Supplementary data

Performance Assessment and Translation of Physiologically Based Pharmacokinetic Models from acslX™ to Berkeley Madonna™, MATLAB®, and R language: Oxytetracycline and Gold Nanoparticles as Case Examples

Zhoumeng Lin,*,[1] Majid Jaberi-Douraki,*,[†] Chunla He,[‡] Shiqiang Jin,*,[§] Raymond S. H. Yang,[¶,||] Jeffrey W. Fisher,[|||] Jim E. Riviere,*

*Institute of Computational Comparative Medicine (ICCM), Department of Anatomy and Physiology, College of Veterinary Medicine, Kansas State University, Manhattan, Kansas 66506, USA; [†]Department of Mathematics, College of Arts and Sciences, Kansas State University, Manhattan, Kansas 66506, USA; [‡]Department of Epidemiology and Biostatistics, College of Public Health, The University of Georgia, Athens, Georgia 30602, USA; [§]Department of Statistics, College of Arts and Sciences, Kansas State University, Manhattan, Kansas 66506, USA; [¶]Department of Environmental and Radiological Health Sciences, College of Veterinary Medicine and Biomedical Sciences, Colorado State University, Fort Collins, Colorado 80523, USA; [||]Ray Yang Consulting LLC, Fort Collins, Colorado 80526, USA; [|||]Division of Biochemical Toxicology, National Center for Toxicological Research, U.S. Food and Drug Administration, Jefferson, Arkansas 72079, USA.

[1] To whom correspondence should be addressed: zhoumeng@ksu.edu; Phone: 785-532-4087; Fax: 785-532-4953; Address: 1800 Denison Avenue, P200 Mosier Hall, Manhattan, KS 66506.

Contents Page

From acslX™ to Berkeley Madonna™

Conversion of comment, constant, section, and method statements

After copying the model code from acslX™ and pasting it to the "Equations" editor in Berkeley Madonna™, there are four general modifications that need to be done for every model. First, change the comment symbol (the exclamation mark "!" used in acslX™) to the semicolon character ";" in Berkeley Madonna™. Alternatively, comments can be placed anywhere in the equations by enclosing them in curly brackets (below as an example).

BW = 11.3 ;Body weight (kg)

BW = 11.3 {Body weight (kg)}

Second, delete the "**CONSTANT**" command for all constants because if a fixed value is assigned to a parameter, it will be recognized as a constant in Berkeley Madonna™ (extra "**CONSTANT**" command is not needed). Third, in acslX™, model code is typically categorized into three sections: **INITIAL**, **DERIVATIVE** and **DYNAMIC** using section statements. For example, the code below represents the **INITIAL** section for declaring constants. On the other hand, in Berkeley Madonna™ all code is edited in an "Equations" window and there is no such categorization. Therefore, all section statements can be deleted.

INITIAL

……

END !INITIAL

Fourth, acslX™ and Berkeley Madonna™ use different syntax to state the integration algorithm method used to run the model. AcslX™ has up to 10 different algorithms and each is assigned an ALGORITHM IALG value (e.g., "2" represents the Gear's Stiff algorithm); while Berkeley Madonna™ contains 6 algorithms and each has a short method name (e.g., "PK4" represents the Runge-Kutta 4 algorithm). In the code conversion, simply delete the acslX™ method statements because when you open the “Equations” editor in Berkeley Madonna$^{\text{TM}}$, the default method statements below will automatically appear at the beginning of the model code, and then just accept the default option or change to another method and adjust the parameter values or add additional parameters as needed.

```
METHOD RK4
STARTTIME = 0
STOPTIME=10
DT = 0.02
DTOUT = 0.1
```

In the above code, *STARTTIME* and *STOPTIME* represent the initiation time and termination time of model simulation, respectively. The values can be changed according to the study design. *DTOUT* represents the simulation interval at which results are stored in memory that can be extracted. Typically, an output interval of 0.1 h is designated in PBPK models. In addition, *DT* represents simulation stepsize and is used only by fixed-stepsize integration methods, including Euler’s method, Runge-Kutta 2 (RK2), and Runge-Kutta 4 (RK4). *DTMIN*, *DTMAX* and *TOLERANCE* should be (and are only) used if variable-stepsize integration methods (i.e., Auto-stepsize and Rosenbrock (Stiff)) are chosen.

Detailed explanation of the differences among these different integration methods refers to the User Guide of Berkeley Madonna$^{\text{TM}}$ (Macey et al., 2009) and relevant literature (Butcher, 2008; Stewart, 2016). In brief, Euler’s approximation method discretizes ordinary differential equations (ODEs):

$$\frac{dy(t)}{dt} = f(t, y(t))$$

by truncating the Taylor expansion (Stewart, 2016) and using the first term in this series. The error will then be quantitated by the first order term $O(\Delta t)$ where Δt denotes the time step. The error quantity $O(\Delta t)$ decays with the size of Δt, requiring to take sufficiently small steps to approximate precisely the ODE system, otherwise, the error can accumulate as the ODE system step forward in time which may not be a good approximation for $\frac{dy(t)}{dt}$. Similarly, Runge-Kutta 2 and 4 are methods of numerically integrating ODEs by using trial steps at the midpoints of the intervals of time steps (Butcher, 2008). This helps to cancel out lower-order error terms. The error quantity for RK2 is obtained by the third order $O(\Delta t^3)$, meaning that this algorithm can estimate roughly three times more accurate than Euler's method. RK4 is even more accurate compared with the other two methods. It can estimate the solution of the ODE system with a fifth order error quantity $O(\Delta t^5)$ and up to five times more accurate than Euler's method. Overall, in terms of accuracy, RK4 method is slightly more accurate than RK2 method and the Euler's method; and in terms of computational complexity and expense, Euler's approximation method runs slightly faster than Runge-Kutta methods.

Compared to RK2, RK4 and Euler's methods that are fixed-stepsize methods, Auto-stepsize and Rosenbrock (stiff) integration methods are variable-stepsize methods. In general, these two methods differ in the way they compute flows (solutions of the model), estimate absolute errors, and adjust the stepsize. Specifically, the Auto-stepsize method employs a fifth-order Runge-Kutta algorithm to compute flows and it estimates errors by comparing this flow with another flow computed with a fourth-order algorithm. The Rosenbrock (stiff) method uses a semi-implicit fourth-order Runge-Kutta algorithm to compute flows and it estimates error by comparing the flow with another flow computed with a third-order approximation. The main difference between these estimated flows is the estimated error. Additional explanation of the differences between these methods refer to the User Guide of Berkeley Madonna^{TM} (Macey et al., 2009).

Conversion of differential integration mass balance equations

In acslX^{TM}, differential equations are implemented using the "**INTEG**" statement in the form of

State = INTEG(deriv, ic)

where *state* represents a simple variable, *deriv* represents an arithmetic expression of arbitrary complexity (i.e., may contain additional statements, functions, or macros), and *ic* can be a real constant or a simple variable representing the initial condition of the *state* variable. In PBPK models, for example, as the code shown below:

AL = INTEG(RL,0.0) *(1)*

AL (*state* variable; unit: mg) represents the amount of a compound in the liver compartment, *RL* (*deriv* variable; unit: mg/h) represents the rate of change in the amount of the compound in the liver compartment, and the constant "0.0" means that the initial amount of this compound in the liver compartment is zero.

In Berkeley Madonna^{TM}, ODEs are defined by two equations: an initializer equation and an integrator equation. The initializer equation determines the initial value of the *state* variable, while the integrator equation simulates how much the *state* variable's value increases or decreases during each simulation time step. Berkeley Madonna^{TM} supports several different forms of initializer and integrator equations. As an example, the code below represents two forms of example initializer equations that are functionally identical and each equation initializes the *state* variable to an initial value denoted by "…".

INIT State = …

INIT (State) = …

In addition, two example forms of integrator equations are provided below, where each equation defines a net flow into a particular compartment denoted by "…".

d/dt(State) = …

State' = …

Therefore, the above acslX^{TM} code (1) is converted to Berkeley Madonna^{TM} code, as below:

d/dt(AL) = RL

INIT AL = 0

This differential integration equation syntax conversion from acslX^{TM} to Berkeley Madonna^{TM} should be done in each compartment of the PBPK model.

Conversion of intravenous, intramuscular, subcutaneous and oral administration equations

Both acslX^{TM} and Berkeley Madonna^{TM} use the **STEP** function to simulate intravenous exposure. In acslX^{TM}, the equation is as follows:

Riv = (DOSEiv/TIMEiv)(1.0-STEP(TIMEiv))*

where *Riv* (unit: mg/h) represents the rate of intravenous injection, *DOSEiv* (unit: mg) represents the amount that is injected, and *TIMEiv* is the injection or infusion time. The **STEP** function output (i.e., the value of *STEP(TIMEiv)*) changes from zero to one at a specified time point of *TIMEiv* (i.e., when *Time < TIMEiv*, *STEP(TIMEiv) = 0.0; when Time ≥ TIMEiv*, *STEP(TIMEiv)*

= 1.0). In Berkeley Madonna^{TM}, a slightly different equation is used describing intravenous injection, as below:

Riv = (DOSEiv/TIMEiv)(1.0-STEP(1.0, TIMEiv))* *(2)*

where *Riv*, *DOSEiv*, and *TIMEiv* have the same meanings as in the acslX^{TM} model, and the **STEP** function output (i.e., the value of *STEP(1.0, TIMEiv)*) means a step of height h (h = 1.0 in this particular equation) at time ≥ *TIMEiv*.

In our oxytetracycline PBPK model (Lin et al., 2015), we used a two-compartment model to simulate intramuscular injection of the conventional or long-acting formulation of oxytetracycline. The equations are shown below:

*Doseimfast = Doseim*Frac*

Doseimslow = Doseim(1-Frac)*

*Rim=Kim*Amtsite*

Absorb=INTEG(Rim,0)

*Rsite=-Rim+Kdiss*Doseimremain*

Amtsite=Integ(Rsite,Doseimfast)

*Rdoseimremain=-Kdiss*Doseimremain*

Doseimremain= INTEG(Rdoseimremain,Doseimslow) *(3)*

where *Doseimfast* and *Doseimslow* (unit: mg) are the doses allocated to the fast absorption and to the depot for slow absorption, respectively; *Frac* (unitless) is the fraction of the dose allocated to the fast absorption; *Rim* (unit: mg/h) is the intramuscular absorption rate; *Kim* (unit: h^{-1}) is the intramuscular absorption rate constant; *Amtsite* (unit: mg) is the amount of absorbable oxytetracycline that remains at the injection site; *Absorb* (unit: mg) is the amount that is absorbed; *Rsite* (unit: mg/h) is the rate of change in the amount of absorbable oxytetracycline in the injection site; *Kdiss* (unit: h^{-1}) is the dissolution rate constant of oxytetracycline from the depot; *Doseimremain* (unit: mg) is the dose remaining in the depot. When converting the above code to Berkeley Madonna$^{\text{TM}}$, only the integration equations need to be changed following the rule mentioned above, all the other equations remain the same. Thus, the above acslX$^{\text{TM}}$ code appeared like this in the Berkeley Madonna$^{\text{TM}}$ code:

```
Doseimfast = Doseim*Frac
Doseimslow = Doseim*(1-Frac)
Rim = Kim*Amtsite
d/dt(Absorb) = Rim
INIT Absorb = 0
Rsite=-Rim+Kdiss*Doseimremain
d/dt(Amtsite) = Rsite
INIT Amtsite = Doseimfast
Rdoseimremain = -Kdiss*Doseimremain
d/dt(Doseimremain) = Rdoseimremain
INIT Doseimremain = Doseimslow
```

where all the parameters have the same meanings as in acslX$^{\text{TM}}$. These equations can also be used to simulate subcutaneous injection of chemicals.

In the oxytetracycline PBPK model (Lin et al., 2015), single or repeated oral exposure was described using the following code:

```
CONSTANT Tlen = 0.001
CONSTANT Tinterval = 6
CONSTANT Dstart = 0.0
CONSTANT Dstop = 1.25
Tsim = TSTOP
DS = Dstart*24
Doff = (Dstop - Dstart)*24
Exposure = PULSE(0,Tinterval,Tlen)*PULSE(DS, Tsim, Doff)
RDoseoral = (Doseoral/Tlen)*Exposure
RAST = RDoseoral-Kst*AST
AST = INTEG(RAST,0)
RAI = Kst*AST-Ka*AI-Kint*AI
Rcolon=Kint*AI
Acolon=INTEG(Rcolon,0)
AI = INTEG(RAI,0)
RAO=Ka*AI
AAO = INTEG(RAO,0)                                                    (4)
```

where *Tlen* (unit: h) is the length of oral gavage; *Tinterval* (unit: h) is the interval between two treatments (varied dependent on the exposure paradigm); *Dstart* (unit: day) is the initiation day of oral gavage; *Dstop* (unit: day) is the termination day of oral gavage; *Tsim* (unit: h) is the terminal simulation time; *DS* (unit: h) is the initiation time point of oral gavage; *Doff* (unit: h) is the oral exposure duration; *Exposure* (unitless) represents the exposure paradigm; *RDoseoral* (unit: mg/h) is the oral exposure rate; *Doseoral* (unit: mg) is the amount that is given; *RAST*, *RAI*, *Rcolon*, and *RAO* (unit: mg/h) are the rates of change in the amounts of the drug in the stomach, intestine, colon, and the oral absorption rate, respectively; *Kst*, *Ka*, and *Kint* (unit: h^{-1}) are the gastric emptying rate constant, the intestinal absorption rate constant, and the intestinal transition rate constant, respectively; *AST*, *AI*, *Acolon*, and *AAO* (unit: mg) are the amounts of the drug in the stomach, intestine, colon, and the amount that is absorbed, respectively. In these codes, the **PULSE** function was used to simulate the exposure paradigm (single vs. repeated exposure) by varying the values of *Tinterval*, *Tlen*, *DS*, *Tsim*, and/or *Doff*. When converting to Berkeley Madonna^{TM}, the **SQUAREPULSE** function could be used, as shown below:

```
Tlen = 0.0025

Tinterval = 6

Exposure5 = SQUAREPULSE(0,Tlen) + SQUAREPULSE(0+Tinterval,Tlen) +
                SQUAREPULSE(0+2*Tinterval,Tlen) + SQUAREPULSE(0+3*Tinterval,Tlen) +
                SQUAREPULSE(0+4*Tinterval,Tlen)

RDoseoral = (Doseoral/Tlen)*Exposure5

RAST = RDoseoral-Kst*AST

d/dt(AST) = RAST

init AST = 0

RAI = Kst*AST-Ka*AI-Kint*AI

Rcolon=Kint*AI

d/dt(Acolon) = Rcolon

init Acolon = 0

d/dt(AI) = RAI

init AI = 0

RAO=Ka*AI
```

```
d/dt(AAO) = RAO
init AAO = 0
```

In the above codes, *Exposure5* represents repeated exposures for 5 times. The output of SQUAREPULSE(*t*, *Tlen*) is a square pulse of height 1 starting at time *t* and lasting for duration of *Tlen*. Thus, equations for single exposure (*Exposure1*) and repeated exposures for two times (*Exposure2*) are as follows:

```
Exposure1 = SQUAREPULSE(0,Tlen)
Exposure2 = SQUAREPULSE(0,Tlen) + SQUAREPULSE(0+Tinterval,Tlen)
```

Conversion of equations describing nanomaterial endocytosis

Compared to small molecular chemicals, one major difference in the PBPK modeling for nanomaterials is that, besides flow-limited or membrane-limited diffusion, nanomaterials can also enter into the tissue compartment via endocytosis by phagocytic cells, which should be described using the Hill function (Liang et al., 2016; Lin et al., 2016a; Lin et al., 2016b). For example, endocytosis rate (unit: h^{-1}) of gold nanoparticles in the liver can be described using acslX^{TM} as follows:

```
CONSTANT KLRESrelease = 0.005
CONSTANT KLRESmax = 1000
CONSTANT KLRES50 = 24
CONSTANT KLRESn = 0.5
CONSTANT ALREScap = 32.5
KLRESUP = (KLRESmax*T^KLRESn)/(KLRES50^KLRESn+T^KLRESn)*(1-
            (ALRES/(ALREScap*VL)))
```

where *KLRESrelease* (unit: h^{-1}) is the release rate constant of phagocytic cells; *KLRESmax* (unit: h^{-1}) is the maximum uptake rate constant of phagocytic cells; *KLRES50* (unit: h) is the time reaching half maximum uptake rate; *KLRESn* (unitless) is the Hill coefficient; *ALREScap* (µg/g tissue) is the uptake capacity per tissue weight. The above code is translated to Berkeley Madonna$^{\text{TM}}$ as shown below:

```
KLRESrelease = 0.005
KLRESmax = 1000
KLRES50 = 24
KLRESn = 0.5
ALREScap = 32.5
KLRESUP = (KLRESmax*TIME^KLRESn)/(KLRES50^KLRESn+TIME^KLRESn)*(1-
          (ALRES/(ALREScap*VL)))
```

The major difference between these codes is the name of the variable simulation time, which is "*T*" in acslX$^{\text{TM}}$ and "*TIME*" in Berkeley Madonna$^{\text{TM}}$. Note that this name is a software-reserved name, thus it is a software-specific and unchangeable name.

Once all the aforementioned changes have been implemented in every compartment, the last thing is to check the values of *DT* (simulation stepsize), *DTOUT* (simulation result output interval), and *Bal* (mass balance equation). The *DT* value should be checked and adjusted accordingly to avoid floating-point exceptions (e.g., numerical overflow) if the value is too big or running out of memory if the value is too small. The default value of output interval in acslX$^{\text{TM}}$ is 0.1, but in Berkeley Madonna$^{\text{TM}}$ is 0.0. Therefore, *DTOUT* value should be changed to 0.1 so that the results are comparable. In addition, the value of *Bal* should be almost 0.0

throughout the simulation period to make sure the input is always equal to the output, i.e., mass balanced.

From acslX™ to MATLAB®

Conversion of comment, constant, section, and method statements

In MATLAB® programming, unlike acslX™, it is worthwhile to mention that constants, equations, etc. are defined easily. There are at least two ways to convert the code into MATLAB®. One reduces the speed of simulation and is not recommended. In this scenario, all the functions and parameters are defined in one function script. This way of coding is very long and less organized. A better approach is to divide the whole script into multiple functions or subsections each of which is managed and run through the Main script (section of Main script below). Similar to the previous subsection, we can copy the model code including the constant parameters and differential equations from acslX™ and then paste it to a specified part of the function file. In MATLAB® all files are assigned as a .m format including the Main and functions scripts.

We can list multiple modifications from acslX™ to MATLAB®; however, here are the most important changes we have done in each script. The comments for each line of definition can be done using percentage symbol “%” as opposed to exclamation mark “!” used in acslX™. To comment a block, one can alternatively employ the block comment operators “%{ … %}” for any number of lines in green color to explain a command line in the script file as follows

model

%{

```
This is a multiple
line block comment
operator …
%}
```

Or

```
% This is a one line block comment operator …
```

Unlike acslX^{TM}, we do not necessarily need to define any type of representing objects with a constant statement for a parameter value with the command **CONSTANT**. Simply we omit the **CONSTANT** and set any new defined symbolic variables without any constraints. For instance, if the cardiac output is equal to 5.0 L/h/kg, we may define by the statement *QCC = 5.0*;. Also notice that we included a semicolon at the end of the command line which shows nothing should be printed out in the command window, otherwise its value will be shown at each iteration. The constant parameters can be stored into a function called Constants.m and can be called in another primary equation function called OTCPBPK.m for oxytetracycline or GoldPBPK.m for gold nanoparticles, but we do not separate constant file from primary PBPK model files in this project. The primary PBPK function files must commence with the command "function" following by the output of this function script which is evidently the system of differential equations defined as variables. This is the main reason why we utilize "*dY*" and is equal to the name of the function with all the inputs from the Main.m script file (described in the section of Main script below).

```
function
dY=GoldPBPK(T,Y,PDOSEORAL,PDOSEIV,PDOSEIM,TINTERVAL,KA,TSTOP,KIM,FRAC)
```

```
%% Constants
          ⋮
%% Variables
          ⋮
% body of the code
                    ⋮
end
```

"*T*" and "*Y*" are the only variables we have in this function and the remaining arguments (*PDOSEORAL,PDOSEIV,* …) in the GoldPBPK function is the constant input from the Main script. The constant block initializes all the constant parameters and is the same as **INITIAL** in acslX^{TM}. The derivatives and dynamical system can also be assigned to the primary function following the statement for variables. In fact, these parameters and variables are used in the primary PBPK function with double percentages symbol "%%" which noticeably separate each function and its job from the rest of the code by highlighting that segment only. Also, it is worth pointing out that in MATLAB$^{®}$ unlike acslX^{TM}, anything used must be previously defined regardless of whether it is a constant or variable otherwise an error will occur. For instance, if *QCC* is defined after the use in the code we receive the following error that

```
Undefined function or variable
"QCC".
Error in GoldPBPK (line 134)
QC = QCC*BW; %^0.75 % Cardiac output
```

It shows what and where the error is in GoldPBPK.m script file. If we click on the error where the name of the function or the line is given, it will take us to exactly where the miscalculation occurs. Also, initializing constants in MATLAB® programming is case sensitive and, for example, {QCC, QCc, Qcc, qcc, …} can be allocated with different values without running into any problem.

Another modification required to be performed is related to the syntax of time series integration. MATLAB® uses 8 different solvers for the integration algorithm method; however, any other approximation techniques for initial value problems can straightforwardly be implemented in MATALB®. For instance, for Runge-Kutta 4 approximation algorithm, there are two different solvers (**ode23** for RK2 and **ode45** for RK4). MATALB® basically supports any systems that can be nonstiff, moderately stiff, stiff, or fully implicit differential equations. One can also utilize other MATALB® functions to evaluate and extend solutions to ODEs with multiple solver options. In the code conversion, simply delete the acslX™ method statements because when we call the ODE solver **ode23** in our Main code function a different syntax needs to initialize or finalize the time points. This will be explained in the next subsection. We can simply remove all the commands after **DYNAMIC** in acslX™:

ALGORITHM IALG = 2

NSTEPS NSTP = 10

MAXTERVAL MAXT = 1.0e9

MINTERVAL MINT = 1.0e-9

CINTERVAL CINT = 0.1

The Variables block consists of all the variables that change over time and are required. This method is, however, not unique. There are several other techniques to let MATLAB® understand which symbols are variables, but for simplicity and consistency with acslX™, we figured out this could be the best way to present variables.

Integration of mass balance equations

As aforementioned, the derivatives for time series integration can be called from a separate .m function or employed directly in PBPK.m function file. Here we implement the derivatives in the primary PBPK function (OTCPBPK.m or GoldPBPK.m). As defined in equation (1), differential equations in acslX™ are implemented using the "**INTEG**" statement. We first need to remove the command "**INTEG**" and the initial conditions "*ic*" inside this command, and then make use of the ODE representing symbol "*dY(i)*" where "*i*" is corresponding to the variable defined in Variables block (e.g., for *dY(1)* = the derivative of AIV which is associated with the total IV dosing). The same technique is applied for the remaining variables which will be used in the Main script and ODE solver for time series integration for a given interval of time and specific arguments. The following command is also used at the bottom of the PBPK function to make sure that the PBPK function returns a column vector of all variables:

dY = dY(:);

Conversion of intravenous, intramuscular, subcutaneous and oral administration equations

Similar to the command **STEP** in acslX™ to simulate intravenous exposure, MATLAB® also supports a step function called **Heaviside**. It is worthwhile to mention that *Heaviside(T-T0)* returns the value 0 for $T < T0$, 1 for $T > T0$, and 1/2 for $T = T0$. As a result, the **STEP** function of acslX^{TM} represented by (2) will be transformed into the following equation:

```
Riv = DOSEiv/timeiv *(1.-heaviside(T-timeiv));
```

Where "*T*" is the input variable time which will be used and varied in the ODE solver. The other parameters (*DOSEiv* and *timeiv*) and variable (*Riv*) have been implemented the same as equation (2) with identical units. The expression *(1.-heaviside(T-timeiv))* guarantees that the rate of intravenous injection (*Riv*) is only nonzero during the injection or infusion time, that is, when $0<T<timeiv$.

Equation (3) for a two-compartment model in oxytetracycline PBPK model (Lin et al., 2015) can be converted in a very similar pattern into MATLAB$^{®}$ as follows:

```
% Dosing, intramuscular, dissolution model
Doseimfast = Doseim*Frac;
Doseimslow = Doseim*(1-Frac);
Rim=Kim*Amtsite;
dY(5)=(Rim);
Rsite=-Rim+Kdiss*Doseimremain;
dY(6)=(Rsite);
Rdoseimremain=-Kdiss*Doseimremain;
dY(7)=(Rdoseimremain);
```

These statements were employed to simulate the IM injection of the conventional or long-acting formulation of oxytetracycline. All the units and quantities are defined previously after the list of comments in (3). The changes here are merely related to the integration function "**INTEG**" and the remaining statements are the same as equation (3). We observe that the derivatives for *dY(5)*, *dY(6)* and *dY(7)* are employed for *Absorb*, *Amtsite*, *Doseimremain*, respectively. As abovementioned; *Absorb* (unit: mg) is the amount that is absorbed; *Amtsite* (unit: mg) is the amount of absorbable oxytetracycline that remains at the injection site; and *Doseimremain* (unit: mg) is the dose remaining in the depot.

Another change required in our MATLAB® coding for the oxytetracycline PBPK model (Lin et al., 2015) was described by command lines (4). These statements were used to compute a single or repeated oral exposure. We need to define all the parameters to be able to estimate these amounts by the length of oral gavage exposure *Tlen* and the repetition frequency *Tinterval*. As a result, we can identically transform the acslX™ code into MATLAB® using the following code conversion:

```
%...Dosing, multiple oral gavage
tlen = 0.002;
tinterval = TINTERVAL;
Dstart = 0.0;
TSTOP = TSTOP;
MAXT = 1.0 ;
CINTC = 0.1;
CINT = CINTC ;
```

```
Tsim = 24*TSTOP;
DS = Dstart*24 ;
Doff = (TSTOP - Dstart)*24;
TimeOn = Dstart*24;
TimeOff = TSTOP*24+tlen;
Exposure = pulstran(T-tlen/2,tinterval,@rectpuls,tlen)*(1-heaviside(T-TimeOff));
RDoseoral = (Doseoral/tlen)*Exposure;
RAST = RDoseoral-Kst*AST;
dY(1) = (RAST);
RAI = Kst*AST-Ka*AI-Kint*AI;
Rcolon=Kint*AI;
dY(2)=(Rcolon);
dY(3) = (RAI);
RAO=Ka*AI;
dY(4) = (RAO);
```

Analogous to the previous modification, we defined the variables *dY(1)*, *dY(2)*, *dY(3)*, and *dY(4)* representing the numerical quantities *AST*, *Acolon*, *AI*, and *AAO* (unit: mg) for the amounts of the drug in the stomach, intestine, colon, and the amount that is absorbed, respectively. The Heaviside step functions were employed in both Gold and oxytetracycline PBPK models. We, however, made use of another MATLAB® function called **PULSTRAN**. **PULSTRAN** generates pulse trains for consecutive pulses from either continuous functions or sampled prototype pulses. The function can be used as *exposure = pulstran(t, d, @func, length)*

which then produces a single pulse train at time *t* with the repetition frequency *d* contingent on samples of a continuous function used by *@func* or *'func'*. The argument *'func'* can have multiple shapes such as triangular, Gaussian-modulated sinusoidal pulse, or aperiodic rectangle signals. The length of each pulse is set to be equal to *tlen*. We should point out that the initiation and stop time points of oral gavage and oral gavage duration were all controlled by another **PULSE** function in acslX^{TM} which was multiplied by the second term *PULSE(DS, Tsim, Doff)*. However, in MATLAB$^{®}$, all will be encrypted in *tinterval* and will be shut down by **Heaviside** function at *TimeOff*. The pulse quantity *Exposure5* representing 5 times repeated exposures.

Conversion of equations describing nanomaterial endocytosis

The conversion of nanomaterial endocytosis equations from acslX^{TM} to MATLAB$^{®}$ is almost identical to that of acslX^{TM} to Berkeley Madonna^{TM}. We make use of the following MATLAB$^{®}$ statements for endocytosis rate (unit: h^{-1}) of gold nanoparticles in the liver using the Hill function (Liang et al., 2016; Lin et al., 2016a; Lin et al., 2016b):

```
KLRESrelease = 0.005;
KLRESmax = 1000;
KLRES50 = 24;
KLRESn = 0.5;
ALREScap = 32.5;
KLRESUP = (KLRESmax*T^KLRESn)/(KLRES50^KLRESn+T^KLRESn)*(1-
        (ALRES/(ALREScap*VL)));
```

Notice that we use semicolon at the end of each line to make sure no output is printed out in the MATLAB® Command Window. This helps to speed up slightly the program and have less disorganized output. The constant parameters including *KLRESrelease, KLRESmax, KLRES50, KLRESn, and ALREScap* have already been described and provided the units above. We also showed that the major difference for these codes between acslX™ and Berkeley Madonna™ was the use of the time as a variable for our simulation purpose. An excellent advantage of MATLAB® is that one can describe the change of time using any type or name for the variable of interest. There is no restriction in choosing a time-varying symbol as long as it is consistent throughout the PBPK script file. In addition, the variable for time can be reassigned to another variable or appropriately recalled in each function or the Main script without any constraints. No command or variable is reserved for "*time*" in MATLAB®.

Main script

Here we describe how one can run the code and how the main code and other functions are organized and called in MATLAB®. We can control all the changes required for each plot and subplot in this script by only varying the necessary parameters and keep the primary PBPK functions for gold nanoparticle and oxytetracycline models entirely unchanged for the whole simulating process. The changes will then be incorporated as arguments into the ODE command and will be fed to the PBPK functions. This way we are certain that we do not require to take care of fixing or tracing the changes in these functions accordingly.

The code should begin by deleting the memory allocated to other variables or script and closing unnecessary figures from the previous simulations. To this end, we take advantage of three commands:

```
clear
clc
close all
```

where the command “**clear**” clears all data from simulation data and “**clc**” stands for Clear Command Window. We can use another command “**clear all**” which clears all objects in the MATLAB® workspace and closes the MuPAD® engine associated with the MATLAB® workspace and resets all its assumptions. In addition, the command “**close all**” deletes all figures whose handles are not hidden.

We now need to define all the controlling input arguments. These can be initialized here and then fed to the PBPK function. This part of the code will be changed for another experiment:

```
%% Input Arguments
PDOSEORAL = 25;
PDOSEIV = 0;
PDOSEIM = 0;
TINTERVAL=0:6:24;
KA=0.012;
KIM=0.15;
TSTOP = 24;
```

```
FRAC=0.5;
BW = 11.3;
Doseim = PDOSEIM*BW;
Doseimfast = Doseim*FRAC;
Doseimslow = Doseim*(1-FRAC);

% Initial Conditions
Initials=zeros(23,1);
Initials(6)=Doseimfast;
Initials(7)=Doseimslow;
```

where *PDOSEORAL*, *PDOSEIV*, *PDOSEIM* are all parameters for various exposure scenarios for oral, IV and IM doses, respectively, and the units are all in mg/kg. The other statement *TINTERVAL* is also given for oral exposure. This will provide a row vector consisting of these elements *TINTERVAL = (0, 6, 12, 18, 24)* which shows where the square pulse must occur. In fact, we will have square pulse train at time 0 when *TINTERVAL* is zero, at 6 hours when *TINTERVAL* is equal to 6, and so forth. The square pulse train takes place in a consecutive order without interruption until *TINTERVAL = 24* or **Heaviside** function is satisfied at *T = TimeOff*. The other parameters have been defined before.

We also initialized the system of differential equations by setting all of them zero for the initial conditions using the function **zeros(23,1)**. This function generates a column vector of 23 zero elements. In the next lines, we changed the values of row 6 and 7 to *Doseimfast* and *Doseimslow* (unit: mg) for the doses allocated to the fast absorption and to the depot for slow absorption, respectively.

MATLAB® software has implemented several packages to numerically solve ODEs. To compile the system of differential equations, here we will only take advantage of the MATLAB® ODE solvers using Runge-Kutta (RK) approximation methods. The built-in functions **ode23** and **ode45** are based on explicit **RK2** *and* **RK4**, respectively, but implemented more efficiently (Dormand and Prince, 1980; Shampine and Reichelt, 1997). In fact, these two solvers implement simultaneously pairs of $RK(2,3) = (RK2, RK3)$ for **ode23** and $RK(4,5) = (RK4, RK5)$ for **ode45** (Hairer et al., 2010). Since $RK(2,3) = (RK2, RK3)$ requires fewer steps, then **ode23** is less computationally expensive than **ode45**. It is worthwhile to point out that both solvers are merely used for nonstiff ODEs. Below is an example of **ode23**:

```
%% Running ODE Solver
ti=0;
tf=40;
tspan=[ti tf];
options=odeset('AbsTol',10e-2,'RelTol',10e-2,'Stats','on', 'NonNegative');
tic;
[t,y] = ode23(@(T,Y)OTCPBPK(T, Y, PDOSEORAL, PDOSEIV, PDOSEIM, TINTERVAL, KA,
            TSTOP, KIM,FRAC), tspan, Initials', options);
toc;
```

where *ti*, *tf* is the initial and final times of simulations which are incorporated in *tspan* for the whole course of the experiment. We can also make use of some options for the ODE solver by using the "**odeset**" function for the absolute tolerance, relative tolerance. Or we can specify that the components of the solution vector must be nonnegative by means of **NonNegative**. The argument **Stats** also determines whether the solver should display statistics about the computations cost of the integration such as the number of successful steps or failed attempts. To start stopwatch timers, we employed **tic** function to measure performance. This function records the internal time at the execution of the "**tic**" command. To display the elapsed time, we then make use of the "**toc**" function.

All the time series integration occurs by the implementation of the command line for **ode23**. This command takes (*T*,*Y*) as the only variables with 23 different differential equations which are the input of the PBPK function and the remaining arguments are incorporated for the simulation purpose and assigned to fix values in the previous lines. Then it is required to provide the duration of run with *tspan*, in this case in the interval of [0, 24] hours. We also need to include the initial conditions with **Initials** with or without the options arguments. This command will be executed until the time reaches the final time of *tspan*. Then all the data integrated over *tspan* time will be assigned to another pair of variables [*t*,*y*] which stand for the time and 23 different variables given in the Variables block.

The final step is to polish the result for illustration purpose. All MATLAB® ODE solvers including **ode23** and **ode45** choose adaptive algorithms for their time-step as the solution progresses in order to satisfy the error tolerance. This indicates that all the time-steps do not track a unique pattern for the solutions which cannot then be straightforwardly compared with our previous results for acslX^{TM} to Berkeley Madonna^{TM}. To be consistent with the previous results, we can, however, use the following code to transform any outputs of the adaptive method into a fixed step size, for this study, equal to 0.1:

```
%% Evaluation for fixed step sizes
t_temp = ti:.1:tf;
y_temp=zeros(length(t_temp),23);
t_temp=t_temp';
for i=1:length(Initials)
        y_temp (:,i) = interp1(t,y(:,i),t_temp);
end
t=t_temp;
y= y_temp;
```

We make use of two temporary variables represented by *t_temp*, *y_temp* for the time and 23 variables. The statement for *t_temp* generates a fixed temp step from initial time *ti* to final time *tf*. Also, *y_temp* is a matrix with 23 columns and have the same row as the length of *t_temp* which, in this case, is a matrix of size 23x241. We then interpolate the same for the variable *y* with the size of *y_temp* and then store each column into *y_temp*. When we pass nonuniformly spaced points (*t*,*y*) in **interp1** with the last argument *t_temp*, it will interpolate the date for the specified time points provided by *t_temp*.

The evaluation for fixed step sizes follows by including the data for the amounts of modeled substance in the organ of interest. For instance, below we call the data file for the amounts of gold nanoparticles in the liver by using the following command to read the text file for this dataset (listed in the Supplementary Data):

```
FentLiver = textread('FentLiver.txt');
```

The columns in this file show time and amount whose unit is in ug.

The remaining part to call data and plot figures is the same as the acslX^{TM} codes and nothing is left to program. We also used another "**for-loop**" statement to calculate, for instance, *cm*, *cl*, *ck*, *cv* for the concentrations in muscle, liver, kidney and venous blood compartment, respectively. As an example, we compute the venous blood concentration using the *y* variable in the following format:

```
cv=zeros(length(t),1);
for i=1:length(t)-1
    cv(i,1)=(y(i+1,10)-y(i,10))/(t(i+1)-t(i));
end
```

Since the tenth variable integrates *AUCCV* for the area under curve concentration, the derivative of this, therefore, gives the venous blood concentration which is why we calculate this using the for-loop statement. There are other ways of calculating the blood concentration over time, but this seems the simplest technique to do so.

From acslX™ to R language

Compared to acslX^{TM}, Berkeley Madonna^{TM}, and MATLAB®, R is less commonly used in the development of PBPK models. However, since R is a free and open-source software program and has a large number of functions for statistical analyses and graphics, some readers may be of interest in using R to develop PBPK models. Below we use gold nanoparticle PBPK model as an example to describe how to convert the code from acslX^{TM} to R.

Conversion of comment, constant, section and method statements

There are several options that you can choose to edit the R codes. Basic editors include R Console or R Editor window that comes with R programming software. RStudio is a wonderful alternative that offers a richer editing environment than R Console and R Editor. Basic version of RStudio is a free and open-source cross-platform integrated development environment (IDE) for the R statistical language. RStudio Commerical version (the price is dependent on the number of users) is also available, but the free basic version is sufficient for our PBPK modeling study. One great advantage of RStudio is that all relevant windows including Editor, Console, Environment, and Plots windows, are decked together. Therefore, it is very convenient for you to keep track of the sources syntax, data and corresponding analyzed results in RStudio. In addition, RStudio has a full-feature text editor which contains everything you would expect from a text editor: syntax highlighting, parenthesis and bracket matching, etc. Under the platform of RStudio, the model can be visualized to become a Graphical User Interface (GUI) using Shiny (a web application framework for R). RStudio is recommended even if you are a casual user or beginner of R.

R's syntax is simple and intuitive. It is easy to define constants and equations in R language. After copying and pasting the acslX^{TM} codes to R Editor in RStudio, you can easily define equations in R by removing **CONSTANT** command in acslX^{TM} syntax. For example, if the cardiac output is equal to 5.0 L/h/kg, we can simply define it as *QCC = 5.0* in R. An alternative way to define equation using the operator "<-": *QCC <- 5.0.* The comments for each line of statement can be done using pound sign "#" as opposed to exclamation mark "!" used in acslX^{TM}. Another difference in R syntax from acslX^{TM} is that there are no section statements in R, so all section statements including **PROGRAM**, **INITIAL**, **DERIVATIVE**, **DYNAMIC**, **TERMINAL**, **END!INTIAL**, **END!DYNAMIC**, **END!TERMINAL**, **END!PROGRAM**, should be removed from R programs. Also, in R there is a requirement of the sequence of the code, i.e., parameters used in the equation should be defined before. The main PBPK model can be performed through the **ode** function, a built-in function of the R package "**deSolve**" (Soetaert et al., 2010).

Conversion of differential integration mass balance equations

Differential equations are solved using **ode** function from the R package **deSolve**. The **deSolve** package was built under R version 3.2.5. R software with a version higher than 3.2.5 (e.g., R version 3.3.1) is required in order to successfully use its built-in functions. The **deSolve** package needs to be installed for the first time using it. The **deSolve** package can be installed either from a repository such as CRAN or from a local .zip file by using install.packages (e.g., install.packages("deSolve", lib="/my/own/R-packages/"). A specific R package needs to be installed only once on a computer. However, it is required to load the package prior to using any functions built with it using R codes as follows.

library(deSolve)

There are at least four sets of inputs required for requesting the **deSolve** function **ode**, including parameter definitions, initial values for state variables, time specification, and model equations. The last input (i.e., model equations), which calculates the rate of change of the state variables, is the most crucial one in solving the differential equations. A user-defined function is used to specify model equations. Note that the other inputs for **ode** function such as model parameters, initial values for state variables and model time are also inputs to the function for model equations. The model equations function is specified using the following R codes

```
ODEPBPK <- function(Time, State, Pars) {
    with(as.list(c(State, Pars)), {
    # rate of change
        ⋮
    # return the rate of change
    list(c(, …, ))
    }) # end with(as.list ...
}
```

Here, “ODEPBPK” is a user defined name for the model equations function. The statement *with(as.list(c(State, Pars)), ...)* is attempt to coerce vector variables into a list as to make available the names in the list. It is worth noting that the state variables and initial parameters are now created as vectors.

```
Pars <- c(, …,)
State<- c(, …,)
```

We can specify the time using the **seq** function:

Time <- seq(from=0, to=48, by=0.1)

where "*from=*" and "*to=*" statements specify the starting and ending times, and "*by=*" statements defines the simulation interval. Finally, we can solve the differential equations using **deSolve** function **ode:**

pbpkout <- ode(func = ODEPBPK, y = State, parms =Pars, times = Time)

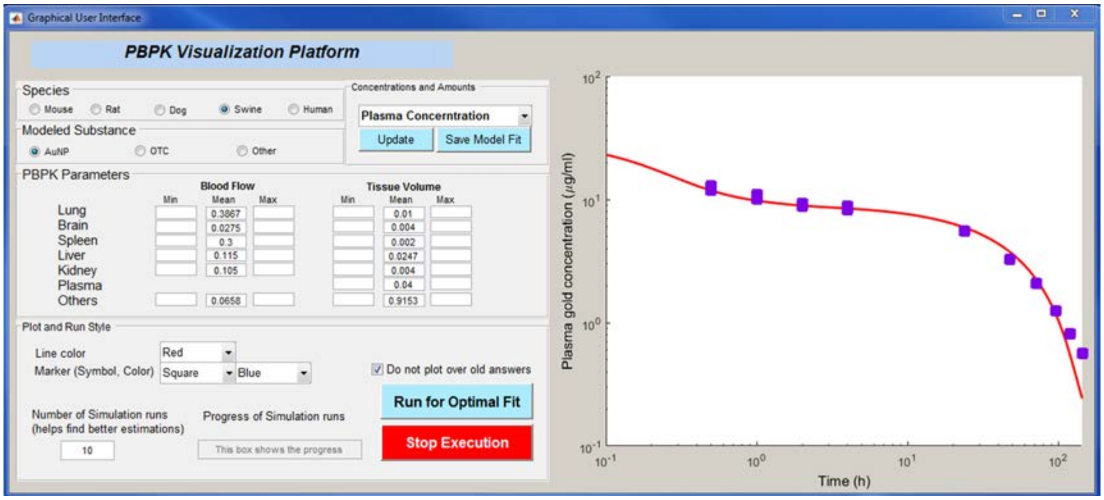
Here, "pbpkout" is a user-defined name that stores the outputs generating from the **ode** function, which is a matrix containing the values of the state variable at the requested output times.

Extract and export R outputs

As aforementioned, the integration results generated from the **ode** function are stored in a matrix. A matrix is computationally efficient, but it can be problematic if the user wants to extract a specific variable among a number of them because no column names are assigned to a matrix. To make the results more readable, we can convert the matrix to data frame through the statement "*pbpkout <- data.frame(pbpkout)*". Thereafter, we can extract a specific variable by using the statement "*pbpkout$Varname*". For instance, we can use the statement "*pbpkout$ALt*" to extract the integration result for the variable "*ALt*". For ease of reading and comparison, we can export the modeling results to an excel file. R software provides a convenient way to achieve this goal through the statement "*write.csv(Mydata, file="Mydata.csv")*", where "*Mydata*" is a data frame containing several variables. It is possible that variables in "*Mydata*" data frame are derived from different equations. We can use a "*cbind*" statement to gather them together into a data frame.

Graphical User Interface (GUI) Development of PBPK Visualization Platform

Briefly, a PBPK model GUI can be developed in MATLAB® based on the MALAB® APP Designer by calling the command "**GUIDE** (GUI development environment)" in the Command Window. GUIDE provides excellent tools to browse and align objects for different layouts to design user interfaces for custom apps. As an example only for the illustration purpose, a GUI design based on our PBPK models for both gold nanoparticles and oxytetracycline in different species using the GUIDE Layout Editor is shown below (Figure S1). The GUIDE Layout Editor can help automatically generate the initial blank MATLAB code using our PBPK model GUI for constructing the user interface, which we then incorporated the gold nanoparticle and oxytetracycline model code into the blank MATLAB® code and further modified to associate the GUI with the MATLAB® code. The codes can contain different modules to program the behavior of our parameter ranges from a lower to upper range for various PBPK parameters such as blood flows and tissue volumes.



Graphical User Interface

PBPK Visualization Platform

Species: ○ Mouse ○ Rat ○ Dog ● Swine ○ Human

Modeled Substance: ● AuNP ○ OTC ○ Other

Concentrations and Amounts: Plasma Concerntration

PBPK Parameters

	Blood Flow			Tissue Volume		
	Min	Mean	Max	Min	Mean	Max
Lung		0.3867			0.01	
Brain		0.0275			0.004	
Spleen		0.3			0.002	
Liver		0.115			0.0247	
Kidney		0.105			0.004	
Plasma					0.04	
Others		0.0658			0.9153	

Plot and Run Style

Line color: Red

Marker (Symbol, Color): Square, Blue

☑ Do not plot over old answers

Number of Simulation runs (helps find better estimations): 10

Progress of Simulation runs: This box shows the progress

Figure S1. A screenshot of the PBPK model GUI. This visualization platform was created for the purpose of illustrating how to make it easier to apply the PBPK model by researchers with no or limited programming experience. The PBPK model GUI provides various options for modeling different species (i.e., the mouse, rat, dog, swine, and human), different substances (gold nanoparticles [AuNP], oxytetracyline, or other substances), and plotting concentrations/amounts in different compartments provided that the species- and chemical-specific parameter values are well defined.

Oxytetracycline PBPK model code in acslX™ format

```
PROGRAM
! Initiated by Zhoumeng Lin on 02/11/2014;
! Last checked by Zhoumeng Lin on 07/28/2014;

INITIAL
! code that is executed once at the beginning of a simulation run goes here
!! Physiological parameters
! Blood flow rates
 CONSTANT QCC = 12.9 ! Cardiac output (L/h/kg) (Brown et al., 1997, Table 22 for mixed-
breed (Mongrel) dogs (8.39 L/h/kg) and beagles (12.9 L/h/kg))
 CONSTANT QLC = 0.297 ! Fraction of blood flow to the liver (Brown et al., 1997, Table 26)
 CONSTANT QKC = 0.173 ! Fraction of blood flow to the kidneys (Brown et al., 1997, Table
26)
 CONSTANT QFC = 0.097 ! Fraction of blood flow to the fat (Vinegar, 2001, Table 2)
 CONSTANT QMC = 0.217 ! Fraction of blood flow to the muscle (Brown et al., 1997, Table
26)

! Tissue volumes
 CONSTANT BW = 11.3 ! Body weight (kg) (Baggot et al., 1977)
 CONSTANT VLC = 0.0329 ! Fractional liver tissue (Brown et al., 1997, Table 6)
 CONSTANT VKC = 0.0055 ! Fractional kidney tissue (Brown et al., 1997, Table 6)
 CONSTANT VFC = 0.15 ! Fractional fat tissue (Vinegar, 2001, Table 2)
 CONSTANT VMC = 0.4565 ! Fractional muscle tissue (Brown et al., 1997, Table 6)
 CONSTANT VbloodC = 0.082 ! Blood volume, fraction of BW (Brown et al., 1997, Table 21)

!Fraction of tissue volumes that is blood ! (Brown et al., 1997; Table 30)
 CONSTANT FVBF = 0.02 ! Blood volume fraction of fat (%)
 CONSTANT FVBM = 0.01 ! Blood volume fraction of muscle (%)
 CONSTANT FVBS = 0.01 ! Blood volume fraction of slowly perfused tissues (%)

! Mass Transfer Parameters (Chemical-specific parameters)
! Partition coefficients (PC, tissue:plasma)
 CONSTANT PL = 1.89 ! Liver:plasma PC (Craigmill et al., 2000, Table 3; Craigmill, 2003,
Table 4, in sheep)
 CONSTANT PK = 4.75 ! Kidney:plasma PC (Craigmill et al., 2000, Table 3; Craigmill, 2003,
Table 4, in sheep)
 CONSTANT PM = 0.85 ! Muscle:plasma PC (Craigmill et al., 2000, Table 3; Craigmill, 2003,
Table 4, in sheep)
 CONSTANT PF = 0.086 ! Fat:plasma PC (Craigmill et al., 2000, Table 3; Craigmill, 2003,
Table 4, in sheep)
 CONSTANT PR = 4.75 ! Richly perfused tissues:plasma PC (Assumed the same as
kidney:plasma PC)
 CONSTANT PS = 0.85 ! Slowly perfused tissues:plasma PC (Assumed the same as
muscle:plasma PC)
```

```
!Permeability constants (L/h/kg tissue) (Permeation area cross products)
 CONSTANT PAFC = 0.012 ! Fat tissue permeability (0.12 in Leavens et al., 2012)
 CONSTANT PAMC = 0.225 ! Muscle tissue permeability (0.45 in Leavens et al., 2012)
 CONSTANT PASC = 0.049 ! Slowly perfused tissue permeability (0.49 in Leavens et al., 2012)

! Kinetic constants
! Oral absorption rate constants
 CONSTANT Kst = 2 ! /h, gastric emptying rate constant
 CONSTANT Ka = 0.012 ! /h, intestinal absorption rate constant, ka=0.05 for experimental
solution; ka=0.012 for tablets or capsules
 CONSTANT Kint = 0.2 ! /h, intestinal transit rate constant

! IM absorption rate constants
 CONSTANT Kim = 0.3 ! 0.15 for conventional formulation; 0.3 for long-acting formulation;
IM absorption rate constant (/h)
 CONSTANT Frac = 0.5 ! 0.95 for conventional formulation; 0.5 for long-acting formulation
 CONSTANT Kdiss = 0.02 ! /h

! IV infusion rate constants
 CONSTANT Timeiv = 0.01 ! IV infusion time (h), based on Leavens et al., 2012

 ! Urinary elimination rate constant
  CONSTANT KurineC = 0.2 ! L/h/kg

!Parameters for various exposure scenarios
 Constant PDOSEoral = 0 ! (mg/kg)
 Constant PDOSEiv = 10 ! (mg/kg)
 Constant PDOSEim = 0 ! (mg/kg)

END ! INITIAL

DYNAMIC

ALGORITHM IALG = 2
NSTEPS    NSTP = 10
MAXTERVAL MAXT = 1.0e9
MINTERVAL MINT = 1.0e-9
CINTERVAL CINT = 0.1

DERIVATIVE
! code for calculating the derivative goes here
! Cardiac output and blood flows to tissues (L/h)
 QC = QCC*BW ! Cardiac output
 QL = QLC*QC ! Liver
 QK = QKC*QC ! Kidney
 QF = QFC*QC ! Fat
 QM = QMC*QC ! Muscle
 QR = 0.626*QC-QK-QL ! Richly perfused tissues
 QS = 0.374*QC-QF-QM ! Slowly perfused tissues

! Tissue volumes (L)
 VL = VLC*BW ! Liver
 VK = VKC*BW ! Kidney
 VF = VFC*BW ! Fat
 VM = VMC*BW ! Muscle
 Vblood = VbloodC*BW ! Blood
 VR = 0.142*BW-VL-VK ! Richly perfused tissues
 VS = 0.776*BW-VF-VM ! Slowly perfused tissues

!Permeability surface area coefficients
 PAF = PAFC*VF ! Fat:blood permeability (L/h)
 PAM = PAMC*VM ! Muscle:blood permeability (L/h)
 PAS = PASC*VS ! Slowly perfused tissue:blood permeability (L/h)

! volume of tissue vs blood
! Fat
 VFb = FVBF*VF ! Fat compartment blood volume
 VFt = VF-VFb ! Fat compartment tissue volume

! Muscle
 VMb = FVBM*VM ! Muscle compartment blood volume
 VMt = VM-VMb ! Muscle compartment tissue volume

! Slowly perfused tissue
 VSb = FVBS*VS ! Slowly perfused compartment blood volume
 VSt = VS-VSb ! Slowly perfused compartment tissue volume

! Dosing
 DOSEoral = PDOSEoral*BW ! (mg)
 DOSEiv = PDOSEiv*BW ! (mg)
 DOSEim = PDOSEim*BW ! (mg)

!...Dosing, multiple oral gavage
 CONSTANT tlen = 0.001 ! Length of oral gavage exposure (h/day)
 CONSTANT tinterval = 6 ! Varied dependent on the exposure paradigm
 CONSTANT Dstart = 0.0 ! Initiation day of oral gavage (day)
 CONSTANT Dstop = 0.2 ! Termination day of oral gavage (day)
 CONSTANT MAXT = 1.0  ! maximum comm. interval
 CONSTANT CINTC = 0.1 ! Communication interval
 CINT = CINTC  ! Communication interval

 Tsim = TSTOP ! Tstop in hours
 DS = Dstart*24  ! Initiation time point of oral gavage (h)
 Doff = (Dstop - Dstart)*24 ! Oral gavage duration (h)

 TimeOn = Dstart*24! Initiation time point of oral gavage (h)
 TimeOff = Dstop*24+tlen ! Termination time point of oral gavage (h)

 Exposure = PULSE(0,tinterval,tlen)*PULSE(DS, Tsim, Doff)
 Rdoseoral = (Doseoral/tlen)*Exposure
 RAST = RDoseoral-Kst*AST
 AST = Integ(RAST,0)
 RAI = Kst*AST-Ka*AI-Kint*AI
 Rcolon=Kint*AI
 Acolon=Integ(Rcolon,0)
 AI = Integ(RAI,0)
 RAO=Ka*AI
 AAO = Integ(RAO,0)

!...Dosing, intramuscular, dissolution model
 Doseimfast = Doseim*Frac
 Doseimslow = Doseim*(1-Frac)

 Rim=Kim*Amtsite
 Absorb=Integ(Rim,0)
 Rsite=-Rim+Kdiss*Doseimremain
 Amtsite=Integ(Rsite,Doseimfast)
 Rdoseimremain=-Kdiss*Doseimremain
 Doseimremain=Integ(Rdoseimremain,Doseimslow)

! OTC iv injection to the venous
 IVR = DOSEiv/timeiv
 RIV = IVR*(1.-step(timeiv))
 Aiv = Integ(Riv,0)

! Urinary elimination rate constant
 Kurine = KurineC*BW

! OTC in blood compartment
 CV = ((QL*CVL+QK*CVK+QF*CVF+QM*CVM+QR*CVR+QS*CVS+Riv+Rim)/QC)
 RA = QC*(CV-CA)
 AA = Integ(RA,0)
 CA = AA/Vblood
 AUCCV = Integ(CV,0.0)

! OTC in liver compartment
 RL = QL*(CA-CVL)+RAO
 AL = Integ(RL,0)
 CL = AL/VL
 CVL = AL/(VL*PL)
 AUCCL = Integ(CL,0.0)

! OTC in kidney compartment
 RK = QK*(CA-CVK)-Rurine
 AK = Integ(RK,0)
 CK = AK/VK
 CVK = AK/(VK*PK)
 AUCCK = Integ(CK,0.0)

! Urinary excretion of OTC
 Rurine = Kurine*CVK
 Aurine = Integ(Rurine,0)

! OTC in fat compartment, permeability-limited model
 RFB = QF*(CA-CVF)-PAF*CVF+(PAF*CFt)/PF
 AFB = Integ(RFB,0)
 CVF = AFB/VFB

 RFt = PAF*CVF-(PAF*CFt)/PF
 AFt = Integ(RFt,0)
 CFt = AFt/VFT
 Aftotal = Aft+Afb
 CF = Aftotal/VF

!! OTC in muscle compartment, permeability-limited model
 RMB = QM*(CA-CVM)-PAM*CVM+(PAM*CMt)/PM
 AMB = Integ(RMB,0)
 CVM = AMB/VMB

 RMt = PAM*CVM-(PAM*CMt)/PM
 AMt = Integ(RMt,0)
 CMt = AMt/VMT
 AMtotal = AMt+AMb
 CM = AMtotal/VM
 AUCCM = Integ(CM,0.0)

! OTC in richly perfused tissue compartment
 RR = QR*(CA-CVR)
 AR = Integ(RR,0)
 CR = AR/VR
 CVR = AR/(VR*PR)

! OTC in slowly perfused tissue compartment, permeability-limited model
 RSB = QS*(CA-CVS)-PAS*CVS+(PAS*CSt)/PS
 ASB = Integ(RSB,0)
 CVS = ASB/VSB

 RSlt = PAS*CVS-(PAS*CSt)/PS
 ASlt = Integ(RSlt,0)
 CSt = ASlt/VST
 AStotal = ASlt+ASb
 CS = AStotal/VS

! Mass balance
Qbal = QC-QL-QK-QF-QM-QR-QS
Tmass = AA+AL+AK+AFtotal+AMtotal+AR+AStotal+Aurine
Bal = AAO+Aiv+Absorb-Tmass ! Permeability-limited model mass balance

END ! DERIVATIVE

! Add discrete events here as needed
!DISCRETE
!END

! code that is executed once at each communication interval goes here

CONSTANT TSTOP = 24.0
TERMT (T .GE. TSTOP, 'checked on communication interval: REACHED TSTOP')

END ! DYNAMIC

TERMINAL
! code that is executed once at the end of a simulation run goes here
END ! TERMINAL

END ! PROGRAM
```

Oxytetracycline PBPK model code in Berkeley Madonna™ format

```
METHOD RK4

STARTTIME = 0
STOPTIME = 105;24
DT = 0.0025
DTOUT = 0.1

; Physiological parameters
; Blood flow rates
QCC = 12.9 ; Cardiac output (L/h/kg) (Brown et al., 1997, Table 22 for mixed-breed (Mongrel)
dogs (8.39 L/h/kg) and beagles (12.9 L/h/kg))
QLC = 0.297 ; Fraction of blood flow to the liver (Brown et al., 1997, Table 26)
QKC = 0.173 ; Fraction of blood flow to the kidneys (Brown et al., 1997, Table 26)
QFC = 0.097 ; Fraction of blood flow to the fat (Vinegar, 2001, Table 2)
QMC = 0.217 ; Fraction of blood flow to the muscle (Brown et al., 1997, Table 26)

; Tissue volumes
BW = 11.3 ; Body weight (kg) (Baggot et al., 1977)
VLC = 0.0329 ; Fractional liver tissue (Brown et al., 1997, Table 6)
VKC = 0.0055 ; Fractional kidney tissue (Brown et al., 1997, Table 6)
VFC = 0.15 ; Fractional fat tissue (Vinegar, 2001, Table 2)
VMC = 0.4565 ; Fractional muscle tissue (Brown et al., 1997, Table 6)
VbloodC = 0.082 ; Blood volume, fraction of BW (Brown et al., 1997, Table 21)

; Fraction of tissue volumes that is blood ! (Brown et al., 1997; Table 30)
FVBF = 0.02 ; Blood volume fraction of fat (%)
FVBM = 0.01 ; Blood volume fraction of muscle (%)
FVBS = 0.01 ; Blood volume fraction of slowly perfused tissues (%)

; Mass Transfer Parameters (Chemical-specific parameters)
; Partition coefficients (PC, tissue:plasma)
PL = 1.89 ; Liver:plasma PC (Craigmill et al., 2000, Table 3; Craigmill, 2003, Table 4, in sheep)
PK = 4.75 ; Kidney:plasma PC (Craigmill et al., 2000, Table 3; Craigmill, 2003, Table 4, in
sheep)
PM = 0.85 ; Muscle:plasma PC (Craigmill et al., 2000, Table 3; Craigmill, 2003, Table 4, in
sheep)
PF = 0.086 ; Fat:plasma PC (Craigmill et al., 2000, Table 3; Craigmill, 2003, Table 4, in sheep)
PR = 4.75 ; Richly perfused tissues:plasma PC (Assumed the same as kidney:plasma PC)
PS = 0.85 ; Slowly perfused tissues:plasma PC (Assumed the same as muscle:plasma PC)

; Permeability constants (L/h/kg tissue) (Permeation area cross products)
PAFC = 0.012 ; Fat tissue permeability (0.12 in Leavens et al., 2012)
PAMC = 0.225 ; Muscle tissue permeability (0.45 in Leavens et al., 2012)
PASC = 0.049 ; Slowly perfused tissue permeability (0.49 in Leavens et al., 2012)

; Kinetic constants
; Oral absorption rate constants
Kst = 2 ; /h, gastric emptying rate constant
Ka = 0.05; 0.012 ; /h, intestinal absorption rate constant, ka=0.05 for experimental solution;
ka=0.012 for tablets or capsules
Kint = 0.2 ; /h, intestinal transit rate constant

; IM absorption rate constants
```

```
Kim = 0.3 ; 0.15 for conventional formulation; 0.3 for long-acting formulation; IM absorption rate constant (/h)
Frac = 0.5 ; 0.95 for conventional formulation; 0.5 for long-acting formulation
Kdiss = 0.02 ; /h

; IV infusion rate constants
Timeiv = 0.01 ; IV infusion time (h), based on Leavens et al., 2012

; Urinary elimination rate constant
KurineC = 0.2 ; L/h/kg
; Parameters for various exposure scenarios
PDOSEoral = 0 ; (mg/kg)
PDOSEiv = 0 ; (mg/kg)
PDOSEim = 20 ; (mg/kg)

; Cardiac output and blood flows to tissues (L/h)
QC = QCC*BW ; Cardiac output
QL = QLC*QC ; Liver
QK = QKC*QC ; Kidney
QF = QFC*QC ; Fat
QM = QMC*QC ; Muscle
QR = 0.626*QC-QK-QL ; Richly perfused tissues
QS = 0.374*QC-QF-QM ; Slowly perfused tissues

; Tissue volumes (L)
VL = VLC*BW ; Liver
VK = VKC*BW ; Kidney
VF = VFC*BW ; Fat
VM = VMC*BW ; Muscle
Vblood = VbloodC*BW ; Blood
VR = 0.142*BW-VL-VK ; Richly perfused tissues
VS = 0.776*BW-VF-VM ; Slowly perfused tissues

; Permeability surface area coefficients
PAF = PAFC*VF ; Fat:blood permeability (L/h)
PAM = PAMC*VM ; Muscle:blood permeability (L/h)
PAS = PASC*VS ; Slowly perfused tissue:blood permeability (L/h)

; volume of tissue vs blood
; Fat
VFb = FVBF*VF ; Fat compartment blood volume
VFt = VF-VFb ; Fat compartment tissue volume

; Muscle
VMb = FVBM*VM ; Muscle compartment blood volume
VMt = VM-VMb ; Muscle compartment tissue volume
```

```
; Slowly perfused tissue
VSb = FVBS*VS ; Slowly perfused compartment blood volume
VSt = VS-VSb ; Slowly perfused compartment tissue volume

; Dosing
DOSEoral = PDOSEoral*BW ; (mg)
DOSEiv = PDOSEiv*BW ; (mg)
DOSEim = PDOSEim*BW ; (mg)

; Dosing, multiple oral gavage
tlen = 0.1 ; Length of oral gavage exposure (h/day)
tinterval = 6 ; Varied dependent on the exposure paradigm
Dstart = 0.0 ; Initiation day of oral gavage (day)
Dstop = 0.2 ; Termination day of oral gavage (day)
MAXT = 1.0 ; maximum comm. interval
CINTC = 0.1 ; Communication interval
CINT = CINTC ; Communication interval

Tsim = STOPTIME ; Tstop in hours
DS = Dstart*24 ; Initiation time point of oral gavage (h)
Doff = (Dstop - Dstart)*24 ; Oral gavage duration (h)

TimeOn = Dstart*24 ; Initiation time point of oral gavage (h)
TimeOff = Dstop*24+tlen ; Termination time point of oral gavage (h)

; Exposure = PULSE(0,tinterval,tlen)*PULSE(DS, Tsim, Doff)
Exposure1 = SQUAREPULSE(0,tlen)
Exposure2 = SQUAREPULSE(0,tlen) + SQUAREPULSE(0+tinterval,tlen)
Exposure5 = SQUAREPULSE(0,tlen) + SQUAREPULSE(0+tinterval,tlen) +
SQUAREPULSE(0+2*tinterval,tlen) + SQUAREPULSE(0+3*tinterval,tlen) +
SQUAREPULSE(0+4*tinterval,tlen)

Rdoseoral = (DOSEoral/tlen)*Exposure5
RAST = Rdoseoral-Kst*AST
d/dt(AST) = RAST
init AST = 0
RAI = Kst*AST-Ka*AI-Kint*AI
Rcolon=Kint*AI
d/dt(Acolon) = Rcolon
init Acolon = 0
d/dt(AI) = RAI
init AI = 0
RAO=Ka*AI
d/dt(AAO) = RAO
init AAO = 0
```

```
; Dosing, intramuscular, dissolution model
Doseimfast = Doseim*Frac
Doseimslow = Doseim*(1-Frac)

Rim = Kim*Amtsite
d/dt(Absorb) = Rim
init Absorb = 0
Rsite=-Rim+Kdiss*Doseimremain
d/dt(Amtsite) = Rsite
init Amtsite = Doseimfast
Rdoseimremain = -Kdiss*Doseimremain
d/dt(Doseimremain) = Rdoseimremain
init Doseimremain = Doseimslow

; OTC iv injection to the venous
IVR = DOSEiv/Timeiv
RIV = IVR*(1.-step(1,Timeiv))
d/dt(Aiv) = Riv
init Aiv = 0

; Urinary elimination rate constant
Kurine = KurineC*BW

; OTC in blood compartment
CV = ((QL*CVL+QK*CVK+QF*CVF+QM*CVM+QR*CVR+QS*CVS+Riv+Rim)/QC)
RA = QC*(CV-CA)
d/dt(AA) = RA
init AA = 0
CA = AA/Vblood
d/dt(AUCCV) = CV
init AUCCV = 0

; OTC in liver compartment
RL = QL*(CA-CVL)+RAO
d/dt(AL) = RL
init AL = 0
CL = AL/VL
CVL = AL/(VL*PL)
d/dt(AUCCL) = CL
init AUCCL = 0

; OTC in kidney compartment
RK = QK*(CA-CVK)-Rurine
d/dt(AK) = RK
init AK = 0
```

```
CK = AK/VK
CVK = AK/(VK*PK)
d/dt(AUCCK) = CK
init AUCCK = 0

; Urinary excretion of OTC
Rurine = Kurine*CVK
d/dt(Aurine) = Rurine
init Aurine = 0

; OTC in fat compartment, permeability-limited model
RFB = QF*(CA-CVF)-PAF*CVF+(PAF*CFt)/PF
d/dt(AFB) = RFB
init AFB = 0
CVF = AFB/VFB

RFt = PAF*CVF-(PAF*CFt)/PF
d/dt(AFt) = RFt
init AFt = 0
CFt = AFt/VFT
Aftotal = Aft+Afb
CF = Aftotal/VF

; OTC in muscle compartment, permeability-limited model
RMB = QM*(CA-CVM)-PAM*CVM+(PAM*CMt)/PM
d/dt(AMB) = RMB
init AMB = 0
CVM = AMB/VMB

RMt = PAM*CVM-(PAM*CMt)/PM
d/dt(AMt) = RMt
init AMt = 0
CMt = AMt/VMT
AMtotal = AMt+AMb
CM = AMtotal/VM
d/dt(AUCCM) = CM
init AUCCM = 0

; OTC in richly perfused tissue compartment
RR = QR*(CA-CVR)
d/dt(AR) = RR
init AR = 0
CR = AR/VR
CVR = AR/(VR*PR)

; OTC in slowly perfused tissue compartment, permeability-limited model
```

```
RSB = QS*(CA-CVS)-PAS*CVS+(PAS*CSt)/PS
d/dt(ASB) = RSB
init ASB = 0
CVS = ASB/VSB

RSlt = PAS*CVS-(PAS*CSt)/PS
d/dt(ASlt) = RSlt
init ASlt = 0
CSt = ASlt/VST
AStotal = ASlt+ASb
CS = AStotal/VS

; Mass balance
Qbal = QC-QL-QK-QF-QM-QR-QS
Tmass = AA+AL+AK+AFtotal+AMtotal+AR+AStotal+Aurine
Bal = AAO+Aiv+Absorb-Tmass ; Permeability-limited model mass balance
```

Oxytetracycline PBPK model code in MATLAB® format

```
function
dY=OTCPBPK(T,Y,PDOSEORAL,PDOSEIV,PDOSEIM,TINTERVAL,KA,TSTOP,KIM,FRAC)

%% Constants
% INITIAL Values
% code that is executed once at the beginning of a simulation run goes here
% Physiological parameters
% Blood flow rates
QCC = 12.9; % Cardiac output (L/h/kg) (Brown et al., 1997, Table 22 for
mixed-breed (Mongrel) dogs (8.39 L/h/kg) and beagles (12.9 L/h/kg))
QLC = 0.297; % Fraction of blood flow to the liver (Brown et al., 1997, Table
26)
QKC = 0.173; % Fraction of blood flow to the kidneys (Brown et al., 1997,
Table 26)
QFC = 0.097; % Fraction of blood flow to the fat (Vinegar, 2001, Table 2)
QMC = 0.217; % Fraction of blood flow to the muscle (Brown et al., 1997,
Table 26)

% Tissue volumes
BW = 11.3; % Body weight (kg) (Baggot et al., 1977)
VLC = 0.0329; % Fractional liver tissue (Brown et al., 1997, Table 6)
VKC = 0.0055; % Fractional kidney tissue (Brown et al., 1997, Table 6)
VFC = 0.15; % Fractional fat tissue (Vinegar, 2001, Table 2)
VMC = 0.4565; % Fractional muscle tissue (Brown et al., 1997, Table 6)
VbloodC = 0.082; % Blood volume, fraction of BW (Brown et al., 1997, Table
21)

%Fraction of tissue volumes that is blood; % (Brown et al., 1997; Table 30)
FVBF = 0.02; % Blood volume fraction of fat (%)
FVBM = 0.01; % Blood volume fraction of muscle (%)
FVBS = 0.01; % Blood volume fraction of slowly perfused tissues (%)
```

```
% Mass Transfer Parameters (Chemical-specific parameters)
% Partition coefficients (PC, tissue:plasma)
PL = 1.89; % Liver:plasma PC (Craigmill et al., 2000, Table 3; Craigmill,
2003, Table 4, in sheep)
PK = 4.75; % Kidney:plasma PC (Craigmill et al., 2000, Table 3; Craigmill,
2003, Table 4, in sheep)
PM = 0.85; % Muscle:plasma PC (Craigmill et al., 2000, Table 3; Craigmill,
2003, Table 4, in sheep)
PF = 0.086; % Fat:plasma PC (Craigmill et al., 2000, Table 3; Craigmill,
2003, Table 4, in sheep)
PR = 4.75; % Richly perfused tissues:plasma PC (Assumed the same as
kidney:plasma PC)
PS = 0.85; % Slowly perfused tissues:plasma PC (Assumed the same as
muscle:plasma PC)

%Permeability constants (L/h/kg tissue) (Permeation area cross products)
PAFC = 0.012; % Fat tissue permeability (0.12 in Leavens et al., 2012)
PAMC = 0.225; % Muscle tissue permeability (0.45 in Leavens et al., 2012)
PASC = 0.049; % Slowly perfused tissue permeability (0.49 in Leavens et al.,
2012)

% Kinetic constants
% Oral absorption rate constants
Kst = 2; % /h, gastric emptying rate constant
Ka = KA; % /h, intestinal absorption rate constant, ka=0.05 for experimental
solution; ka=0.012 for tablets or capsules
Kint = 0.2; % /h, intestinal transit rate constant

% IM absorption rate constants
Kim = KIM; % 0.15 for conventional formulation; 0.3 for long-acting
formulation; IM absorption rate (/h)
Frac = FRAC; % 0.95 for conventional formulation; 0.5 for long-acting
formulation
Kdiss = 0.02; % /h

% IV infusion rate constants
timeiv = 0.01; % IV infusion time (h), based on Leavens e al., 2012

% Urinary elimination rate constant
KurineC = 0.2; % L/h/kg

%Parameters for various exposure scenarios
PDOSEoral = PDOSEORAL; % (mg/kg)
PDOSEiv = PDOSEIV; % (mg/kg)
PDOSEim = PDOSEIM; % (mg/kg)

% Cardiac output and blood flows to tissues (L/h)
QC = QCC*BW; % Cardiac output
QL = QLC*QC; % Liver
QK = QKC*QC; % Kidney
QF = QFC*QC; % Fat
QM = QMC*QC; % Muscle
QR = 0.626*QC-QK-QL; % Richly perfused tissues
QS = 0.374*QC-QF-QM; % Slowly perfused tissues
```

```
% Tissue volumes (L)
VL = VLC*BW; % Liver
VK = VKC*BW; % Kidney
VF = VFC*BW; % Fat
VM = VMC*BW; % Muscle
Vblood = VbloodC*BW; % Blood
VR = 0.142*BW-VL-VK; % Richly perfused tissues
VS = 0.776*BW-VF-VM; % Slowly perfused tissues

%Permeability surface area coefficients
PAF = PAFC*VF; % Fat:blood permeability (L/h)
PAM = PAMC*VM; % Muscle:blood permeability (L/h)
PAS = PASC*VS; % Slowly perfused tissue:blood permeability (L/h)

% volume of tissue vs blood
% Fat
VFb = FVBF*VF; % Fat compartment blood volume
VFt = VF-VFb; % Fat compartment tissue volume

% Muscle
VMb = FVBM*VM; % Muscle compartment blood volume
VMt = VM-VMb; % Muscle compartment tissue volume

% Slowly perfused tissue
VSB = FVBS*VS; % Slowly perfused compartment blood volume
VSt = VS-VSB; % Slowly perfused compartment tissue volume

% Dosing
Doseoral = PDOSEoral*BW; % (mg)
DOSEiv = PDOSEiv*BW; % (mg)
Doseim = PDOSEim*BW; % (mg)

%...Dosing, multiple oral gavage
tlen = 0.002; % Length of oral gavage exposure (h/day)
tinterval = TINTERVAL; % Varied dependent on the exposure paradigm
Dstart = 0.0; % Initiation day of oral gavage (day)
TSTOP = TSTOP; % Termination day of oral gavage (day)
MAXT = 1.0 ; % maximum comm. interval
CINTC = 0.1; % Communication interval
CINT = CINTC ; % Communication interval

Tsim = 24*TSTOP; % Tstop in hours
DS = Dstart*24 ; % Initiation time point of oral gavage (h)
Doff = (TSTOP - Dstart)*24; % Oral gavage duration (h)

TimeOn = Dstart*24; % Initiation time point of oral gavage (h)
TimeOff = TSTOP*24+tlen; % Termination time point of oral gavage (h)

% END INITIAL

%% Variables
AST=Y(1);
Acolon=Y(2);
AI=Y(3);
```

```
AAO=Y(4);
Absorb=Y(5);
Amtsite=Y(6);
Doseimremain=Y(7);
Aiv=Y(8);
AA=Y(9);
AUCCV=Y(10);
AL=Y(11);
AUCCL=Y(12);
AK=Y(13);
AUCCK=Y(14);
Aurine=Y(15);
AFB=Y(16);
AFt=Y(17);
AMB=Y(18);
AMt=Y(19);
AUCCM=Y(20);
AR=Y(21);
ASB=Y(22);
ASlt=Y(23);

%%

Exposure = pulstran(T-tlen/2,tinterval,@rectpuls,tlen)*(1-heaviside(T-
TimeOff));
RDoseoral = (Doseoral/tlen)*Exposure;
RAST = RDoseoral-Kst*AST;
dY(1) = (RAST);
RAI = Kst*AST-Ka*AI-Kint*AI;
Rcolon=Kint*AI;
dY(2)=(Rcolon);
dY(3) = (RAI);
RAO=Ka*AI;
dY(4) = (RAO);

%...Dosing, intramuscular, dissolution model
Doseimfast = Doseim*Frac;
Doseimslow = Doseim*(1-Frac);

Rim=Kim*Amtsite;
dY(5)=(Rim);
Rsite=-Rim+Kdiss*Doseimremain;
dY(6)=(Rsite); %%% ,Doseimfast
Rdoseimremain=-Kdiss*Doseimremain;
dY(7)=(Rdoseimremain); %%% ,Doseimslow

% OTC iv injection to the venous
IVR = DOSEiv/timeiv;
Riv = IVR*(1.-heaviside(T-timeiv));
dY(8) = (Riv);

% Urinary elimination rate constant
Kurine = KurineC*BW;

CA = AA/Vblood;
```

```
% OTC in kidney compartment
CVK = AK/(VK*PK);
Rurine = Kurine*CVK;
RK = QK*(CA-CVK)-Rurine;
dY(13) = (RK);
CK = AK/VK;
dY(14) = (CK);

% Urinary excretion of OTC
dY(15) = (Rurine);

% OTC in fat compartment, permeability-limited model
CFt = AFt/VFt;
CVF = AFB/VFb;
RFB = QF*(CA-CVF)-PAF*CVF+(PAF*CFt)/PF;
dY(16) = (RFB);

RFt = PAF*CVF-(PAF*CFt)/PF;
dY(17) = (RFt);
Aftotal = AFt+AFB;
CF = Aftotal/VF;

% OTC in muscle compartment, permeability-limited model
CMt = AMt/VMt;
CVM = AMB/VMb;
RMB = QM*(CA-CVM)-PAM*CVM+(PAM*CMt)/PM;
dY(18) = (RMB);

CVL = AL/(VL*PL);
% OTC in liver compartment
RL = QL*(CA-CVL)+RAO;
dY(11) = (RL);
CL = AL/VL;
dY(12) = (CL);

RMt = PAM*CVM-(PAM*CMt)/PM;
dY(19) = (RMt);
AMtotal = AMt+AMB;
CM = AMtotal/VM;
dY(20) = (CM);

% OTC in richly perfused tissue compartment
CVR = AR/(VR*PR);
RR = QR*(CA-CVR);
dY(21) = (RR);
CR = AR/VR;

% OTC in slowly perfused tissue compartment, permeability-limited model
CSt = ASlt/VSt;
CVS = ASB/VSB;
RSB = QS*(CA-CVS)-PAS*CVS+(PAS*CSt)/PS;
dY(22) = (RSB);

RSlt = PAS*CVS-(PAS*CSt)/PS;
dY(23) = (RSlt);
```

```
AStotal = ASlt+ASB;
CS = AStotal/VS;

% OTC in blood compartment
CV = ((QL*CVL+QK*CVK+QF*CVF+QM*CVM+QR*CVR+QS*CVS+Riv+Rim)/QC);
RA = QC*(CV-CA);
dY(9) = (RA);
dY(10) = (CV);

% Mass balance
Qbal = QC-QL-QK-QF-QM-QR-QS;
Tmass = AA+AL+AK+Aftotal+AMtotal+AR+AStotal+Aurine;
Bal = AAO+Aiv+Absorb-Tmass; %Permeability-limited model mass balance


dY = dY(:);
```

Oxytetracycline PBPK model code in R language format

```
library(deSolve) #load and attach add-on packages.
# time parameter
starttime <- 0
stoptime <- 105
dtout <- 0.001 # resolution of output time
Times <- seq(starttime, stoptime, dtout)
#############################################################################
######################################
## parameters to be changed as
follows:#####################################################################
#######

# Parameters for exposure scenario
PDOSEiv = 0 # (mg/kg)
PDOSEim = 0 #(mg/kg)
PDOSEoral = 44 # (mg/kg)
#IM absorption rate constats
Kim = 0.3 # 0.15 for conventional formulation# 0.3 for long-acting
formulation# IM absorption rate constant(/h)
Frac = 0.5 # 0.95 for conventional formulation#0.5 for long-acting
formulation#
Kdiss = 0.02 #/h
#Dosing, multiple oral gavage
tlen = 0.001 # Length of oral gavage exposure (h/day)
tinterval = 6 # varied dependent on the exposure paradigm
tdose = 1 # dose times
Ka = 0.012 # for tablets or capsules
#Ka = 0.05 # Ka = 0.05 for experimental solution/h, intestinal absorption
rate constant,


#############################################################################
######################################
```

```
## Fixed parameters as
follows:#####################################################################
###############
# Physiological parameters
# Blood flow rates
QCC = 12.9 # Cardiac output (L/h/kg)
QLC = 0.297 # Fraction of flow to the liver
QKC  = 0.173 # Fraction of flow to the kidneys
QFC = 0.097 # Fraction of flow to the fat
QMC = 0.217 # Fraction of flow to the muscle
# Tissue  volumes
BW = 11.3 # Body weight(kg)
VLC = 0.0329 # Fractional liver tissue
VKC = 0.0055 # Fractional kidney tissue
VFC = 0.15 # Fractional fat tissue
VMC = 0.4565 # Fractional muscle tissue
VbloodC = 0.082 # Blood volume, fractional of BW

#Fraction of tissue volumes that is blood!
FVBF = 0.02# Blood volume fraction of fat (%)
FVBM = 0.01# Blood volume fraction of muscle (%)
FVBS = 0.01# Blood volume fraction of slowly perfused tissue (%)

# Mass Transfer Parameters (Chemical-specific parameters)
# Partition coefficients(PC, tissue:plasma)
PL = 1.89 # Liver: plasma PC
PK = 4.75 # Kidney:plasma PC
PM = 0.85# Muscle:plasma PC
PF = 0.086 #Fat:plasma PC
PR = 4.75# Richly perfused tissues:plasma PC
PS = 0.85# Slowly perfused tissues:plasma PC

#Permeability constatns (L/h/kg tissue) (Permeation area cross products)
PAFC = 0.012# Fat tissue permeability constant
PAMC = 0.225# Muscle tissue permeability constant
PASC = 0.049# Slowly perfused tissue permeability constant

#Kinetic constats
#Oral absorbtion rate constants
Kst = 2 # /h, gastric emptying rate constant
Kint = 0.2 #/h, intestinal transit rate constant.

# IV infusion rate constants
Timeiv = 0.01 # IV injection/infusion time (h)

# Urinary elimination rate constant adjusted by bodyweight
KurineC = 0.2 # L/h/kg
# Urinary elimination rate constant
Kurine = KurineC * BW #L/h


# Cardiac output and blood flows to tissues(L/h)
QC = QCC * BW # Cardiac output
QL = QLC * QC # Liver
QK = QKC * QC # Kidney
QF = QFC * QC # Fat
QM = QMC * QC # Muscle
```

```
QR = 0.626 * QC - QK - QL # Richly perfused tissues
QS = 0.374* QC - QF - QM # Slowly perfused tissues

#  Tissue volumes (L)
VL = VLC * BW # Liver
VK = VKC * BW # Kidney
VF = VFC * BW # Fat
VM = VMC * BW # Muscle
Vblood = VbloodC * BW# Blood
VR = 0.142 * BW - VL - VK # Richly perfused tissues
VS = 0.776 * BW - VF -VM # Slowly perfused tissues

#Permeability surface area coefficients
PAF = PAFC * VF# Fat:blood permeability (L/h)
PAM = PAMC * VM# Muscle : blood permeability (L/h)
PAS = PASC * VS# Slowly perfused tissue:blood permeability (L/h)

#Volume of tissue vs blood
#Fat
VFB = FVBF * VF # fat compartment blood volume
VFT = VF - VFB# Fat compartment tissue volume

#Muscle
VMB = FVBM * VS# Muscle compartment blood volume
VMT = VM - VMB# Muslce compartment blood volume
#Slowly perfused tissue
VSB = FVBS * VS# Slowly perfused compartment blood volume
VST = VS - VSB#  Slowly perfused compartment blood volume

# Dosing
DOSEoral = PDOSEoral * BW # (mg)
DOSEiv = PDOSEiv * BW # (mg)
DOSEim = PDOSEim * BW # (mg)
# Dosing, intramuscular, dissolution model
DOSEimfast = DOSEim * Frac
DOSEimslow = DOSEim * (1 - Frac)

# IVR & oral rate constant
IVR = DOSEiv/Timeiv
oralR = DOSEoral/tlen
## PBPK model
pbpkmodel <- function(Time, State, Parmeters) {
  with(as.list(c(State, Paras)), {

############################################################################
######################################
    ## Concentration of the chemical in vein compartment
    CVL = AL/(VL * PL) # con' of chem in liver  / PC of plasma: liver
    CVK = AK/(VK * PK) # con' of chem in Kidney / PC of plasma: kidney
    CVF = AFB/VFB
    CVR = AR/(VR * PR) # con' of chem in RPT    / PC of plasma: richly
perfused tissue
    CVS = ASB/VSB # con' of chem in SPT: slowly perfused tissue
    CVM = AMB/VMB
    CMT = AMT/VMT
    CFT = AFT/VFT
    CST = ASLT/VST
    # OTCiv injection to the venous
    RIV = IVR * (Time < Timeiv)
    dAIV = RIV
    # OTC oral to the stomach
    RDOSEoral <- oralR * (Time <= tdose * tinterval) * (Time %% tinterval <
tlen)
    dADOSEoral = RDOSEoral
    RAST = RDOSEoral - Kst * AST
    dAST = RAST
    # OTCim injetion to the muscle
    RDOSEimremain = -Kdiss * DOSEimremain
    dDOSEimremain = RDOSEimremain
    Rim = Kim * Amtsite
    dAbsorb = Rim
    Rsite = - Rim + Kdiss * DOSEimremain
    dAmtsite = Rsite

#############################################################################
######################################
    ## OTC in Intestine and Colon
    RAI = Kst * AST - Kint * AI - Ka * AI
    dAI = RAI
    Rcolon = Kint * AI
    dAcolon = Rcolon
    RAO = Ka * AI
    dAAO = RAO

#############################################################################
######################################
    ## OTC in blood compartment
    #con' of chemical in the vein
    CV = ((QL * CVL + QK  * CVK + QF * CVF + QM * CVM + QR * CVR + QS * CVS +
RIV + Rim) / QC)

    CA = AA/Vblood # con' of chem in artery = amount of chem in artery /
volume of blood
    RA = QC * (CV - CA) # rate of change in amount of chem in tissue of blood
compartment
    dAA = RA
    dAUCCV = CV

#############################################################################
######################################
    ## OTC in liver compartment
    RL = QL * (CA - CVL) + RAO # rate of change in amount of the chem in
liver
    dAL = RL # amount of chemical in liver
    CL = AL/VL # con' of chem in liver
    dAUCCL = CL

#############################################################################
######################################
    ## OTC in kidney compartment
    # Urinary excretion of OTC
    Rurine = Kurine * CVK
    dAurine = Rurine
    # kidney
    RK = QK * (CA - CVK) - Rurine
    dAK = RK
    CK = AK/VK # con' of chem in kieney
    dAUCCK = CK

#############################################################################
######################################
    ## OTC in muslce compartment
    RMB = QM * (CA - CVM)- PAM * CVM + (PAM * CMT)/PM
    dAMB = RMB # amount of chemical in muscle
    RMT = PAM * CVM - (PAM * CMT)/PM
    dAMT = RMT
    AMtotal = AMT + AMB
    CM =AMtotal/VM
    dAUCCM = CM

#############################################################################
######################################
    ## OTC in fat compartment
    RFT = PAF * CVF - (PAF * CFT)/PF
    dAFT = RFT
    RFB = QF * (CA - CVF) - PAF * CVF + (PAF * CFT)/PF
    dAFB = RFB # amount of chemical in fat
    AFtotal = AFT + AFB
    CF =AFtotal/VF

#############################################################################
######################################
    ## OTC in RPT of body compartment
    RR = QR * (CA - CVR)
    dAR = RR
    CR = AR/VR

#############################################################################
######################################
    ## OTC in SPT of body compartment
    RSB = QS * (CA - CVS) - PAS * CVS + (PAS * CST)/PS
    dASB = RSB
    RSLT = PAS * CVS - (PAS * CST)/PS
    dASLT = RSLT
    AStotal = ASLT + ASB
    CS =AStotal/VS

#############################################################################
######################################
    ## Mass balance
    Qbal = QC - QL - QK - QM - QF - QR - QS
    Tmass = AA + AL + AK + Aurine + AMtotal + AFtotal + AR + AStotal
    list(c(dAbsorb, dAmtsite, dDOSEimremain, dAIV, dADOSEoral, dAST,
           dAI, dAcolon, dAAO, dAA, dAL, dAK, dAurine, dAMB,
           dAMT, dAUCCV, dAUCCL, dAUCCK, dAUCCM, dAFB, dAFT,
           dAR, dASB, dASLT))
  })
}
State <- c(Absorb = 0, Amtsite = DOSEimfast, DOSEimremain = DOSEimslow, AIV =
0, ADOSEoral = 0, AST = 0,
           AI = 0, Acolon = 0, AAO = 0, AA = 0, AL = 0, AK = 0,  Aurine = 0, 
AMB = 0, 
           AMT = 0, AUCCV = 0, AUCCL = 0, AUCCK = 0, AUCCM = 0, AFB = 0, AFT 
= 0, 
           AR = 0, ASB = 0, ASLT = 0)
Paras <- c(Kst, Ka, Kint, Kim, Kdiss, QC, QL, QK, QM, QF, QR, QS, PAF, PAM, 
PAS)

# PBPK output
out <- ode(y = State,
           times = Times,
           func = pbpkmodel,
           parms = Paras)
pbpkout <- as.data.frame(out[seq(from = 1, to = dim(out)[1], by = 
0.1/dtout),])
newtime <- Times[seq(from = 1, to = dim(out)[1], by = 0.1/dtout)]
## Check balance
Tmass = pbpkout$AA + pbpkout$AL + pbpkout$AK + pbpkout$Aurine + pbpkout$AMT + 
pbpkout$AMB + 
  pbpkout$AFT + pbpkout$AFB + pbpkout$AR + pbpkout$ASLT + pbpkout$ASB 
Bal = pbpkout$AAO + pbpkout$AIV + pbpkout$Absorb - Tmass #permeability-
limited model mass balance
plot(newtime, Bal, col = 'red')

#############################################################################
######################################
#############################################################################
######################################
#############################################################################
######################################
# You should change the path bellow according to your computer, and change 
the name in quotes of write.csv function
# so that for each PBPK model, the name would not be replaced by latter 
generated files.
#!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
setwd('Z:/AP 873/module7/files') # path
amoumt.drug <- cbind(pbpkout, Tmass, Bal)
write.csv(amoumt.drug, 'amount of drug.csv') # change file name here for 
different dosing
#!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
#############################################################################
######################################
#############################################################################
######################################
#############################################################################
######################################


## plot concentration in all compartments
# OTC in blood compartment
## Concentration of the chemical in vein compartment
CVL = pbpkout$AL/(VL * PL) # liver
CVK = pbpkout$AK/(VK * PK) # kidney
CVM = pbpkout$AMB/VMB # muscle
CVF = pbpkout$AFB/VFB # Fat
```

```
CVR = pbpkout$AR/(VR * PR) # richly perfused tissue
CVS = pbpkout$ASB/VSB # slowly perfused tissue
# concentration of chemical in the vein
RIV = IVR * (newtime < Timeiv)
Rim = Kim * pbpkout$Amtsite
# OTC in vein compartment
CV = ((QL * CVL + QK  * CVK + QM * CVM + QF * CVF + QR * CVR + QS * CVS + RIV
+ Rim) / QC)
plot(newtime, CV, type = 'l', xlab = c('Time'),  col = 1)
# OTC in artery compartment
CA = pbpkout$AA/Vblood
plot(newtime, CA, type = 'l', xlab = c('Time'),  col = 2)

# OTC in liver compartment
CL = pbpkout$AL/VL
plot(newtime, CL, type = 'l', xlab = c('Time'),  col = 3)

# OTC in kidney compartment
CK = pbpkout$AK/VK
plot(newtime, CK, type = 'l', xlab = c('Time'),  col = 4)

# OTC in muslce compartment
AMtotal = pbpkout$AMT + pbpkout$AMB
CM = AMtotal/VM
plot(newtime, CM, type = 'l', xlab = c('Time'),  col = 5)

# OTC in fat compartment
AFtotal = pbpkout$AFT + pbpkout$AFB
CF = AFtotal/VF
plot(newtime, CF, type = 'l', xlab = c('Time'),  col = 6)

# OTC in RPT compartment
CR = pbpkout$AR/VR
plot(newtime, CR, type = 'l', xlab = c('Time'),  col = 7)

# OTC in SPT compartment
AStotal = pbpkout$ASLT + pbpkout$ASB
CS = AStotal/VS
plot(newtime, CS, type = 'l', xlab = c('Time'),  col = 8)

#############################################################################
######################################
#############################################################################
######################################
#############################################################################
######################################
# you should change the path bellow according to your computer, and change
the name in quotes of write.csv function
# so that for each PBPK model, the name would not be replaced by latter
generated files.
#!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
concentration.durg <- data.frame(newtime, CV, CA, CL, CK, CM, CF, CR, CS,
CVL, CVK, CVM, CVF, CVR, CVS)
write.csv(concentration.durg, 'concentration of drug-XXmg.csv') # change file
name here for different dosing
```

```
#!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
####################################################################################
######################################
####################################################################################
######################################
####################################################################################
######################################
```

A representative runtime file from the oxytetracycline PBPK model in acslX™ format

```
% Data corresponds to Figure 2a in Lin et al. J. Pharm. Sci. 2015, 104:233-243.

prepare @all
PDOSEORAL = 0
PDOSEIV = 5 % mg/kg
PDOSEIM = 0
start @NoCallback

% OTC concentration in the serum, the column respectively is time, mean concentration, time,
% mean concentration plus SD, time, mean concentration minus SD.
% The unit of the concentration is ug/ml (mg/L).
SerumIV5 = [0.163   9.482   0.162   10.118 0.163   3.308
0.235   6.933   0.235   6.933   0.235   6.933
0.307   5.587   0.328   6.788   0.334   3.911
0.495   4.409   0.516   5.299   0.521   3.188
0.775   3.596   0.773   4.183   0.778   2.686
1.008   2.964   1.031   3.303   1.011   2.262
1.520   2.500   1.495   2.940   1.522   1.929
2.054   2.086   2.053   2.480   2.057   1.558
3.076   1.654   3.051   2.075   3.056   1.170
4.051   1.311   4.095   1.718   4.079   0.815
5.049   1.097   5.070   1.421   5.054   0.704
6.047   0.908   6.044   1.215   6.052   0.552
8.042   0.723   8.015   1.011   8.001   0.421
10.014 0.564   9.987   0.832   9.997   0.298
12.009 0.426   11.982 0.635   11.992 0.225]
plot(_t,_cv,SerumIV5(:,1),SerumIV5(:,2),'+',SerumIV5(:,3),SerumIV5(:,4),'+',SerumIV5(:,5),SerumIV5(:,6),'+',"SerumIV5.aps")
```

A representative Main script file from the oxytetracycline PBPK model in MATLAB® format

```
clear
clc
close all

%% Input Arguments
PDOSEORAL = 0;
PDOSEIV = 5; % mg/kg
PDOSEIM = 0;

TINTERVAL=0:6:24;
KA=0.012;
KIM = 0.3;
TSTOP=.2;
FRAC = 0.5;

BW = 11.3; % Body weight (kg) (Baggot et al., 1977)

% Dosing
 Doseim = PDOSEIM*BW; % (mg)

%...Dosing, intramuscular, dissolution model
 Doseimfast = Doseim*FRAC;
 Doseimslow = Doseim*(1-FRAC);


% Initial Conditions
Initials=zeros(23,1);
Initials(6)=Doseimfast;
Initials(7)=Doseimslow;

%% Running ODE Solver
ti=0;
tf=12;
tspan=[ti tf];
options=odeset('AbsTol',10e-2,'RelTol',10e-2,'Stats','on');
tic;
[t,y] =
ode23(@(T,Y)OTCPBPK(T,Y,PDOSEORAL,PDOSEIV,PDOSEIM,TINTERVAL,KA,TSTOP,KIM,FRAC
), tspan, Initials');
toc;

%% Evaluation for fixed step sizes
t_temp = ti:.1:tf;
y_temp=zeros(length(t_temp),27);
t_temp=t_temp';
for i=1:length(Initials)
y_temp(:,i) = interp1(t,y(:,i),t_temp);
end
t=t_temp;
y=y_temp;
```

```
%%
% OTC concentration in the serum, the column respectively is time, mean
concentration, time,
% mean concentration plus SD, time, mean concentration minus SD.
% The unit of the concentration is ug/ml (mg/L).

SerumIV5 = textread('SerumIV5.txt');



cv=zeros(length(t),1);
for i=1:length(t)-1
    cv(i,1)=(y(i+1,10)-y(i,10))/(t(i+1)-t(i));
end

figure;
plot(t,cv,SerumIV5(:,1),SerumIV5(:,2),'+',SerumIV5(:,3),SerumIV5(:,4),'+',Ser
umIV5(:,5),SerumIV5(:,6),'+')
set(gca,'yscale','log');
box off
xlabel('Time','fontsize',12)
ylabel('OTC Concentration','fontsize',12)
title('IV Serum 5mgkg','fontweight','bold','fontsize',13)
```

A representative DATA file used in the Main script file from the oxytetracycline PBPK model

Save the following data file as a .txt file with the name `'SerumIV5.txt'`.

0.163	9.482	0.162	10.118	0.163	3.308
0.235	6.933	0.235	6.933	0.235	6.933
0.307	5.587	0.328	6.788	0.334	3.911
0.495	4.409	0.516	5.299	0.521	3.188
0.775	3.596	0.773	4.183	0.778	2.686
1.008	2.964	1.031	3.303	1.011	2.262
1.52	2.5	1.495	2.94	1.522	1.929
2.054	2.086	2.053	2.48	2.057	1.558
3.076	1.654	3.051	2.075	3.056	1.17
4.051	1.311	4.095	1.718	4.079	0.815
5.049	1.097	5.07	1.421	5.054	0.704
6.047	0.908	6.044	1.215	6.052	0.552
8.042	0.723	8.015	1.011	8.001	0.421
10.014	0.564	9.987	0.832	9.997	0.298
12.009	0.426	11.982	0.635	11.992	0.225

Gold nanoparticle PBPK model code in acslX™ format

```
PROGRAM
! Gold Nanoparticle PBPK model, initiated by Zhoumeng Lin on June 23, 2014;
INITIAL
! code that is executed once at the beginning of a simulation run goes here
! Blood flow rate (Fraction of cardiac output)
CONSTANT QCC = 5.0!Cardiac output(L/h/kg) (Upton, 2008; Yuan et al., 2011)
CONSTANT QLC = 0.2725! Fraction of blood flow to liver (Average of Buur et al., 2005 and Upton, 2008)
CONSTANT QBRC = 0.03! Fraction of blood flow to brain (Upton,2008)
CONSTANT QKC = 0.12! Fraction of blood flow to kidneys (Average of Buur et al., 2005 and Upton, 2008)
CONSTANT QSC = 0.0151!Davies and Morris (1993)Table III! Fraction of blood flow to spleen (average of 6 species)

! Tissue volumes (Fraction of body weight)
CONSTANT BW0 = 9.6! Body weight (kg) (Upton, 2008)
CONSTANT VLC = 0.0247! Liver (Average of Buur et al., 2005 and Upton, 2008)
CONSTANT VBRC = 0.004! Brain (Upton, 2008)
CONSTANT VKC = 0.004! Kidneys (Average of Buur et al., 2005 and Upton, 2008)
CONSTANT VSC = 0.002! Spleen (Upton, 2008)
CONSTANT VLuC = 0.01! Lungs (Upton, 2008)
CONSTANT VBloodC = 0.06! Blood (Average of Buur et al., 2005 and Upton, 2008)
CONSTANT VPlasmaC = 0.04!0.04! Buur et al. 2005 (Hematocrit is 0.33)

! Blood volume fraction in organs and tissues (percentage of tissues) All rat values are from Table 30 in Brown et al. (1997).
CONSTANT BVL = 0.115! Liver (Buur et al. 2005)
CONSTANT BVBR = 0.0275! Brain (Brown et al., 1997; Table 30, average of 4 species)
CONSTANT BVK = 0.105! Kidneys (Buur et al., 2005)
CONSTANT BVS = 0.3! Spleen (Brown et al., 1997; Table 30, average of 3 species)
CONSTANT BVLu = 0.3867! Lungs (Brown et al., 1997; Table 30, average of 3 species)
CONSTANT BVrest = 0.026! Rest of body (Assume the same as the muscle in Buur et al. 2005)

! Partition coefficients (PC), unitless
CONSTANT PL = 0.08! Liver:blood PC (calculated using AUC method, 7-day AUC data from Cho et al., 2010, data for 13-nm AuNPs)
CONSTANT PBR = 0.15!Brain:blood PC (generic value, Li et al., 2014)
CONSTANT PK = 0.15!Kidneys:blood PC (calculated using AUC method, 7-day AUC data from Cho et al., 2010, data for 13-nm AuNPs)
CONSTANT PS = 0.15!Spleen:blood PC (calculated using AUC method, 7-day AUC data from Cho et al., 2010, data for 13-nm AuNPs)
CONSTANT PLu = 0.15!Lungs:blood PC (calculated using AUC method, 7-day AUC data from Cho et al., 2010, data for 13-nm AuNPs)
CONSTANT Prest = 0.15! Rest of body:blood PC (generic value, Li et al., 2014)
```

```
! Diffusion limitation coefficient constants, unitless (Based on Li et al., 2014)
CONSTANT PALC = 0.001! Permeability coefficient between blood and liver, generic value from Li et al., 2014
CONSTANT PABRC = 0.000001! Permeability coefficient between blood and brain, set a very low values
CONSTANT PAKC = 0.001!0.001! Permeability coefficient between blood and kidneys, generic value from Li et al., 2014
CONSTANT PASC =0.001! Permeability coefficient between blood and spleen, generic value from Li et al., 2014
CONSTANT PALuC = 0.001! Permeability coefficient between blood and lungs, generic value from Li et al., 2014
CONSTANT PArestC = 0.000001! Permeability coefficient between blood and rest of body, from Li et al., 2014

CONSTANT KLRESrelease = 0.005 ! Release rate constant of phagocytic cells, (h-1)
CONSTANT KLRESmax = 1000 ! Maximum uptake rate constant of phagocytic cells, (h-1)
CONSTANT KLRES50 = 24 ! Time reaching half maximum uptake rate, (h)
CONSTANT KLRESn = 0.5 ! Hill coefficient, (unitless)
CONSTANT ALREScap = 32.5 ! Uptake capacity per tissue weight (ug/g tissue)

CONSTANT KSRESrelease = 0.02 ! Release rate constant of phagocytic cells, (h-1)
CONSTANT KSRESmax = 500 ! Maximum uptake rate constant of phagocytic cells, (h-1)
CONSTANT KSRES50 = 24 ! Time reaching half maximum uptake rate, (h)
CONSTANT KSRESn = 0.5 ! Hill coefficient, (unitless)
CONSTANT ASREScap = 25 ! Uptake capacity per tissue weight (ug/g tissue)

CONSTANT KKRESrelease = 0.04 ! Release rate constant of phagocytic cells, (h-1)
CONSTANT KKRESmax = 500 ! Maximum uptake rate constant of phagocytic cells, (h-1)
CONSTANT KKRES50 = 24 ! Time reaching half maximum uptake rate, (h)
CONSTANT KKRESn = 0.5 ! Hill coefficient, (unitless)
CONSTANT AKREScap = 55 ! Uptake capacity per tissue weight (ug/g tissue)

CONSTANT KLuRESrelease = 0.02 ! Release rate constant of phagocytic cells, (h-1)
CONSTANT KLuRESmax = 300 ! Maximum uptake rate constant of phagocytic cells, (h-1)
CONSTANT KLuRES50 = 24 ! Time reaching half maximum uptake rate, (h)
CONSTANT KLuRESn = 0.5 ! Hill coefficient, (unitless)
CONSTANT ALuREScap = 25 ! Uptake capacity per tissue weight (ug/g tissue)

CONSTANT KrestRESrelease = 0.0001 ! Release rate constant of phagocytic cells, (h-1)
CONSTANT KrestRESmax = 0.00025 ! Maximum uptake rate constant of phagocytic cells, (h-1)
CONSTANT KrestRES50 = 24 ! Time reaching half maximum uptake rate, (h)
CONSTANT KrestRESn = 0.5 ! Hill coefficient, (unitless)
CONSTANT ArestREScap =0.05 ! Uptake capacity per tissue weight (ug/g tissue)

! Biliary excretion
```

```
CONSTANT KbileC = 0.0008!0.00008! Biliary clearance (L/hr/kg^0.75)
!CONSTANT Kbile = 0.0012! Biliary clearance (L/hr/kg)
! changed to L/h/kg^0.75 for interspecies extrapolation 0.0012/0.02^0.75 =
! Urine excretion
CONSTANT KurineC = 0.0008!0.0008! ! Urine clearance (L/hr/kg^0.75)
!CONSTANT Kurine = 0.00012! Urine clearance (L/hr)
! changed to L/h/kg^0.75 for interspecies extrapolation 0.00012/0.02^0.75 =


! IV dosing
CONSTANT Timeiv = 0.05!0.005 ! IV infusion time (h), set, approximately 15-20 seconds, on average 18 sec
CONSTANT PDOSEiv = 2!1!2 ! mg/kg

END ! INITIAL

DYNAMIC

ALGORITHM IALG = 2
NSTEPS    NSTP = 10
MAXTERVAL MAXT = 1.0e9
MINTERVAL MINT = 1.0e-9
CINTERVAL CINT = 0.1

DERIVATIVE
! code for calulating the derivative goes here
! Scaled papameters
! Cardiac output and regional blood blow (L/h)
BW = 9.6+0.015*t
QC = QCC*BW!**0.75 ! Cardiac output
QL = QC*QLC ! Blood flow to liver
QBR = QC*QBRC ! Blood flow to brain
QK = QC*QKC ! Blood flow to kidney
QS = QC*QSC ! Blood flow to spleen
Qrest = QC-QL-QBR-QK-QS! Blood flow to rest of body
Qbal = QC-QL-QBR-QK-QS-Qrest !Blood flow balance equation

! Tissue volumes (L)
VL = BW*VLC ! Liver
VBR = BW*VBRC ! Brain
VK = BW*VKC ! Kidney
VS = BW*VSC ! Spleen
VLu = BW*VLuC ! Lungs
VBlood = BW*VBloodC
VPlasma = BW*VPlasmaC
Vrest = BW-VL-VBR-VK-VS-VLu-VPlasma.....................................................................
```

```
Vbal = BW-VL-VBR-VK-VS-VLu-VPlasma-Vrest
VLb = VL*BVL ! Weight/volume of capillary blood in liver compartment
VLt = VL-VLb ! Weight/volume of tissue in liver compartment
VBRb = VBR*BVBR ! Weight/volume of capillary blood in brain compartment
VBRt = VBR-VBRb ! Weight/volume of tissue in brain compartment
VKb = VK*BVK ! Weight/volume of capillary blood in kidney compartment
VKt = VK-VKb ! Weight/volume of tissue in kidney compartment
VSb = VS*BVS ! Weight/volume of capillary blood in spleen compartment
VSt = VS-VSb ! Weight/volume of tissue in spleen compartment
VLub = VLu*BVLu ! Weight/volume of capillary blood in Lung compartment
VLut = VLu-VLub ! Weight/volume of tissue in Lung compartment
Vrestb = Vrest*BVrest ! Weight/volume of capillary blood in rest of body compartment
Vrestt = Vrest-Vrestb ! Weight/volume of tissue in rest of body compartment

! Permeability coefficient-surface area cross-product
PAL = PALC*QL
PABR = PABRC*QBR
PAK = PAKC*QK
PAS = PASC*QS
PALu = PALuC*QC
PArest = PArestC*Qrest

KLRESUP = (KLRESmax*T^KLRESn)/(KLRES50^KLRESn+T^KLRESn)*(1-
(ALRES/(ALREScap*VL)))
KSRESUP = (KSRESmax*T^KSRESn)/(KSRES50^KSRESn+T^KSRESn)*(1-
(ASRES/(ASREScap*VS)))
KKRESUP = (KKRESmax*T^KKRESn)/(KKRES50^KKRESn+T^KKRESn)*(1-
(AKRES/(AKREScap*VK)))
KLuRESUP = (KLuRESmax*T^KLuRESn)/(KLuRES50^KLuRESn+T^KLuRESn)*(1-
(ALuRES/(ALuREScap*VLu)))
KrestRESUP = (KrestRESmax*T^KrestRESn)/(KrestRES50^KrestRESn+T^KrestRESn)*(1-
(ArestRES/(ArestREScap*Vrest)))

! Dosing
DOSEiv = PDOSEiv*BW0 ! mg
IVR = DOSEiv/Timeiv ! mg/h
RIV = IVR*(1.-step(Timeiv))
AIV = Integ(RIV, 0.0)

! Elimination
Kbile = KbileC*BW**0.75 ! L/h
Kurine = KurineC*BW**0.75 ! L/h

! Blood compartment
! CA = Arterial blood concentration (mg/L or ug/ml)
RA = QC*CVLu - QC*CA
```

```
AA = Integ(RA, 0.0)
!CA = AA/VBlood
CA = AA/(VPlasma*0.2)
AUCCA = Integ(CA,0.0)
CA1000 = CA*1000 ! ng/g, ng/ml, ug/L
AUCCA1000 = Integ(CA1000,0.0)
! CV = Venous blood concentration (mg/L or ug/ml)
RV = QL*CVL + QBR*CVBR + QK*CVK + Qrest*CVrest + RIV - QC*CV
AV = Integ(RV, 0.0)
!CV = AV/VBlood
CV = AV/(VPlasma*0.8)
CV1000 = CV*1000
APlasma = AA+AV
APlasmaperc = 100*(APlasma/Doseiv)/(VPlasma*1000)
Abloodperc = 100*(APlasma/Doseiv)/(VBlood*1000)

!! Lung compartment
! Diffusion limited model
RLub = QC*(CV-CVLu) - PALu*CVLu + (PALu*CLut)/PLu + RLuRESrelease -
KLuRESup*ALub!
        ALub = Integ(RLub,0.0)
CVLu = ALub/VLub

RLut = PALu*CVLu - (PALu*CLut)/PLu
ALut = Integ(RLut,0.0)
CLut = ALut/VLut
ALutotal = ALub+ALut
CLu = ALutotal/VLu
CLu1000 = CLu*1000 ! ng/g, ng/ml, ug/L

RLuRES = KLuRESUP*ALub-KLuRESrelease*ALuRES
RLuRESUP = KLuRESUP*ALub!*1000
RLuRESrelease = KLuRESrelease*ALuRES
ALuRES = INTEG(RLuRES,0.0)
CLung = (ALutotal+ALuRES)/VLu
CLungtissue = (ALut+ALuRES)/VLut
Clungtissue1000 = 1000*(ALut+ALuRES)/VLut
Alungtissue1000 = 1000*(ALut+ALuRES)
CLung1000 = CLung*1000

ALungtissue = ALut+ALuRES
ALungtissueperc = 100*(ALungtissue/Doseiv)/(VLut*1000)

!! Brain compartment
! Diffusion limited model
RBRb = QBR*(CA-CVBR) - PABR*CVBR + (PABR*CBRt)/PBR
```

```
ABRb = Integ(RBRb,0.0)
CVBR = ABRb/VBRb

RBRt = PABR*CVBR - (PABR*CBRt)/PBR
ABRt = Integ(RBRt,0.0)
CBRt = ABRt/VBRt
ABRtotal = ABRb+ABRt
CBR = ABRtotal/VBR

!! Rest of body compartment
! Diffusion limited model
Rrestb = Qrest*(CA-CVrest) - PArest*CVrest + (PArest*Crestt)/Prest + RrestRESrelease -
KrestRESUP*Arestb!
Arestb = Integ(Rrestb,0.0)
CVrest = Arestb/Vrestb

Rrestt = PArest*CVrest - (PArest*Crestt)/Prest
Arestt = Integ(Rrestt,0.0)
Crestt = Arestt/Vrestt
Aresttotal = Arestb+Arestt
Crest = Aresttotal/Vrest
Crest1000 = Crest*1000 ! ng/g, ng/ml, ug/L

RrestRES = KrestRESUP*Arestb-KrestRESrelease*ArestRES
RrestRESUP = KrestRESUP*Arestb!*1000
RrestRESrelease = KrestRESrelease*ArestRES
ArestRES = INTEG(RrestRES,0.0)
Crestall = (Aresttotal+ArestRES)/Vrest
Cresttissue1000 = 1000*(Arestt+ArestRES)/Vrestt
Cresttissue = (Arestt+ArestRES)/Vrestt
Aresttissue1000 = 1000*(Arestt+ArestRES)
Crestall1000 = Crestall*1000

Aresttissue = Arestt+ArestRES
Aresttissueperc = 100*(Aresttissue/Doseiv)/(Vrestt*1000)

!! Kidney compartment
! Diffusion limited model
RKb = QK*(CA-CVK) - PAK*CVK + (PAK*CKt)/PK - Rurine + RKRESrelease -
KKRESUP*AKb!
AKb = Integ(RKb,0.0)
CVK = AKb/VKb

RKt = PAK*CVK - (PAK*CKt)/PK
AKt = Integ(RKt,0.0)
CKt = AKt/VKt
```

```
AKtotal = AKb+AKt
CK = AKtotal/VK
CK1000 = CK*1000 ! ng/g, ng/ml, ug/L

! Urinary excretion
Rurine = Kurine*CVK ! mg/h
Aurine = Integ(Rurine,0.0)
Aurine2 = Aurine*1000

!RKRES = KKRESUP*AA-KKRESrelease*AKRES
RKRES = KKRESUP*AKb-KKRESrelease*AKRES
RKRESUP = KKRESUP*AKb!*1000
RKRESrelease = KKRESrelease*AKRES
AKRES = INTEG(RKRES,0.0)
CKidney = (AKtotal+AKRES)/VK
CKidneytissue = (AKt+AKRES)/VKt
CKidneytissue1000 = 1000*(AKt+AKRES)/VKt
AKidneytissue1000 = 1000*(AKt+AKRES)
CKidney1000 = CKidney*1000

AKidneytissue = AKt+AKRES
AKidneytissueperc = 100*(AKidneytissue/Doseiv)/(VKt*1000)

!! Spleen compartment
! Diffusion limited model
RSb = QS*(CA-CVS) - PAS*CVS + (PAS*CSt)/PS + RSRESrelease - KSRESUP*ASb!
ASb = Integ(RSb,0.0)
CVS = ASb/VSb

RSt = PAS*CVS - (PAS*CSt)/PS
ASt = Integ(RSt,0.0)
CSt = ASt/VSt
AStotal = ASb+ASt
CS = AStotal/VS
CS1000 = CS*1000 ! ng/g, ng/ml, ug/L

!RSRES = KSRESUP*AA-KSRESrelease*ASRES
RSRES = KSRESUP*ASb-KSRESrelease*ASRES
RSRESUP = KSRESUP*ASb!*1000
RSRESrelease = KSRESrelease*ASRES
ASRES = INTEG(RSRES,0.0)
CSpleen = (AStotal+ASRES)/VS
Cspleentissue = (ASt+ASRES)/VSt
CSpleentissue1000 = 1000*(ASt+ASRES)/VSt
AUCCSpleentissue1000 = Integ(CSpleentissue1000,0.0)
ASpleentissue1000 = 1000*(ASt+ASRES)
```

```
CSpleen1000 = CSpleen*1000
ASpleentissue = ASt+ASRES
ASpleentissueperc = 100*(ASpleentissue/Doseiv)/(VSt*1000)

!! Liver compartment

! Diffusion limited model
RLb = QL*(CA-CVL) + QS*CVS - PAL*CVL + (PAL*CLt)/PL - Rbile + RLRESrelease -
KLRESUP*ALb!- KLRESUP*AA
ALb = Integ(RLb,0.0)
CVL = ALb/VLb

RLt = PAL*CVL - (PAL*CLt)/PL
ALt = Integ(RLt,0.0)
CLt = ALt/VLt
ALtotal = ALb+ALt
CL = ALtotal/VL
CL1000 = CL*1000 ! ng/g, ng/ml, ug/L

!RLRES = KLRESUP*AA-KLRESrelease*ALRES
RLRES = KLRESUP*ALb-KLRESrelease*ALRES
RLRESUP = KLRESUP*ALb!*1000
RLRESrelease = KLRESrelease*ALRES
ALRES = INTEG(RLRES,0.0)
CLiver = (ALtotal+ALRES)/VL
CLivertissue1000 = 1000*(ALt+ALRES)/VLt
AUCCLivertissue1000 = Integ(CLivertissue1000,0.0)
CLivertissue = (ALt+ALRES)/VLt
ALivertissue1000 = 1000*(ALt+ALRES)
CLiver1000 = CLiver*1000

ALivertissue = ALt+ALRES
ALivertissueperc = 100*(ALivertissue/Doseiv)/(VLt*1000)

! Biliary excretion
Rbile = Kbile*CVL ! mg/h
Abile = Integ(Rbile,0.0)

! Mass balance
Tmass                                                                        =
AA+AV+ALtotal+ABRtotal+AKtotal+ALutotal+Aresttotal+AStotal+Abile+Aurine+ALRES+A
SRES+ALuRES+AKRES+ArestRES!
Bal = AIV-Tmass

END ! DERIVATIVE
```

```
! Add discrete events here as needed
!  DISCRETE
!  END

! code that is executed once at each communication interval goes here

CONSTANT TSTOP = 770!4325!240.0
TERMT (T .GE. TSTOP, 'checked on communication interval: REACHED TSTOP')

END ! DYNAMIC


TERMINAL
! code that is executed once at the end of a simulation run goes here
END ! TERMINAL

END ! PROGRAM
```

Gold nanoparticle PBPK model code in Berkeley Madonna™ format

```
METHOD RK4

STARTTIME = 0
STOPTIME = 770 ;10
DT = 0.00125
DTOUT = 0.1

; Blood flow rate (Fraction of cardiac output, unitless)
QCC = 5.0; Cardiac output (L/h/kg) (Upton, 2008; Yuan et al., 2011)
QLC = 0.2725 ; Fraction of blood flow to liver (Average of Buur et al., 2005 and Upton, 2008)
QBRC = 0.03 ; Fraction of blood flow to brain (Upton, 2008)
QKC = 0.12 ; Fraction of blood flow to kidneys (Average of Buur et al., 2005 and Upton, 2008)
QSC = 0.0151 ; Fraction of blood flow to spleen (Davies and Morris, 1993) Table III! Fraction
of blood flow to spleen (average of 6 species)

; Tissue volumes (Fraction of body weight, unitless)
BW0 = 9.6 ; Body weight (kg) (Upton, 2008)
VLC = 0.0247 ; Liver (Average of Buur et al., 2005 and Upton, 2008)
VBRC = 0.004 ; Brain (Upton, 2008)
VKC = 0.004 ; Kidneys (Average of Buur et al., 2005 and Upton, 2008)
VSC = 0.002 ; Spleen (Upton, 2008)
VLuC = 0.01 ; Lungs (Upton, 2008)
VBloodC = 0.06 ; Blood (Average of Buur et al., 2005 and Upton, 2008)
VPlasmaC = 0.04 ; Buur et al. 2005 (Hematocrit is 0.33)

; Blood volume fraction in organs and tissues (percentage of organs/tissues, unitless)
BVL = 0.115 ; Liver (Buur et al., 2005)
BVBR = 0.0275 ; Brain (Brown et al., 1997; Table 30, average of 4 species)
BVK = 0.105 ; Kidneys (Buur et al., 2005)
BVS = 0.3 ; Spleen (Brown et al., 1997, Table 30, average of 3 species)
BVLu = 0.3867 ; Lungs (Brown et al., 1997, Table 30, average of 3 species)
BVrest = 0.026 ; Rest of body (Assume the same as the muscle in Buur et al. 2005)

; Tissue:plasma distribution coefficients (PCs), unitless; these values were from our published
mouse PBPK model for gold nanoparticles (Lin et al., in press)
PL = 0.08 ; Liver
PBR = 0.15 ; Brain
PK = 0.15 ; Kidneys
PS = 0.15 ; Spleen
PLu = 0.15 ; Lungs
Prest = 0.15 ; Rest of body

; Membrane-limited permeability coefficient constants, unitless; these values were from our
published mouse PBPK model for gold nanoparticles (Lin et al., in press)
```

```
PALC = 0.001 ; Liver
PABRC = 0.000001 ; Brain
PAKC = 0.001 ; Kidneys
PASC =0.001 ; Spleen
PALuC = 0.001 ; Lungs
PArestC = 0.000001 ; Rest of body

; Endocytic parameters; RES represent phagocytic cells; L, S, K, Lu, rest represent liver, spleen, kidneys, lungs, and rest of body, respectively.
; Compared to the original mouse model, uptake capacity parameters were added to simulate the dose-dependence;
; Endocytosis in the rest of body compartment was added because this compartment contains certain tissues (e.g., bone marrow) that can also uptake NPs.
KLRESrelease = 0.005 ; Release rate constant of phagocytic cells, (h-1)
KLRESmax = 1000 ; Maximum uptake rate constant of phagocytic cells, (h-1)
KLRES50 = 24 ; Time reaching half maximum uptake rate, (h)
KLRESn = 0.5 ; Hill coefficient, (unitless)
ALREScap = 32.5 ; Uptake capacity per tissue weight (ug/g tissue)

KSRESrelease = 0.02 ; Release rate constant of phagocytic cells, (h-1)
KSRESmax = 500 ; Maximum uptake rate constant of phagocytic cells, (h-1)
KSRES50 = 24 ; Time reaching half maximum uptake rate, (h)
KSRESn = 0.5 ; Hill coefficient, (unitless)
ASREScap = 25 ; Uptake capacity per tissue weight (ug/g tissue)

KKRESrelease = 0.04 ; Release rate constant of phagocytic cells, (h-1)
KKRESmax = 500 ; Maximum uptake rate constant of phagocytic cells, (h-1)
KKRES50 = 24 ; Time reaching half maximum uptake rate, (h)
KKRESn = 0.5 ; Hill coefficient, (unitless)
AKREScap = 55 ; Uptake capacity per tissue weight (ug/g tissue)

KLuRESrelease = 0.02 ; Release rate constant of phagocytic cells, (h-1)
KLuRESmax = 300 ; Maximum uptake rate constant of phagocytic cells, (h-1)
KLuRES50 = 24 ; Time reaching half maximum uptake rate, (h)
KLuRESn = 0.5 ; Hill coefficient, (unitless)
ALuREScap = 25 ; Uptake capacity per tissue weight (ug/g tissue)

KrestRESrelease = 0.0001 ; Release rate constant of phagocytic cells, (h-1)
KrestRESmax = 0.00025 ; Maximum uptake rate constant of phagocytic cells, (h-1)
KrestRES50 = 24 ; Time reaching half maximum uptake rate, (h)
KrestRESn = 0.5 ; Hill coefficient, (unitless)
ArestREScap = 0.05 ; Uptake capacity per tissue weight (ug/g tissue)

; Biliary excretion
KbileC = 0.0008 ; Biliary clearance (L/hr/kg^0.75)
; L/hr/kg changed to L/h/kg^0.75 for interspecies extrapolation
; Urine excretion
KurineC = 0.0008 ; Urine clearance (L/hr/kg^0.75)
; L/hr changed to L/h/kg^0.75 for interspecies extrapolation

; IV dosing
Timeiv = 0.005 ; IV infusion time (h), set, approximately 15-20 seconds, on average 18 sec
PDOSEiv = 2 ; mg/kg

; Scaled parameters
; Cardiac output and regional blood blow (L/h)
BW = 9.6+0.015*TIME
QC = QCC*BW ; Cardiac output
QL = QC*QLC ; Blood flow to liver
QBR = QC*QBRC ; Blood flow to brain
QK = QC*QKC ; Blood flow to kidneys
QS = QC*QSC ; Blood flow to spleen
Qrest = QC-QL-QBR-QK-QS ; Blood flow to rest of body
Qbal = QC-QL-QBR-QK-QS-Qrest ; Blood flow balance equation

; Tissue volumes (L)
VL = BW*VLC ; Liver
VBR = BW*VBRC ; Brain
VK = BW*VKC ; Kidneys
VS = BW*VSC ; Spleen
VLu = BW*VLuC ; Lungs
VBlood = BW*VBloodC
VPlasma = BW*VPlasmaC
Vrest = BW-VL-VBR-VK-VS-VLu-VPlasma.......................................................................
Vbal = BW-VL-VBR-VK-VS-VLu-VPlasma-Vrest
VLb = VL*BVL ; Weight/volume of capillary blood in liver compartment
VLt = VL-VLb ; Weight/volume of tissue in liver compartment
VBRb = VBR*BVBR ; Weight/volume of capillary blood in brain compartment
VBRt = VBR-VBRb ; Weight/volume of tissue in brain compartment
VKb = VK*BVK ; Weight/volume of capillary blood in kidney compartment
VKt = VK-VKb ; Weight/volume of tissue in kidney compartment
VSb = VS*BVS ; Weight/volume of capillary blood in spleen compartment
VSt = VS-VSb ; Weight/volume of tissue in spleen compartment
VLub = VLu*BVLu ; Weight/volume of capillary blood in lung compartment
VLut = VLu-VLub ; Weight/volume of tissue in lung compartment
Vrestb = Vrest*BVrest ; Weight/volume of capillary blood in rest of body compartment
Vrestt = Vrest-Vrestb ; Weight/volume of tissue in rest of body compartment

; Permeability coefficient-surface area cross-product (L/h)
PAL = PALC*QL ; Liver
PABR = PABRC*QBR ; Brain
```

```
PAK = PAKC*QK ; Kidneys
PAS = PASC*QS ; Spleen
PALu = PALuC*QC ; Lungs
PArest = PArestC*Qrest ; Rest of body

; Endocytosis rate (h-1)
KLRESUP = (KLRESmax*TIME^KLRESn)/(KLRES50^KLRESn+TIME^KLRESn)*(1-
(ALRES/(ALREScap*VL))) ; Liver
KSRESUP = (KSRESmax*TIME^KSRESn)/(KSRES50^KSRESn+TIME^KSRESn)*(1-
(ASRES/(ASREScap*VS))) ; Spleen
KKRESUP = (KKRESmax*TIME^KKRESn)/(KKRES50^KKRESn+TIME^KKRESn)*(1-
(AKRES/(AKREScap*VK))) ; Kidneys
KLuRESUP =
(KLuRESmax*TIME^KLuRESn)/(KLuRES50^KLuRESn+TIME^KLuRESn)*(1-
(ALuRES/(ALuREScap*VLu))) ; Lungs
KrestRESUP =
(KrestRESmax*TIME^KrestRESn)/(KrestRES50^KrestRESn+TIME^KrestRESn)*(1-
(ArestRES/(ArestREScap*Vrest))) ; Rest of body

; Dosing
DOSEiv = PDOSEiv*BW0 ; mg
IVR = DOSEiv/Timeiv ; mg/h
RIV = IVR*(1.-step(1,Timeiv))
d/dt(AIV) = RIV
init AIV = 0

; Elimination
Kbile = KbileC*BW**0.75 ; L/h
Kurine = KurineC*BW**0.75 ; L/h

; Blood compartment
; CA = Arterial blood concentration (mg/L or ug/ml)
RA = QC*CVLu - QC*CA
d/dt(AA) = RA
init AA = 0
;CA = AA/(VBlood*0.2)
CA = AA/(VPlasma*0.2)
d/dt(AUCCA) = CA
init AUCCA = 0
CA1000 = CA*1000 ; ng/g, ng/ml, ug/L
d/dt(AUCCA1000) = CA1000
init AUCCA1000 = 0

; CV = Venous blood concentration (mg/L or ug/ml)
RV = QL*CVL + QBR*CVBR + QK*CVK + Qrest*CVrest + RIV - QC*CV
d/dt(AV) = RV
```

```
init AV = 0
;CV = AV/(VBlood*0.8)
CV = AV/(VPlasma*0.8)
CV1000 = CV*1000
APlasma = AA+AV
APlasmaperc = 100*(APlasma/Doseiv)/(VPlasma*1000)
Abloodperc = 100*(APlasma/Doseiv)/(VBlood*1000)

;; Lung compartment
; Membrane-limited model
RLub = QC*(CV-CVLu) - PALu*CVLu + (PALu*CLut)/PLu + RLuRESrelease - KLuRESup*ALub
d/dt(ALub) = RLub
init ALub = 0
CVLu = ALub/VLub

RLut = PALu*CVLu - (PALu*CLut)/PLu
d/dt(ALut) = RLut
init ALut = 0
CLut = ALut/VLut
ALutotal = ALub+ALut
CLu = ALutotal/VLu
CLu1000 = CLu*1000 ; ng/g, ng/ml, ug/L

RLuRES = KLuRESUP*ALub-KLuRESrelease*ALuRES
RLuRESUP = KLuRESUP*ALub ;*1000
RLuRESrelease = KLuRESrelease*ALuRES
d/dt(ALuRES) = RLuRES
init ALuRES = 0
CLung = (ALutotal+ALuRES)/VLu
CLungtissue = (ALut+ALuRES)/VLut
Clungtissue1000 = 1000*(ALut+ALuRES)/VLut
Alungtissue1000 = 1000*(ALut+ALuRES)
CLung1000 = CLung*1000

ALungtissue = ALut+ALuRES
ALungtissueperc = 100*(ALungtissue/Doseiv)/(VLut*1000)

;; Brain compartment
; Membrane-limited model
RBRb = QBR*(CA-CVBR) - PABR*CVBR + (PABR*CBRt)/PBR
d/dt(ABRb) = RBRb
init ABRb = 0
CVBR = ABRb/VBRb

RBRt = PABR*CVBR - (PABR*CBRt)/PBR
```

```
d/dt(ABRt) = RBRt
init ABRt = 0
CBRt = ABRt/VBRt
ABRtotal = ABRb+ABRt
CBR = ABRtotal/VBR

;; Rest of body compartment
; Membrane-limited model, endocytosis is included in this compartment because it contains
certain tissues (e.g., bone marrow) that can uptake non-PEG NPs.
Rrestb = Qrest*(CA-CVrest) - PArest*CVrest + (PArest*Crestt)/Prest + RrestRESrelease -
KrestRESUP*Arestb
d/dt(Arestb) = Rrestb
init Arestb = 0
CVrest = Arestb/Vrestb

Rrestt = PArest*CVrest - (PArest*Crestt)/Prest
d/dt(Arestt) = Rrestt
init Arestt = 0
Crestt = Arestt/Vrestt
Aresttotal = Arestb+Arestt
Crest = Aresttotal/Vrest
Crest1000 = Crest*1000 ; ng/g, ng/ml, ug/L

RrestRES = KrestRESUP*Arestb-KrestRESrelease*ArestRES
RrestRESUP = KrestRESUP*Arestb ;*1000
RrestRESrelease = KrestRESrelease*ArestRES
d/dt(ArestRES) = RrestRES
init ArestRES = 0
Crestall = (Aresttotal+ArestRES)/Vrest
Cresttissue1000 = 1000*(Arestt+ArestRES)/Vrestt
Cresttissue = (Arestt+ArestRES)/Vrestt
Aresttissue1000 = 1000*(Arestt+ArestRES)
Crestall1000 = Crestall*1000

Aresttissue = Arestt+ArestRES
Aresttissueperc = 100*(Aresttissue/Doseiv)/(Vrestt*1000)

;; Kidney compartment
; Membrane-limited model
RKb = QK*(CA-CVK) - PAK*CVK + (PAK*CKt)/PK - Rurine + RKRESrelease -
KKRESUP*AKb
d/dt(AKb) = RKb
init AKb = 0
CVK = AKb/VKb

RKt = PAK*CVK - (PAK*CKt)/PK
```

```
d/dt(AKt) = RKt
init AKt = 0
CKt = AKt/VKt
AKtotal = AKb+AKt
CK = AKtotal/VK
CK1000 = CK*1000 ; ng/g, ng/ml, ug/L

; Urinary excretion
Rurine = Kurine*CVK ;mg/h
d/dt(Aurine) = Rurine
init Aurine = 0
Aurine2 = Aurine*1000

;RKRES = KKRESUP*AA-KKRESrelease*AKRES
RKRES = KKRESUP*AKb-KKRESrelease*AKRES
RKRESUP = KKRESUP*AKb;*1000
RKRESrelease = KKRESrelease*AKRES
d/dt(AKRES) = RKRES
init AKRES = 0
CKidney = (AKtotal+AKRES)/VK
CKidneytissue = (AKt+AKRES)/VKt
CKidneytissue1000 = 1000*(AKt+AKRES)/VKt
AKidneytissue1000 = 1000*(AKt+AKRES)
CKidney1000 = CKidney*1000

AKidneytissue = AKt+AKRES
AKidneytissueperc = 100*(AKidneytissue/Doseiv)/(VKt*1000)

;; Spleen compartment
; Membrane-limited model
RSb = QS*(CA-CVS) - PAS*CVS + (PAS*CSt)/PS + RSRESrelease - KSRESUP*ASb
d/dt(ASb) = RSb
init ASb = 0
CVS = ASb/VSb

RSt = PAS*CVS - (PAS*CSt)/PS
d/dt(ASt) = RSt
init ASt = 0
CSt = ASt/VSt
AStotal = ASb+ASt
CS = AStotal/VS
CS1000 = CS*1000 ; ng/g, ng/ml, ug/L

;RSRES = KSRESUP*AA-KSRESrelease*ASRES
RSRES = KSRESUP*ASb-KSRESrelease*ASRES
RSRESUP = KSRESUP*ASb;*1000
```

```
RSRESrelease = KSRESrelease*ASRES
d/dt(ASRES) = RSRES
init ASRES = 0
CSpleen = (AStotal+ASRES)/VS
Cspleentissue = (ASt+ASRES)/VSt
CSpleentissue1000 = 1000*(ASt+ASRES)/VSt
d/dt(AUCCSpleentissue1000) = CSpleentissue1000
init AUCCSpleentissue1000 = 0
ASpleentissue1000 = 1000*(ASt+ASRES)
CSpleen1000 = CSpleen*1000
ASpleentissue = ASt+ASRES
ASpleentissueperc = 100*(ASpleentissue/Doseiv)/(VSt*1000)

;; Liver compartment
; Membrane-limited model
RLb = QL*(CA-CVL) + QS*CVS - PAL*CVL + (PAL*CLt)/PL - Rbile + RLRESrelease -
KLRESUP*ALb
d/dt(ALb) = RLb
init ALb = 0
CVL = ALb/VLb

RLt = PAL*CVL - (PAL*CLt)/PL
d/dt(ALt) = RLt
init ALt = 0
CLt = ALt/VLt
ALtotal = ALb+ALt
CL = ALtotal/VL
CL1000 = CL*1000 ; ng/g, ng/ml, ug/L

;RLRES = KLRESUP*AA-KLRESrelease*ALRES
RLRES = KLRESUP*ALb-KLRESrelease*ALRES
RLRESUP = KLRESUP*ALb;*1000
RLRESrelease = KLRESrelease*ALRES
d/dt(ALRES) = RLRES
init ALRES = 0
CLiver = (ALtotal+ALRES)/VL
CLivertissue1000 = 1000*(ALt+ALRES)/VLt
d/dt(AUCCLivertissue1000) = CLivertissue1000
init AUCCLivertissue1000 = 0
CLivertissue = (ALt+ALRES)/VLt
ALivertissue1000 = 1000*(ALt+ALRES)
CLiver1000 = CLiver*1000

ALivertissue = ALt+ALRES
ALivertissueperc = 100*(ALivertissue/Doseiv)/(VLt*1000)
```

```
; Biliary excretion
Rbile = Kbile*CVL ; mg/h
d/dt(Abile) = Rbile
init Abile = 0

; Mass balance
Tmass =
AA+AV+ALtotal+ABRtotal+AKtotal+ALutotal+Aresttotal+AStotal+Abile+Aurine+ALRES+A
SRES+ALuRES+AKRES+ArestRES
Bal = AIV-Tmass
```

Gold nanoparticle PBPK model code in MATLAB® format

```
function
dY=GoldPBPK(T,Y,PDOSEORAL,PDOSEIV,PDOSEIM,TINTERVAL,KA,TSTOP,KIM,FRAC)
% Gold Nanoparticle PBPK model, initiated by Zhoumeng Lin on June 23, 2014;
%% Constants
%%INITIAL
% code that is executed once at the beginning of a simulation run goes here
% Blood flow rate (Fraction of cardiac output)
QCC = 5.0; %Cardiac output(L/h/kg) (Upton, 2008; Yuan et al., 2011)
QLC = 0.2725; % Fraction of blood flow to liver (Average of Buur et al., 2005
and Upton, 2008)
QBRC = 0.03; % Fraction of blood flow to brain (Upton,2008)
QKC = 0.12; % Fraction of blood flow to kidneys (Average of Buur et al., 2005
and Upton, 2008)
QSC = 0.0151; %Davies and Morris (1993)Table III% Fraction of blood flow to
spleen (average of 6 species)

% Tissue volumes (Fraction of body weight)
BW0 = 9.6; % Body weight (kg) (Upton, 2008)
VLC = 0.0247; % Liver (Average of Buur et al., 2005 and Upton, 2008)
VBRC = 0.004; % Brain (Upton, 2008)
VKC = 0.004; % Kidneys (Average of Buur et al., 2005 and Upton, 2008)
VSC = 0.002; % Spleen (Upton, 2008)
VLuC = 0.01; % Lungs (Upton, 2008)
VBloodC = 0.06; % Blood (Average of Buur et al., 2005 and Upton, 2008)
VPlasmaC = 0.04; %0.04% Buur et al. 2005 (Hematocrit is 0.33)

% Blood volume fraction in organs and tissues (percentage of tissues) All rat
values are from Table 30 in Brown et al. (1997).
BVL = 0.115; % Liver (Buur et al. 2005)
BVBR = 0.0275; % Brain (Brown et al., 1997; Table 30, average of 4 species)
BVK = 0.105; % Kidneys (Buur et al., 2005)
BVS = 0.3; % Spleen (Brown et al., 1997; Table 30, average of 3 species)
BVLu = 0.3867; % Lungs (Brown et al., 1997; Table 30, average of 3 species)
BVrest = 0.026; % Rest of body (Assume the same as the muscle in Buur et al.
2005)
```

```
% Partition coefficients (PC), unitless
PL = 0.08; % Liver:blood PC (calculated using AUC method, 7-day AUC data from
Cho et al., 2010, data for 13-nm AuNPs)
PBR = 0.15; %Brain:blood PC (generic value, Li et al., 2014)
PK = 0.15; %Kidneys:blood PC (calculated using AUC method, 7-day AUC data
from Cho et al., 2010, data for 13-nm AuNPs)
PS = 0.15; %Spleen:blood PC (calculated using AUC method, 7-day AUC data from
Cho et al., 2010, data for 13-nm AuNPs)
PLu = 0.15; %Lungs:blood PC (calculated using AUC method, 7-day AUC data from
Cho et al., 2010, data for 13-nm AuNPs)
Prest = 0.15; % Rest of body:blood PC (generic value, Li et al., 2014)

% Diffusion limitation coefficient constants, unitless (Based on Li et al.,
2014)
PALC = 0.001; % Permeability coefficient between blood and liver, generic
value from Li et al., 2014
PABRC = 0.000001; % Permeability coefficient between blood and brain, set a
very low values
PAKC = 0.001; %0.001% Permeability coefficient between blood and kidneys,
generic value from Li et al., 2014
PASC =0.001; % Permeability coefficient between blood and spleen, generic
value from Li et al., 2014
PALuC = 0.001; % Permeability coefficient between blood and lungs, generic
value from Li et al., 2014
PArestC = 0.000001; % Permeability coefficient between blood and rest of
body, from Li et al., 2014

KLRESrelease = 0.005;
KLRESmax = 1000;
KLRES50 = 24;
KLRESn = 0.5;
ALREScap = 32.5;

KSRESrelease = 0.02;
KSRESmax = 500;
KSRES50 = 24;
KSRESn = 0.5;
ASREScap = 25;

KKRESrelease = 0.04;
KKRESmax = 500;
KKRES50 = 24;
KKRESn = 0.5;
AKREScap = 55;

KLuRESrelease = 0.02;
KLuRESmax = 300;
KLuRES50 = 24;
KLuRESn = 0.5;
ALuREScap = 25;

KrestRESrelease = 0.0001;
KrestRESmax = 0.00025;
KrestRES50 = 24;
```

```
KrestRESn = 0.5;
ArestREScap =0.05;

% Biliary excretion
KbileC = 0.0008; %0.00008% Biliary clearance (L/hr/kg^0.75)
%Kbile = 0.0012% Biliary clearance (L/hr/kg)
% changed to L/h/kg^0.75 for interspecies extrapolation 0.0012/0.02^0.75 =
% Urine excretion
KurineC = 0.0008; %0.0008% % Urine clearance (L/hr/kg^0.75)
%Kurine = 0.00012% Urine clearance (L/hr)
% changed to L/h/kg^0.75 for interspecies extrapolation 0.00012/0.02^0.75 =



% IV dosing
Timeiv = 0.05; %0.005 % IV infusion time (h), set, approximately 15-20
seconds, on average 18 sec
PDOSEiv = 2; %1%2 % mg/kg

% END % INITIAL

% DYNAMIC
%
% ALGORITHM IALG = 2
% NSTEPS    NSTP = 10
% MAXTERVAL MAXT = 1.0e9
% MINTERVAL MINT = 1.0e-9
% CINTERVAL CINT = 0.1

%% Variables
 AIV=Y(1);
 AA=Y(2);
 AUCCA=Y(3);
 AUCCA1000=Y(4);
 AV=Y(5);
 ALub=Y(6);
 ALut=Y(7);
 ALuRES=Y(8);
 ABRb=Y(9);
 ABRt=Y(10);
 Arestb=Y(11);
 Arestt=Y(12);
 ArestRES=Y(13);
 AKb=Y(14);
 AKt=Y(15);
 Aurine=Y(16);
 AKRES=Y(17);
 ASb=Y(18);
 ASt=Y(19);
 ASRES=Y(20);
 AUCCSpleentissue1000=Y(21);
 ALb=Y(22);
 ALt=Y(23);
 ALRES=Y(24);
 AUCCLivertissue1000=Y(25);
 Abile=Y(26);
 CV=Y(27);
```

```
%% code for calulating the derivative goes here
% Scaled papameters
% Cardiac output and regional blood blow (L/h)
BW = 9.6+0.015*T;
QC = QCC*BW; %^0.75 % Cardiac output
QL = QC*QLC ; % Blood flow to liver
QBR = QC*QBRC ; % Blood flow to brain
QK = QC*QKC ; % Blood flow to kidney
QS = QC*QSC ; % Blood flow to spleen
Qrest = QC-QL-QBR-QK-QS; % Blood flow to rest of body
Qbal = QC-QL-QBR-QK-QS-Qrest ; %Blood flow balance equation

% Tissue volumes (L)
VL = BW*VLC ; % Liver
VBR = BW*VBRC ; % Brain
VK = BW*VKC ; % Kidney
VS = BW*VSC ; % Spleen
VLu = BW*VLuC ; % Lungs
VBlood = BW*VBloodC;
VPlasma = BW*VPlasmaC;
Vrest = BW-VL-VBR-VK-VS-VLu-VPlasma ;
Vbal = BW-VL-VBR-VK-VS-VLu-VPlasma-Vrest  ;
VLb = VL*BVL ; % Weight/volume of capillary blood in liver compartment
VLt = VL-VLb ; % Weight/volume of tissue in liver compartment
VBRb = VBR*BVBR ; % Weight/volume of capillary blood in brain compartment
VBRt = VBR-VBRb ; % Weight/volume of tissue in brain compartment
VKb = VK*BVK ; % Weight/volume of capillary blood in kidney compartment
VKt = VK-VKb ; % Weight/volume of tissue in kidney compartment
VSb = VS*BVS ; % Weight/volume of capillary blood in spleen compartment
VSt = VS-VSb ; % Weight/volume of tissue in spleen compartment
VLub = VLu*BVLu ; % Weight/volume of capillary blood in Lung compartment
VLut = VLu-VLub ; % Weight/volume of tissue in Lung compartment
Vrestb = Vrest*BVrest ; % Weight/volume of capillary blood in rest of body
compartment
Vrestt = Vrest-Vrestb ; % Weight/volume of tissue in rest of body compartment

% Permeability coefficient-surface area cross-product
PAL = PALC*QL;
PABR = PABRC*QBR;
PAK = PAKC*QK;
PAS = PASC*QS;
PALu = PALuC*QC;
PArest = PArestC*Qrest;


KLRESUP = (KLRESmax*T^KLRESn)/(KLRES50^KLRESn+T^KLRESn)*(1-
(ALRES/(ALREScap*VL)));
KSRESUP = (KSRESmax*T^KSRESn)/(KSRES50^KSRESn+T^KSRESn)*(1-
(ASRES/(ASREScap*VS)));
KKRESUP = (KKRESmax*T^KKRESn)/(KKRES50^KKRESn+T^KKRESn)*(1-
(AKRES/(AKREScap*VK)));
KLuRESUP = (KLuRESmax*T^KLuRESn)/(KLuRES50^KLuRESn+T^KLuRESn)*(1-
(ALuRES/(ALuREScap*VLu)));
KrestRESUP = (KrestRESmax*T^KrestRESn)/(KrestRES50^KrestRESn+T^KrestRESn)*(1-
(ArestRES/(ArestREScap*Vrest)));
```

```
%% DERIVATIVE
% Dosing
DOSEiv = PDOSEiv*BW0 ; % mg
IVR = DOSEiv/Timeiv ; % mg/h
RIV = IVR*(1.-heaviside(T-Timeiv));
d(1) = (RIV);

% Elimination
Kbile = KbileC*BW^0.75 ; % L/h
Kurine = KurineC*BW^0.75 ; % L/h

% Blood compartment
%CA = AA/VBlood
CA = AA/(VPlasma*0.2);
dY(3) = (CA);
CA1000 = CA*1000 ; % ng/g, ng/ml, ug/L
dY(4) = (CA1000);
% CA = Arterial blood concentration (mg/L or ug/ml)
CVLu = ALub/VLub;
RA = QC*CVLu - QC*CA ;
dY(2) = (RA);
%CV = AV/VBlood
CV = AV/(VPlasma*0.8);
dY(27)=(CV);
CV1000 = CV*1000;
APlasma = AA+AV;
APlasmaperc = 100*(APlasma/DOSEiv)/(VPlasma*1000);
Abloodperc = 100*(APlasma/DOSEiv)/(VBlood*1000);
% CV = Venous blood concentration (mg/L or ug/ml)

CVL = ALb/VLb;
CVBR = ABRb/VBRb;
CVK = AKb/VKb;
CVrest = Arestb/Vrestb;
RV = QL*CVL + QBR*CVBR + QK*CVK + Qrest*CVrest + RIV - QC*CV ;
dY(5) = (RV);

%% Lung compartment
% Diffusion limited model
CLut = ALut/VLut;
RLuRESrelease = KLuRESrelease*ALuRES;
RLub = QC*(CV-CVLu) - PALu*CVLu + (PALu*CLut)/PLu + RLuRESrelease -
KLuRESUP*ALub; %
        dY(6) = (RLub);

RLut = PALu*CVLu - (PALu*CLut)/PLu;
dY(7) = (RLut);
ALutotal = ALub+ALut;
CLu = ALutotal/VLu;
CLu1000 = CLu*1000 ; % ng/g, ng/ml, ug/L

RLuRES = KLuRESUP*ALub-KLuRESrelease*ALuRES;
RLuRESUP = KLuRESUP*ALub; %*1000
dY(8) = (RLuRES);
```

```
CLung = (ALutotal+ALuRES)/VLu;
CLungtissue = (ALut+ALuRES)/VLut;
Clungtissue1000 = 1000*(ALut+ALuRES)/VLut;
Alungtissue1000 = 1000*(ALut+ALuRES);
CLung1000 = CLung*1000;

ALungtissue = ALut+ALuRES;
ALungtissueperc = 100*(ALungtissue/DOSEiv)/(VLut*1000);

%% Brain compartment
% Diffusion limited model
CBRt = ABRt/VBRt;
RBRb = QBR*(CA-CVBR) - PABR*CVBR + (PABR*CBRt)/PBR;
dY(9) = (RBRb);

RBRt = PABR*CVBR - (PABR*CBRt)/PBR;
dY(10) = (RBRt);
ABRtotal = ABRb+ABRt;
CBR = ABRtotal/VBR;

%% Rest of body compartment
% Diffusion limited model
Crestt = Arestt/Vrestt;
RrestRESrelease = KrestRESrelease*ArestRES;
Rrestb = Qrest*(CA-CVrest) - PArest*CVrest + (PArest*Crestt)/Prest +
RrestRESrelease - KrestRESUP*Arestb; %
dY(11) = (Rrestb);

Rrestt = PArest*CVrest - (PArest*Crestt)/Prest;
dY(12) = (Rrestt);
Aresttotal = Arestb+Arestt;
Crest = Aresttotal/Vrest;
Crest1000 = Crest*1000 ; % ng/g, ng/ml, ug/L

RrestRES = KrestRESUP*Arestb-KrestRESrelease*ArestRES;
RrestRESUP = KrestRESUP*Arestb; %*1000
dY(13) = (RrestRES);
Crestall = (Aresttotal+ArestRES)/Vrest;
Cresttissue1000 = 1000*(Arestt+ArestRES)/Vrestt;
Cresttissue = (Arestt+ArestRES)/Vrestt;
Aresttissue1000 = 1000*(Arestt+ArestRES);
Crestall1000 = Crestall*1000;

Aresttissue = Arestt+ArestRES;
Aresttissueperc = 100*(Aresttissue/DOSEiv)/(Vrestt*1000);

%% Kidney compartment

AKtotal = AKb+AKt;

%RKRES = KKRESUP*AA-KKRESrelease*AKRES
RKRES = KKRESUP*AKb-KKRESrelease*AKRES;
RKRESUP = KKRESUP*AKb; %*1000
RKRESrelease = KKRESrelease*AKRES;
dY(17) = (RKRES);
```

```
CKidney = (AKtotal+AKRES)/VK;
CKidneytissue = (AKt+AKRES)/VKt;
CKidneytissue1000 = 1000*(AKt+AKRES)/VKt;
AKidneytissue1000 = 1000*(AKt+AKRES);
CKidney1000 = CKidney*1000;

% Urinary excretion
Rurine = Kurine*CVK ; % mg/h
dY(16) = (Rurine);
Aurine2 = Aurine*1000;

% Diffusion limited model
CKt = AKt/VKt;
RKb = QK*(CA-CVK) - PAK*CVK + (PAK*CKt)/PK - Rurine + RKRESrelease -
KKRESUP*AKb; %
dY(14) = (RKb);

RKt = PAK*CVK - (PAK*CKt)/PK;
dY(15) = (RKt);
CK = AKtotal/VK;
CK1000 = CK*1000 ; % ng/g, ng/ml, ug/L

AKidneytissue = AKt+AKRES;
AKidneytissueperc = 100*(AKidneytissue/DOSEiv)/(VKt*1000);

%% Spleen compartment
% Diffusion limited model
CSt = ASt/VSt;
CVS = ASb/VSb;
RSRESrelease = KSRESrelease*ASRES;
RSb = QS*(CA-CVS) - PAS*CVS + (PAS*CSt)/PS + RSRESrelease - KSRESUP*ASb; %
dY(18) = (RSb);

RSt = PAS*CVS - (PAS*CSt)/PS;
dY(19) = (RSt);
AStotal = ASb+ASt;
CS = AStotal/VS;
CS1000 = CS*1000 ; % ng/g, ng/ml, ug/L

%RSRES = KSRESUP*AA-KSRESrelease*ASRES
RSRES = KSRESUP*ASb-KSRESrelease*ASRES;
RSRESUP = KSRESUP*ASb; %*1000
dY(20) = (RSRES);
CSpleen = (AStotal+ASRES)/VS;
Cspleentissue = (ASt+ASRES)/VSt;
CSpleentissue1000 = 1000*(ASt+ASRES)/VSt;
dY(21) = (CSpleentissue1000);
ASpleentissue1000 = 1000*(ASt+ASRES);
CSpleen1000 = CSpleen*1000;
ASpleentissue = ASt+ASRES;
ASpleentissueperc = 100*(ASpleentissue/DOSEiv)/(VSt*1000);

%% Liver compartment

% Biliary excretion
```

```
Rbile = Kbile*CVL ; % mg/h
dY(26) = (Rbile);

ALtotal = ALb+ALt;

%RLRES = KLRESUP*AA-KLRESrelease*ALRES
RLRES = KLRESUP*ALb-KLRESrelease*ALRES;
RLRESUP = KLRESUP*ALb; %*1000
RLRESrelease = KLRESrelease*ALRES;
dY(24) = (RLRES);
CLiver = (ALtotal+ALRES)/VL;
CLivertissue1000 = 1000*(ALt+ALRES)/VLt;
dY(25) = (CLivertissue1000);
CLivertissue = (ALt+ALRES)/VLt;
ALivertissue1000 = 1000*(ALt+ALRES);
CLiver1000 = CLiver*1000;

ALivertissue = ALt+ALRES;
ALivertissueperc = 100*(ALivertissue/DOSEiv)/(VLt*1000);

% Diffusion limited model
CLt = ALt/VLt;
RLb = QL*(CA-CVL) + QS*CVS - PAL*CVL + (PAL*CLt)/PL - Rbile + RLRESrelease -
KLRESUP*ALb; %- KLRESUP*AA
dY(22) = (RLb);

RLt = PAL*CVL - (PAL*CLt)/PL;
dY(23) = (RLt);
CL = ALtotal/VL;
CL1000 = CL*1000 ; % ng/g, ng/ml, ug/L

% Mass balance
Tmass =
AA+AV+ALtotal+ABRtotal+AKtotal+ALutotal+Aresttotal+AStotal+Abile+Aurine+ALRES
+ASRES+ALuRES+AKRES+ArestRES; %
Bal = AIV-Tmass;


dY = dY(:);
 end
```

Gold nanoparticle PBPK model code in R language format

```
## Gold Nanoparticle PBPK model, initiated by Zhoumeng Lin on June 23, 2014; ##
# Gold Nanoparticle model code in R® format
# Physiological parameters
# Blood flow rate (Fraction of cardiac output)
QCC = 5.0     # Cardiac output(L/h/kg) (Upton, 2008; Yuan et al., 2011)
QLC = 0.2725  # Fraction of blood flow to liver (Average of Buur et al., 2005 and Upton, 2008)
QBRC = 0.03   # Fraction of blood flow to brain (Upton,2008)
QKC = 0.12     # Fraction of blood flow to kidneys (Average of Buur et al., 2005 and Upton,
2008)
QSC = 0.0151  # Davies and Morris (1993)Table III% Fraction of blood flow to spleen (average
of 6 species)

# Tissue volumes (Fraction of body weight)
BW0 = 9.6       # Body weight (kg) (Upton, 2008)
VLC = 0.0247    # Liver (Average of Buur et al., 2005 and Upton, 2008)
VBRC = 0.004    # Brain (Upton, 2008)
VKC = 0.004     # Kidneys (Average of Buur et al., 2005 and Upton, 2008)
VSC = 0.002     # Spleen (Upton, 2008)
VLuC = 0.01     # Lungs (Upton, 2008)
VBloodC = 0.06  # Blood (Average of Buur et al., 2005 and Upton, 2008)
VPlasmaC = 0.04 # 0.04% Buur et al. 2005 (Hematocrit is 0.33)

# Blood volume fraction in organs and tissues (percentage of tissues) All rat values are from
Table 30 in Brown et al. (1997).
BVL = 0.115        # Liver (Buur et al. 2005)
BVBR = 0.0275      #  Brain (Brown et al., 1997; Table 30, average of 4 species)
BVK = 0.105        # Kidneys (Buur et al., 2005)
BVS = 0.3          # Spleen (Brown et al., 1997; Table 30, average of 3 species)
BVLu = 0.3867      # Lungs (Brown et al., 1997; Table 30, average of 3 species)
BVrest = 0.026     # Rest of body (Assume the same as the muscle in Buur et al. 2005)

# Partition coefficients (PC), unitless
PL = 0.08       # Liver:blood PC (calculated using AUC method, 7-day AUC data from Cho et al.,
2010, data for 13-nm AuNPs)
PBR = 0.15      # Brain:blood PC (generic value, Li et al., 2014)
PK = 0.15       # Kidneys:blood PC (calculated using AUC method, 7-day AUC data from Cho et
al., 2010, data for 13-nm AuNPs)
PS = 0.15        # Spleen:blood PC (calculated using AUC method, 7-day AUC data from Cho et
al., 2010, data for 13-nm AuNPs)
PLu = 0.15       # Lungs:blood PC (calculated using AUC method, 7-day AUC data from Cho et
al., 2010, data for 13-nm AuNPs)
Prest = 0.15    # Rest of body:blood PC (generic value, Li et al., 2014)

# Diffusion limitation coefficient constants, unitless (Based on Li et al., 2014)
```

```
PALC = 0.001        # Permeability coefficient between blood and liver, generic value from Li et al., 2014
PABRC = 0.000001    # Permeability coefficient between blood and brain, set a very low values
PAKC = 0.001        # 0.001% Permeability coefficient between blood and kidneys, generic value from Li et al., 2014
PASC =0.001         # Permeability coefficient between blood and spleen, generic value from Li et al., 2014
PALuC = 0.001       # Permeability coefficient between blood and lungs, generic value from Li et al., 2014
PArestC = 0.000001  # Permeability coefficient between blood and rest of body, from Li et al., 2014

KLRESrelease = 0.005   # Release rate constant of phagocytic cells, (h-1)
KLRESmax = 1000        # Maximum uptake rate constant of phagocytic cells, (h-1)
KLRES50 = 24           # Time reaching half maximum uptake rate, (h)
KLRESn = 0.5           # Hill coefficient, (unitless)
ALREScap = 32.5        # Uptake capacity per tissue weight (ug/g tissue)

KSRESrelease = 0.02    # Release rate constant of phagocytic cells, (h-1)
KSRESmax = 500         # Maximum uptake rate constant of phagocytic cells, (h-1)
KSRES50 = 24           # Time reaching half maximum uptake rate, (h)
KSRESn = 0.5           # Hill coefficient, (unitless)
ASREScap = 25          # Uptake capacity per tissue weight (ug/g tissue)

KKRESrelease = 0.04    # Release rate constant of phagocytic cells, (h-1)
KKRESmax = 500         # Maximum uptake rate constant of phagocytic cells, (h-1)
KKRES50 = 24           # Time reaching half maximum uptake rate, (h)
KKRESn = 0.5           # Hill coefficient, (unitless)
AKREScap = 55          # Uptake capacity per tissue weight (ug/g tissue)

KLuRESrelease = 0.02   # Release rate constant of phagocytic cells, (h-1)
KLuRESmax = 300        # Maximum uptake rate constant of phagocytic cells, (h-1)
KLuRES50 = 24          # Time reaching half maximum uptake rate, (h)
KLuRESn = 0.5          # Hill coefficient, (unitless)
ALuREScap = 25         # Uptake capacity per tissue weight (ug/g tissue)

KrestRESrelease = 0.0001  # Release rate constant of phagocytic cells, (h-1)
KrestRESmax = 0.00025     # Maximum uptake rate constant of phagocytic cells, (h-1)
KrestRES50 = 24           # Time reaching half maximum uptake rate, (h)
KrestRESn = 0.5           # Hill coefficient, (unitless)
ArestREScap =0.05         # Uptake capacity per tissue weight (ug/g tissue)

# Biliary excretion
KbileC = 0.0008;     # Biliary clearance (L/hr/kg^0.75)
# Kbile = 0.0012% Biliary clearance (L/hr/kg)
# changed to L/h/kg^0.75 for interspecies extrapolation 0.0012/0.02^0.75 =
# Urine excretion
KurineC = 0.0008  # Urine clearance (L/hr/kg^0.75)
#Kurine = 0.00012% Urine clearance (L/hr)
# changed to L/h/kg^0.75 for interspecies extrapolation 0.00012/0.02^0.75 =


# IV dosing
Timeiv = 0.05  # IV infusion time (h), set, approximately 15-20 seconds, on average 18 sec
PDOSEiv = 2    # mg/kg



library(deSolve)   # Loading the deSolve package
############################################################################
# Conversion of differential integration mass balance equations
############################################################################
# Obtain the derivatives
ODEPBPK <- function(Time, State, Pars) {
  with(as.list(c(State, Pars)), {
    # Cardiac output and regional blood blow (L/h)
    BW = 9.6+0.015*Time
    QC = QCC*BW                   # Cardiac output
    QL = QC*QLC                   # Blood flow to liver
    QBR = QC*QBRC                 # Blood flow to brain
    QK = QC*QKC                   # Blood flow to kidney
    QS = QC*QSC                   # Blood flow to spleen
    Qrest = QC-QL-QBR-QK-QS       # Blood flow to rest of body
    Qbal = QC-QL-QBR-QK-QS-Qrest   # Blood flow balance equation

    # Tissue volumes (L)
    VL = BW*VLC      # Liver
    VBR = BW*VBRC    # Brain
    VK = BW*VKC      # Kidney
    VS = BW*VSC      # Spleen
    VLu = BW*VLuC    # Lungs
    VBlood = BW*VBloodC
    VPlasma = BW*VPlasmaC
    Vrest = BW-VL-VBR-VK-VS-VLu-VPlasma..........................................................................
    Vbal = BW-VL-VBR-VK-VS-VLu-VPlasma-Vrest
    VLb = VL*BVL        # Weight/volume of capillary blood in liver compartment
    VLt = VL-VLb        # Weight/volume of tissue in liver compartment
    VBRb = VBR*BVBR     # Weight/volume of capillary blood in brain compartment
    VBRt = VBR-VBRb     # Weight/volume of tissue in brain compartment
    VKb = VK*BVK        # Weight/volume of capillary blood in kidney compartment
    VKt = VK-VKb        # Weight/volume of tissue in kidney compartment
    VSb = VS*BVS        # Weight/volume of capillary blood in spleen compartment
```

```
   VSt = VS-VSb        # Weight/volume of tissue in spleen compartment
   VLub = VLu*BVLu     # Weight/volume of capillary blood in Lung compartment
   VLut = VLu-VLub     # Weight/volume of tissue in Lung compartment
   Vrestb = Vrest*BVrest   # Weight/volume of capillary blood in rest of body compartment
   Vrestt = Vrest-Vrestb   # Weight/volume of tissue in rest of body compartment

   # Permeability coefficient-surface area cross-product
   PAL = PALC*QL
   PABR = PABRC*QBR
   PAK = PAKC*QK
   PAS = PASC*QS
   PALu = PALuC*QC
   PArest = PArestC*Qrest

   KLRESUP = (KLRESmax*Time^KLRESn)/(KLRES50^KLRESn+Time^KLRESn)*(1-
(ALRES/(ALREScap*VL)))
   KSRESUP = (KSRESmax*Time^KSRESn)/(KSRES50^KSRESn+Time^KSRESn)*(1-
(ASRES/(ASREScap*VS)))
   KKRESUP = (KKRESmax*Time^KKRESn)/(KKRES50^KKRESn+Time^KKRESn)*(1-
(AKRES/(AKREScap*VK)))
   KLuRESUP                                                        =
(KLuRESmax*Time^KLuRESn)/(KLuRES50^KLuRESn+Time^KLuRESn)*(1-
(ALuRES/(ALuREScap*VLu)))
   KrestRESUP                                                      =
(KrestRESmax*Time^KrestRESn)/(KrestRES50^KrestRESn+Time^KrestRESn)*(1-
(ArestRES/(ArestREScap*Vrest)))

   # Dosing
   DOSEiv = PDOSEiv*BW0     # mg
   IVR = DOSEiv/Timeiv      # mg/h
   Riv = IVR * (Time < Timeiv)
   dAiv = Riv

   # Elimination
   Kbile = KbileC*BW^0.75    # L/h
   Kurine = KurineC*BW^0.75  # L/h

   # Blood compartment
   CA = AA/(VPlasma*0.2)
   dAUCCA = CA
   CA1000 = CA*1000  # ng/g, ng/ml, ug/L
   dAUCCA1000 = CA1000
   CVLu = ALub/VLub
   RA = QC*CVLu - QC*CA
   dAA = RA
   CV = AV/(VPlasma*0.8)
```

```
dAUCCV = CV
CV1000 = CV*1000
APlasma = AA+AV
APlasmaperc = 100*(APlasma/DOSEiv)/(VPlasma*1000)
Abloodperc = 100*(APlasma/DOSEiv)/(VBlood*1000)

CVL = ALb/VLb
CVBR = ABRb/VBRb
CVK = AKb/VKb
CVrest = Arestb/Vrestb
RV = QL*CVL + QBR*CVBR + QK*CVK + Qrest*CVrest + Riv - QC*CV
dAV = RV

# Lung compartment
# Diffusion limited model
CLut = ALut/VLut
RLuRESrelease = KLuRESrelease*ALuRES
RLub = QC*(CV-CVLu) - PALu*CVLu + (PALu*CLut)/PLu + RLuRESrelease - KLuRESUP*ALub
dALub = RLub

RLut = PALu*CVLu - (PALu*CLut)/PLu
dALut = RLut
ALutotal = ALub+ALut
CLu = ALutotal/VLu
CLu1000 = CLu*1000  # ng/g, ng/ml, ug/L

RLuRES = KLuRESUP*ALub-KLuRESrelease*ALuRES
RLuRESUP = KLuRESUP*ALub  # *1000
dALuRES = RLuRES
CLung = (ALutotal+ALuRES)/VLu
CLungtissue = (ALut+ALuRES)/VLut
Clungtissue1000 = 1000*(ALut+ALuRES)/VLut
Alungtissue1000 = 1000*(ALut+ALuRES)
CLung1000 = CLung*1000

ALungtissue = ALut+ALuRES;
ALungtissueperc = 100*(ALungtissue/DOSEiv)/(VLut*1000)

# Brain compartment
# Diffusion limited model
CBRt = ABRt/VBRt
RBRb = QBR*(CA-CVBR) - PABR*CVBR + (PABR*CBRt)/PBR
dABRb = RBRb

RBRt = PABR*CVBR - (PABR*CBRt)/PBR
```

```
   dABRt = RBRt
   ABRtotal = ABRb+ABRt
   CBR = ABRtotal/VBR

   # Rest of body compartment
   # Diffusion limited model
   Crestt = Arestt/Vrestt
   RrestRESrelease = KrestRESrelease*ArestRES
   Rrestb = Qrest*(CA-CVrest) - PArest*CVrest + (PArest*Crestt)/Prest + RrestRESrelease -
KrestRESUP*Arestb
   dArestb = Rrestb

   Rrestt = PArest*CVrest - (PArest*Crestt)/Prest
   dArestt = Rrestt
   Aresttotal = Arestb+Arestt
   Crest = Aresttotal/Vrest
   Crest1000 = Crest*1000   ## ng/g, ng/ml, ug/L

   RrestRES = KrestRESUP*Arestb-KrestRESrelease*ArestRES
   RrestRESUP = KrestRESUP*Arestb # *1000
   dArestRES = RrestRES
   Crestall = (Aresttotal+ArestRES)/Vrest
   Cresttissue1000 = 1000*(Arestt+ArestRES)/Vrestt
   Cresttissue = (Arestt+ArestRES)/Vrestt
   Aresttissue1000 = 1000*(Arestt+ArestRES)
   Crestall1000 = Crestall*1000

   Aresttissue = Arestt+ArestRES
   Aresttissueperc = 100*(Aresttissue/DOSEiv)/(Vrestt*1000)

   # Kidney compartment
   AKtotal = AKb+AKt

   RKRES = KKRESUP*AKb-KKRESrelease*AKRES
   RKRESUP = KKRESUP*AKb # *1000
   RKRESrelease = KKRESrelease*AKRES
   dAKRES = RKRES
   CKidney = (AKtotal+AKRES)/VK
   CKidneytissue = (AKt+AKRES)/VKt
   CKidneytissue1000 = 1000*(AKt+AKRES)/VKt
   AKidneytissue1000 = 1000*(AKt+AKRES)
   CKidney1000 = CKidney*1000

   # Urinary excretion
   Rurine = Kurine*CVK   # mg/h
   dAurine = Rurine
```

```
  Aurine2 = Aurine*1000

  # Diffusion limited model
  CKt = AKt/VKt
  RKb = QK*(CA-CVK) - PAK*CVK + (PAK*CKt)/PK - Rurine + RKRESrelease - KKRESUP*AKb
  dAKb = RKb

  RKt = PAK*CVK - (PAK*CKt)/PK
  dAKt = RKt
  CK = AKtotal/VK
  CK1000 = CK*1000  # ng/g, ng/ml, ug/L

  AKidneytissue = AKt+AKRES
  AKidneytissueperc = 100*(AKidneytissue/DOSEiv)/(VKt*1000)

  ## Spleen compartment
  # Diffusion limited model
  CSt = ASt/VSt
  CVS = ASb/VSb
  RSRESrelease = KSRESrelease*ASRES
  RSb = QS*(CA-CVS) - PAS*CVS + (PAS*CSt)/PS + RSRESrelease - KSRESUP*ASb
  dASb = RSb

  RSt = PAS*CVS - (PAS*CSt)/PS
  dASt = RSt
  AStotal = ASb+ASt
  CS = AStotal/VS
  CS1000 = CS*1000   # ng/g, ng/ml, ug/L

  RSRES = KSRESUP*ASb-KSRESrelease*ASRES
  RSRESUP = KSRESUP*ASb  # *1000
  dASRES = RSRES
  CSpleen = (AStotal+ASRES)/VS
  Cspleentissue = (ASt+ASRES)/VSt
  CSpleentissue1000 = 1000*(ASt+ASRES)/VSt
  dAUCCSpleentissue1000 = CSpleentissue1000
  ASpleentissue1000 = 1000*(ASt+ASRES)
  CSpleen1000 = CSpleen*1000
  ASpleentissue = ASt+ASRES
  ASpleentissueperc = 100*(ASpleentissue/DOSEiv)/(VSt*1000)

  ## Liver compartment
  # Biliary excretion
  Rbile = Kbile*CVL   # mg/h
  dAbile = Rbile
```

```
   ALtotal = ALb+ALt

   RLRES = KLRESUP*ALb-KLRESrelease*ALRES
   RLRESUP = KLRESUP*ALb  # *1000
   RLRESrelease = KLRESrelease*ALRES
   dALRES = RLRES
   CLiver = (ALtotal+ALRES)/VL
   CLivertissue1000 = 1000*(ALt+ALRES)/VLt
   dAUCCLivertissue1000 = CLivertissue1000
   CLivertissue = (ALt+ALRES)/VLt
   ALivertissue1000 = 1000*(ALt+ALRES)
   CLiver1000 = CLiver*1000

   ALivertissue = ALt+ALRES
   ALivertissueperc = 100*(ALivertissue/DOSEiv)/(VLt*1000)

   # Diffusion limited model
   CLt = ALt/VLt
   RLb = QL*(CA-CVL) + QS*CVS - PAL*CVL + (PAL*CLt)/PL - Rbile + RLRESrelease -
KLRESUP*ALb # KLRESUP*AA
   dALb= RLb

   RLt = PAL*CVL - (PAL*CLt)/PL
   dALt = RLt
   CL = ALtotal/VL
   CL1000 = CL*1000  # ng/g, ng/ml, ug/L


return(list(c(dAiv,dAUCCA,dAUCCA1000,dAA,dAUCCV,dAV,dALub,dALut,dALuRES,dAB
Rb,dABRt,dArestb,dArestt,dArestRES,

dAKRES,dAurine,dAKb,dAKt,dASb,dASt,dASRES,dAUCCSpleentissue1000,dAbile,dALRES,
dAUCCLivertissue1000,
               dALb,dALt)))
  })
}

Time <- seq(from=0, to=770, by=0.1)      # Observation time
Pars                                                                          <-
c(QCC,QLC,QBRC,QKC,QSC,VLC,VBRC,VKC,VSC,VLuC,VBloodC,VPlasmaC,BW0,VPlas
maC,Timeiv,KbileC,KurineC,KLRESrelease,

KLRESmax,KLRES50,KLRESn,ALREScap,KSRESrelease,KSRESmax,KSRES50,KSRESn,A
SREScap,KKRESrelease,KKRESmax,
KKRES50,KKRESn,AKREScap,KLuRESrelease,KLuRESmax,KLuRES50,KLuRESn,ALuRES
cap,KrestRESrelease,KrestRESmax,

KrestRES50,KrestRESn,ArestREScap,BVL,BVBR,BVK,BVS,BVLu,BVrest,PL,PBR,PK,PS,PL
u,Prest,PALC,PABRC,PAKC,
            PASC,PALuC,PArestC)
State <-
c(Aiv=0,AUCCA=0,AUCCA1000=0,AA=0,AUCCV=0,AV=0,ALub=0,ALut=0,ALuRES=0,AB
Rb=0,ABRt=0,Arestb=0,Arestt=0,

ArestRES=0,AKRES=0,Aurine=0,AKb=0,AKt=0,ASb=0,ASt=0,ASRES=0,AUCCSpleentissue1
000=0,Abile=0,
            ALRES=0,AUCCLivertissue1000=0,ALb=0,ALt=0) #Initial values
pbpkout <- ode(func = ODEPBPK, y = State , parms =Pars, times = Time) # Obtain integration
summary(pbpkout)

# Generate plots for each state by time
par(mar=c(2.0,2.0,2.0,2.0))
plot(pbpkout, xlab = "Time", ylab = "-")
mtext(outer = TRUE, side = 3,  cex = 1.5)

# Covert matrix into data frame
pbpkout<-data.frame(pbpkout)

# Mass balance
ALtotal = pbpkout$ALb + pbpkout$ALt
ABRtotal = pbpkout$ABRb + pbpkout$ABRt
AKtotal = pbpkout$AKb + pbpkout$AKt
ALutotal = pbpkout$ALub + pbpkout$ALut
Aresttotal = pbpkout$Arestb + pbpkout$Arestt
AStotal = pbpkout$ASb + pbpkout$ASt
ALivertissue1000 = 1000*(pbpkout$ALt+pbpkout$ALRES)
ASpleentissue1000 = 1000*(pbpkout$ASt+pbpkout$ASRES)
AKidneytissue1000 = 1000*(pbpkout$AKt+pbpkout$AKRES)
Alungtissue1000 = 1000*(pbpkout$ALut+pbpkout$ALuRES)


Tmass =
pbpkout$AA+pbpkout$AV+ALtotal+ABRtotal+AKtotal+ALutotal+Aresttotal+AStotal+pbpkou
t$Abile+pbpkout$Aurine+pbpkout$ALRES+pbpkout$ASRES+pbpkout$ALuRES+pbpkout$AK
RES+pbpkout$ArestRES
Bal = pbpkout$Aiv-Tmass

# Merge all output variable into one dataset
```

```
pbpkout = cbind(pbpkout, ALtotal,ABRtotal,AKtotal,ALutotal,Aresttotal,AStotal,
ALivertissue1000, ASpleentissue1000,
            AKidneytissue1000, Alungtissue1000, Tmass, Bal)
# Set physical directory of the folder
setwd("C:/Users/zhoumeng/Desktop/Code Conversion project/Chunla R language")
# Export the data
write.csv(pbpkout, file="Gold nanoparticle .csv")


#par(mfrow=c(1,1))

####################################
### Obtain values of concentrations
# CA: concentration in the artery
#device.off()
par(mfrow=c(3,2))
par(mar=c(2.0,2.0,2.0,2.0))
BW = 9.6+0.015*Time
VPlasma = BW*VPlasmaC
CA =  (pbpkout$AA)/(VPlasma*0.2)
plot (y=CA, x=Time,xlab = "Time", main = "CA", type='l')

CA1000 =  (pbpkout$AA)/(VPlasma*0.2)*1000
plot (y=CA1000, x=Time,xlab = "Time", main = "CA1000", type='l')
# CV: concentration in the venous
CV = (pbpkout$AV)/(VPlasma*0.8)
plot (y=CV, x=Time,xlab = "Time", main = "CV", type='l')

CV1000 = (pbpkout$AV)/(VPlasma*0.8)*1000
plot (y=CV1000, x=Time,xlab = "Time", main = "CV1000", type='l')
# Clungtissue1000: concentration in the lung tissue
VLu = BW*VLuC
VLub = VLu*BVLu
VLut = VLu-VLub
Clungtissue1000 = 1000*(pbpkout$ALut+pbpkout$ALuRES)/VLut
plot (y=Clungtissue1000, x=Time,xlab = "Time", main = "Clungtissue1000", type='l')
# Clung1000: concentration in the lung
Clung1000 = 1000*(ALutotal+pbpkout$ALuRES)/VLu
plot (y=Clung1000, x=Time,xlab = "Time", main = "Clung1000", type='l')
# CKidneytissue1000: concentration in the kidney tissue
VK = BW*VKC
VKb = VK*BVK
VKt = VK-VKb
CKidneytissue1000 = 1000*(pbpkout$AKt+pbpkout$AKRES)/VKt
plot (y=CKidneytissue1000, x=Time,xlab = "Time", main = "CKidneytissue1000", type='l')
# CKidney1000: concentration in the kidney
```

```
CKidney1000 = 1000*(AKtotal+pbpkout$AKRES)/VK
plot (y=CKidney1000, x=Time,xlab = "Time", main = "CKidney1000", type='l')
# CSpleentissue1000: concentration in the spleen tissue
VS = BW*VSC
VSb = VS*BVS
VSt = VS-VSb
CSpleentissue1000 = 1000*(pbpkout$ASt+pbpkout$ASRES)/VSt
plot (y=CSpleentissue1000, x=Time,xlab = "Time", main = "CSpleentissue1000", type='l')
# CSpleen1000: concentration in the spleen
CSpleen1000 = 1000*(AStotal+pbpkout$ASRES)/VS
plot (y=CSpleen1000, x=Time,xlab = "Time", main = "CSpleen1000", type='l')
# CLivertissue1000: concentration in the liver tissue
VL = BW*VLC
VLb = VL*BVL
VLt = VL-VLb
CLivertissue1000 = 1000*(pbpkout$ALt+pbpkout$ALRES)/VLt
plot (y=CLivertissue1000, x=Time,xlab = "Time", main = "CLivertissue1000", type='l')
# CLiver1000: concentration in the liver
CLiver1000 = 1000*(ALtotal+pbpkout$ALRES)/VL
plot (y=CLiver1000, x=Time,xlab = "Time", main = "CLiver1000", type='l')


# Export concentration
concentrat_gold = cbind(CA1000,CV1000,Clungtissue1000,Clung1000,CKidneytissue1000,
                CKidney1000,CSpleentissue1000,CSpleen1000,CLivertissue1000,
                CLiver1000)
write.csv(concentrat_gold, file="nanoparticle concentration.csv")
```

A representative runtime file from the gold nanoparticle PBPK model in acslX™ format

```
% Data corresponds to Figure 2E in Lin et al. Nanomedicine (Lond). 2016, 11:107-119.

prepare @all
PDOSEIV = 2 % mg/kg
start @NoCallback

% Concentrations of 15-20nm gum arabic-coated AuNPs in the plasma. The columns are time
and conc. Unit: ug/ml.

FentPlasma = [0.5      11.91339405
0.5      12.95868791
1        10.07896699
1        10.9342156
2        9.321418012
2        8.846327386
4        8.817240205
4        8.215452746
24       5.525799568
48       3.279540313
72       2.07861991
96       1.257677525
120      0.816848133
144      0.566108983]

plot(_t,_cv,FentPlasma(:,1),FentPlasma(:,2),'+',"Plasma.aps")
```

A representative Main script file from the gold nanoparticle PBPK model in MATLAB® format

```
clear
clc
% close all

%% Input Arguments
PDOSEORAL = 0;
PDOSEIV = 2; % mg/kg
PDOSEIM = 20;% mg/kg
TINTERVAL=0:6:24;
KA=0.012;
KIM=0.15;
TSTOP = 770; %4325%240.0;
FRAC=0.95;

BW = 11.3; % Body weight (kg) (Baggot et al., 1977)

% Dosing
 Doseim = PDOSEIM*BW; % (mg)

%...Dosing, intramuscular, dissolution model
 Doseimfast = Doseim*FRAC;
 Doseimslow = Doseim*(1-FRAC);


Initials=zeros(27,1);
% Initials(6)=Doseimfast;
% Initials(7)=Doseimslow;

%% Running ODE Solver
ti=0;
tf=1000;
tspan=[ti tf];
options=odeset('AbsTol',10e-2,'RelTol',10e-2,'Stats','on');
tic;
[t,y] =
ode23(@(T,Y)GoldPBPK(T,Y,PDOSEORAL,PDOSEIV,PDOSEIM,TINTERVAL,KA,TSTOP,KIM,FRA
C), tspan, Initials);
toc;

%% Evaluation for fixed step sizes
t0 = ti:.1:tf;
y0=zeros(length(t0),27);
t0=t0';
for i=1:length(Initials)
y0(:,i) = interp1(t,y(:,i),t0);
end
t=t0;
y=y0;
ALivertissue1000 = 1000*(y(:,23)+y(:,24));

%%
```

```
% Amounts of 15-20nm gum arabic-coated AuNPs in the liver. The columns are
time and amount. Unit: ug.

FentLiver = textread('FentLiver.txt');

cv=zeros(length(t),1);
for i=1:length(t)-1
    cv(i,1)=(y(i+1,10)-y(i,10))/(t(i+1)-t(i));
end

figure;
plot(t,ALivertissue1000,FentLiver(:,1),FentLiver(:,2),'+')
set(gca,'yscale','log');
box off
xlabel('Time','fontsize',12)
ylabel('Amounts of AuNPs in liver','fontsize',12)
% title('IV Serum 5mgkg','fontweight','bold','fontsize',13)
```

A representative DATA file used in the Main script file from the gold nanoparticle PBPK model

Save the following data file as a .txt file with the name `'FentLiver.txt'`.

168	8460.51
168	15112.73
768	11465.65
768	11828.9

Optimized PBPK model code for gold nanoparticle in MATLAB® format

```
function [dT,CV]=OptimizedGoldPBPK2(FinalTime)

 dT=.0002;
 N=FinalTime/dT;
% Gold Nanoparticle PBPK model
% code that is executed once at the beginning of a simulation run goes here
% Blood flow rate (Fraction of cardiac output)
QCC = 5.0; %Cardiac output(L/h/kg) (Upton, 2008; Yuan et al., 2011)
QLC = 0.2725; % Fraction of blood flow to liver (Average of Buur et al., 2005
and Upton, 2008)
QBRC = 0.03; % Fraction of blood flow to brain (Upton,2008)
QKC = 0.12; % Fraction of blood flow to kidneys (Average of Buur et al., 2005
and Upton, 2008)
QSC = 0.0151; %Davies and Morris (1993)Table III% Fraction of blood flow to
spleen (average of 6 species)

% Tissue volumes (Fraction of body weight)
BW0 = 9.6; % Body weight (kg) (Upton, 2008)
VLC = 0.0247; % Liver (Average of Buur et al., 2005 and Upton, 2008)
VBRC = 0.004; % Brain (Upton, 2008)
VKC = 0.004; % Kidneys (Average of Buur et al., 2005 and Upton, 2008)
VSC = 0.002; % Spleen (Upton, 2008)
VLuC = 0.01; % Lungs (Upton, 2008)
VBloodC = 0.06; % Blood (Average of Buur et al., 2005 and Upton, 2008)
VPlasmaC = 0.04; %0.04% Buur et al. 2005 (Hematocrit is 0.33)

% Blood volume fraction in organs and tissues (percentage of tissues) All rat
values are from Table 30 in Brown et al. (1997).
BVL = 0.115; % Liver (Buur et al. 2005)
BVBR = 0.0275; % Brain (Brown et al., 1997; Table 30, average of 4 species)
BVK = 0.105; % Kidneys (Buur et al., 2005)
BVS = 0.3; % Spleen (Brown et al., 1997; Table 30, average of 3 species)
BVLu = 0.3867; % Lungs (Brown et al., 1997; Table 30, average of 3 species)
BVrest = 0.026; % Rest of body (Assume the same as the muscle in Buur et al.
2005)

% Partition coefficients (PC), unitless
PL = 0.08; % Liver:blood PC (calculated using AUC method, 7-day AUC data from
Cho et al., 2010, data for 13-nm AuNPs)
PBR = 0.15; %Brain:blood PC (generic value, Li et al., 2014)
PK = 0.15; %Kidneys:blood PC (calculated using AUC method, 7-day AUC data
from Cho et al., 2010, data for 13-nm AuNPs)
PS = 0.15; %Spleen:blood PC (calculated using AUC method, 7-day AUC data from
Cho et al., 2010, data for 13-nm AuNPs)
PLu = 0.15; %Lungs:blood PC (calculated using AUC method, 7-day AUC data from
Cho et al., 2010, data for 13-nm AuNPs)
Prest = 0.15; % Rest of body:blood PC (generic value, Li et al., 2014)

% Diffusion limitation coefficient constants, unitless (Based on Li et al.,
2014)
PALC = 0.001; % Permeability coefficient between blood and liver, generic
value from Li et al., 2014
```

```
PABRC = 0.000001; % Permeability coefficient between blood and brain, set a
very low values
PAKC = 0.001; %0.001% Permeability coefficient between blood and kidneys,
generic value from Li et al., 2014
PASC =0.001; % Permeability coefficient between blood and spleen, generic
value from Li et al., 2014
PALuC = 0.001; % Permeability coefficient between blood and lungs, generic
value from Li et al., 2014
PArestC = 0.000001; % Permeability coefficient between blood and rest of
body, from Li et al., 2014

KLRESrelease = 0.005;
KLRESmax = 1000;
KLRES50 = 24;
KLRESn = 0.5;
ALREScap = 32.5;

KSRESrelease = 0.02;
KSRESmax = 500;
KSRES50 = 24;
KSRESn = 0.5;
ASREScap = 25;

KKRESrelease = 0.04;
KKRESmax = 500;
KKRES50 = 24;
KKRESn = 0.5;
AKREScap = 55;

KLuRESrelease = 0.02;
KLuRESmax = 300;
KLuRES50 = 24;
KLuRESn = 0.5;
ALuREScap = 25;

KrestRESrelease = 0.0001;
KrestRESmax = 0.00025;
KrestRES50 = 24;
KrestRESn = 0.5;
ArestREScap =0.05;

% Biliary excretion
KbileC = 0.0008; %0.00008% Biliary clearance (L/hr/kg^0.75)
%Kbile = 0.0012% Biliary clearance (L/hr/kg)
% changed to L/h/kg^0.75 for interspecies extrapolation 0.0012/0.02^0.75 =
% Urine excretion
KurineC = 0.0008; %0.0008% % Urine clearance (L/hr/kg^0.75)
%Kurine = 0.00012% Urine clearance (L/hr)
% changed to L/h/kg^0.75 for interspecies extrapolation 0.00012/0.02^0.75 =


% IV dosing
Timeiv = 0.05; %0.005 % IV infusion time (h), set, approximately 15-20
seconds, on average 18 sec
PDOSEiv = 2; %1%2 % mg/kg
```

```
%% Variables
%
AIV=zeros(N+1,1);
AA=zeros(N+1,1);
AUCCA=zeros(N+1,1);
AUCCA1000=zeros(N+1,1);
AV=zeros(N+1,1);
ALub=zeros(N+1,1);
ALut=zeros(N+1,1);
Arestb=zeros(N+1,1);
ALuRES=zeros(N+1,1);
ABRb=zeros(N+1,1);
ABRt=zeros(N+1,1);
Arestt=zeros(N+1,1);
ArestRES=zeros(N+1,1);
AKb=zeros(N+1,1);
AKt=zeros(N+1,1);
Aurine=zeros(N+1,1);
AKRES=zeros(N+1,1);
ASb=zeros(N+1,1);
ASt=zeros(N+1,1);
ASRES=zeros(N+1,1);
AUCCSpleentissue1000=zeros(N+1,1);
ALb=zeros(N+1,1);
ALt=zeros(N+1,1);
ALRES=zeros(N+1,1);
AUCCLivertissue1000=zeros(N+1,1);
Abile=zeros(N+1,1);
IntCV=zeros(N+1,1);

CV=zeros(N+1,1);
error=1e-23;

%% DERIVATIVE
for t=1:N
    % code for calulating the derivative goes here
    % Scaled papameters
    % Cardiac output and regional blood blow (L/h)
    T=(t-1)*dT;
    BW = 9.6+0.015*T;
    QC = QCC*BW; %^0.75 % Cardiac output
    QL = QC*QLC ; % Blood flow to liver
    QBR = QC*QBRC ; % Blood flow to brain
    QK = QC*QKC ; % Blood flow to kidney
    QS = QC*QSC ; % Blood flow to spleen
    Qrest = QC-QL-QBR-QK-QS; % Blood flow to rest of body
%     Qbal = QC-QL-QBR-QK-QS-Qrest ; %Blood flow balance equation

    % Tissue volumes (L)
    VL = BW*VLC ; % Liver
    VBR = BW*VBRC ; % Brain
    VK = BW*VKC ; % Kidney
    VS = BW*VSC ; % Spleen
    VLu = BW*VLuC ; % Lungs
%     VBlood = BW*VBloodC;
```

```
    VPlasma = BW*VPlasmaC;
    Vrest = BW-VL-VBR-VK-VS-VLu-VPlasma ;
%     Vbal = BW-VL-VBR-VK-VS-VLu-VPlasma-Vrest  ;
    VLb = VL*BVL ; % Weight/volume of capillary blood in liver compartment
    VLt = VL-VLb ; % Weight/volume of tissue in liver compartment
    VBRb = VBR*BVBR ; % Weight/volume of capillary blood in brain compartment
    VBRt = VBR-VBRb ; % Weight/volume of tissue in brain compartment
    VKb = VK*BVK ; % Weight/volume of capillary blood in kidney compartment
    VKt = VK-VKb ; % Weight/volume of tissue in kidney compartment
    VSb = VS*BVS ; % Weight/volume of capillary blood in spleen compartment
    VSt = VS-VSb ; % Weight/volume of tissue in spleen compartment
    VLub = VLu*BVLu ; % Weight/volume of capillary blood in Lung compartment
    VLut = VLu-VLub ; % Weight/volume of tissue in Lung compartment
    Vrestb = Vrest*BVrest ; % Weight/volume of capillary blood in rest of
body compartment
    Vrestt = Vrest-Vrestb ; % Weight/volume of tissue in rest of body
compartment

    % Permeability coefficient-surface area cross-product
    PAL = PALC*QL;
    PABR = PABRC*QBR;
    PAK = PAKC*QK;
    PAS = PASC*QS;
    PALu = PALuC*QC;
    PArest = PArestC*Qrest;

    KLRESUP = (KLRESmax*T^KLRESn)/(KLRES50^KLRESn+T^KLRESn)*(1-
(ALRES(t)/(ALREScap*VL)));
    KSRESUP = (KSRESmax*T^KSRESn)/(KSRES50^KSRESn+T^KSRESn)*(1-
(ASRES(t)/(ASREScap*VS)));
    KKRESUP = (KKRESmax*T^KKRESn)/(KKRES50^KKRESn+T^KKRESn)*(1-
(AKRES(t)/(AKREScap*VK)));
    KLuRESUP = (KLuRESmax*T^KLuRESn)/(KLuRES50^KLuRESn+T^KLuRESn)*(1-
(ALuRES(t)/(ALuREScap*VLu)));
    KrestRESUP =
(KrestRESmax*T^KrestRESn)/(KrestRES50^KrestRESn+T^KrestRESn)*(1-
(ArestRES(t)...
        /(ArestREScap*Vrest)));

    % Dosing
    DOSEiv = PDOSEiv*BW0 ; % mg
    IVR = DOSEiv/Timeiv ; % mg/h
    RIV = IVR*(1.-heaviside(T-Timeiv));
    AIV(t+1)=AIV(t)+dT*RIV; % dy(1) = (RIV);

    % Elimination
    Kbile = KbileC*BW^0.75 ; % L/h
    Kurine = KurineC*BW^0.75 ; % L/h

    % Blood compartment
    %CA = AA/VBlood
    CA = AA(t)/(VPlasma*0.2);
    AUCCA(t+1)=(AUCCA(t)+dT*CA); % dy(3) = (CA);
    CA1000 = CA*1000 ; % ng/g, ng/ml, ug/L
    AUCCA1000(t+1)=(AUCCA1000(t)+dT*CA1000); %dy(4) = (CA1000);
    % CA = Arterial blood concentration (mg/L or ug/ml)
    CVLu = ALub(t)/VLub;
    RA = QC*CVLu - QC*CA ;
    Temp=0;% QC*CA;
    AA(t+1)=(AA(t)+dT*(RA+Temp))/(1+dT*Temp/(AA(t)+error)); % dy(2) = (RA);

    %CV = AV/VBlood
    CV(t) = AV(t)/(VPlasma*0.8);
    IntCV(t+1)=IntCV(t)+dT*CV(t); % dy(27)=(CV);
%     CV1000 = CV(t)*1000;
%     APlasma = AA(t+1)+AV(t);
%     APlasmaperc = 100*(APlasma/DOSEiv)/(VPlasma*1000);
%     Abloodperc = 100*(APlasma/DOSEiv)/(VBlood*1000);
    % CV = Venous blood concentration (mg/L or ug/ml)

    CVL = ALb(t)/VLb;
    CVBR = ABRb(t)/VBRb;
    CVK = AKb(t)/VKb;
    CVrest = Arestb(t)/Vrestb;
    RV = QL*CVL + QBR*CVBR + QK*CVK + Qrest*CVrest + RIV - QC*CV(t) ;
    Temp=0;% QC*CV(t);
    AV(t+1)=(AV(t)+dT*(RV+Temp))/(1+Temp/(AV(t)+error)); % dy(5) = (RV);

    % Lung compartment
    % Diffusion limited model
    CLut = ALut(t)/VLut;
    RLuRESrelease = KLuRESrelease*ALuRES(t);
    RLub = QC*(CV(t)-CVLu) - PALu*CVLu + (PALu*CLut)/PLu + RLuRESrelease -
KLuRESUP*ALub(t); %
    Temp=0;% QC*CVLu+PALu*CVLu+KLuRESUP*ALub(t);
    ALub(t+1)=(ALub(t)+dT*(RLub+Temp))/(1+dT*Temp/(ALub(t)+error)); % dy(6) =
(RLub);

    RLut = PALu*CVLu - (PALu*CLut)/PLu;
    Temp=0;% (PALu*CLut)/PLu;
    ALut(t+1)=(ALut(t)+dT*(RLut+Temp))/(1+dT*Temp/(ALut(t)+error)); % dy(7) =
(RLut);
    ALutotal = ALub(t)+ALut(t);
%     CLu = ALutotal/VLu;
%     CLu1000 = CLu*1000 ; % ng/g, ng/ml, ug/L

    RLuRES = KLuRESUP*ALub(t)-KLuRESrelease*ALuRES(t);
%     RLuRESUP = KLuRESUP*ALub(t); %*1000
    Temp=0;% KLuRESrelease*ALuRES(t);
    ALuRES(t+1)=(ALuRES(t)+dT*(RLuRES+Temp))/(1+dT*Temp/(ALuRES(t)+error));
%     CLung = (ALutotal+ALuRES)/VLu;
%     CLungtissue = (ALut(t+1)+ALuRES)/VLut;
%     Clungtissue1000 = 1000*(ALut(t+1)+ALuRES)/VLut;
%     Alungtissue1000 = 1000*(ALut+ALuRES);
%     CLung1000 = CLung*1000;

%     ALungtissue = ALut(t+1)+ALuRES(t+1);
%     ALungtissueperc = 100*(ALungtissue/DOSEiv)/(VLut*1000);

    % Brain compartment
    % Diffusion limited model
    CBRt = ABRt(t)/VBRt;
    RBRb = QBR*(CA-CVBR) - PABR*CVBR + (PABR*CBRt)/PBR;
    Temp=0;% QBR*(CVBR) + PABR*CVBR;
    ABRb(t+1) = (ABRb(t)+dT*(RBRb+Temp))/(1+dT*Temp/(ABRb(t)+error)); % dy(9)
= (RBRb);

    RBRt = PABR*CVBR - (PABR*CBRt)/PBR;
    Temp=0;% (PABR*CBRt)/PBR;
    ABRt(t+1) = (ABRt(t)+dT*(RBRt+Temp))/(1+dT*Temp/(ABRt(t)+error)); %
dy(10) = (RBRt);
    ABRtotal = ABRb(t)+ABRt(t);
%     CBR = ABRtotal/VBR;

    % Rest of body compartment
    % Diffusion limited model
    Crestt = Arestt(t)/Vrestt;
    RrestRESrelease = KrestRESrelease*ArestRES(t);
    Rrestb = Qrest*(CA-CVrest) - PArest*CVrest + (PArest*Crestt)/Prest +
RrestRESrelease - KrestRESUP*Arestb(t); %
    Temp=0;% Qrest*(CVrest) + PArest*CVrest + KrestRESUP*Arestb(t);
    Arestb(t+1) = (Arestb(t)+dT*(Rrestb+Temp))/(1+dT*Temp/(Arestb(t)+error));
% dy(11) = (Rrestb);

    Rrestt = PArest*CVrest - (PArest*Crestt)/Prest;
    Temp=0;% (PArest*Crestt)/Prest;
    Arestt(t+1) = (Arestt(t)+dT*(Rrestt+Temp))/(1+dT*Temp/(Arestt(t)+error));
% dy(12) = (Rrestt);
    Aresttotal = Arestb(t)+Arestt(t);
%     Crest = Aresttotal/Vrest;
%     Crest1000 = Crest*1000 ; % ng/g, ng/ml, ug/L

    RrestRES = KrestRESUP*Arestb(t)-KrestRESrelease*ArestRES(t);
%     RrestRESUP = KrestRESUP*Arestb(t); %*1000
    Temp=0;% KrestRESrelease*ArestRES(t);
    ArestRES(t+1) =
(ArestRES(t)+dT*(RrestRES+Temp))/(1+dT*Temp/(ArestRES(t)+error)); % dy(13) =
(RrestRES);
%     Crestall = (Aresttotal+ArestRES(t+1))/Vrest;
%     Cresttissue1000 = 1000*(Arestt(t+1)+ArestRES(t+1))/Vrestt;
%     Cresttissue = (Arestt(t+1)+ArestRES(t+1))/Vrestt;
%     Aresttissue1000 = 1000*(Arestt(t+1)+ArestRES(t+1));
%     Crestall1000 = Crestall*1000;

%     Aresttissue = Arestt(t+1)+ArestRES(t+1);
%     Aresttissueperc = 100*(Aresttissue/DOSEiv)/(Vrestt*1000);

    % Kidney compartment

    AKtotal = AKb(t)+AKt(t);

    %RKRES = KKRESUP*AA-KKRESrelease*AKRES
    RKRES = KKRESUP*AKb(t)-KKRESrelease*AKRES(t);
%     RKRESUP = KKRESUP*AKb; %*1000
    RKRESrelease = KKRESrelease*AKRES(t);
    Temp=0;% KKRESrelease*AKRES(t);
    AKRES(t+1) = (AKRES(t)+dT*(RKRES+Temp))/(1+dT*Temp/(AKRES(t)+error)); %
dy(17) = (RKRES);
%     CKidney = (AKtotal+AKRES(t+1))/VK;
%     CKidneytissue = (AKt+AKRES(t+1))/VKt;
%     CKidneytissue1000 = 1000*(AKt+AKRES(t+1))/VKt;
%     AKidneytissue1000 = 1000*(AKt+AKRES(t+1));
%     CKidney1000 = CKidney*1000;

    % Urinary excretion
    Rurine = Kurine*CVK ; % mg/h
    Aurine(t+1) = (Aurine(t)+dT*Rurine); % dy(16) = (Rurine);
%     Aurine2 = Aurine(t+1)*1000;

    % Diffusion limited model
    CKt = AKt(t)/VKt;
    RKb = QK*(CA-CVK) - PAK*CVK + (PAK*CKt)/PK - Rurine + RKRESrelease -
KKRESUP*AKb(t); %
    Temp=0;% QK*(CVK) + PAK*CVK + Rurine + KKRESUP*AKb(t);
    AKb(t+1) = (AKb(t)+dT*(RKb+Temp))/(1+dT*Temp/(AKb(t)+error)); % dy(14) =
(RKb);

    RKt = PAK*CVK - (PAK*CKt)/PK;
    Temp=0;% (PAK*CKt)/PK;
    AKt(t+1) = (AKt(t)+dT*(RKt+Temp))/(1+dT*Temp/(AKt(t)+error)); % dy(15) =
(RKt);
%     CK = AKtotal/VK;
%     CK1000 = CK*1000 ; % ng/g, ng/ml, ug/L

%     AKidneytissue = AKt(t+1)+AKRES(t+1);
%     AKidneytissueperc = 100*(AKidneytissue/DOSEiv)/(VKt*1000);

    % Spleen compartment
    % Diffusion limited model
    CSt = ASt(t)/VSt;
    CVS = ASb(t)/VSb;
    RSRESrelease = KSRESrelease*ASRES(t);
    RSb = QS*(CA-CVS) - PAS*CVS + (PAS*CSt)/PS + RSRESrelease -
KSRESUP*ASb(t); %
    Temp=0;% QS*(CVS) + PAS*CVS + KSRESUP*ASb(t);
    ASb(t+1) = (ASb(t)+dT*(RSb+Temp))/(1+dT*Temp/(ASb(t)+error)); % dy(18) =
(RSb);

    RSt = PAS*CVS - (PAS*CSt)/PS;
    Temp=0;% (PAS*CSt)/PS;
    ASt(t+1) = (ASt(t)+dT*(RSt+Temp))/(1+dT*Temp/(ASt(t)+error)); % dy(19) =
(RSt);
    AStotal = ASb(t)+ASt(t);
%     CS = AStotal/VS;
%     CS1000 = CS*1000 ; % ng/g, ng/ml, ug/L

    %RSRES = KSRESUP*AA-KSRESrelease*ASRES
    RSRES = KSRESUP*ASb(t)-KSRESrelease*ASRES(t);
%     RSRESUP = KSRESUP*ASb(t); %*1000
    Temp=0;% KSRESrelease*ASRES(t);
    ASRES(t+1) = (ASRES(t)+dT*(RSRES+Temp))/(1+dT*Temp/(ASRES(t)+error)); %
dy(20) = (RSRES);
%     CSpleen = (AStotal+ASRES(t+1))/VS;
%     Cspleentissue = (ASt(t+1)+ASRES(t+1))/VSt;
    CSpleentissue1000 = 1000*(ASt(t)+ASRES(t))/VSt;
    AUCCSpleentissue1000(t+1) =
(AUCCSpleentissue1000(t)+dT*CSpleentissue1000); % dy(21) =
(CSpleentissue1000);
%     ASpleentissue1000 = 1000*(ASt(t+1)+ASRES(t+1));
%     CSpleen1000 = CSpleen*1000;
%     ASpleentissue = ASt(t+1)+ASRES(t+1);
%     ASpleentissueperc = 100*(ASpleentissue/DOSEiv)/(VSt*1000);

    % Liver compartment

    % Biliary excretion
    Rbile = Kbile*CVL ; % mg/h
    Abile(t+1) = (Abile(t)+dT*Rbile); % dy(26) = (Rbile);

    ALtotal = ALb(t)+ALt(t);

    %RLRES = KLRESUP*AA-KLRESrelease*ALRES
    RLRES = KLRESUP*ALb(t)-KLRESrelease*ALRES(t);
%     RLRESUP = KLRESUP*ALb(t); %*1000
    RLRESrelease = KLRESrelease*ALRES(t);
    Temp=0;% KLRESrelease*ALRES(t);
    ALRES(t+1) = (ALRES(t)+dT*(RLRES+Temp))/(1+dT*Temp/(ALRES(t)+error)); %
dy(24) = (RLRES);
%     CLiver = (ALtotal+ALRES(t+1))/VL;
    CLivertissue1000 = 1000*(ALt(t)+ALRES(t))/VLt;
    AUCCLivertissue1000(t+1) = (AUCCLivertissue1000(t)+dT*CLivertissue1000);
% dy(25) = (CLivertissue1000);
%     CLivertissue = (ALt+ALRES(t+1))/VLt;
%     ALivertissue1000 = 1000*(ALt+ALRES(t+1));
%     CLiver1000 = CLiver*1000;

%     ALivertissue = ALt(t)+ALRES(t+1);
%     ALivertissueperc = 100*(ALivertissue/DOSEiv)/(VLt*1000);

    % Diffusion limited model
    CLt = ALt(t)/VLt;
    RLb = QL*(CA-CVL) + QS*CVS - PAL*CVL + (PAL*CLt)/PL - Rbile +
RLRESrelease - KLRESUP*ALb(t); %- KLRESUP*AA
    Temp=0;% QL*(CVL) + PAL*CVL + Rbile + KLRESUP*ALb(t);
    ALb(t+1) = (ALb(t)+dT*(RLb+Temp))/(1+dT*Temp/(ALb(t)+error)); % dy(22) =
(RLb);

    RLt = PAL*CVL - (PAL*CLt)/PL;
    Temp=0;% (PAL*CLt)/PL;
    ALt(t+1) = (ALt(t)+dT*(RLt+Temp))/(1+dT*Temp/(ALt(t)+error)); % dy(23) =
(RLt);
%     CL = ALtotal/VL;
%     CL1000 = CL*1000 ; % ng/g, ng/ml, ug/L

    % Mass balance
    Tmass = AA(t+1)+AV(t+1)+ALtotal+ABRtotal+AKtotal+ALutotal+Aresttotal+...

AStotal+Abile(t+1)+Aurine(t+1)+ALRES(t+1)+ASRES(t+1)+ALuRES(t+1)+AKRES(t+1)+A
restRES(t+1); %
    Bal = AIV(t+1)-Tmass;

end
```

A representative Main script file from the optimized gold nanoparticle PBPK model in MATLAB®

```
%%
clc;
clear
close all

FinalTime=160;

tic;
[dT,CV]=OptimizedGoldPBPK2(FinalTime);
toc;
CV(1,1)=CV(2,1);

figure(1);
hold on
Time=0:dT:160;
T=1:1:length(Time);
plot(Time(T),CV(T),'r')
set(gca,'yscale','log');
set(gca,'xscale','log');
xlim([.1,144])
box off
xlabel('Time (h)','fontsize',12)
ylabel('Plasma gold concentration (\mug/ml)','fontsize',12)

legend('Type your legend here', 'Type your second legend here')

ax = gca;
c = ax.Color;
ax.FontSize = 16;
```

References

Butcher JC (2008). *Numerical methods for ordinary differential equations (Second Edition)*. John Wiley & Sons.

Dormand JR, Prince PJ. A family of embedded Runge-Kutta formulae. Journal of computational and applied mathematics 1980;6:19-26.

Hairer E, Nrsett SP, Wanner G (2010). *Solving Ordinary Differential Equations: Nonstiff problems. v. 2: Stiff and differential-algebraic problems*. Springer Verlag.

Liang X, Wang H, Grice JE, Li L, Liu X, Xu ZP, Roberts MS. Physiologically Based Pharmacokinetic Model for Long-Circulating Inorganic Nanoparticles. Nano Letters 2016;16:939-45.

Lin Z, Li M, Gehring R, Riviere JE. Development and application of a multiroute physiologically based pharmacokinetic model for oxytetracycline in dogs and humans. Journal of Pharmaceutical Sciences 2015;104:233-43.

Lin Z, Monteiro-Riviere NA, Kannan R, Riviere JE. A computational framework for interspecies pharmacokinetics, exposure and toxicity assessment of gold nanoparticles. Nanomedicine (Lond) 2016a;11:107-19.

Lin Z, Monteiro-Riviere NA, Riviere JE. A physiologically based pharmacokinetic model for polyethylene glycol-coated gold nanoparticles of different sizes in adult mice. Nanotoxicology 2016b;10:162-72.

Macey R, Oster G, Zahnley T. Berkeley Madonna User's Guide. available at: http://www.berkeleymadonna.com/BM%20User%27s%20Guide%208.0.2.pdf. p. 1-70. 2009.

Shampine LF, Reichelt MW. The matlab ode suite. SIAM journal on scientific computing 1997;18:1-22.

Soetaert K, Petzoldt T, Setzer RW. Solving differential equations in R: package deSolve. Journal of Statistical Software 2010;33:1-25.

Stewart J (2016). *Calculus (8th edition)*. Cengage Learning.