
COMKAT: Compartment Model Kinetic Analysis Tool

Raymond F. Muzic, Jr., and Shawn Cornelius

Radiology, School of Medicine, and Biomedical Engineering, School of Engineering, Case Western Reserve University, Cleveland, Ohio

Compartment models are the basis for most physiologically based quantification of nuclear medicine data. Although some software packages are available for this purpose, many are expensive, run on relatively few types of computers or are of limited capability, and cannot be extended because of the unavailability of source code. Consequently, institutions with modeling expertise often develop software for themselves, which has the disadvantages of lack of standardization and possible replication of effort. Therefore, general-purpose compartment-modeling software distributed with source code would be a welcome resource for the nuclear medicine community. **Methods:** We formulated a mathematic framework within which compartment models containing unimolecular and bimolecular (receptor saturation) kinetics can be described. We implemented this framework within MATLAB and call the resultant software COMKAT (Compartment Model Kinetic Analysis Tool). **Results:** COMKAT simplifies the process of defining and solving standard blood flow, ^{18}F -FDG, and receptor models as well as models of a user's own design. In particular, COMKAT automatically defines and implements state, analytic sensitivity, and Jacobian equations. Given these, COMKAT can perform simulations in which model outputs are solved for specified parameter values, thereby allowing the user to predict how sensitive data are to these parameters. In addition, COMKAT can be used to estimate values for the parameters by fitting model output to experimental data. COMKAT is equipped with command-line and graphic user interfaces from which the user can access these features. Examples of these applications are presented along with validation and performance summaries. **Conclusion:** COMKAT is a useful software tool and is available without cost to researchers, at www.nuclear.uhrad.com/comkat.

Key Words: compartment model; PET; software

J Nucl Med 2001; 42:636–645

Compartment models are the basis for the majority of the methods used in quantitative physiologic analyses of nuclear medicine data. In PET there is a particular emphasis on such models, because tracer concentrations can be measured in vivo in absolute terms. Models used in PET include

those for blood flow, oxygen metabolism, glucose metabolism, and receptor concentration estimations. Particularly for the more complex models, implementation requires considerable effort. Even for people with expert mathematic and computer skills, derivation of equations and attendance to details make model implementation, at best, a time-consuming task. Although this effort might be easily justified for a model that will be used in a large number of patient studies or experiments, it is more difficult to justify when the model is experimental in nature and one must implement numerous models as part of the model selection process. An additional issue is one of validation. Software developed within an organization and used by a few individuals is not as thoroughly tested as that used globally by numerous users. For these reasons, we have developed a general approach to compartment modeling, implemented it in software, and made it available without cost to researchers.

We formulated a mathematic framework within which compartment models incorporating unimolecular and bimolecular (receptor saturation) kinetics can be described. We implemented this framework within MATLAB software (MathWorks, Inc., Natick, MA) and call the resultant program COMKAT—Compartment Model Kinetic Analysis Tool (1). COMKAT can be used to perform simulations in which model outputs are determined for specified parameter values as well as to perform parameter estimation in which model output is fitted to experimental data. COMKAT can be used to build PET blood flow, ^{18}F -FDG, and receptor models. It is probably more important that COMKAT can, without requiring mathematic derivations, be used to build compartment models of a user's own design; users are not constrained to picking models from a predefined list of configurations. (For a detailed description of COMKAT usage including model building, input, and tissue curve specification, please refer to the *COMKAT System User Manual* (1).)

We are not the first group to develop and make available software for compartment modeling. Some of the better known packages in the field of nuclear medicine are BLD, KMZ/PKIN, and SAAM/SAAM II (2–4). However, many of these have drawbacks that limit their usefulness for a large audience. For example, BLD runs only on VAX computers (Digital Equipment Corp., Maynard, MA), and

Received Aug. 9, 2000; revision accepted Dec. 12, 2000.

For correspondence or reprints contact: Raymond F. Muzic, Jr., PhD, Nuclear Medicine, University Hospitals of Cleveland, 11100 Euclid Ave., Cleveland, OH 44106.

both SAAM II and PKIN require payment for licensing and do not come with source code. Moreover, with PKIN, users cannot build their own models but rather are restricted to picking from a list of predefined models.

In comparison with these, COMKAT is useful to a wider audience for several reasons. First, COMKAT is distributed through the Internet, with its source code and documentation and without cost to researchers. Second, COMKAT is written in MATLAB, using exclusively m-files, and thus it will run on any type of computer for which MATLAB is available (e.g., Windows 95/98/NT [Microsoft, Redmond, WA], AIX [International Business Machines Corp., Armonk, NY], Digital UNIX [Compaq Computer Corp., Houston, TX], HP-UX [Hewlett-Packard Co., Palo Alto, CA], IRIX [SGI, Mountain View, CA], Linux and Power Macintosh [Apple Computer, Inc., Cupertino, CA], and Solaris [Sun Microsystems, Palo Alto, CA]). Third, the availability of source code enables users to add features or improvements to suit their specific applications. Indeed, in some respects COMKAT is similar to SPM (5,6)—MATLAB implementation, including free distribution with source code—and we hope to likewise foster a user community.

This article describes a general mathematic framework within which we can quantitatively describe, to the best of our knowledge, all compartmental models in use in nuclear medicine from the simplest to the most complex, including those used in multiple-injection receptor studies (presented in the Theory section below). We also report on COMKAT-MATLAB software, with which models are implemented according to the theory described. In particular, we present a general description of the software and its performance for solving three common PET models. We particularly note that the software makes it easy to implement arbitrary, user-specified models by either COMKAT's graphic user or command-line interface.

MATERIALS AND METHODS

Theory

State Equation. The mathematic framework for COMKAT is based on a building block approach for compartment models, which is shown in Figure 1. In the figure, C_j denotes the concentration in compartment j , and I_n denotes the concentration of input n . Compartment j may exchange with zero or more other compartments and zero or more inputs. As an example, one might consider the FDG PET model to be composed of two such building blocks, with C_j from one unit corresponding to FDG in the tissue and C_j from the other unit corresponding to FDG-6-PO₄ in the tissue.

Compartment j is formally the recipient of fluxes from zero or more compartments i and zero or more inputs n , and it is the source of fluxes into zero or more compartments q . Each such flux is considered either unimolecular or bimolecular. For a unimolecular flux, the rate is dependent on the concentration in the source compartment or input. More specifically, the flux equals the unimolecular rate constant multiplied by the concentration of the source compartment or input. In contrast, in a bimolecular flux, the rate is dependent on both the concentration of the source compartment or input and the concentration of the receiving compartment. The flux equals the bimolecular rate constant multiplied by the concentration of the driving compartment or input and the concentration of open binding sites in the receiving compartment. This relationship is calculated as $(C_q^{\max} - C_q)$, where C_q^{\max} is the total concentration of binding sites and C_q is the concentration of occupied sites. Note that C_q^{\max} is the value at which C_q saturates, because the flux approaches zero as C_q approaches C_q^{\max} . In some cases, more than one compartment corresponds to a single saturable pool of binding sites. Such is the case when the model includes multiple parallel sets of compartments to account for multiple injections of material, each with possibly a different specific activity (7–9). Thus, we generalize the calculation of available binding sites to $(C_q^{\max} - \sum_v B_{vq}^c C_v)$, where B_{vq}^c is the coupling coefficient. B_{vq}^c equals unity if compartments v and q correspond to a common pool of binding sites and equals zero otherwise.

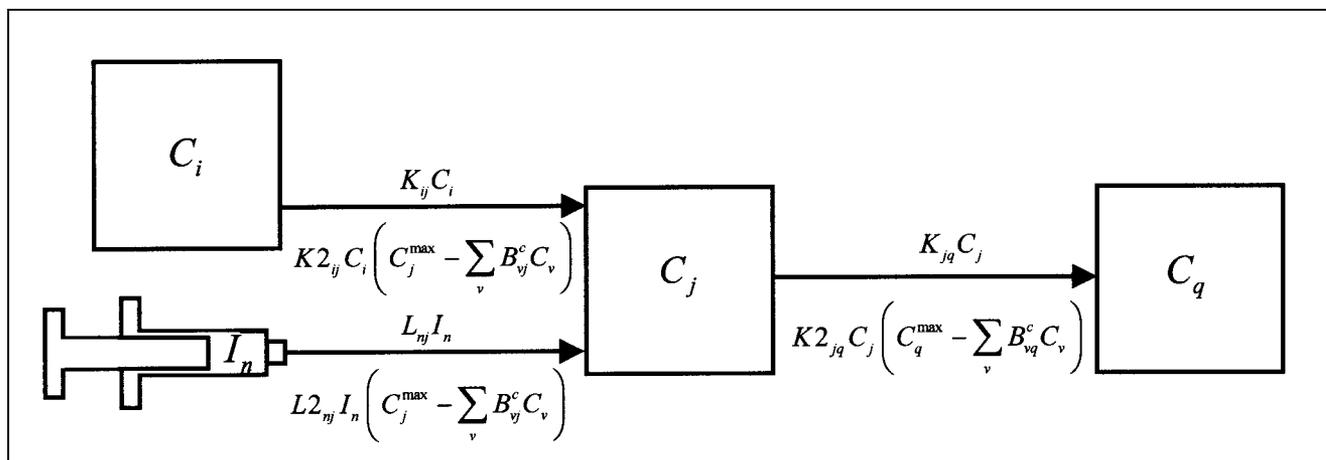


FIGURE 1. Building blocks used to define compartment models. Compartments are shown as rectangles. Arrows indicate unimolecular (labeled above arrow) and bimolecular (labeled below arrow) fluxes.

In agreement with this framework, the equation describing the pharmacokinetics of compartment j is

$$\begin{aligned} \frac{dC_j}{dt} = & \left(C_j^{\max} - \sum_v B_{vj}^c C_v \right) \sum_i L_{2ij} I_i + \\ & \left(C_j^{\max} - \sum_v B_{vj}^c C_v \right) \sum_i K_{2ij} C_i + \sum_i L_{ij} I_i + \\ & \sum_i K_{ij} C_i - C_j \sum_q \left[K_{2jq} \left(C_q^{\max} - \sum_v B_{vq}^c C_v \right) \right] - \\ & C_j \sum_q K_{jq}, \end{aligned} \quad \text{Eq. 1}$$

where K_{ij} is the unimolecular rate constant from compartment i to j , K_{2ij} is the bimolecular rate constant from compartment i to j , L_{ij} is the unimolecular rate constant from input i to compartment j , and L_{2ij} is the bimolecular rate constant from input i to compartment j . Note that for a given compartment j , one expects physiologic uptake to be either nonsaturable or saturable. In the former case, K_{2ij} will equal zero for all i . In the latter case, K_{ij} will equal zero for all i . Analogous expectations hold for L and L_2 .

Output Equation. The output equation relates compartment and input concentrations to the measurable quantities. The output equation appropriate for the majority of nuclear medicine data is

$$M_l(t_b, t_c) = \frac{1}{t_c - t_b} \int_{t_b}^{t_c} \left(\sum_u W_{ul} A_u C_u + \sum_u X_{ul} I_u \right) dt, \quad \text{Eq. 2}$$

where M_l is output l of the model, W_{ul} is the relative contribution of compartment u to output l , A_u is the exponentially decaying specific activity of material in compartment u , and X_{ul} is the relative contribution of input i to output l . The indicated integration accounts for the temporal averaging inherent to experimental measurements. The equation relates C and I , concentrations at specific points in time, to these values averaged over the time interval of a scan that begins at t_b and ends at t_c . In the simple case, values of W_{ul} might be set to one for compartments contributing to output l and zero otherwise. Alternatively, other values could be used to account for partial volume, scatter, or other effects. Analogously, X_{ul} could be set to a value reflecting the contribution of intravascular radioactivity to the measurement.

To facilitate computation of M_l , we define and integrate the next equation along with the state (Eq. 1):

$$\frac{dO_l}{dt} = \sum_u W_{ul} A_u C_u + \sum_u X_{ul} I_u, \quad \text{Eq. 3}$$

so that M_l may be calculated as

$$M_l(t_b, t_c) = \frac{O_l(t_c) - O_l(t_b)}{t_c - t_b}. \quad \text{Eq. 4}$$

Example. For the sake of concreteness, we digress briefly with an example of how the general form of the state and output equations may be used in a PET receptor model wherein there is a single output corresponding to the concentration of radioactivity in a region of interest. First, one defines compartment

concentrations in terms of molar concentrations, because this is what drives ligand–receptor interactions. Second, one defines separate inputs I for the plasma concentration of ligand and for the blood concentration of radioactivity. The former is the source of radioactivity to the compartments, whereas the latter is used to account for vascular radioactivity by setting the appropriate X_{ul} equal to the tissue vascular space. Third, values for W_{ul} corresponding to free and bound compartments are set to equal the tissue extravascular space and thus define the contributions to the output. Finally, A_u , the specific activity, is used to convert the molar concentration in compartment u to the measured quantity, which is radioactivity concentration.

Sensitivity Equations. Often one uses the model output to fit experimental data to estimate values of model parameters that have physiologic significance. Typically, the parameter estimates are obtained by minimizing a weighted least-squares objective function

$$\phi(\theta) = \sum_l \sum_i \frac{1}{\sigma_{li}^2} (M_l(t_i, t_{i+1}) - P_l(t_i, t_{i+1}))^2, \quad \text{Eq. 5}$$

where P is the experimental data, σ_{li}^2 is (an estimate of) the variance of $P_l(t_i, t_{i+1})$, and θ is the vector of model parameters. To minimize ϕ , it is advantageous to use the output sensitivities—the derivatives of M with respect to elements of θ . To derive the appropriate expressions, we differentiate analytically the output equation

$$\frac{dM_l}{d\theta_j} = \frac{1}{t_{i+1} - t_i} \left[\frac{d}{d\theta_j} O_l(t_{i+1}) - \frac{d}{d\theta_j} O_l(t_i) \right]. \quad \text{Eq. 6}$$

To evaluate $dO_l/d\theta_j$ we differentiate analytically (Eq. 3) with respect to parameter θ_j and then change the order of differentiation (10) to obtain

$$\frac{d}{dt} \left(\frac{dO_l}{d\theta_j} \right) = \sum_u \frac{d}{d\theta_j} (W_{ul} A_u C_u) + \sum_u \frac{d}{d\theta_j} (X_{ul} I_u), \quad \text{Eq. 7}$$

which is an ordinary differential equation (ODE) that can be solved simultaneously with Equations 1 and 2 to obtain $dO_l/d\theta_j$. To evaluate Equation 7, the indicated derivatives on the right hand side are evaluated analytically as follows:

$$\begin{aligned} \frac{d}{d\theta_j} (W_{ul} A_u C_u) = & W_{ul} A_u \frac{dC_u}{d\theta_j} \\ & + W_{ul} C_u \frac{dA_u}{d\theta_j} + A_u C_u \frac{dW_{ul}}{d\theta_j} \end{aligned} \quad \text{Eq. 8}$$

$$\frac{d}{d\theta_j} (X_{ul} I_u) = X_{ul} \frac{dI_u}{d\theta_j} + I_u \frac{dX_{ul}}{d\theta_j}. \quad \text{Eq. 9}$$

With the exception of $dC_u/d\theta_j$, the derivatives on the right hand sides of Equations 8 and 9 are evaluated analytically. For example, if W_{ul} is vascular fraction, $dW_{ul}/d\theta_j$ equals unity when element j of the parameter vector θ is vascular fraction and equals zero if W_{ul} is independent of θ_j . To evaluate $dC_u/d\theta_j$ analytically, the state equation (Eq. 1) is differentiated, and the order of differentiation is changed to obtain

$$\frac{d}{dt} \left(\frac{dC_u}{d\theta_j} \right) = \frac{d}{d\theta_j} \left\{ \left(C_u^{\max} - \sum_v B_{vu}^c C_v \right) \sum_i L_{2iu} I_i + \left(C_q^{\max} - \sum_v B_{vu}^c C_v \right) \sum_u K_{2iu} C_i + \sum_i L_{iu} I_i + \sum_i K_{iu} C_i - C_u \sum_q \left[K_{2uq} \left(C_q^{\max} - \sum_v B_{vq}^c C_v \right) \right] - C_u \sum_q K_{uq} \right\}. \quad \text{Eq. 10}$$

This too is an ODE that can be integrated simultaneously with the other equations to obtain $dC_u/d\theta_j$. To evaluate the right hand side of Equation 10, we proceed analytically with the differentiation to yield expressions of the form $dC_n/d\theta_j$, $dC_n^{\max}/d\theta_j$, $dK_{in}/d\theta_j$, $dK_{2in}/d\theta_j$, $dL_{in}/d\theta_j$, and $dL_{2in}/d\theta_j$. Terms of the first type are solutions of the ODEs being solved, and the other terms have trivial values. For example, $dK_{in}/d\theta_j$ equals unity if parameter θ_j is K_{in} and zero if K_{in} is independent of θ_j .

Jacobian Equation. An ODE solver is used to solve numerically the state, output, and sensitivity equations, which are in the form

$$\frac{dy}{dt} = F(t, y). \quad \text{Eq. 11}$$

F is the vector-valued function defined according to Equations 1 through 10, and y is a vector of values of state, output, and sensitivity functions. To solve these equations, we used the stiff

ODE solvers written by Shampine and Reichelt (11) that are distributed with MATLAB. As is common for stiff ODE solvers, these require use of the Jacobian $J = dF/dy$, which can be estimated by numeric differentiation or by differentiating F analytically. We chose the analytic route, because, among other things, it is more computationally efficient. These equations are omitted here for the sake of brevity but are included in the *COMKAT System User Manual*.

Implementation

COMKAT was developed within MATLAB 5.0 and consists of functions for defining compartment models, solving them to obtain model output, and estimating parameter values. These features are supported both by command line and graphic user interfaces. All functions were written as m-files, which maximizes portability, because they will run, without compilation or modification, on any computer with MATLAB 5.0 or newer.

RESULTS

Examples

Blood Flow. Figure 2 depicts a screen snapshot for the PET blood flow model (12–14). It contains one tissue compartment corresponding to an extravascular concentration of the tracer. It also contains two inputs—one corresponding to the molar concentration of the tracer and one corresponding to the blood radioactivity concentration of the tracer. This latter input is used to account for vascular radioactiv-

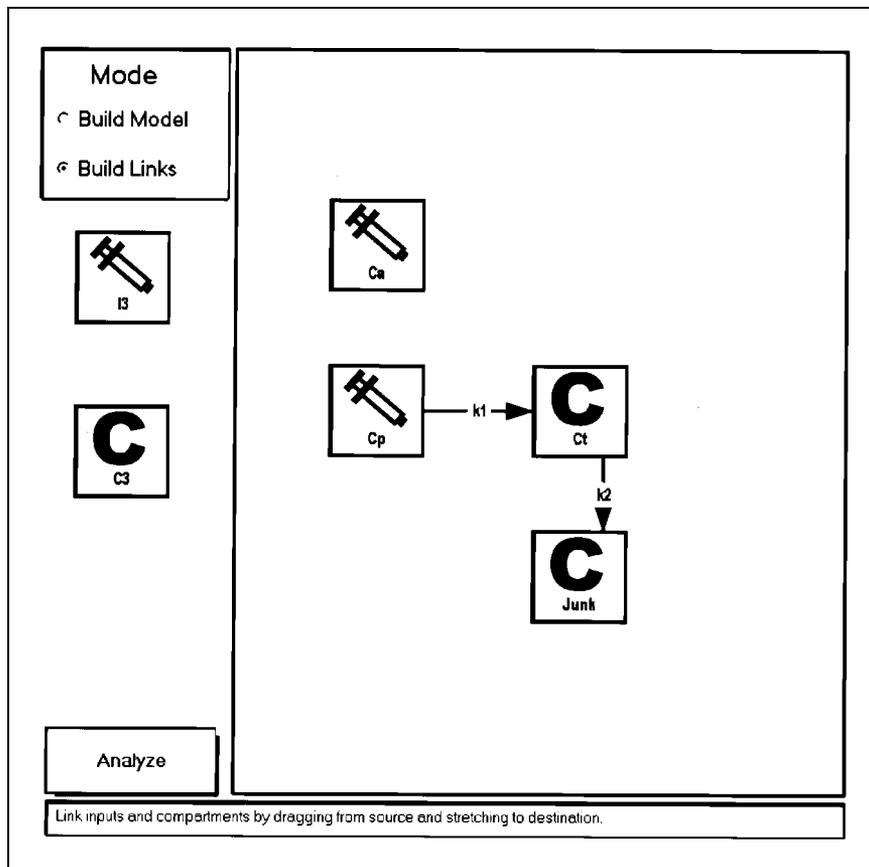


FIGURE 2. Graphic user interface to define the blood flow model. Model was implemented by dragging compartments and inputs from left (labeled I3 and C3), switching to “Build Links” mode, and dragging arrows between inputs and compartments. For blood flow model, output includes contributions from C_t but not from “Junk,” which serves as a sink.

ity. While this model is sufficiently simple that vascular radioactivity could be modeled without separate inputs, we chose this representation to illustrate the approach that is necessary in more complex models.

FDG Metabolism. Figure 3 depicts a screen snapshot for the FDG metabolism model (15–19). It contains two tissue compartments corresponding to FDG and to FDG-6-PO₄. Inputs were defined for both the plasma molar concentration of FDG and for the blood concentration of radioactivity. This latter input is used to account for vascular radioactivity.

Receptor Model. Figure 4 depicts a screen snapshot for the receptor model of Delforge et al. (7,8) which properly accounts for multiple injections of ligand at different specific activities (9). Example solutions are shown in Figure 5, and a MATLAB script showing command-line interface is included in Figure 6. For these solutions, model parameters were set to values defined as “solution 1” and “solution 2” (8), and the inputs were defined using data kindly provided by Jacques Delforge. The model output in Figure 5 agrees well with that shown elsewhere (Fig. 7) (7). Note that to reproduce this result, the contribution of vascular radioactivity to model output X was set to zero, and the model output was expressed in a molar concentration by setting initial specific activity to unity and the decay constant (ln 2/half-life) to zero.

Validation

Model outputs for the three example models were validated by comparison with prior model-specific implementations of receptor models developed previously by us (20, 21), by comparison with model solutions published by other institutions (8,9) and with analytic solutions of blood flow and FDG models.

Performance

Solving Model Equations. Although COMKAT is written to solve compartment models of arbitrary, user-specified configurations, it is nevertheless efficient. This is illustrated in Table 1, which lists time to solve the three example models both with and without sensitivity functions, on a Pentium 2/366 (Intel Corp., Santa Clara, CA) notebook computer running MATLAB 5.3 and Windows NT 4 (SP4). Times were measured with MATLAB’s clock function, and the set of ODEs was solved with ODE15s (11), using absolute and relative tolerances of 10⁻⁴ and 10⁻³, respectively.

Parameter Estimation. To evaluate COMKAT’s performance in parameter estimation, we created simulation data using the blood flow, FDG, and receptor models. Using the same input functions used for *Solving Model Equations*, simulated noise-free data were generated by solving the model using the “True” parameter values listed in Table 2 to obtain the outputs $M_1(t_i, t_{i+1})$. Noisy simulation data were

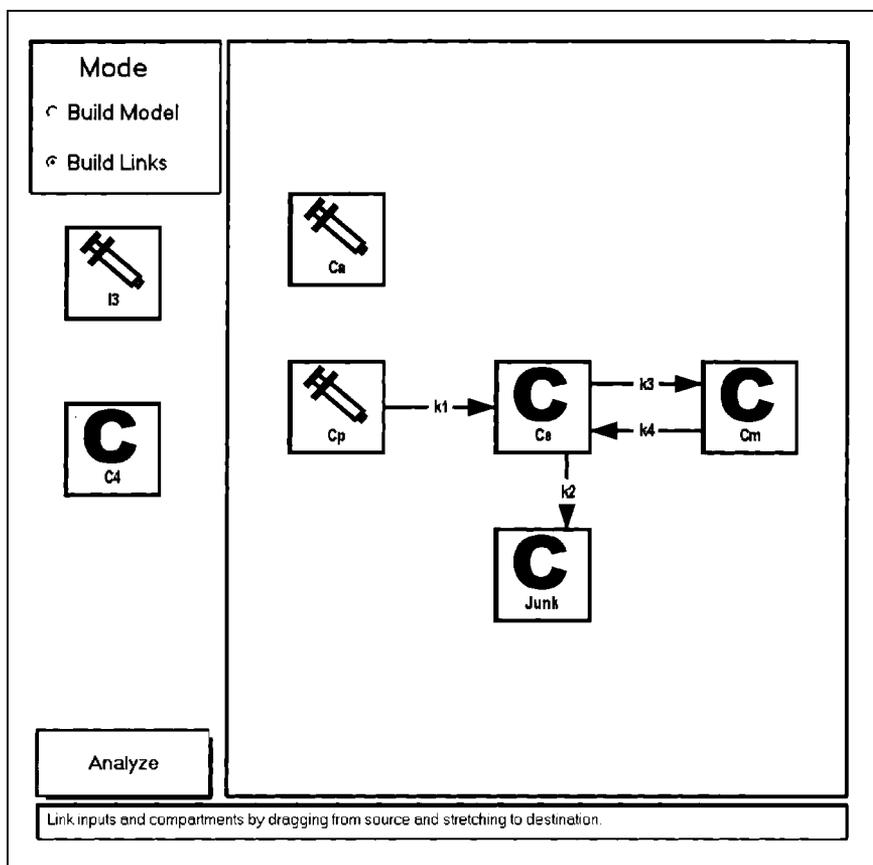


FIGURE 3. Graphic user interface used to define FDG metabolism model. Implementation is analogous to that of blood flow model in Figure 2.

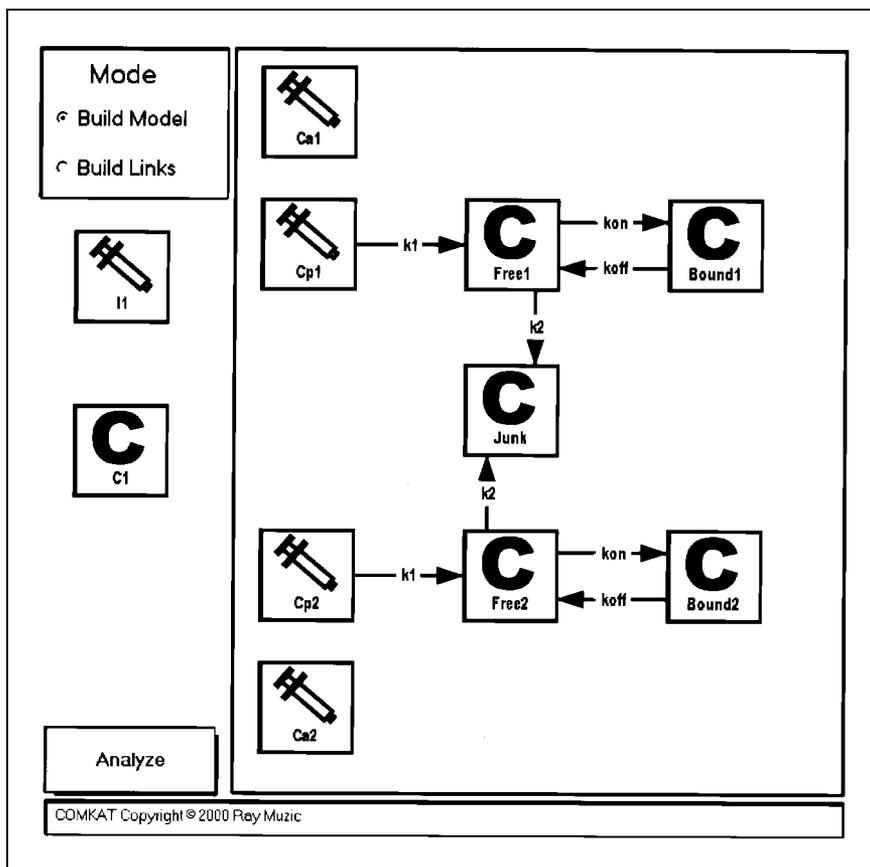


FIGURE 4. Graphic user interface used to define receptor model. Implementation is analogous to that of blood flow model in Figure 2, except that saturation values are defined for “Bound1” and “Bound2” compartments using model parameter B_{max} . Because same parameter was referenced as saturation value of both compartments, COMKAT treats this as indication of common pool of receptors.

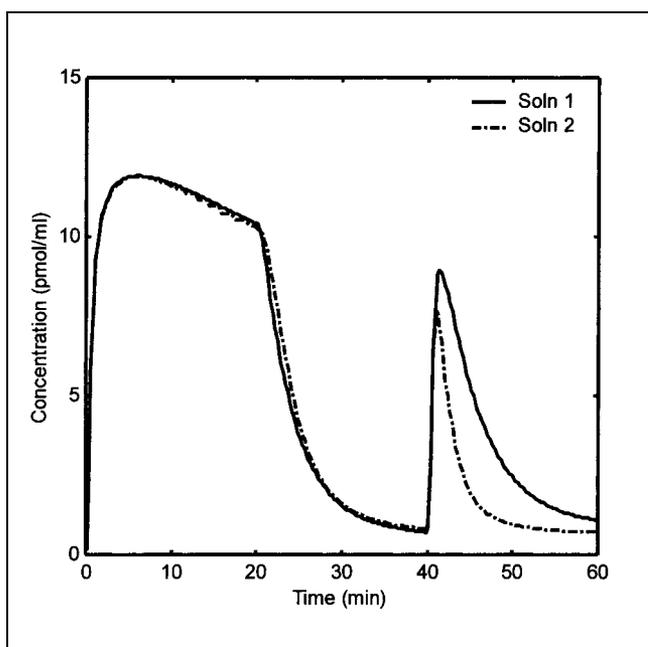


FIGURE 5. Model outputs for receptor model in Figure 4. Solutions were generated using parameter values for solution (“Soln”) 1 and solution 2 (defined elsewhere [8]; these curves closely agree with curves shown elsewhere [8, Fig. 6]).

created by adding to the noise-free data a sample from a normal distribution with zero mean and $0.02M_1(t_i, t_{i+1})/\sqrt{t_{i+1} - t_i}$ SD. Using the weighted least-squares objective function (Eq. 5) with σ_{ii} set equal to this SD, the parameter values were estimated from the noisy simulation data beginning with initial guesses listed in Table 2. In this process COMKAT’s “Fit()” function made use of MATLAB’s “lsqcurvefit” (24) function to minimize $\phi(\theta)$ with the lower and upper bound constraints listed in Table 2. For the FDG and receptor models, termination criteria for the minimization were set to 10^{-4} for TolFun and 10^{-6} for TolX. For the blood flow model, the same value for TolX was used, whereas the value for TolFun was reduced to 10^{-5} because of the relatively fewer data points in its simulated data.

As an example, Figure 7 shows the noisy simulation dataset, the initial guess, and the fit from the receptor model. For all three models, 10 runs with independent noise realizations were performed. The mean (SD) of time required to estimate the parameters was 0.09 (0.03), 0.86 (0.16), and 12 (1.0) min for the blood flow, FDG, and receptor models, respectively. In all cases, the fits converged, and the parameter estimates were not significantly different from their true values (paired t test) in spite of a relatively poor initial guess. The MATLAB commands to perform this parameter estimation for the receptor model are given in Figure 8. The parameter estimation example shows robustness.

```

cm = CompartmentModel; % create empty model object

% define inputs piecewise polynomials (protocol 4 of Delforge 1990)
% (code removed for brevity)

% add compartments to model
% Bound1 & 2 are saturable, sharing a common pool of Bmax receptors
cm = AddCompartment(cm, 'Free1', '');
cm = AddCompartment(cm, 'Bound1', 'Bmax');
cm = AddCompartment(cm, 'Junk', '');
cm = AddCompartment(cm, 'Free2', '');
cm = AddCompartment(cm, 'Bound2', 'Bmax');

% define interconnections
cm = AddLink(cm, 'K', 'Free1', 'Junk', 'k2');
cm = AddLink(cm, 'K', 'Free2', 'Junk', 'k2');
cm = AddLink(cm, 'K', 'Bound1', 'Free1', 'koff');
cm = AddLink(cm, 'K', 'Bound2', 'Free2', 'koff');
cm = AddLink(cm, 'K2', 'Free1', 'Bound1', 'kon');
cm = AddLink(cm, 'K2', 'Free2', 'Bound2', 'kon');
cm = AddLink(cm, 'L', 'Cp1', 'Free1', 'K1');
cm = AddLink(cm, 'L', 'Cp2', 'Free2', 'K1');

% define parameter values
cm = AddParameter(cm, 'K1', 0.293); % ml tiss/ml plasma*min
cm = AddParameter(cm, 'k2', 0.273); % 1/min
cm = AddParameter(cm, 'kon', 5.1); % ml/pmol*min
cm = AddParameter(cm, 'koff', 2.8); % 1/min
cm = AddParameter(cm, 'Bmax', 15.8); % pmol/ml
cm = AddParameter(cm, 'Fv', .08); % ml blood/ml tissue
cm = AddParameter(cm, 'sa1', 1); % spec act(t) = sa1*exp(-dk*t)
cm = AddParameter(cm, 'sa2', 0);
cm = AddParameter(cm, 'dk', 0);
% note Delforge paper uses decay corrected quantities
% so, for replication purposes, don't include decay here

% contributions of compartments to output
Wlist = {'Free1', '1'; 'Bound1', '1'; 'Free2', '1'; 'Bound2', '1'};
% contributions of inputs to output
Xlist = {'Ca1', 'Fv'; 'Ca2', 'Fv'};
cm = AddOutput(cm, 'PET', Wlist, Xlist);

cm = AddSensitivity(cm, 'K1', 'k2', 'kon', 'koff', 'Bmax'); % if sensitivity

cm = set(cm, 'ScanTime', [[0:0.25:59.75] [0.25:0.25:60]]); % 0.25 min frames

% set the options for the ODE solver
solver_options.name = 'ode15s';
solver_options.options = odeset('InitialStep',1/60,'AbsTol',1e-4,...
    'Jacobian','on','RelTol',1e-3);
cm = set(cm, 'SolverOptions', solver_options);

% solve model and plot output
[PET, PETIndex, Output, OutputIndex]=Solve(cm);
plot(0.5*sum(PET(:, [1 2]),2),PET(:,3)); % [1 2] frame times, [3] uCi/ml

```

FIGURE 6. MATLAB commands for implementing receptor model shown in Figure 4 to produce output curves shown in Figure 5.

DISCUSSION

A key point regarding the model formulation and implementation is that arbitrary model configurations are supported. By this we mean that a user is not restricted to using predefined models distributed with COMKAT. Rather, any model of the user's design can be set up. Moreover, COMKAT can support models with a large number of compartments. For example, we have already implemented models with 12 compartments. Thus, COMKAT might be useful beyond nuclear medicine in applications such as drug development, in which whole-body distribution models are used.

Several features of the model formulation are noteworthy. First, it is based on molar concentrations for the compartments because this quantity and not radioactivity governs the pharmacokinetics. Second, the model directly accounts for radioactive decay through the use of the time-varying specific activity A_u . This approach is preferable to the alternative of precorrecting data for radioactive decay

because this method facilitates use of optimal statistical weights in parameter estimation (25). Moreover, our approach of accounting for decay in the output equation can be shown analytically to produce model output identical to that obtained when radioactive decay is included directly in the state equations. Notably, our approach avoids the added complexity of including decay terms in the state equations. Third, the formulation supports multiple-injection receptor models in that both saturation and injections with different specific activities are addressed. Fourth, analytically derived sensitivity and Jacobian equations are used. This improves the numeric accuracy and computational efficiency compared with approximating these by numeric differentiation (10). It is important that, because we derived the sensitivity and Jacobian equations for the general case, no derivation is required on the part of the software user. Fifth, the formulation supports multiple output equations. This feature could be used, for example, to analyze data simultaneously from

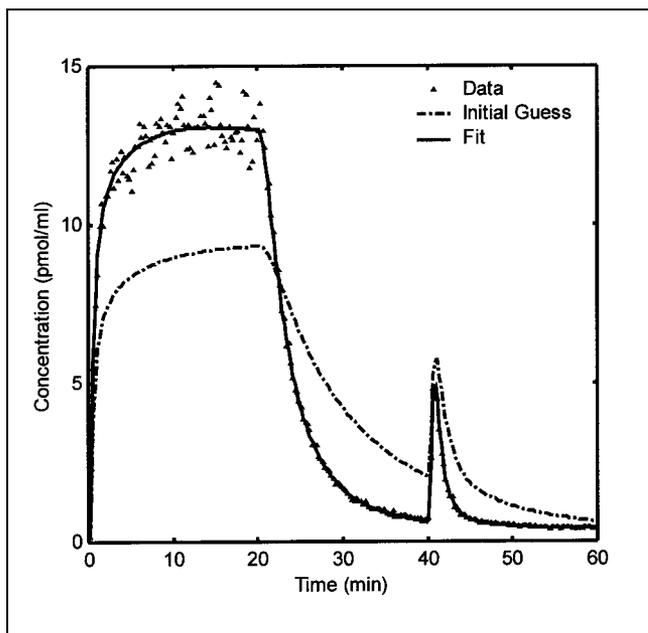


FIGURE 7. Example plot from parameter estimation. Plot shows simulation data including “noise” (triangles), initial guess (dashed curve), and final fit (solid curve).

multiple regions, wherein some of the rate constants for some of the regions might be required to have a common value.

The examples illustrate that models of a range of complexities can be solved. Although we placed particular emphasis on the Delforge receptor model, because it is one of the more complex models, it is important to emphasize that the user is not constrained to use the models provided. Using the graphic user interface and without any mathematics derivations, a user can implement any model configuration he or she defines by simply positioning rectangles for the compartments and inputs and arrows for the connec-

tions. Alternatively, models may be implemented and solved from the command line (Figs. 6 and 8). This is particularly useful for automating data analysis tasks such as parameter estimation.

Figures 7 and 8 list the commands used in evaluating performance in solving model equations and in parameter estimation. A few points become apparent from these listings. First, no derivations were required on the part of the user. Second, an object-oriented syntax is used to define and solve models and to estimate parameters. This provides a simple syntax that is easily extendable while minimizing backward-compatibility problems.

One might expect that the computational performance would be poor because MATLAB is an interpreted language. The timing results show that this is not the case. Indeed, in implementing Equations 1 through 10, care was taken to avoid loops and to instead use MATLAB’s highly optimized matrix-vector operations to implement the summations.

In our timing results, we used ode15s to solve the system of differential equations. This solver is intended for solving stiff systems. One method for ascertaining whether a stiff solver is necessary is to simply compare solution times of a stiff solver and a nonstiff solver while keeping all other factors unchanged. The presence of stiffness can be diagnosed if the stiff solver solves the equations more efficiently. Indeed, this is the recommended approach for detecting stiffness in the MATLAB ODE Suite documentation (11). Applying this technique, we ascertained that the blood flow, FDG, and receptor models are stiff. Notably, because the ODE solvers in the ODE Suite use the same call syntax, switching between stiff and nonstiff solvers is nearly trivial; it requires only changing the name of the solver to call.

Other software for compartment modeling includes BLD (2), KMZ/PKIN (3), and SAAM/SAAM II (4), to name a few. In comparison with each of these, COMKAT has a

TABLE 1
Computational Performance Summary

Model	Sensitivity equations	Frames	Without sensitivity equations		With sensitivity equations	
			ODEs	Time (s)	ODEs	Time (s)
Blood flow	K_1, k_2	18×5 s	3	0.426 (0.010)	9	0.627 (0.005)
FDG	K_1, k_2, k_3, k_4	$12 \times 0.16, 10 \times 0.5, 10 \times 2,$ $10 \times 5, 4 \times 10$ min	4	0.504 (0.012)	20	1.144 (0.010)
Receptor	$K_1, k_2, k_{on}, k_{off}, B_{max}$	240×0.25 min	6	3.904 (0.071)	66	13.629 (0.258)

Times are mean (SD) of 10 runs. ODEs is number of ODEs solved and includes output, state, and, if indicated, sensitivity equations. These equations were automatically generated by COMKAT. For blood flow, plasma input function was specified as $C_p(t) = 0.055(t/60)^6 e^{-t/60/1.8} + 10(t/60)e^{-t/60/22} + 0.7(t/60)e^{-t/60/180}$, $t \geq 0$, and blood radioactivity input function as $C_a(t) = C_p(t)e^{-2.033t}$, with t in minutes. For FDG, plasma input function was specified as $C_p(t) = 650 e^{-6.7t} + 146e^{-0.25t} + 105e^{-0.03t} + 21e^{-0.0001t}$, $t \geq 0$, with t in minutes (26). For receptor, plasma input function was obtained by linearly interpolating input function data for protocol 4 from (8). For both FDG and receptor models, contribution from vascular radioactivity was not included, because these input functions were not available.

TABLE 2
Parameter Estimation Results

Model	Parameter*	True value	Initial guess	Bounds		Estimate†		Error‡ (%)	
				Lower	Upper	Mean	SD	Mean	SD
Blood flow	K ₁	0.5	0.4	0.1	5.0	0.491	0.013	-1.83	2.68
	k ₂	0.5	0.6	0.1	5.0	0.467	0.056	-6.70	11.87
FDG	K ₁	0.102	0.05	0.01	2.0	0.102	0.0013	-0.08	1.30
	k ₂	0.13	0.2	0.001	1.0	0.130	0.0050	0.01	3.86
	k ₃	0.062	0.1	0.01	0.2	0.0618	0.0022	-0.29	3.64
	k ₄	0.0068	0.005	0.001	0.01	0.00667	0.00023	-1.93	3.43
Receptor	K ₁	0.29	0.2	0.1	2.0	0.292	0.0026	0.62	0.91
	k ₂	1.8	1.0	0.05	5.0	1.81	0.016	0.57	0.89
	k _{on}	2.3	1.0	0.5	10.	2.26	0.19	-1.78	8.43
	k _{off}	0.294	0.1	0.05	10.	0.293	0.002	-0.28	0.69
	B _{max}	37.1	25	10.	100.	36.98	0.35	-0.31	0.93

*Values for parameter k_{on} are expressed in units of mL/(pmol min), for parameter B_{max} in units of pmol/mL, and for all other parameters in units of min⁻¹.

†Estimates are obtained from fitting model to 10 noise realizations as described in text and using input functions described in Table 1.

‡Errors are calculated as mean (estimate - true)/true and SD(estimate - true)/true.

unique combination of features that makes it particularly well suited for research. COMKAT supports saturable and nonsaturable kinetics, has analytic sensitivity equations, and

can be run on any computer for which MATLAB is available. Moreover, users may integrate their own application-specific pre- and postprocessing tasks (e.g., input function

```
% create model and define input as shown in Figure 6
% (commands omitted for brevity)

% request sensitivity functions for parameters to be estimated
cm = AddSensitivity(cm, 'K1', 'k2', 'kon', 'koff', 'Bmax');
cm = set(cm, 'ScanTime', [[0:0.25:59.75] [0.25:0.25:60]']); % 0.25 min frames

% set the options for the ODE solver
solver_options.name = 'ode15s';
solver_options.options = odeset('InitialStep',1/60,'AbsTol',1e-4,...
    'Jacobian','on','RelTol',1e-3);
cm = set(cm, 'SolverOptions', solver_options);

% solve model to obtain noise-free data
[PET, PETIndex, Output, OutputIndex]=Solve(cm);
% PET(:, [1 2]) are frame start and end times
% PET(:,3) are corresponding radioactivity concentrations

% set options for lsqcurvefit (used by Fit)
cm = set(cm, 'OptimizerOptions', ...
    optimset('Jacobian','on', 'TolFun', 1e-4, 'TolX', 1e-5));

et=zeros(10,1); % elapsed time
nit= zeros(10,1); % number of iterations
pfit = zeros(10,5); % parameter estimates

% parameter true values and initial guess
ptrue = [pxeval(cm,'K1') pxeval(cm,'k2') pxeval(cm,'kon') ...
    pxeval(cm,'koff') pxeval(cm,'Bmax')];
pinit = [ .2 1. 1 .1 25];
lb = [ .1 .05 .5 .05 10]; % lower bounds
ub = [2. 5. 10. 10. 100]; % upper bounds

for i=1:10,
    sd = 0.02*PET(:,3)./sqrt(PET(:,2)-PET(:,1));
    data=sd.*randn(size(PET(:,3)))+PET(:,3);
    plot(0.5*sum(PET(:, [1 2]),2),PET(:,3),0.5*sum(PET(:, [1 2]),2),data, '.')
    cm=set(cm, 'ExperimentalData', data);
    cm=set(cm, 'ExperimentalDataSD', sd);
    tic; % start timer
    [pp,PETfit, resnorm, residual, exitflag, output]=Fit(cm,pinit,lb,ub);
    et(i) = toc; % record elapsed time
    pfit(i,:) = pp; % store parameter estimate
    nit(i) = output.iterations;
end

fprintf('Fit time: %3.3f (%3.3f)\n', mean(et), std(et));
fprintf('Number of iterations: %3.3f (%3.3f)\n', mean(nit), std(nit));
```

FIGURE 8. MATLAB commands for implementing receptor model shown in Figure 4 to obtain parameter estimates summarized in Table 2.

smoothing, statistical summary) with COMKAT's modeling analysis by implementing such tasks in MATLAB. Finally, MATLAB is distributed in source-code form without cost to interested researchers.

CONCLUSION

COMKAT is a MATLAB-based compartment- modeling package. We believe it has many features desirable for analysis of nuclear medicine data, and we use it routinely in our own research. COMKAT is available at www.nuclear.uhrad.com/comkat. We hope others find it useful and share improvements and new features.

ACKNOWLEDGMENTS

The authors thank Daniel Pendergast, Joan Zhang, and Jigar Patel for contributions to the programming effort; Evan D. Morris for constructive comments made on a preliminary draft of the manuscript; Nathaniel Alpert for valuable discussions and for providing feedback on the software; and Larry Shampine for discussions on numerically solving sets of differential equations. This study was supported in part by National Institutes of Health grant R01 HL62399.

REFERENCES

- COMKAT Web site. Available at: <http://www.nuclear.uhrad.com/comkat/>. Accessed February 2, 2001.
- Carson RE, Huang SC, Phelps M. BLD: a software system for physiological data handling and model analysis. In: *Proceedings of the 5th Annual Symposium on Computer Application in Medical Care, Washington, DC, 1-4 November 1981*. Los Alamitos, CA: IEEE; 1981:62-565.
- Burger C, Buck A. Requirements and implementation of a flexible kinetic modeling tool. *J Nucl Med*. 1997;38:1818-1823.
- Barrett PH, Bell BM, Cobelli C, et al. SAAM II: simulation, analysis, and modeling software for tracer and pharmacokinetic studies. *Metabolism*. 1998;47:484-492.
- Friston KJ, Worsley KJ, Frackowiak RSJ, Mazziotta JC, Evans AC. Assessing the significance of focal activations using their spatial extent. *Hum Brain Mapp*. 1994;1:214-220.
- Friston KJ, Holmes AP, Worsley KJ, Poline JP, Frith CD, Frackowiak RSJ. Statistical parametric maps in functional imaging: a general linear approach. *Hum Brain Mapp*. 1995;2:189-210.
- Delforge J, Syrota A, Mazoyer BM. Experimental design optimisation: theory and application to estimation of receptor model parameters using dynamic positron emission tomography. *Phys Med Biol*. 1989;34:419-435.
- Delforge J, Syrota A, Mazoyer BM. Identifiability analysis and parameter identification of an in vivo ligand-receptor model from PET data. *IEEE Trans Biomed Eng*. 1990;37:653-661.
- Morris ED, Alpert NM, Fischman AJ. Comparison of two compartmental models for describing receptor ligand kinetics and receptor availability in multiple injection PET studies. *J Cereb Blood Flow Metab*. 1996;16:841-853.
- Bard Y. *Nonlinear Parameter Estimation*. New York, NY: Academic Press; 1974:225-231.
- Shampine LF, Reichelt MW. *The MATLAB ODE Suite*. The MathWorks, Inc.; 1997: <ftp://ftp.mathworks.com/pub/mathworks/toolbox/matlab/funfun/odesuite.ps>.
- Herscovitch P, Markham J, Raichle ME. Brain blood flow measured with intravenous $H_2^{15}O$. I. Theory and error analysis. *J Nucl Med*. 1983;24:782-789.
- Raichle ME, Martin WR, Herscovitch P, Mintun MA, Markham J. Brain blood flow measured with intravenous $H_2^{15}O$. II. Implementation and validation. *J Nucl Med*. 1983;24:790-798.
- Kety SS. The theory and applications of the exchange of inert gas at the lungs and tissues. *Pharmacol Rev*. 1951;3:1-41.
- Huang SC, Phelps ME, Hoffman EJ, Sideris K, Selin CJ, Kuhl DE. Noninvasive determination of local cerebral metabolic rate of glucose in man. *Am J Physiol*. 1980;238:E69-E82.
- 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*. 1979;6:371-388.
- Reivich M, Kuhl D, Wolf A, et al. Measurement of local cerebral glucose metabolism in man with ^{18}F -2-fluoro-2-deoxy-d-glucose. *Acta Neurol Scand Suppl*. 1977;64:190-191.
- Sokoloff L. [$1-^{14}C$]-2-deoxy-d-glucose method for measuring local cerebral glucose utilization: mathematical analysis and determination of the "lumped" constants. *Neurosci Res Program Bull*. 1976;14:466-468.
- Sokoloff L, Reivich M, Kennedy C, et al. The [^{14}C]deoxyglucose method for the measurement of local cerebral glucose utilization: theory, procedure, and normal values in the conscious and anesthetized albino rat. *J Neurochem*. 1977;28:897-916.
- Muzic RFJ, Saidel GM, Zhu N, Nelson AD, Zheng L, Berridge MS. Iterative optimal design of PET experiments for estimating β -adrenergic receptor concentration. *Med Biol Eng Comput*. 2000; 38:595-602.
- Muzic RF Jr, Berridge MS, Friedland RP, et al. PET quantification of specific binding of carbon-11-nicotine in human brain. *J Nucl Med*. 1998;39:2048-2054.
- Muzic RF Jr, Berridge MS, Zheng L, et al. PET measurement of beta-adrenergic receptor concentration using [^{18}F]fluorocarazolol [abstract]. *J Nucl Med*. 1996; 37(suppl):174.
- Muzic RF Jr, Nelson AD, Saidel GM, et al. Optimal experimental design for receptor modeling [abstract]. *Ann Biomed Eng*. 1994;22:43.
- Coleman T, Branch M, Grace A. *Optimization Toolbox for Use with MATLAB, User's Guide*. South Natick, MA: The MathWorks, Inc., 1999.
- Glatting G, Reske SN. Treatment of radioactive decay in pharmacokinetic modeling: influence on parameter estimation in cardiac ^{15}N -PET. *Med Phys*. 1999; 26:616-621.
- Huang SC, Phelps ME, Hoffman EJ, Kuhl DE. Error sensitivity of fluorodeoxyglucose method for measurement of cerebral metabolic rate of glucose. *J Cereb Blood Flow Metab*. 1981;1:391-401.