Solving Delay Differential Equations in S-ADAPT by Method of Steps

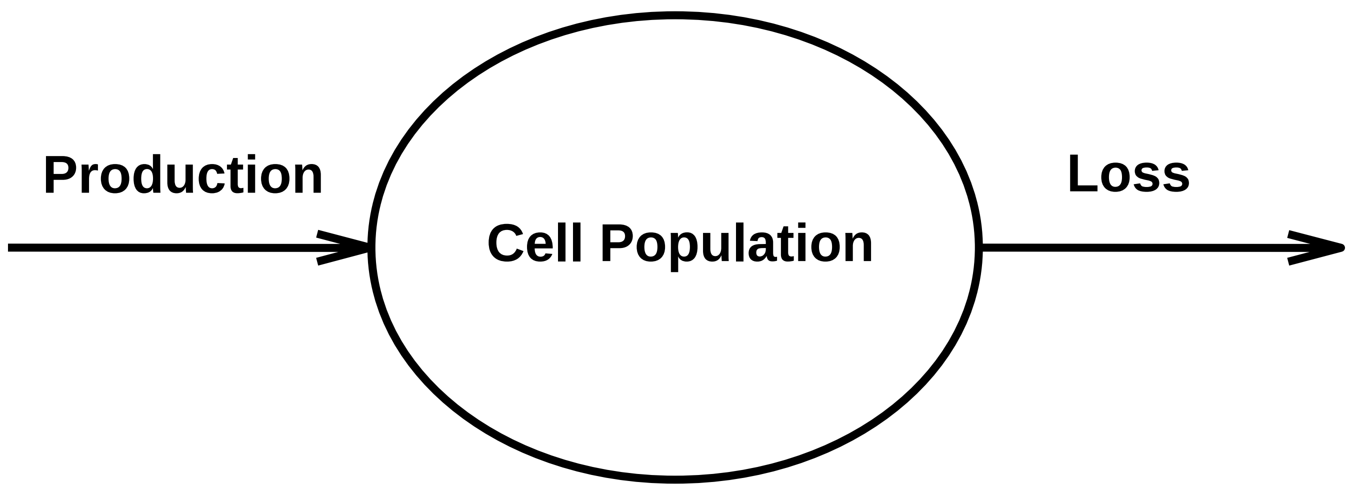
Wojciech Krzyzanski[1]

Robert J. Bauer[2]

[1]Department of Pharmaceutical Sciences, University at Buffalo, Buffalo, NY, USA
[2]ICON Development Solutions, Ellicott City, USA

Cell Populations of Pharmacodynamic Interest

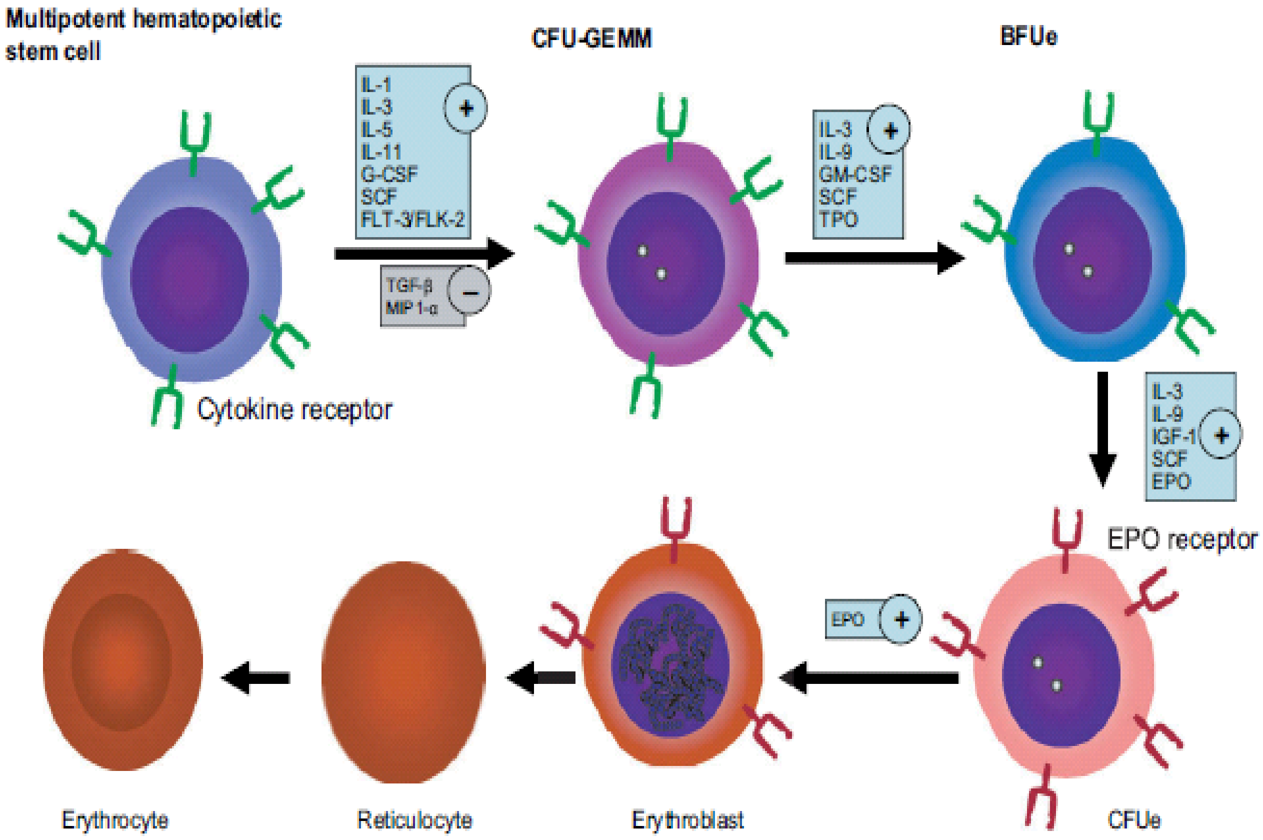
- hematopoietic cells (RBC, WBC, PLT)
- bacteria and viruses
- neoplastic cells
- parasites



mitosis
differentiation
maturation

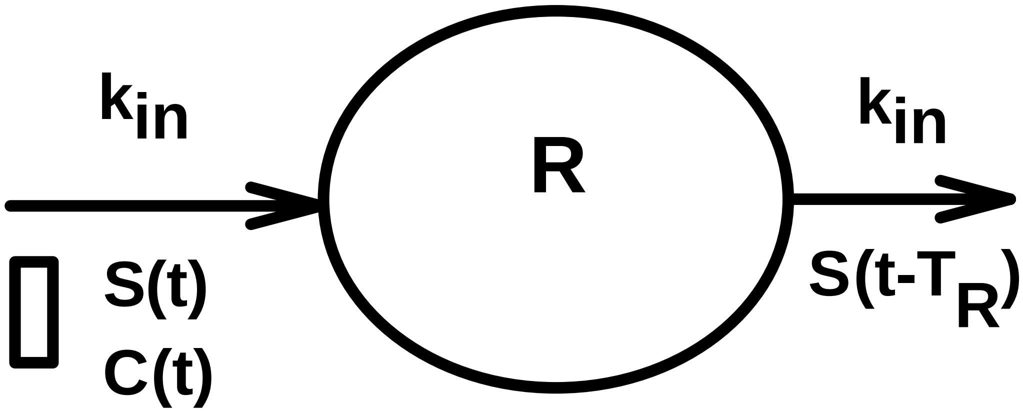
tissue uptake
senescence
random destruction
maturation

Erythropoiesis



Eliott et al., Exp. Hematol. 36:1573 (2008)

Basic Lifespan-based Indirect Response

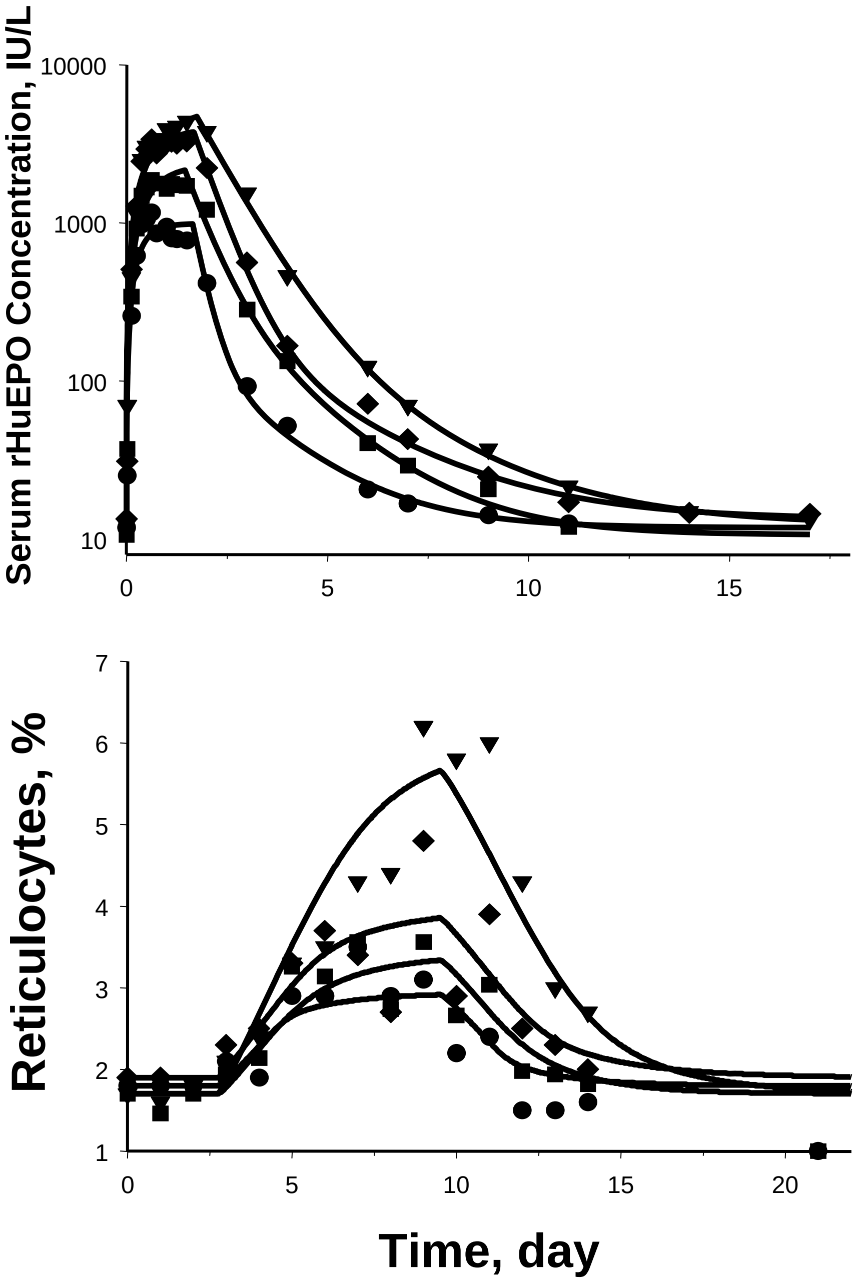


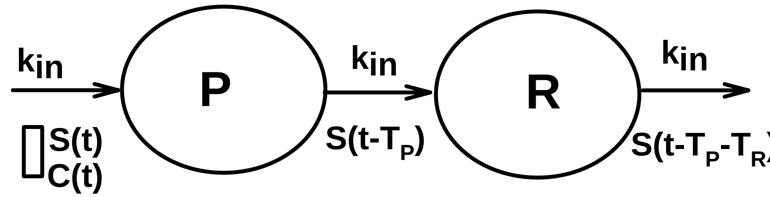
$$\frac{dR}{dt} = k_{in} \cdot S(t) - k_{in} \cdot S(t - T_R)$$

$$S(t) = 1 + \frac{S_{max} C(t)^{\gamma}}{SC_{50}^{\lambda} + C(t)^{\gamma}}$$

Krzyzanski and Jusko, JPB 27: 467 (1999).

Stimulation of Reticulocyte Production by Epoetin Alfa





$$\frac{dR}{dt} = k_{in} \cdot S(t - T_P) - k_{in} \cdot S(t - T_P - T_R)$$

EPO serum concentrations (upper panel) and reticulocyte counts (lower panel) after subcutaneous administration of EPO doses of 450, 900, 1350, and 1800 IU/kg.

Data from K Cheung et al. Clin. Pharmacol. Ther. 64: 412 (1998)

General Form of DDE System

$$\frac{dx}{dt} = f(t, x(t), x(t - T_1), x(t - T_2), \ldots, x(t - T_p)) \quad \text{for } t > 0$$

$x(t) = x_0$, for $t \leq 0$

$x(t)$ = vector of states at time t.
$T_1, \ldots, T_p$ = delay times.
x_0 = state vector in the past.

The rate of change of x depends not only on the current value $x(t)$, but also on the system states before times $T_1, \ldots, T_p$: $x(t-T_1), \ldots, x(t-T_p)$.

DDE Solvers

Algorithms based on the Runge-Kutta Method:

- DELSOL (Fortran)

Willie DR and Baker CTH. DELSOL – a numerical code for the solutions of systems of delay-differential equations. Appl. Numer. Math. 9:223-234 (1992).

- RETARD, RADAR5 (Fortran)

Hairer E, Norsett SP, and Wanner G. Solving Ordinary Differential Equations I: Nonstiff Problems. . Berlin, Springer, 1993.

http://univaq.it/~guglielm/guglielmi_eng.html#SO

- dde23 (MATLAB)

Shampine LF and Thompson S. Solving DDEs in MATLAB. Appl. Num. Math. 37:441-458 (2001).

http://www.radford.edu/~thompson/webddes/index.html

Methods of Steps: Example

$$\frac{dx}{dt} = -x(t-1) \quad \text{for } 0< t \qquad\qquad x(t) = 10 \quad \text{for } t \le 0$$

1st step: Find solution for **0< t <1**:

$0< t <1 \Rightarrow -1 < t-1 < 0 \Rightarrow x(t-1)= 10$

$$\frac{dx}{dt} = -10$$

$$x(t) = 10 - 10t$$

2nd step: Find solution for **1< t <2**:

$1< t <2 \Rightarrow 0 < t-1 < 1 \Rightarrow x(t-1)= 10 - 10(t-1)$

$$\frac{dx}{dt} = -(10 - 10(t-1))$$

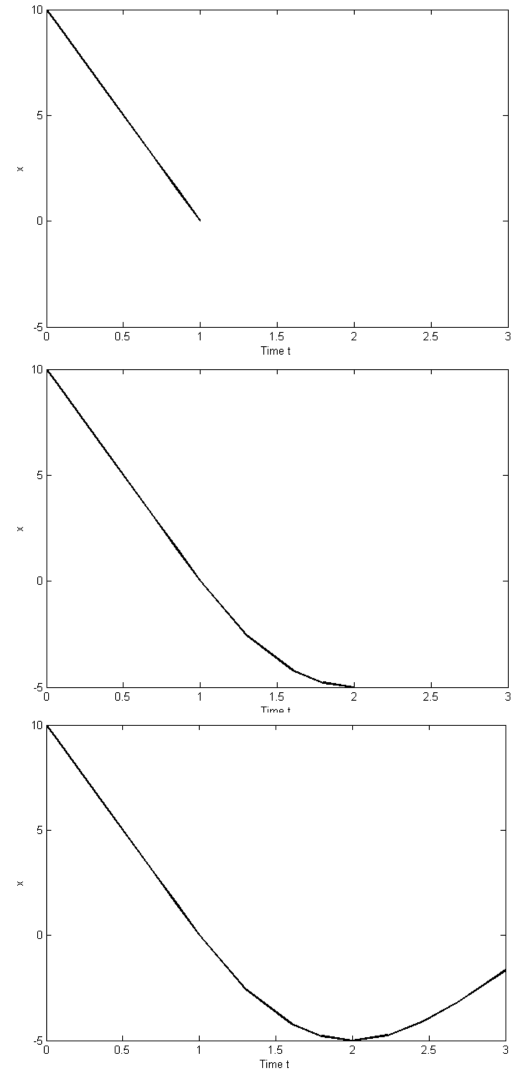
$$x(t) = -10(t-1) + 5(t-1)^2$$

3rd step: Find solution for **2 < t <3**:

$1< t <2 \Rightarrow 1 < t-1 < 2 \Rightarrow x(t-1)= -10(t-2) + 5(t-2)^2$

$$\frac{dx}{dt} = -(-10(t-2) + 5(t-2)^2)$$

$$x(t) = -5 + 5(t-2)^2 - \frac{5}{3}(t-1)^3$$



Methods of Steps: Numerical

$$\frac{dx}{dt} = -x(t-1) \quad \text{for } 0< t \qquad x(t) = 10 \quad \text{for } t \leq 0$$

To find a solution for **0< t <3**: $y_1(t)=x(t)$, $y_2(t)=x(t-1)$

$$\boxed{\frac{dy_1}{dt} = -y_2(t)}$$

$$\frac{dy_2}{dt} = -x(t-2)$$

Since t -2 > 0 for some t, we need to calculate x(t-2): $y_3(t)=x(t-2)$

$$\boxed{\frac{dy_2}{dt} = -y_3(t)}$$

$$\frac{dy_3}{dt} = -x(t-3)$$

Since t -3 < 0 for all 0 < t <3 : x(t-3) = 10 and

$$\boxed{\frac{dy_3}{dt} = -10}$$

Initial conditions: $y_1(0)= 10$, $y_2(0) = 10$, $y_3(0) = 10$

Method of Steps

- **If for a time interval $t_i < t < t_{i+1}$ ("a step") the delay t_{i+1}- T is less than t_i, then delayed state y(t-T) defined by its values for times less than t_i, which makes y(t-T) a "known" variable.**

- **If all delayed variables become "known" over the time interval $t_i < t < t_{i+1}$, then for this time interval the system does not have unknown delay variables and becomes an ODE system.**

- **Methods of steps transforms a system of DDEs into a system of ODEs.**

Driver RD. Ordinary and Delay Differential Equations. New York, Springer-Verlag, 1977.
http://www.radford.edu/~thompson/webddes/tutorial.html

Methods of Steps: NONMEM

Journal of Pharmacokinetics and Pharmacodynamics, Vol. 32, Nos. 5–6, December 2005 (© 2005)
DOI: 10.1007/s10928-005-0019-1

Population Cell Life Span Models for Effects of Drugs Following Indirect Mechanisms of Action[1]

Juan J. Perez-Ruixo,[2,*] Hui C. Kimko,[3] Andrew T. Chow,[3] Vladimir Piotrovsky,[2] Wojciech Krzyzanski,[4] and William J. Jusko[4]

- Delay is introduced to the state variables using ALAG.
- Limited number of ODEs (99).
- Limited number of ALAGs (20).

S-ADAPT

ADAPT

Pharmacokinetic/Pharmacodynamic Systems Analysis

BMSR | ADAPT | DOWNLOAD | INSTALLATION | USER'S GUIDE | CITATIONS | MODEL LIBRARY

S-ADAPT

S-ADAPT is a version of ADAPT II Release 3 that contains an augmented interface as well as additional simulation and optimization abilities. Some features include: command-line interface for entering parameters and executing commands, screen forms and editors for entering data and parameters, evaluation of algebraic expressions at the command line. It also performs parametric population analysis, including maximum likelihood estimation (via the EM algorithm with sampling as implemented in the Monte Carlo Expectation Maximization (MCPEM) algorithm), as well as Bayesian estimation. The program runs on Windows NT through XP and Linux., with various Fortran compilers including Compaq Visual Fortran 6.x, Intel 9.x-11.x and g77.

University of Southern California

Biomedical Simulations Resource

Biomedical Engineering

http://bmsr.usc.edu/Software/BMSRsoftware.html

Implementation of Methods of Steps in S-ADAPT

$$\frac{dx}{dt} = \sum_{j=1}^{m_b} b_j(t) + \sum_{k=1}^{n_r} r_k(t) + h(t, x(t), x(t-T_1), ..., x(t-T_p))$$

Vector of all possible delay times:

$$\tau_i = i_1 T_1 + i_2 T_2 + \ldots + i_p T_p$$

Vector of all possible delay states:

$$y_{i_1 i_2 \ldots i_p}(t) = \begin{cases} x(t - i_1 T_1 \ldots - i_p T_p), & \text{if } i_1 T_1 + \ldots + i_p T_p < t \leq t_{last} \\ x_0, & \text{if } t \leq i_1 T_1 + \ldots + i_p T_p \end{cases}$$

S-ADAPT:

- Determines the derivatives of $y_{i1\ldots ip}$ from the original DDE systems.
- Creates a new ODE systems for $y_{i1\ldots ip}$.
- Uses LSODA to calculate the solution.
- Reports $y_{0\ldots 0}$ as the solution to the DDE system.

Implementation of Methods of Steps in S-ADAPT

- Mapping integer combinations of the delay times.
- Filtering and sorting delay times.
- Creating the ODE system.
- Bolus and zero-order infusion input.
- Handling the time dependent DDE systems.

S-ADAPT Library DDE Routines

Tdroutines.for

```
subroutine tdmapping1(td,nn,tstop,tdc,xvalmax)

subroutine tetd_process(gg)

subroutine xdtd(i,x)

Subroutine DOSE_TIME_DELAY(td,nntd,nbb,nrr)
```

Tdelay.inc

This files contains global variables specific to DDE routines that are not in globals.inc

S-ADAPT User Defined DDE Routines

Model_Del.for

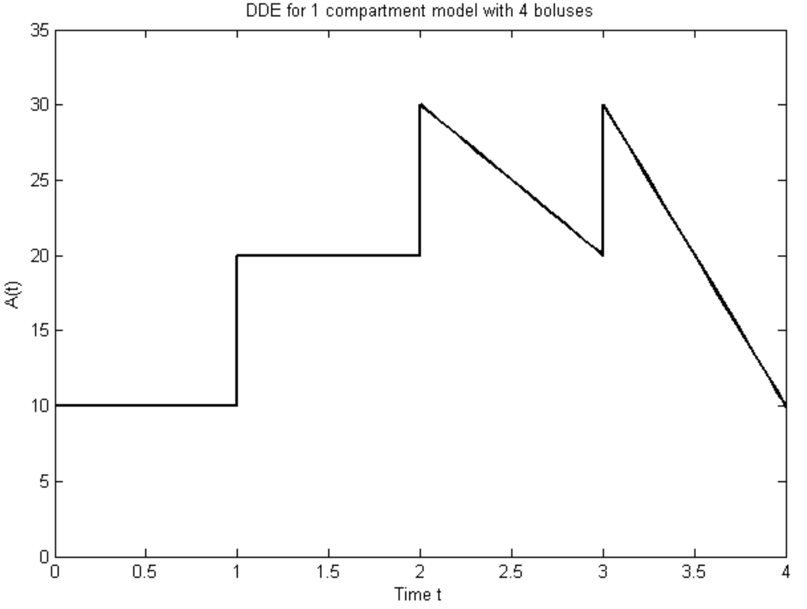
```
Subroutine DEL_DIFFEQ(T,X,XP,RD)

Subroutine DEL_OUTPUT(Y,T,X)

Subroutine SYMBOL

Subroutine DEL_VARMOD(V,T,X,Y,J)
```

Multiple IV Boluses to One Compartment with Delayed Elimination

$$\frac{dA}{dt} = \sum_{i=1}^{m} Dose \cdot \delta(t - t_i) - k \cdot A(t - T) \qquad \text{for } t > 0$$

$$A(t) = A_0, \quad \text{for } t \leq 0$$

D = 10 dose
Bolus times
$t_1 = 0$,
$t_2 = 1$,
$t_3 = 2$,
$t_4 = 3$
k = 1 elimination rate constant.
T = 2 delay
$A_0 = 0$ past



MATLAB dde23 generated solution

User Defined Model

```
Subroutine DEL_DIFFEQ(T,X,XP,RD)
Implicit None

Include 'globals.inc'
Include 'model.inc'
include 'tdelay.inc'

Real*8 T,X(*),XP(*),RD(*)
real*8 k

TD(1)=P(1)
k    =P(2)

XP(1)=-k*XD(1,1)

Return
End
```

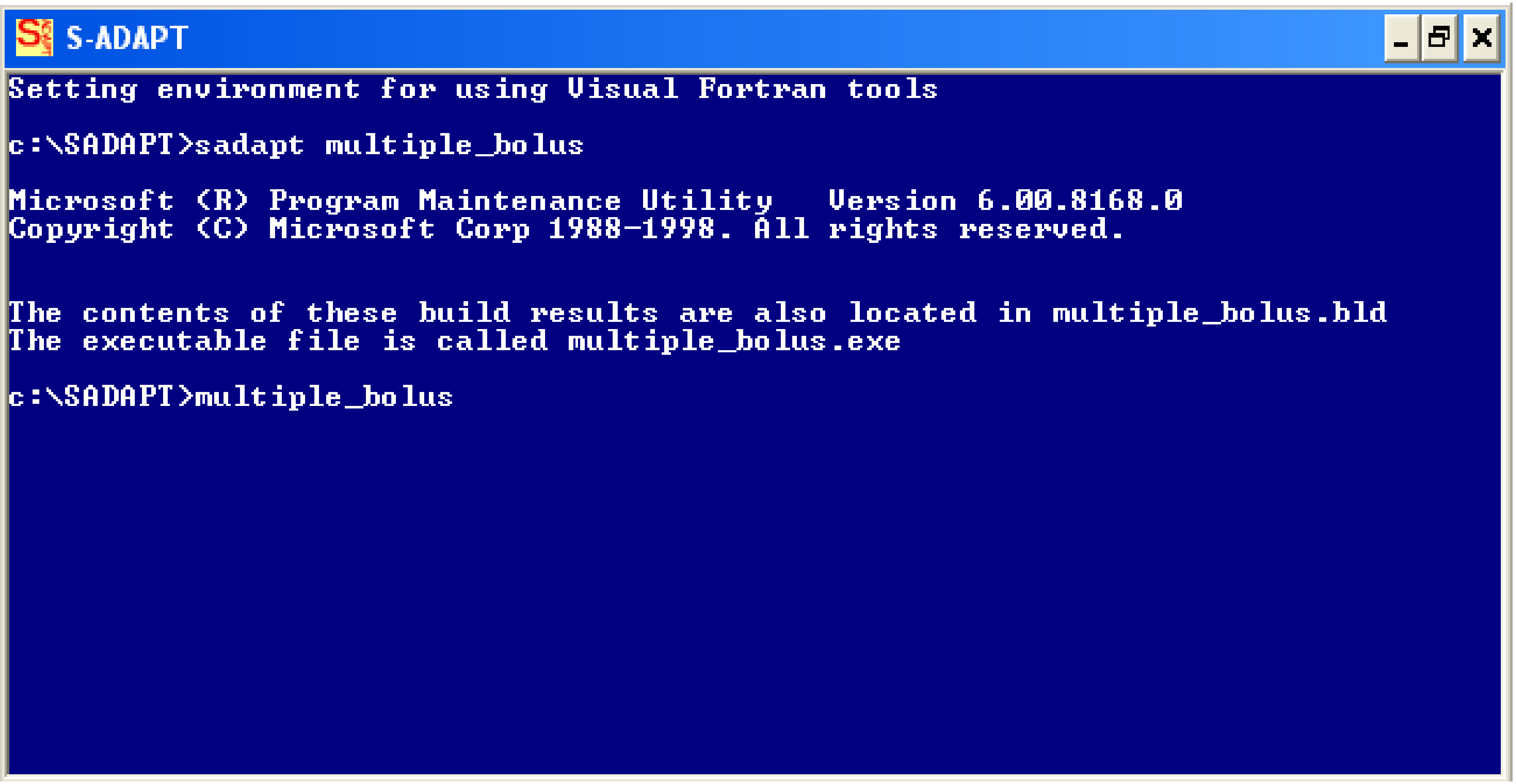
Output Subroutine

```
Subroutine DEL_OUTPUT(Y,T,X)
Implicit None

Include 'globals.inc'
Include 'model.inc'
include 'tdelay.inc'

Real*8 Y(*),T,X(*)
integer k

TD(1)=P(1)

X0(1)=P(3)

noeqs=1

Y(1)=X(1)

Return
End
```

Symbol Subroutine

```
Subroutine SYMBOL
Implicit None

Include 'globals.inc'
Include 'model.inc'
include 'tdelay.inc'
character*60 descr
common /descr/  Descr

NDE=1            ! Enter # of Diff. Eqs.
NDEL=1           !Enter # of time delays
NDEqs   = NDE*(NTD+1)
NSParam = 3    ! Enter # of Sys. Param.
NVparam = 0    ! Enter # of Var. Param.
Ieqsol  = 1    ! Model type.
NTPARAM = 0    ! Enter # of Tran.Param.
Descr   = ' Multi-Bolus'

Psym(1)='T1'
psym(2)='k'
Psym(3)='X0'

Return
End
```

Model Compilation in S-ADAPT



S-ADAPT Table for Simulations

```
Page             1     @SESS CALC
0:PROCESS                   1:INFORMATION

1:DEFINE ROOT NAME          Root name is
2:EDIT DOSE                 Edit Dose information
3:MODEL PARAMETERS          Modify model parameters
4:PLOT PARAMETERS           Set up plot parameters
5:PLOT                      plot the curve
6:PRINT PLOT                Print the plot
7:SAVE CONFIG               store configuration information in .s00
8:PRINT CONFIG              Print configuration file .s00
9:WRITE SIMULATION          Write simulation values to .wrt
10:RESTORE CONFIG           obtain configuration info from .s00
11:BROWSE                   Browse directory for *.dat files
12:STOP                     Exit Adapt program
13:COMMAND                  Enter any other command
14:
15:
16:
/EX=exit /RED to redisplay /HELP for more commands
14:0:
```

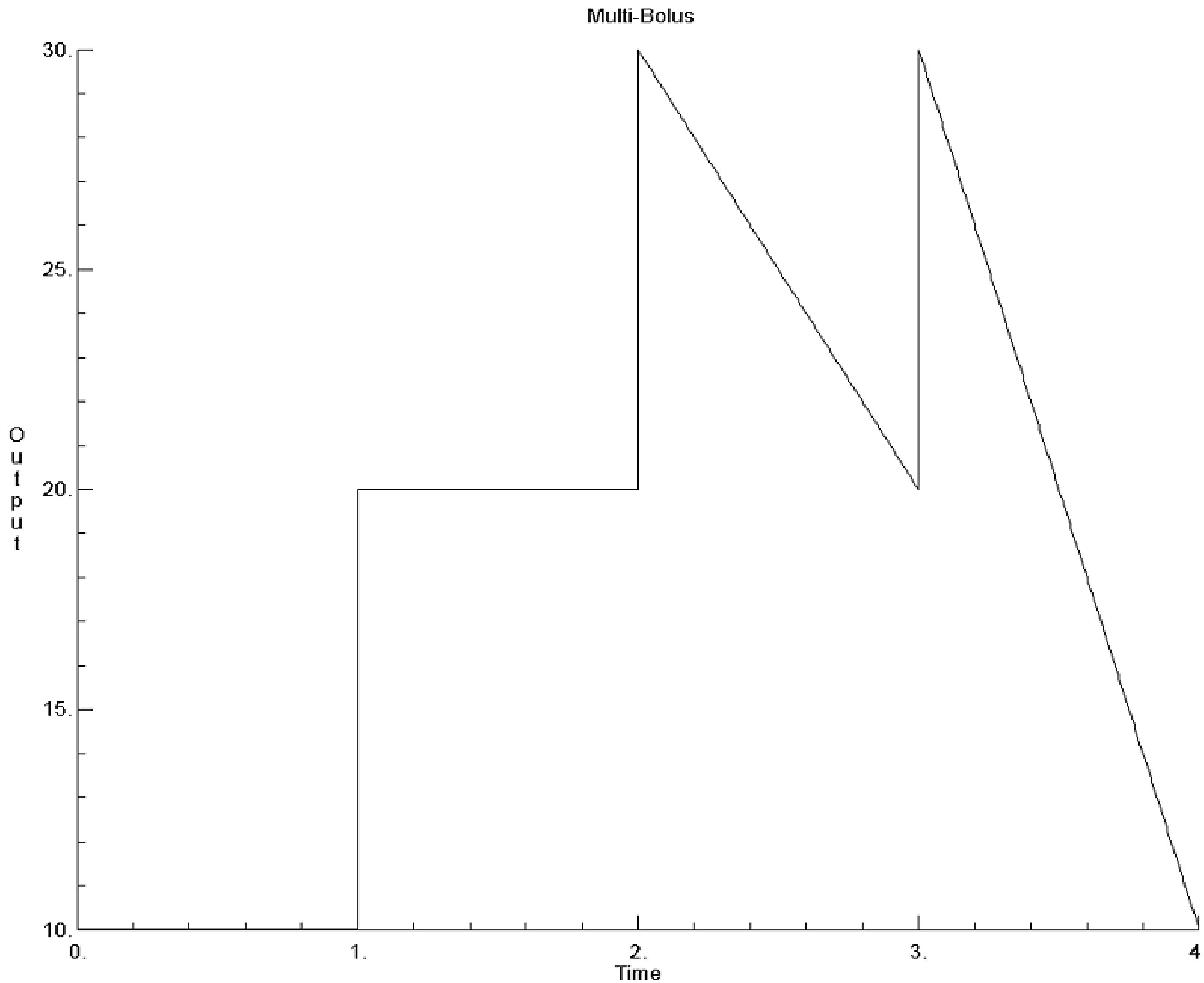
Model Input

Page	1	Dosing Data					
Time	Period	Dose	Repeat	Counts	N	S	E
0.000000	0.000000	10.00000	1.000000	4	0	1	1

Page	1		Model Parameters	
	Parameter	R	Lower	Upper
T1	2.0000000	Y		
k	1.0000000	Y		
X0	0.0000000	Y		

Page	1	Plot Parameters	
Tstart	0.000	Tstop	4.000

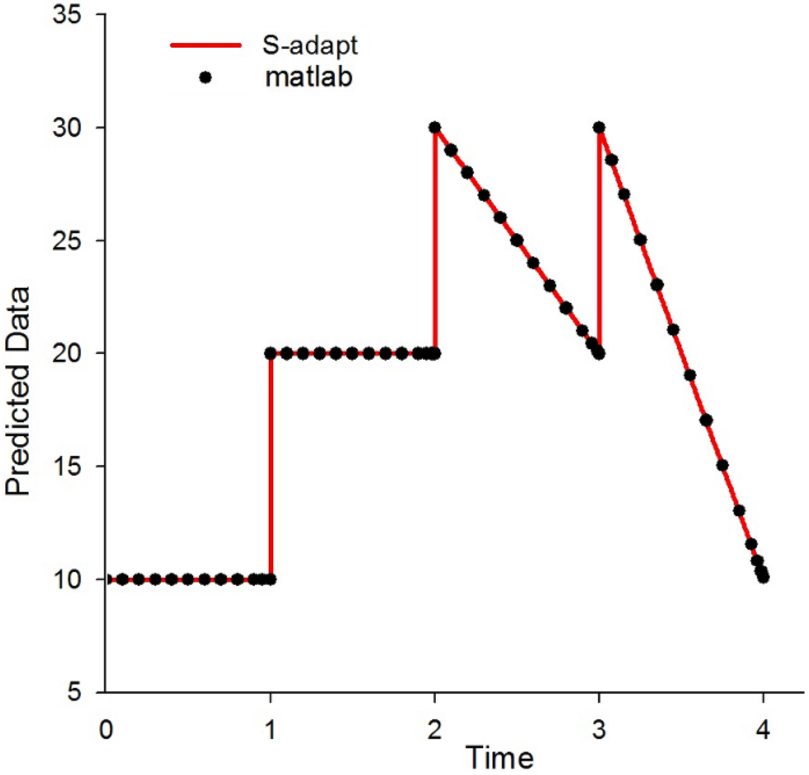
Model Output



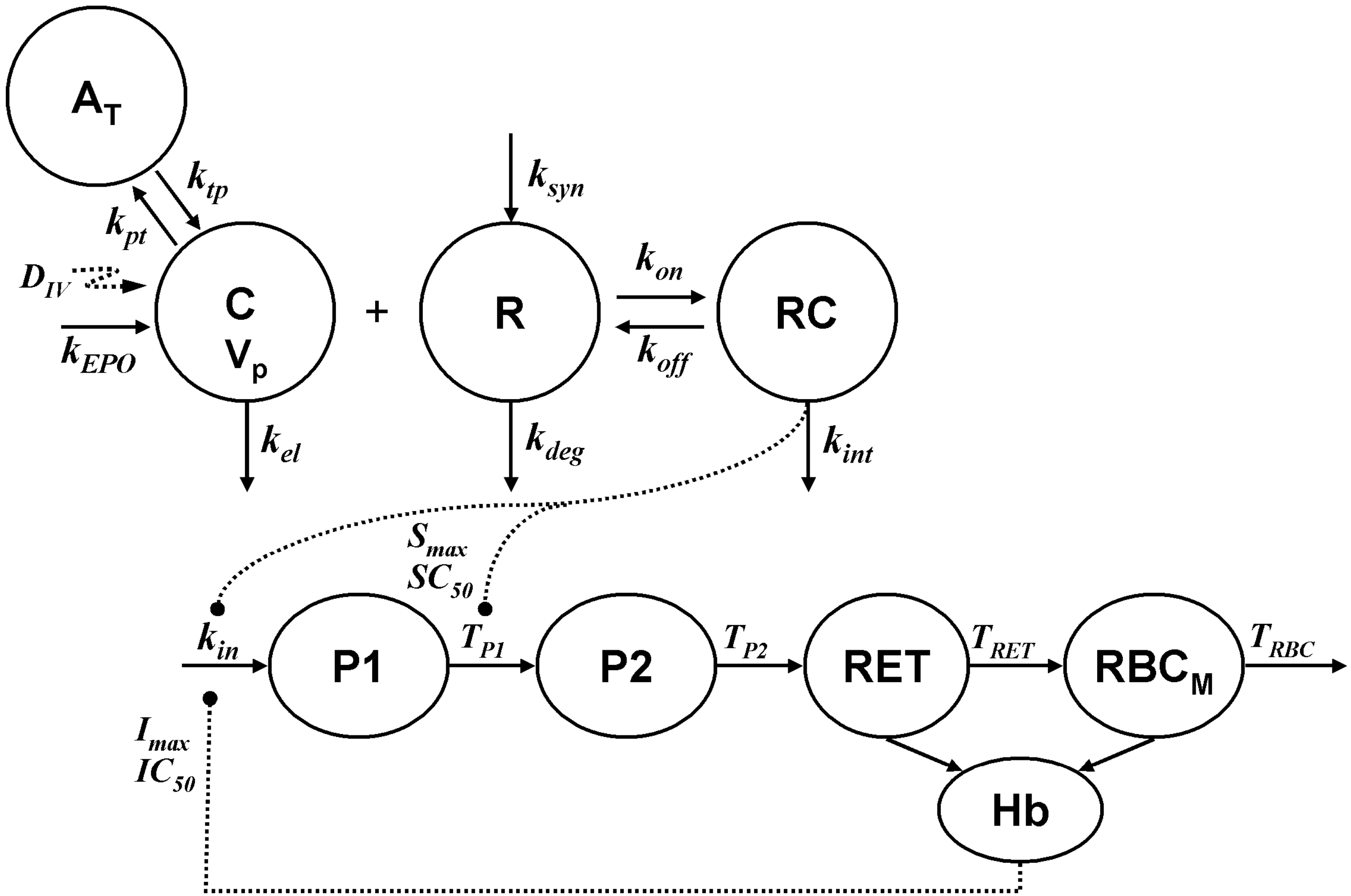
S-ADAPT vs. MATLAB

Time	S-ADAPT	MATLAB
0	10.0000	10.0000
0.5	10.0000	10.0000
1.0	10.0000	10.0000
1.0	20.0000	20.0000
1.5	20.0000	20.0000
2.0	20.0000	20.0000
2.0	30.0000	25.0000
2.5	25.0000	25.0000
3.0	20.0000	20.0000
3.0	30.0000	25.0000
3.5	20.0000	20.0052
4.0	10.0000	10.0052

MATLAB: RelTol=10^{-4} AbsTol=10^{-6}
S-ADAPT: RelTol=10^{-8} AbsTol=10^{-8}



PK/PD Model of rHuEPO Stimulatory Effect on RBC Production



S. Woo et al., J. Pharmacokinet. Pharmacodyn. 34:849-868 (2007).

Model Equations

$$\frac{dC}{dt} = k_{EPO} - k_{on} \cdot R \cdot C + k_{off} \cdot RC - \left(k_{el} + k_{pt}\right) \cdot C + k_{tp} \cdot A_T / V_p$$

$$\frac{dA_T}{dt} = k_{pt} \cdot C \cdot V_p - k_{tp} \cdot A_T$$

$$\frac{dR}{dt} = k_{syn} - k_{on} \cdot R \cdot C + k_{off} \cdot RC - k_{deg} \cdot R$$

$$\frac{dRC}{dt} = k_{on} \cdot R \cdot C - \left(k_{off} + k_{int}\right) \cdot RC$$

$$\frac{dRET}{dt} = k_{in} \cdot S(t - T_{P1} - T_{P2}) \cdot S(t - T_{P2}) \cdot I(t - T_{P1} - T_{P2}) -$$
$$k_{in} \cdot S(t - T_{P1} - T_{P2} - T_{RET}) \cdot S(t - T_{P2} - T_{RET}) \cdot I(t - T_{P1} - T_{P2} - T_{RET})$$

$$\frac{dRBC_M}{dt} = k_{in} \cdot S(t - T_{P1} - T_{P2} - T_{RET}) \cdot S(t - T_{P2} - T_{RET}) \cdot I(t - T_{P1} - T_{P2} - T_{RET}) -$$
$$k_{in} \cdot S(t - T_{P1} - T_{P2} - T_{RET} - T_{RBC}) \cdot S(t - T_{P2} - T_{RET} - T_{RBC}) \cdot I(t - T_{P1} - T_{P2} - T_{RET} - T_{RBC})$$

$$S(t) = 1 + \frac{S_{max} \cdot RC(t)}{SC_{50} + RC(t)} \qquad I(t) = 1 - \frac{I_{max} \cdot \Delta Hb(t)}{IC_{50} + \Delta Hb(t)} \qquad Hb(t) = MCH \cdot RBC(t)$$

S-ADAPT Del_Diffeq and Del_Otput

```
TD(1)=P(1)
TD(2)=P(2)
TD(3)=P(3)
TD(4)=P(4)
TRBC =P(5)
kon=P(7)
koff=P(8)
kel=P(9)
kpt=P(10)
ktp=P(11)
```

```
Vp=P(12)
kint=P(13)
Smax=P(14)
SC50=P(15)
Imax=P(16)
IC50=P(17)
MCH=P(18)
C0=P(19)
R0=P(20)
kdeg=P(21)
RBC0=P(22)
```

```
TRET=TD(4)-TD(2)
RET0=TRET*RBC0/(TRET+TRBC)
RBCM0=RBC0-RET0
Hb0=MCH*RBC0
AT0=kpt*C0*Vp/ktp
RC0=kon*R0*C0/(koff+kint)
kEPO=kel*C0*Vp+kint*RC0*Vp
ksyn=kdeg*R0+kint*RC0
kin=RET0/(TRET*(1+Smax*RC0/(SC50+RC0))**2)
```

```
C   =X(1)/Vp
AT  =X(2)
RR   =X(3)
RC  =X(4)
RET =X(5)
RBCM=X(6)
```

```
S1 = 1+Smax*XD(4,1)/(SC50+XD(4,1))
S2 = 1+Smax*XD(4,2)/(SC50+XD(4,2))
S3 = 1+Smax*XD(4,3)/(SC50+XD(4,3))
S4 = 1+Smax*XD(4,4)/(SC50+XD(4,4))
S0 = 1+Smax*RC0/(SC50+RC0)
```

```
X0(1)=C0*Vp
X0(2)=AT0
X0(3)=R0
X0(4)=RC0
X0(5)=RET0
X0(6)=RBCM0
```

```
I1 = 1-Imax*(MCH*(XD(5,1)+XD(6,1))-HB0)/(IC50+(MCH*(XD(5,1)+XD(6,1))-HB0))
I3 = 1-Imax*(MCH*(XD(5,3)+XD(6,3))-HB0)/(IC50+(MCH*(XD(5,3)+XD(6,3))-HB0))
```

```
XP(1)=kEPO-kon*C*RR*Vp+koff*RC*Vp-(kel+kpt)*C*Vp+ktp*AT
XP(2)=kpt*C*Vp-ktp*AT
XP(3)=ksyn-kon*C*RR+koff*RC-kdeg*RR
XP(4)=kon*C*RR-(koff+kint)*RC
XP(5)=kin*S1*S2*I1-kin*S3*S4*I3
XP(6)=kin*S3*S4*I3-kin*S0*S0
```

S-DAPT Dosing and Parameters

Page	1	Dosing Data					
Time	Period	Dose	Repeat	Counts	N	S	E
0.000000	0.000000	92.51000	0.000000	1	0	1	1

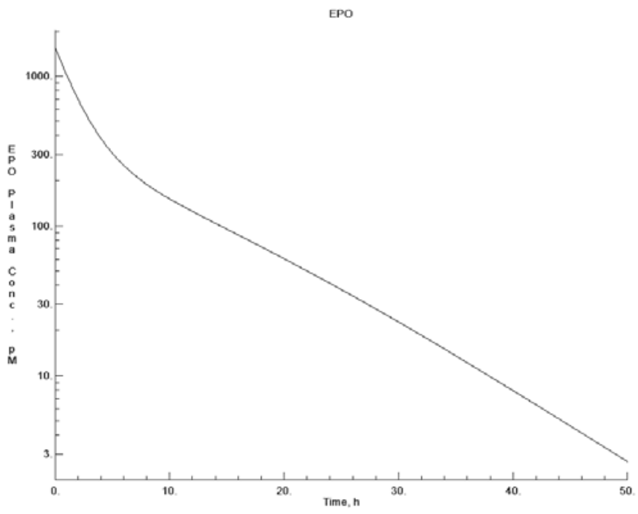
Page	1
	Parameter
T1	76.570000
T2	33.600000
T3	148.90000
T4	105.93000
TRBC	1440.0000
kon	1.13200000E-02
koff	1.2970000
kel	0.22560000
kpt	0.20920000
ktp	0.17210000
Vp	5.69400000E-02
kint	0.82280000
Smax	3.4800000
SC50	1.7000000
Imax	1.0000000
IC50	1.7000000

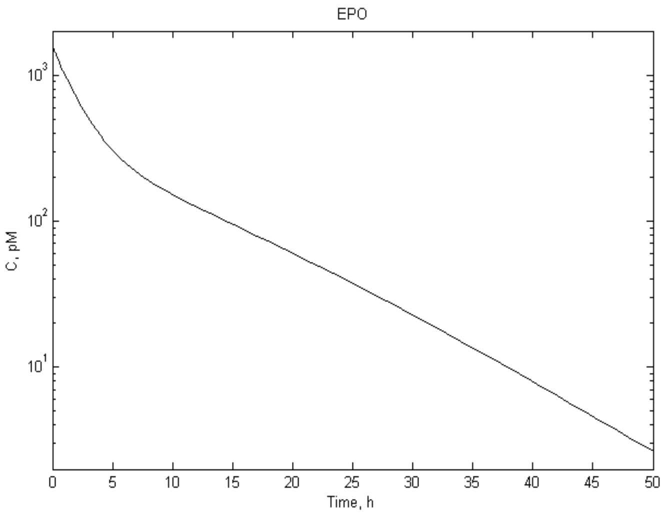
Page	2
	Parameter
MCH	20.000000
C0	0.0000000
R0	63.200000
kdeg	0.11330000
RBC0	6.1280000

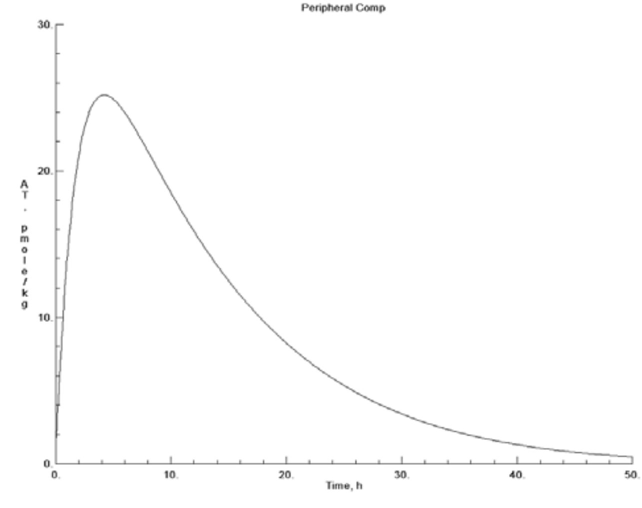
C and A_T

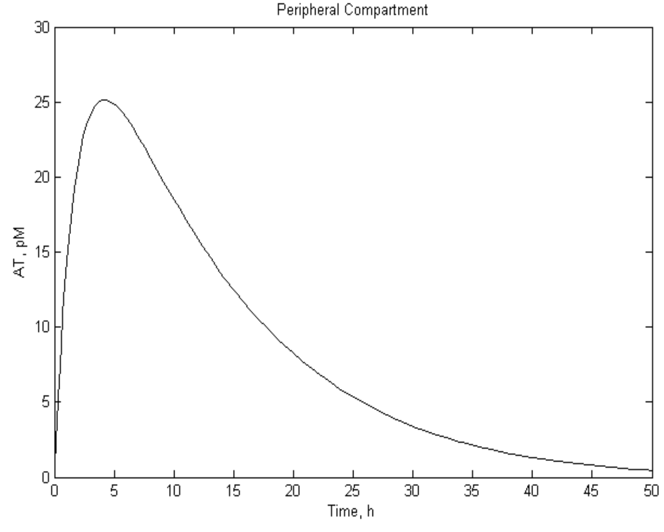
S-ADAPT

MATLAB



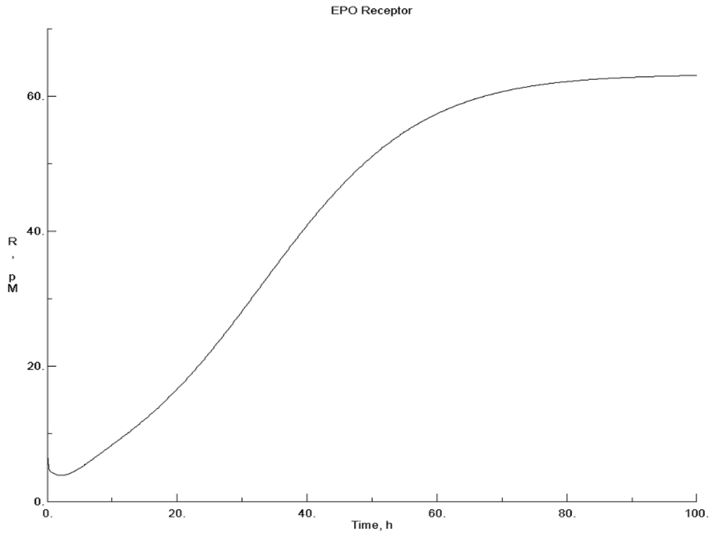


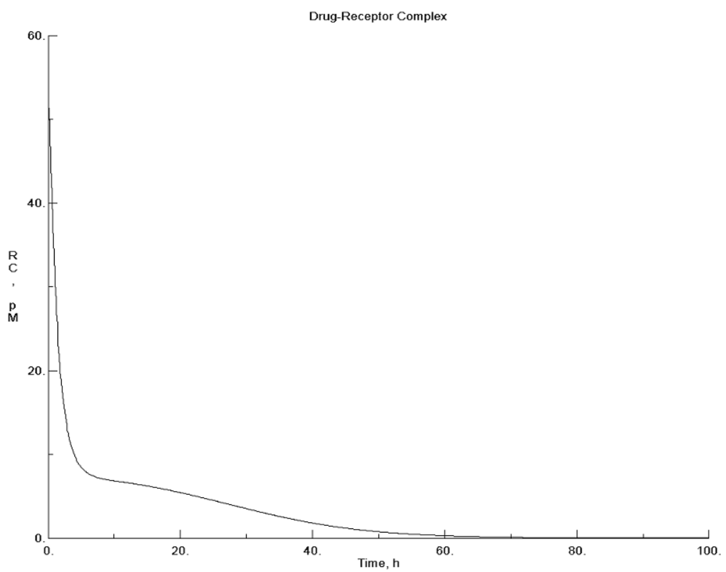




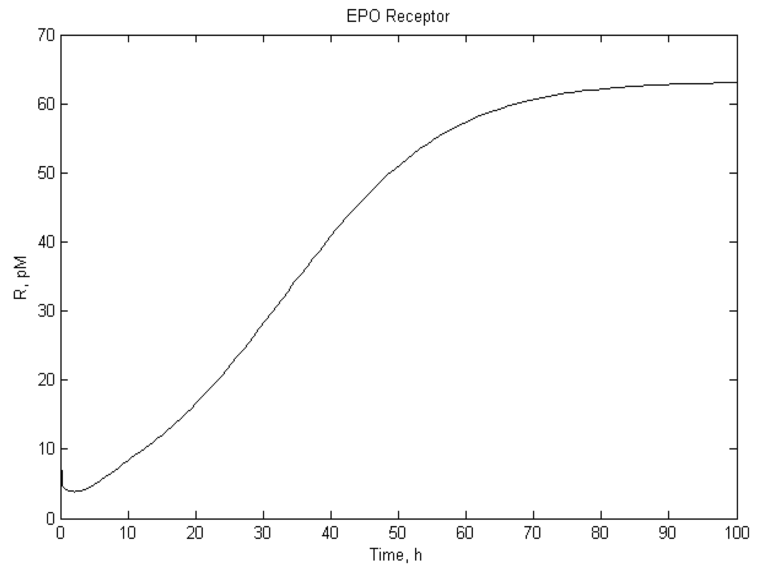
R and RC

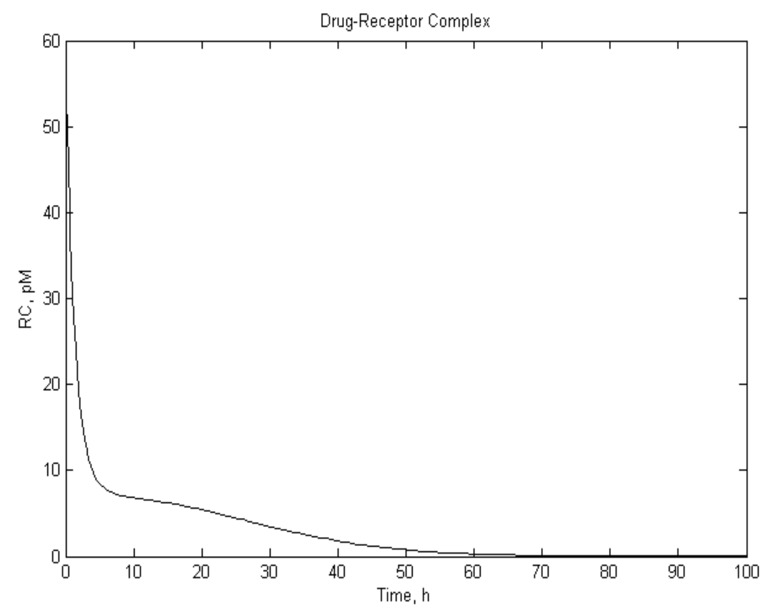
S-ADAPT





MATLAB

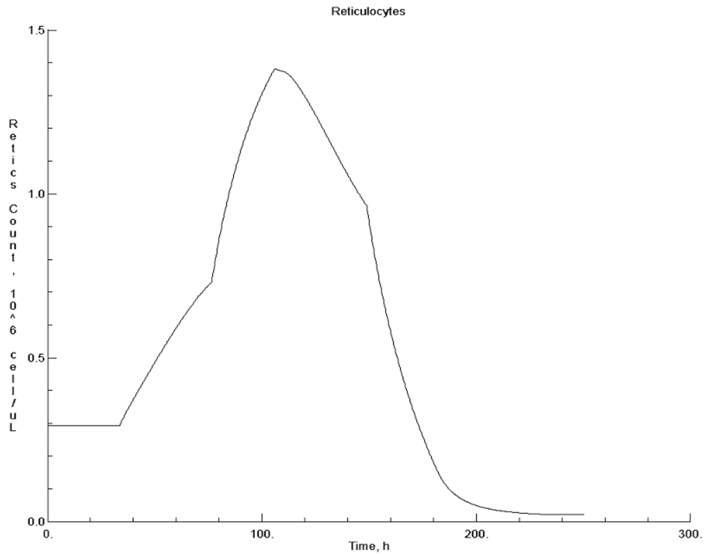


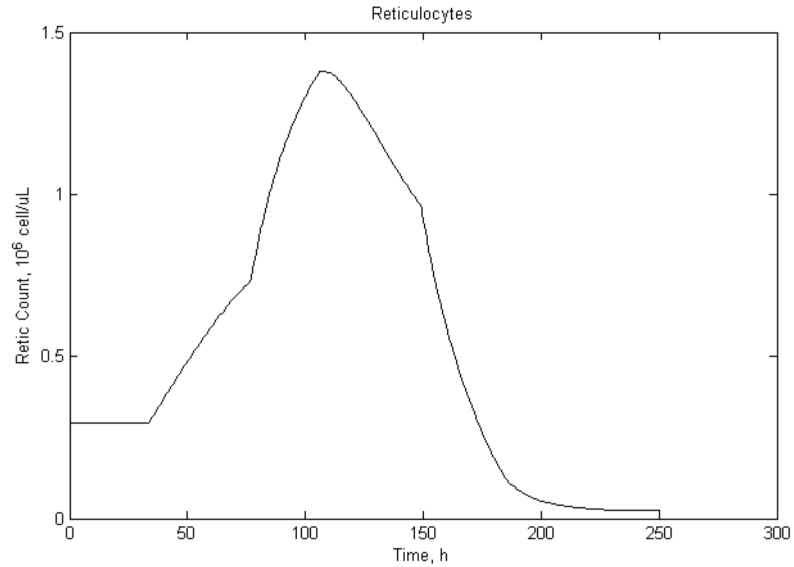


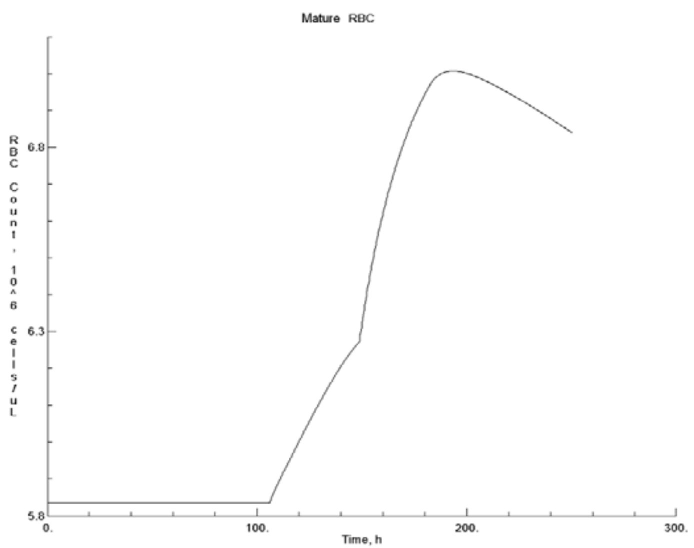
RET and RBC_M

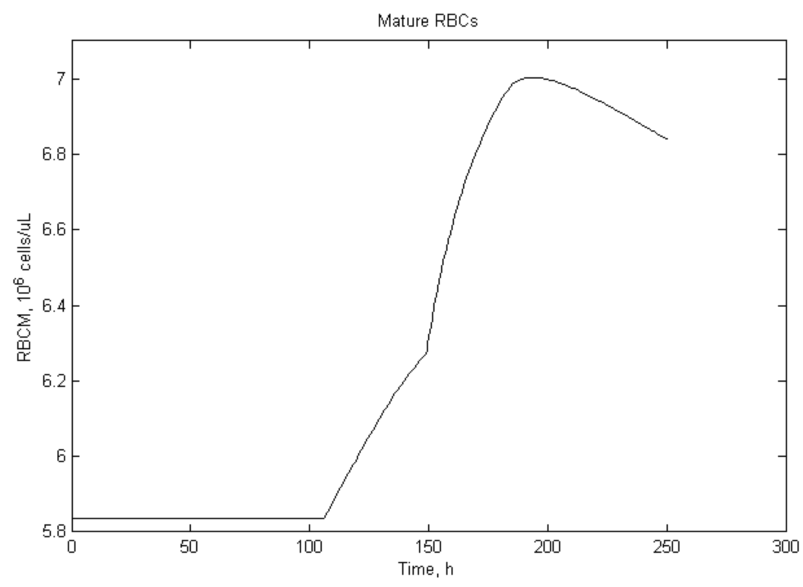
S-ADAPT

MATLAB









Conclusions

- The methods of steps transforms a system of DDEs into a systems of ODEs.
- S-ADAPT implementation of the methods of steps allows solving DDEs with a constant past and bolus inputs.
- S-ADAPT solutions are identical with MATLAB dde23 generated solutions given the same tolerance levels for numerical error.
- S-ADAPT is the first program designed for population PK/PD analysis that is capable of solving arbitrary DDE models

Acknowledgments

- Aman Singh, Department of Pharmaceutical Sciences, University at Buffalo.
- NIH Grant GM57980 from the National Institutes of General Medical Sciences.

Please e-mail wk@buffalo.edu for

- S-ADAPT DDE solver routines.
- S-ADAPT codes for presented examples.