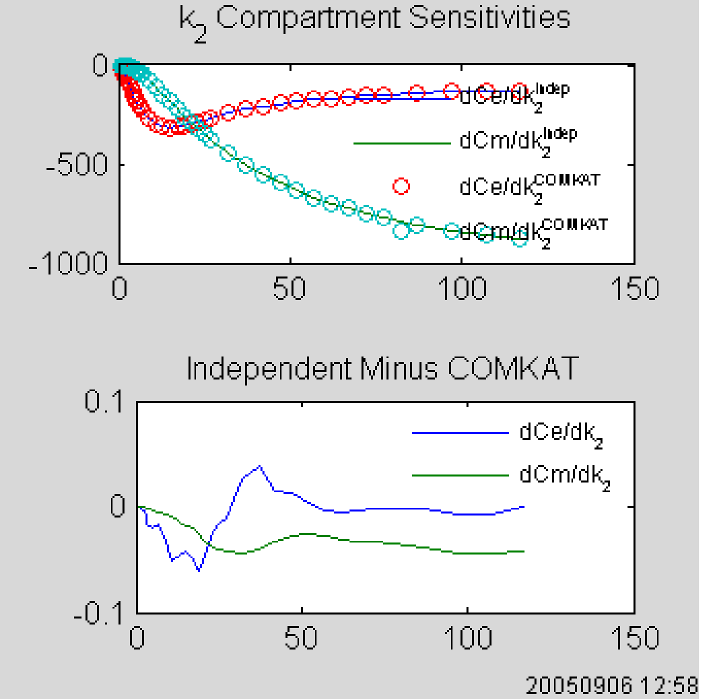
Your Computer



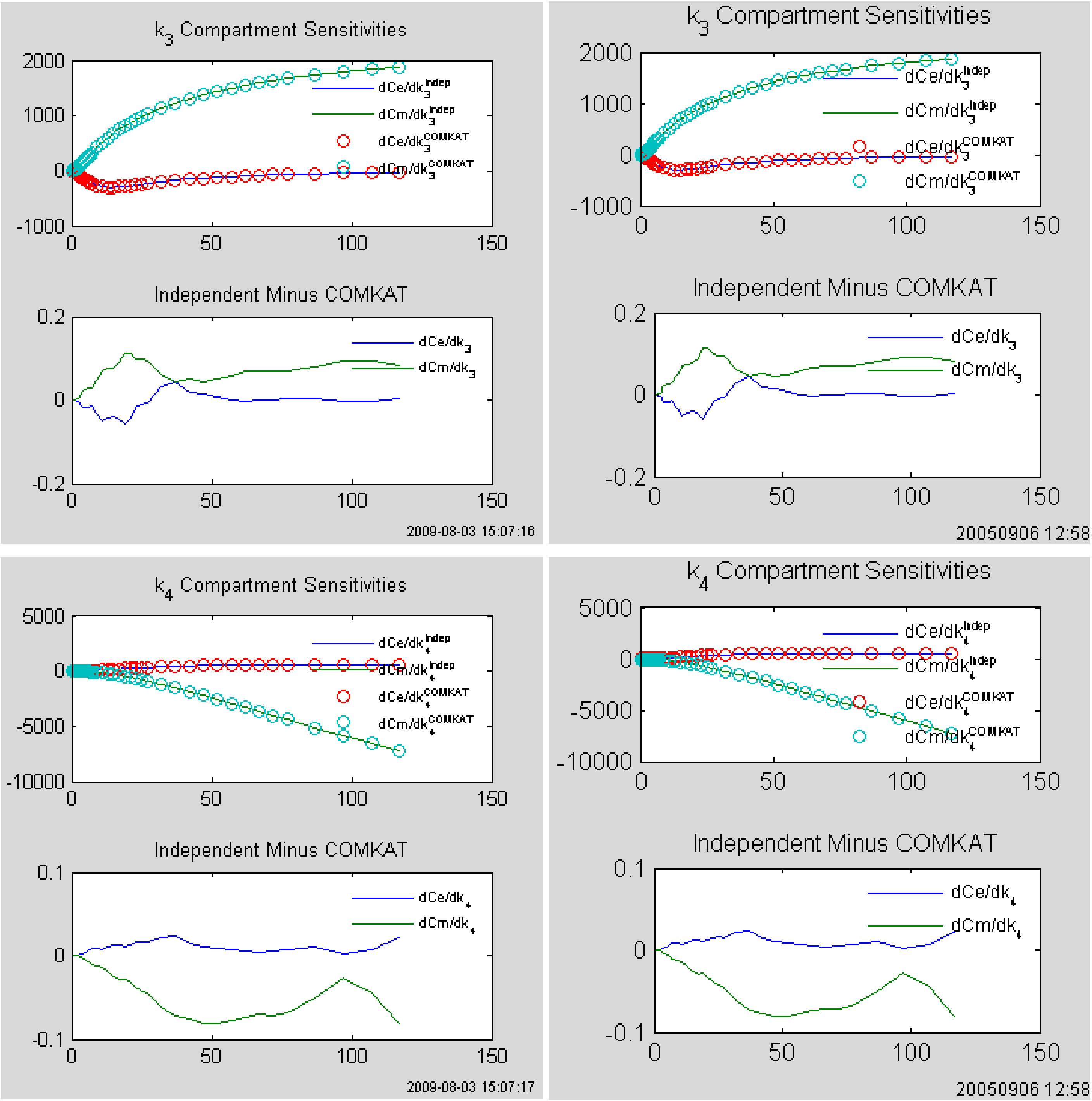


Developer's Computer





Your Computer **Developer's Computer**



Appendix: Analytic Solution

Compartment Concentrations

The compartment concentrations are described by these differential equations

$dC_e/dt = k_1\ C_p - k_2C_e - k_3C_e + k_4C_m$

$dC_m/dt = k_3C_e - k_4C_m$

In matrix form,

$C' = KC + LC_p$ (**Eq. 1**)

where

$$C = \begin{vmatrix} C_e \\ C_m \end{vmatrix}$$

$$K = \begin{vmatrix} -(k_2+k_3) & k_4 \\ k_3 & -k_4 \end{vmatrix}$$

and

$$L = \begin{vmatrix} k_1 \\ 0 \end{vmatrix}$$

Here I solve the set of differential equations using Laplace transforms.

Taking the Laplace transform of **Eq. 1** and letting š *s* denote the Laplace domain variable, yields the algebraic equation

$\check{s}C = KC + L\, C_p$

which has solution

$C = (\check{s}I - K)^{-1}\, L\, C_p$ (**Eq. 2**)

where

$$(\check{s}I - K) = \begin{vmatrix} \check{s}+k_2+k_3 & -k_4 \\ -k_3 & \check{s}+k_4 \end{vmatrix}$$

and

$$(\check{s}I - K)^{-1} = \Delta^{-1} \begin{vmatrix} \check{s}+k_4 & k_4 \\ k_3 & \check{s}+k_2+k_3 \end{vmatrix}$$

with

$\Delta = \check{s}^2 + \check{s}(k_2 + k_3 + k_4) + k_2k_4$

To facilitate calculating the inverse Laplace transform, it is advantageous to express this in an alternate form

$\Delta = (\check{s} - \lambda_1)(\check{s} - \lambda_2)$

where

$\lambda_{1,2} = -(k_2+k_3+k_4)/2 \pm \mathrm{sqrt}[(k_2+k_3+k_4)^2 - 4k_2k_4]/2$

Hence, the above expession (Eq. 2) for *C* becomes

$$C = \Delta^{-1} \left| \begin{array}{ccc} \check{s}+k_4 & k_4 & k_1 \\ k_3 & \check{s}+k_2+k_3 & 0 \end{array} \right| C_p$$

Proceeding algebraically

$$C = \Delta^{-1} \left| \begin{array}{c} C_p\, k_1(\check{s}+k_4) \\ C_p\, k_1 k_3 \end{array} \right|$$

and continuing further

$$C = \left| \begin{array}{c} C_p\, k_1\, (\check{s}+k_4)(\check{s} - \lambda_1)^{-1}(\check{s} - \lambda_2)^{-1} \\ C_p\, k_1\, k_3\, (\check{s} - \lambda_1)^{-1}(\check{s} - \lambda_2)^{-1} \end{array} \right|$$

$$= \left| \begin{array}{c} C_p\, k_1\, (\lambda_1 - \lambda_2)^{-1}[(k_4+\lambda_1)(\check{s}-\lambda_1)^{-1} - (k_4+\lambda_2)(\check{s}-\lambda_2)^{-1}] \\ C_p\, k_1 k_3\, (\lambda_1 - \lambda_2)^{-1}[(\check{s}-\lambda_1)^{-1} - (\check{s}-\lambda_2)^{-1}] \end{array} \right|$$

Taking the inverse Laplace transform yields the time-domain expression

$$C = \left| \begin{array}{c} C_p * \{k_1\, (\lambda_1 - \lambda_2)^{-1}[(k_4+\lambda_1)e^{\lambda_1 t} - (k_4+\lambda_2)e^{\lambda_2 t}]\} \\ C_p * \{k_1 k_3\, (\lambda_1 - \lambda_2)^{-1}[e^{\lambda_1 t} - e^{\lambda_2 t}]\} \end{array} \right|$$

where * denotes convolution.

Compartment Sensitivities

The compartment sensitivity equations can be obtained by differentiating the compartment concentration equations with respect to the parameters.

$$dC_e/dk_1 = C_p * \{(\lambda_1 - \lambda_2)^{-1}[(k_4+\lambda_1)e^{\lambda_1 t} - (k_4+\lambda_2)e^{\lambda_2 t}]\}$$

$$dC_m/dk_1 = C_p * \{k_3\, (\lambda_1 - \lambda_2)^{-1}[e^{\lambda_1 t} - e^{\lambda_2 t}]\}$$

$$\begin{aligned} dC_e/dk_2 = C_p * \{&-k_1\, (\lambda_1 - \lambda_2)^{-2}\, (d\lambda_1/dk_2 - d\lambda_2/dk_2)\, [(k_4 + \lambda_1)\, e^{\lambda_1 t} - (k_4 + \lambda_2)\, e^{\lambda_2 t}] \\ &+ k_1\, (\lambda_1 - \lambda_2)^{-1}\, [d\lambda_1/dk_2\, e^{\lambda_1 t} + (k_4 + \lambda_1)\, t\, d\lambda_1/dk_2 e^{\lambda_1 t} - d\lambda_2/dk_2\, e^{\lambda_2 t} - (k_4 + \lambda_2) t\, d\lambda_2/dk_2\, e^{\lambda_2 t}]\ \} \end{aligned}$$

$$\begin{aligned} dC_m/dk_2 = C_p * \{&-k_1 k_3\, (\lambda_1 - \lambda_2)^{-2}\, (d\lambda_1/dk_2 - d\lambda_2/dk_2)\, [e^{\lambda_1 t} - e^{\lambda_2 t}] \\ &+ k_1 k_3\, (\lambda_1 - \lambda_2)^{-1}\, (t\, d\lambda_1/dk_2\, e^{\lambda_1 t} - t\, d\lambda_2/dk_2\, e^{\lambda_2 t})\ \} \end{aligned}$$

$$\begin{aligned} dC_e/dk_3 = C_p * \{&-k_1\, (\lambda_1 - \lambda_2)^{-2}\, (d\lambda_1/dk_3 - d\lambda_2/dk_3)\, [(k_4 + \lambda_1)\, e^{\lambda_1 t} - (k_4 + \lambda_2)\, e^{\lambda_2 t}] \\ &+ k_1\, (\lambda_1 - \lambda_2)^{-1}\, [d\lambda_1/dk_3\, e^{\lambda_1 t} + (k_4 + \lambda_1)\, t\, d\lambda_1/dk_3 e^{\lambda_1 t} - d\lambda_2/dk_3\, e^{\lambda_2 t} - (k_4 + \lambda_2) t\, d\lambda_2/dk_3\, e^{\lambda_2 t}]\ \} \end{aligned}$$

$$\begin{aligned} dC_m/dk_3 = C_p * \{&k_1\, (\lambda_1 - \lambda_2)^{-1}\, [e^{\lambda_1 t} - e^{\lambda_2 t}] \\ &- k_1 k_3\, (\lambda_1 - \lambda_2)^{-2}\, (d\lambda_1/dk_3 - d\lambda_2/dk_3)\, [e^{\lambda_1 t} - e^{\lambda_2 t}] \\ &+ k_1 k_3\, (\lambda_1 - \lambda_2)^{-1}\, (t\, d\lambda_1/dk_3\, e^{\lambda_1 t} - t\, d\lambda_2/dk_3\, e^{\lambda_2 t})\ \} \end{aligned}$$

$$dC_e/dk_4 = C_p * \{-k_1 (\lambda_1 - \lambda_2)^{-2} (d\lambda_1/dk_4 - d\lambda_2/dk_4) [(k_4 + \lambda_1) e^{\lambda_1 t} - (k_4 + \lambda_2) e^{\lambda_2 t}]$$
$$+ k_1 (\lambda_1 - \lambda_2)^{-1} [(1 + d\lambda_1/dk_4) e^{\lambda_1 t} + (k_4 + \lambda_1) t\, d\lambda_1/dk_4 e^{\lambda_1 t} - (1 + d\lambda_2/dk_4) e^{\lambda_2 t} - (k_4 + \lambda_2)t\, d\lambda_2/dk_4\, e^{\lambda_2 t}] \}$$

$$dC_m/dk_4 = C_p * \{-k_1k_3 (\lambda_1 - \lambda_2)^{-2} (d\lambda_1/dk_4 - d\lambda_2/dk_4) [e^{\lambda_1 t} - e^{\lambda_2 t}]$$
$$+ k_1k_3 (\lambda_1 - \lambda_2)^{-1} (t\, d\lambda_1/dk_4\, e^{\lambda_1 t} - t\, d\lambda_2/dk_4\, e^{\lambda_2 t}) \}$$

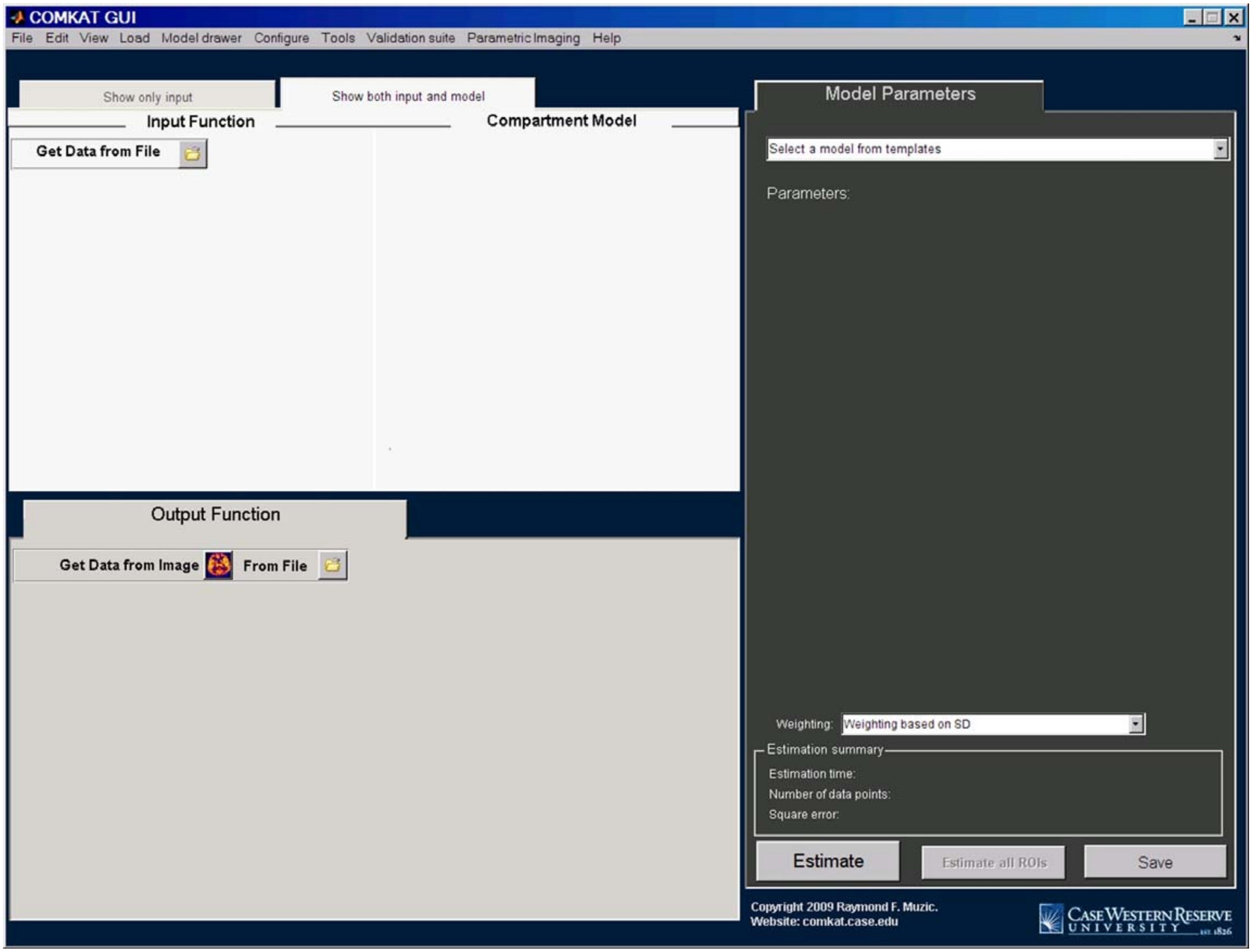
where

$$d\lambda_{1,2}/dk_1 = 0$$

$$d\lambda_{1,2}/dk_2 = -1/2 \pm [2(k_2 + k_3 + k_4) - 4k_4] / \mathrm{sqrt}[(k_2+k_3+k_4)^2 - 4k_2k_4] /4$$

$$d\lambda_{1,2}/dk_3 = -1/2 \pm [2(k_2 + k_3 + k_4)] / \mathrm{sqrt}[(k_2+k_3+k_4)^2 - 4k_2k_4] /4$$

$$d\lambda_{1,2}/dk_4 = -1/2 \pm [2(k_2 + k_3 + k_4) - 4k_2] / \mathrm{sqrt}[(k_2+k_3+k_4)^2 - 4k_2k_4] /4$$



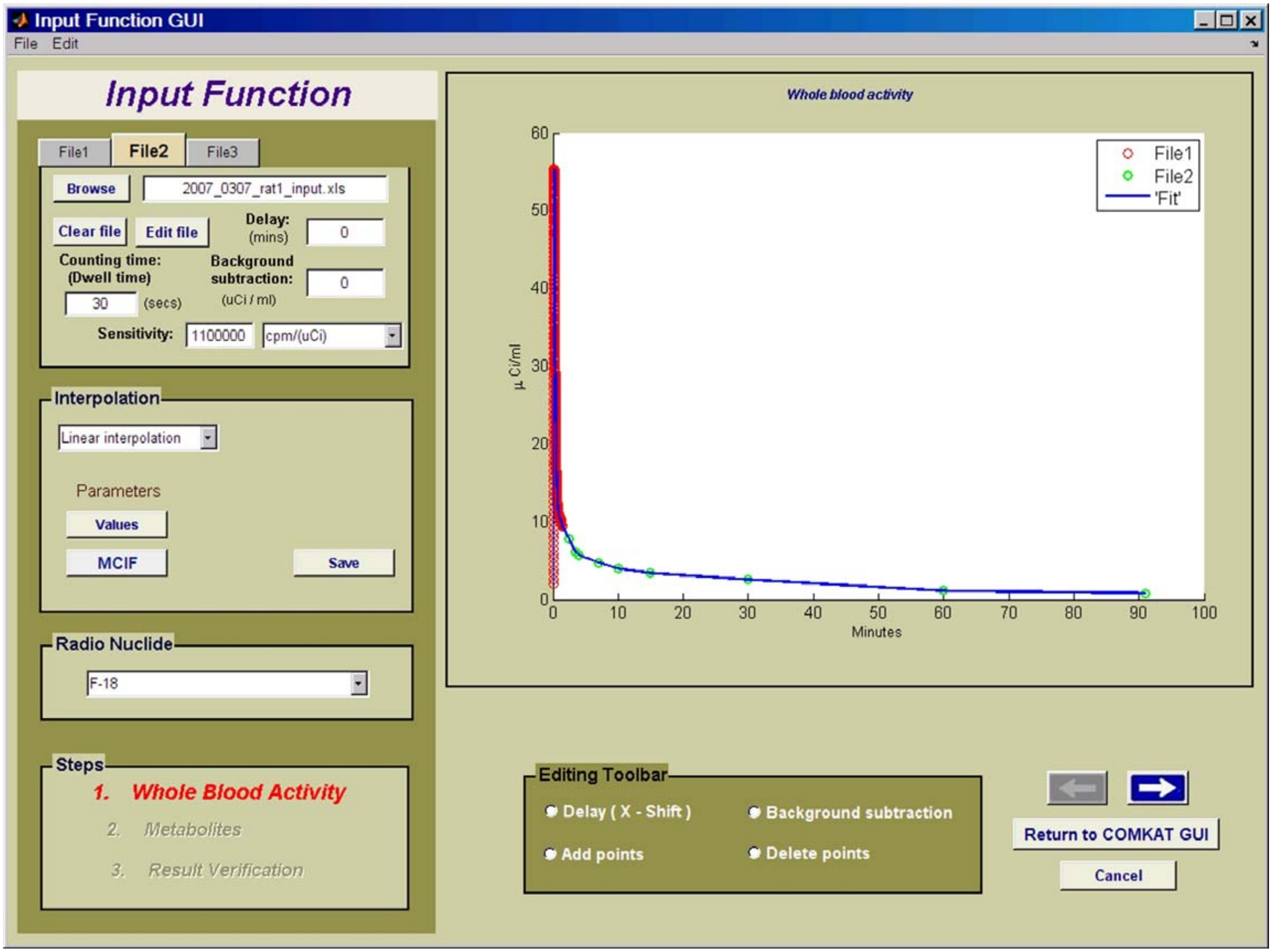
Select a model from templates

Weighting: Weighting based on SD

Estimation time:
Number of data points:
Square error:

Copyright 2009 Raymond F. Muzic.
Website: comkat.case.edu

Supplemental Figure 1



2007_0307_rat1_input.xls

Delay: (mins) 0

Counting time: (Dwell time) 30 (secs)

Background subtraction: 0 (uCi / ml)

Sensitivity: 1100000 cpm/(uCi)

Linear interpolation

F-18

Supplemental Figure 2