Application of DDDEs for Modeling Absorption Delays

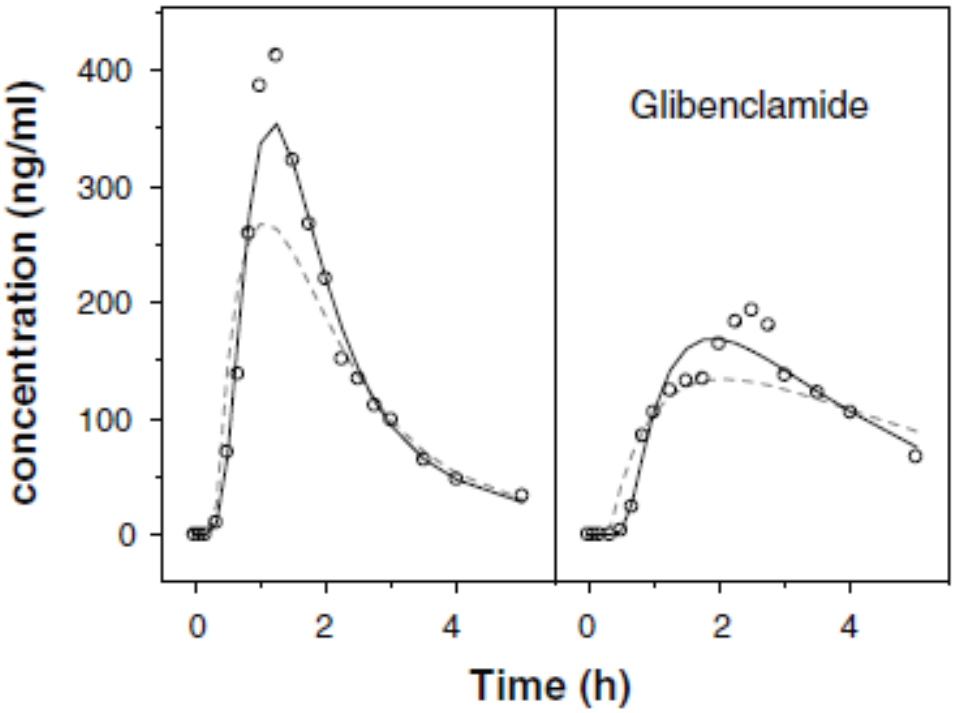
Wojciech Krzyzanski, PhD

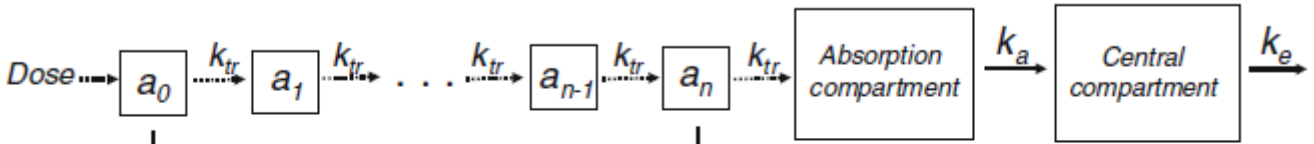
Rosa's Webinar

August 12 2026

University at Buffalo

Gamma Distributed Absorption PK Model for Glibenclamide





$$\frac{da_0}{dt} = -k_{tr} \cdot a_0 \qquad \cdots \qquad \frac{da_n}{dt} = k_{tr} \cdot a_{(n-1)} - k_{tr} \cdot a_n$$

$$a_n(t) = F \cdot \text{Dose} \cdot \frac{(k_{\text{tr}} \cdot t)^n}{n!} \cdot e^{-k_{\text{tr}} \cdot t}$$

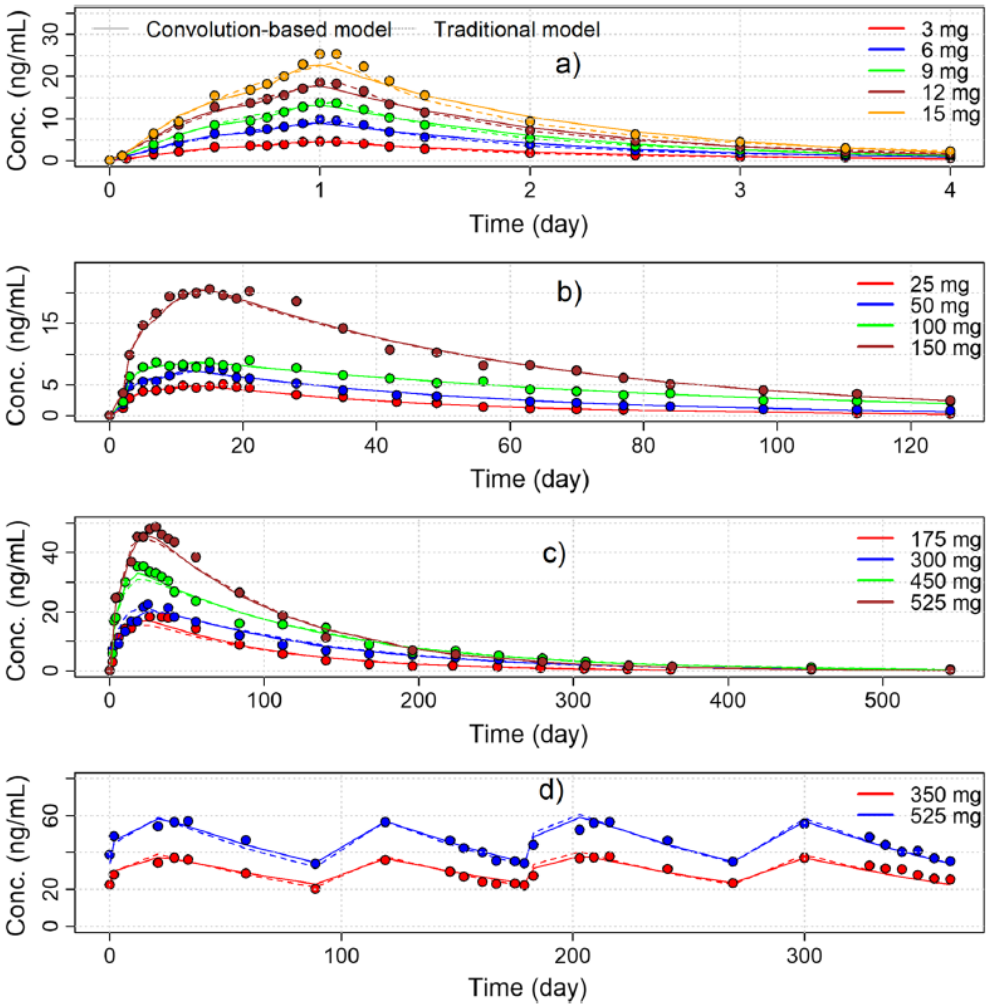
$$\frac{dAa}{dt} = \text{Dose} \cdot F \cdot k_{\text{tr}} \cdot \frac{(k_{\text{tr}} \cdot t)^n \cdot e^{-k_{\text{tr}} \cdot t}}{\sqrt{2\pi} \cdot n^{n+0.5} \cdot e^{-n}} - k_a \cdot A_a$$

Individual plasma concentrations for two subjects after an oral dose of glibenclamide, a sulfonylurea antidiabetic drug. The time courses exhibit a visible lag time before the onset of the data.

$k_{tr} \cdot a_n(t)$ is the gamma p.d.f.

Savic et al., JPKPD 34:711-726 (2007).

Mixture Weibull Absorption PK Model for Paliperidone



$$r(t) = 1 - \left(FF \cdot e^{-\left(\left(\frac{t}{TD}\right)^{ss}\right)} + (1 - FF) \cdot e^{-\left(\left(\frac{t}{TD1}\right)^{ss1}\right)} \right)$$

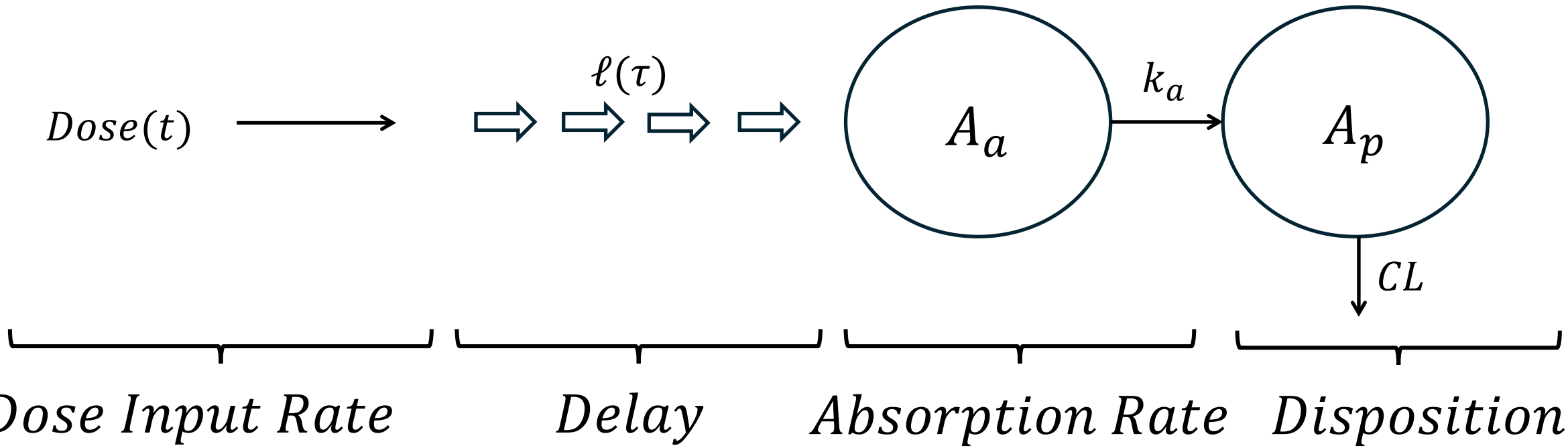
$$\frac{dAp(t)}{dt} = F \times Dose \times \frac{dr(t)}{dt} - kel \times Ap$$

r(t) is the c.d.f. for the mixture distribution of two Weibull distributions.

Mean plasma concentrations of an antipsychotic drug paliperidone in schizophrenic patients following single oral doses of extended-release formulation a), single dose injections of long-acting formulations b) and c), and repeated dose of long-acting formulations injections every three months.

Gomeni and Bresolle-Gomeni, JPKPD 50:89-96 (2023).

ODE Absorption PK Model



$$Dose(t) = \sum_{i=1}^{N_{dose}} F\mathrm{Dose}_i \delta(t - t_i)$$

$$\frac{dA_a}{dt} = \sum_{i=1}^{N_{dose}} F\mathrm{Dose}_i \ell(t - t_i) - k_a A_a$$

$$\frac{dA_p}{dt} = k_a A_a - \frac{CL}{V} A_p$$

Properties of Convolution Integral

$$f(t) * g(t) = \int_0^t f(t-\tau)g(\tau)d\tau = \int_0^t f(\tau)g(t-\tau)d\tau = g(t) * f(t)$$

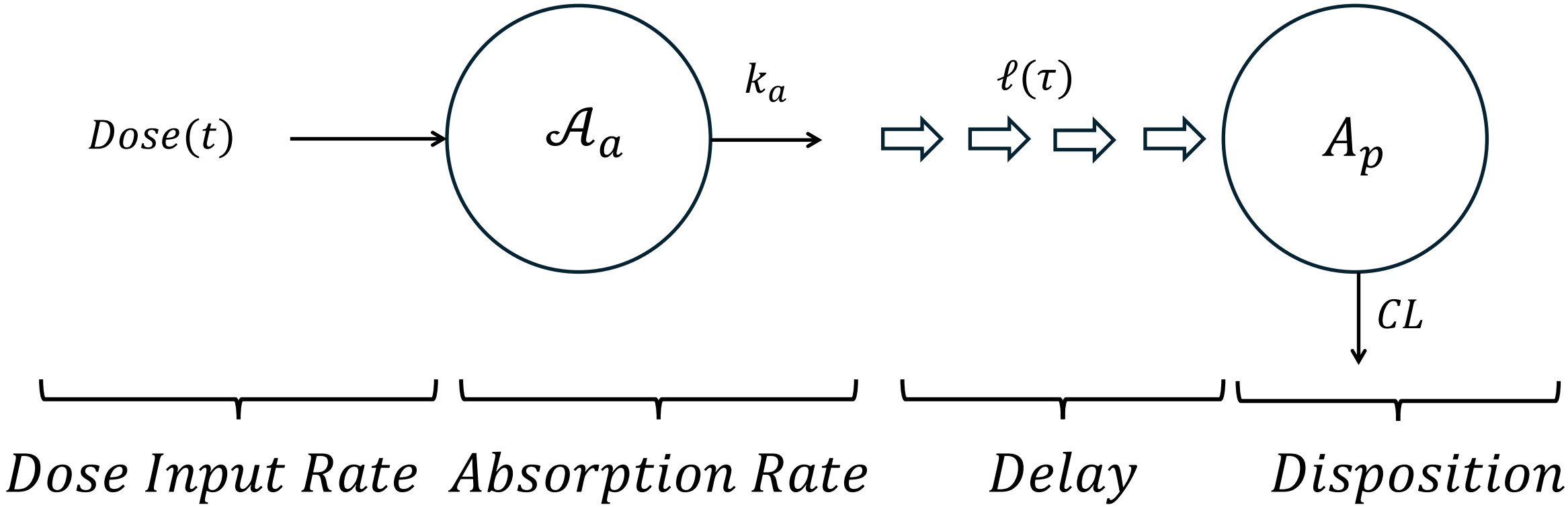
$$\sum_{i=1}^{N_{dose}} F\mathrm{Dose}_i \ell(t-t_i) = Dose(t) * \ell(t)$$

$$A_a(t) = Dose(t) * \ell(t) * \exp(-k_a t)$$

$$\frac{dA_p}{dt} = k_a Dose(t) * \ell(t) * \exp(-k_a t) - \frac{CL}{V} A_p$$

$$\frac{dA_p}{dt} = k_a \underbrace{Dose(t) * \exp(-k_a t)}_{\mathcal{A}_a(t)} * \ell(t) - \frac{CL}{V} A_p$$

Distributed Delay Differential Equation Absorption PK Model



$$\frac{d\mathcal{A}_a}{dt} = Dose(t) - k_a\mathcal{A}_a \qquad \frac{dA_p}{dt} = k_a\mathcal{A}_a(t) * \ell(t) - \frac{CL}{V}A_p$$

Objectives

- **To demonstrate how to implement the absorption PK models with distributed lag times in available DDDE solvers**
  - Phoenix 8.6 delay()
  - NONMEM 7.6 ADVAN16

- **To assess differences in performance between ODE and DDDE models using simulated data**

Distributed Delay Differential Equations

A system of differential equations where at least one state variable is convoluted with the delay kernel $\boldsymbol{\ell(t)}$ is called the system of distributed delay differential equations (DDDEs):

$$\frac{dX}{dt} = g(t, X(t), X(t-T), X * \ell(t)) \text{ for } t > 0$$

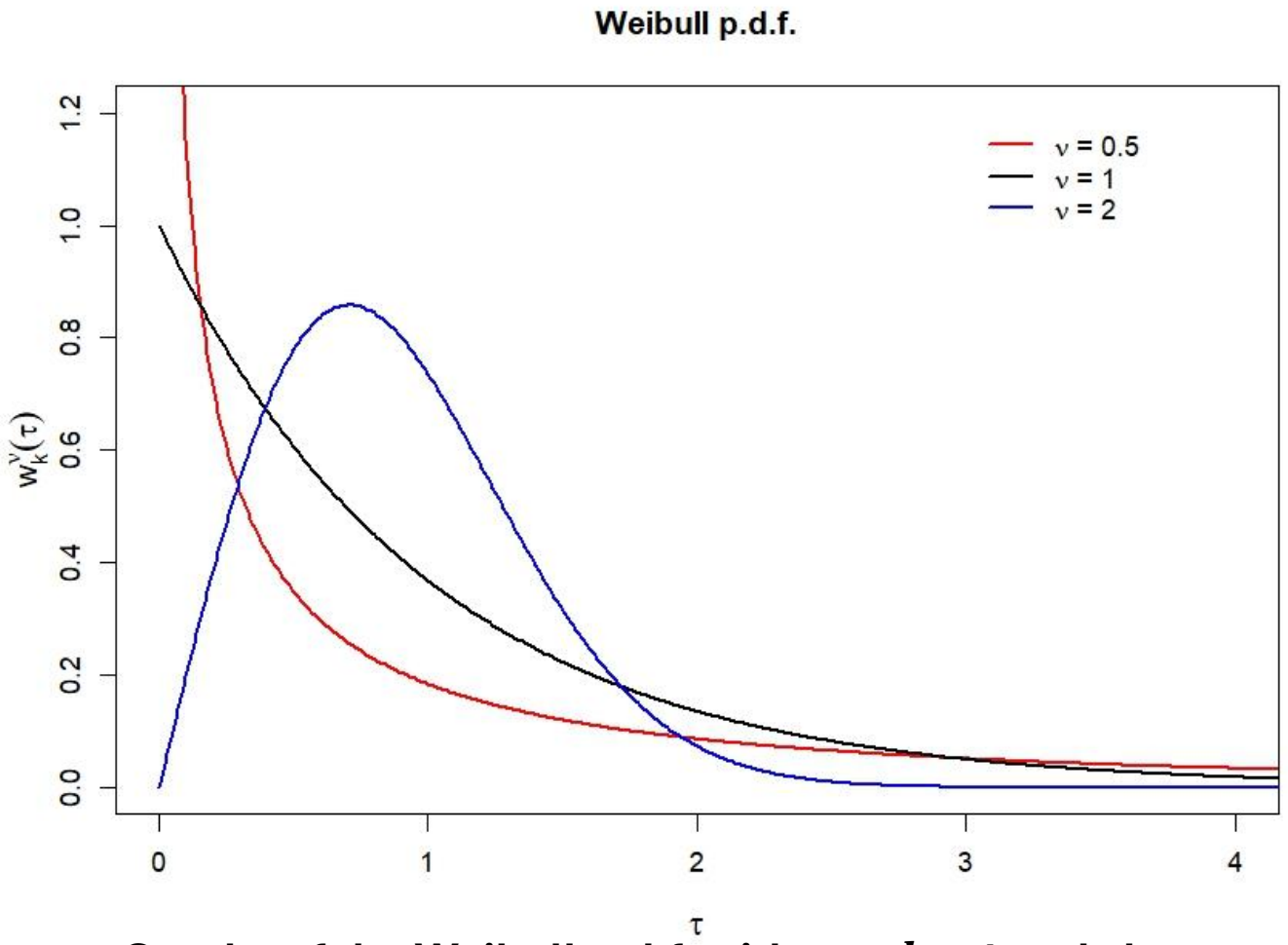
$$X(t) = \varphi(t) \text{ for } t \leq 0 \qquad \text{past condition}$$

where $X = (X_1, \ldots, X_n)$, $\varphi = (\varphi_1, \ldots, \varphi_n)$, and g $= (g_1, \ldots, g_n)$

- Past condition is necessary only for states X_i which are convoluted with the delay kernel $\ell(t)$ or delayed states $X_i(t-T)$
- For states X_i that are not convoluted, the past condition reduces to the initial condition:

$$X_i(0) = X_{i0} \qquad \text{initial condition}$$

Weibull Probability Density Function



Graphs of the Weibull p.d.f. with rate k =1 and shape $\nu = 0.5, 1,$ and 2.

$$w_k^\nu(\tau) = \begin{cases} k\nu(k\tau)^{\nu-1} \exp\big(-(k\tau)^\nu\big), & \text{if} \quad \tau > 0 \\ 0, & \text{if} \quad \tau \le 0 \end{cases}$$

ν = shape factor
k = rate
$1/k$ = scale

Mean= $\frac{\Gamma(1+1/\nu)}{k}$

Variance= $\frac{\Gamma(1+2/\nu)-\Gamma(1+1/\nu)^2}{k^2}$

Weibull Distributed Absorption One-Compartment Model

$$\frac{dA_a}{dt} = \sum_{i=1}^{N_{dose}} F\text{Dose} w_k^{\nu}(t - t_i) - k_a A_a, \quad \text{and} \quad A_a(0) = 0$$

$$\frac{dA_p}{dt} = k_a A_a - \frac{CL}{V} A_p, \quad \text{and} \quad A_p(0) = 0$$

ODE Absorption One Compartment Model

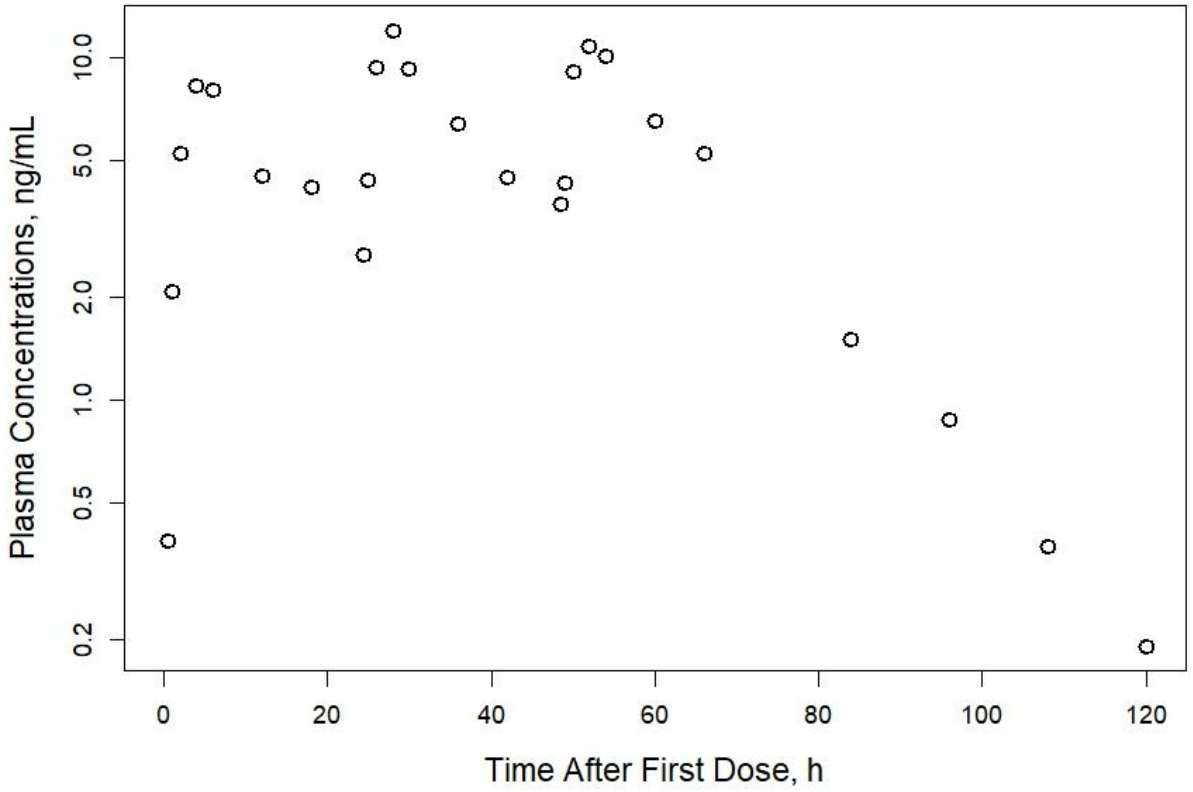
$$\frac{d\mathcal{A}_a}{dt} = \sum_{i=1}^{N_{dose}} F\text{Dose}\delta(t - t_i) - k_a \mathcal{A}_a, \quad \text{and} \quad \mathcal{A}_a(0) = 0$$

$$\frac{d\mathcal{A}_p}{dt} = k_a \mathcal{A}_a(t) * w_k^{\nu}(t) - \frac{CL}{V} \mathcal{A}_p, \quad \text{and} \quad \mathcal{A}_p(0) = 0$$

DDDE Absorption One Compartment Model

$$C_p(t) = \frac{A_p(t)}{V}$$

Simulated Individual Plasma Concentrations



- **The one-compartment model with first-order absorption rate and Weibull distributed lag times was used to simulate drug plasma concentrations in a single subject following administration of 100 µg at 0, 24, and 48 h.**
- **k = k_a for identifiability reasons**
- **The proportional residual error model was used to add the variability of CV=10% to the simulated data.**
- **Simulations were performed using R package deSolve.**

Parameter	Value
CL, L/h	0.6
V, L	10
k_a, 1/h	1
F	1
ν	2
σ	0.1

Implementation of DDDE Models in Phoenix

```
delay(S, MeanDelayTime
     [, shape = ShapeParam]
    [, hist = HistExpression]
    [, dist = NameOfDistribution]
    )
```

- delay() returns $\int_0^\infty S(t-\tau)\ell(\tau)d\tau$ where $S(t)$ is a state to be delayed by lag-times distributed by the p.d.f $\ell(\tau)$
- `MeanDelayTime` = mean of the distribution $\ell(\tau)$
- `ShapeParam` = shape of the distribution $\ell(\tau)$
- `HistExpression` = past for the state $S(t)$
- `NameOfDistribution` = `Gamma`, `Weibull`, and `InverseGaussian`
- If `hist` is omitted then the past for $S(t)$ is equal 0

https://onlinehelp.certara.com/phoenix/8.3/topics/Discrete_and_distributed_delays.htm

Hu et al. J Pharmacokinet Pharmacodyn 45:285-308 (2018) https://doi.org/10.1007/s10928-018-9570-4

Phoenix: Weibull Distributed Absorption One-Compartment Model

```
ABS_ODE(){
    deriv(Aa = 100*dweibull(t,NU,1/KA)+
               100*dweibull(t-24,NU,1/KA)+
               100*dweibull(t-48,NU,1/KA)- KA * Aa)
    deriv(A =  KA *Aa-CL/V*A)
    C = A / V
    error(CEps = 0.1)
    observe(CObs = C*(1 + CEps))
    fixef(CL = c(0, 0.6, ))
    fixef(V = c(0, 10, ))
    fixef(KA = c(0,1.0, ))
    fixef(NU = c(0, 2.0, ))
}
```

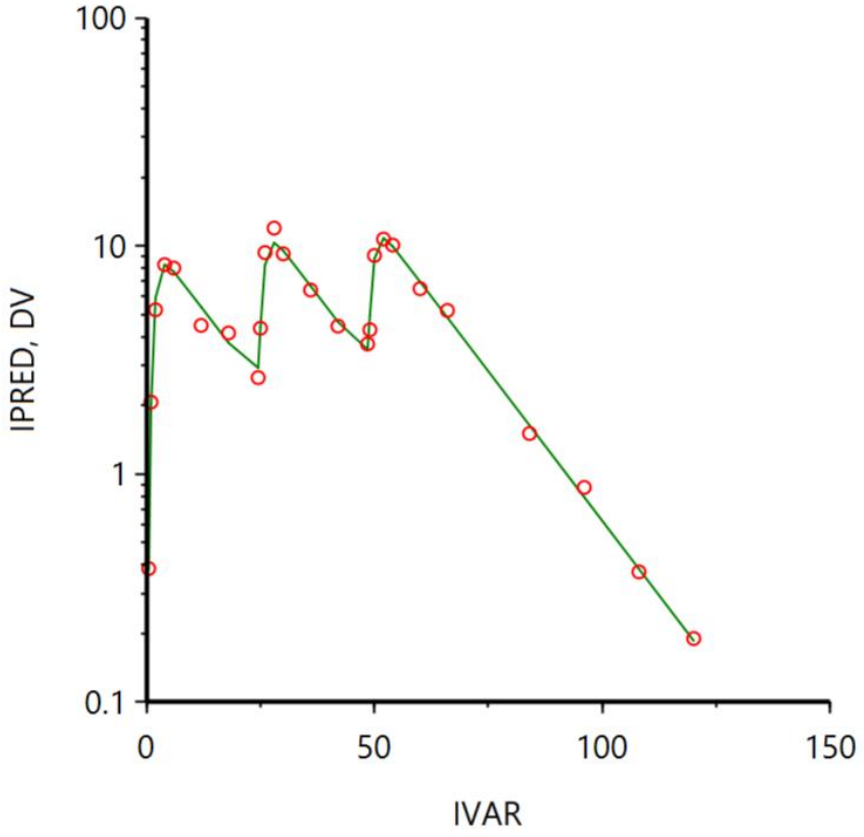
ODE model

```
ABS_DDDE(){
    deriv(Aa = - KA * Aa)
    deriv(A =  KA *delay(Aa,MIT,shape=NU,dist=Weibull)
          -CL/V*A)
    dosepoint(Aa)
    C = A / V
    MIT= 1/KA*exp(lgamm(1+1/NU))
    error(CEps = 0.1)
    observe(CObs = C*(1 + CEps))
    fixef(CL = c(0, 0.6, ))
    fixef(V = c(0, 10, ))
    fixef(KA = c(0,1.0, ))
    fixef(NU = c(0, 2.0, ))
    secondary(MIT=1/KA*exp(lgamm(1+1/NU)))
}
```

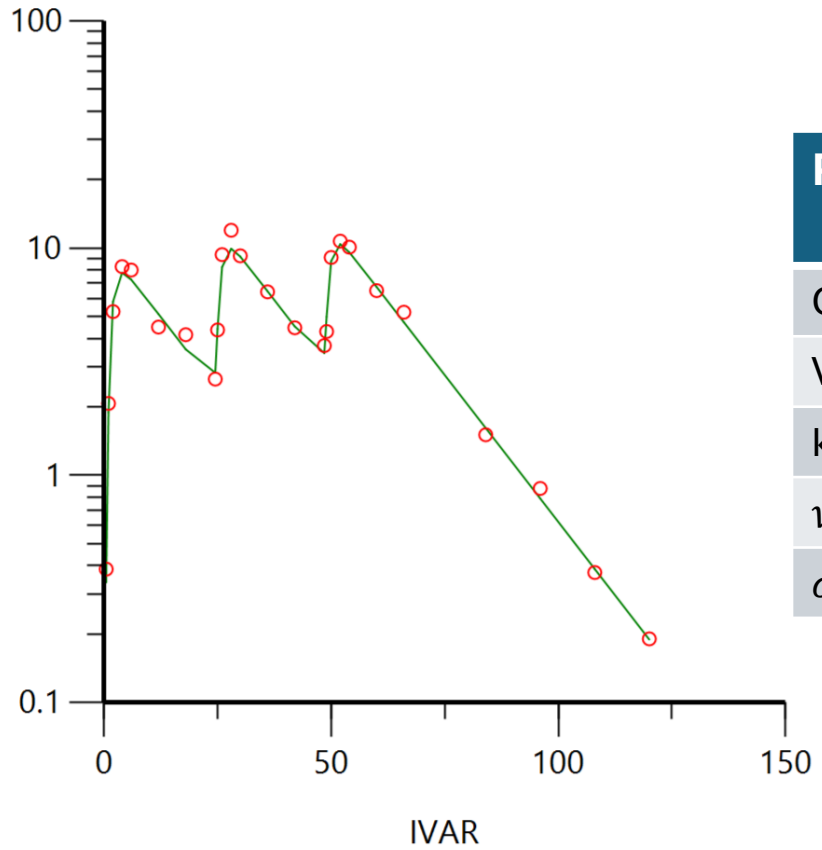
DDDE model

Comparison of Fittings Between ODE and DDDE Models

ODE model



DDDE model



- Naïve-pooled estimation
- Stiff ODE solver

Parameter	ODE (%RSE)	DDDE (%RSE)
CL/F, L/h	0.609(1.8)	0.636 (0.01)
V/F, L	10.1 (2.6)	10.78 (0.01)
k_a, 1/h	1.0 (3.2)	1.02 (0.02)
ν	1.88 (5.5)	2.0 (0.02)
σ	0.082 (14.2)	0.1 (21.4)

Negligible differences in fittings and parameter estimates between ODE and DDDE models

Implementation of DDDE Models in NONMEM

Required statements in NONMEM control stream

; DDE Inform ddexpand utility to supply code for a DDE solver

; NNy Number of quadrature positions to be used in convolution of distributed time delay system y (y = 1,2, …)

; DISTRIBy = name of distribution vector (vector, number of parameters) Distribution to be used for packed quadrature delay method, vector name to be used for parameter inputs to distribution function, and number of parameters to define distribution

;ALGy Optional. Define quadrature method for NNy time delay positions and weights of distributed delay system y

APG_x_y Provide a general distribution time. Variable defining the past for state variable A(x), delay system y

ADG_x_y State variable A(x) time delayed by general distributed delay system y

Execution of control stream

```
..\nmfe76  control_stream_file  output_file -prdefault -dde
```


Bauer and Krzyzanski, CPT Pharmacometrics and Systems Pharmacology 14:1133-1155 (2025) https://doi.org/10.1002/psp4.70046

Delay Kernels and DDE Solvers Supported by NONMEM

P.d.f.s to be used as delay kernels:

- LOGNORMAL, GAMMA, INVGAMMA, WEIBULL
- More p.d.f.s in I.35 NONMEM Uses Guide. Introduction to NONMEM 7.6.1

DDE solvers capable of performing Gauss quadrature integration:

- ADVAN9, ADVAN13, ADVAN14, ADVAN15, ADVAN16, ADVAN17, ADVAN18
- ADVAN16 solves stiff DDEs

Bauer, https://nmusers.github.io/docs/guides/intro7.pdf

NONMEM: Weibull Distributed Absorption One-Compartment Model

```
$PROBLEM ABSORPTION ODE
$ABBR DERIV2=NO
$ABBR FUNCTION WEIBULL(X,4)
$INPUT ID TIME CMT AMT DV EVID MDV
$DATA ABS_ODE_7.csv IGNORE=C
$SUBROUTINES ADVAN13 TOL=6 ATOL=6
$MODEL NCOMPARTMENTS=2

$PK
CL      = THETA(1)
V       = THETA(2)
KA      = THETA(3)
NU      = THETA(4)
K=KA
DOSE=100

A_0(1)=0
A_0(2)=0

$DES
X(2)=NU
X(3)=1/K
X(1)=T
INP1=DOSE*EXP(-WEIBULL(X))
INP2=0
IF(T.GT.24) THEN
  X(1)=T-24.0
  INP2=DOSE*EXP(-WEIBULL(X))
ENDIF
INP3=0
IF(T.GT.48) THEN
 X(1)=T-48.0
 INP3=DOSE*EXP(-WEIBULL(X))
ENDIF
```

```
DADT(1) = INP1+INP2+INP3-KA*A(1)
DADT(2) = KA*A(1)-CL/V*A(2)

$ERROR
IPRED=-10
IF (CMT.EQ.2) IPRED = A(2)/V

Y = IPRED*(1+EPS(1))

$THETA
(0,0.6,)            ; 1: CL
(0,10.0)            ; 2: V
(0,1.0, )           ; 3: KA
(0, 2.0, )          ; 4: NU

$SIGMA
0.01

$EST METHOD=0 MAXEVAL=9999 PRINT=1
SIG=3 NOABORT SIGL=9
$COV PRINT=E
$TABLE ID TIME IPRED MDV EVID INP1
NOPRINT ONEHEADER FILE=ABS_ODE.tab
```

ODE model

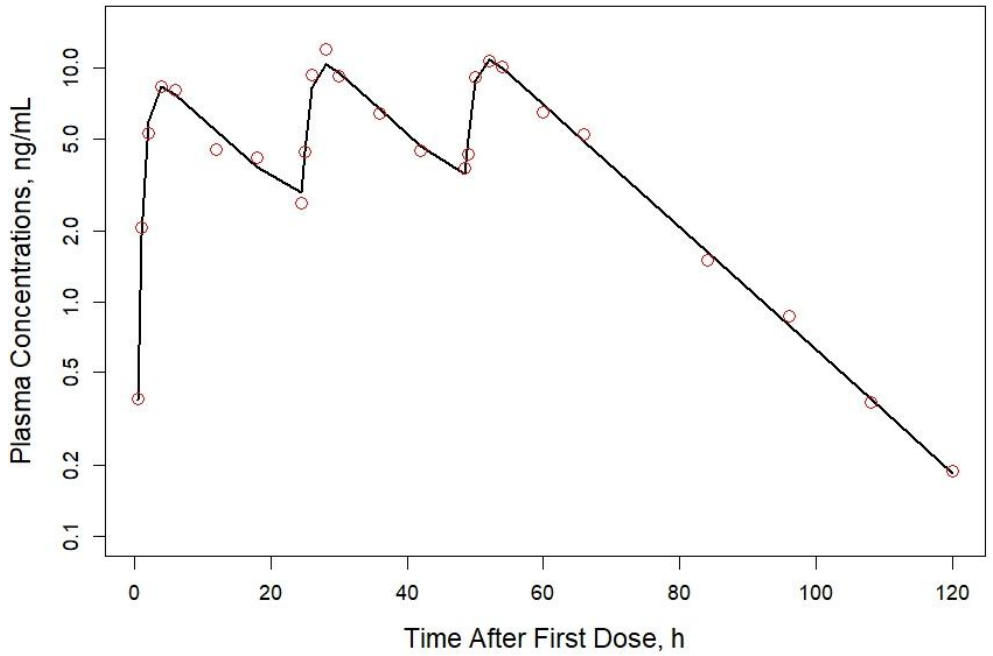
```
;DDE
;NN1=15
;DISTRIB1=WEIBULL(VG,2)
;ALG1=GQ(U)
$PROBLEM ABSORPTION DDDE
$ABBR DERIV2=NO
$INPUT ID TIME CMT AMT DV EVID MDV
$DATA ABS_DDDE.csv IGNORE=C
$SUBROUTINES ADVAN16 TOL=12 ATOL=12
$MODEL NCOMPARTMENTS=2

$PK
CL      = THETA(1)
V       = THETA(2)
KA      = THETA(3)
NU1     = THETA(4)
K=KA

A_0(1)=0
A_0(2)=0

$DES
VG(2)=NU1
VG(3)=1/K
APG_1_1=0

DADT(1) = -KA*A(1)
DADT(2) = KA*ADG_1_1-CL/V*A(2)
```

```
$ERROR
IPRED=-10
IF (CMT.EQ.2) IPRED = A(2)/V

Y = IPRED*(1+EPS(1))

$THETA
(0,0.6,)            ; 1: CL
(0,10.0)            ; 2: V
(0,1.0, )           ; 3: KA
(0, 2.0, )          ; 4: NU1

$SIGMA
0.01

$COV MATRIX=R UNCONDITIONAL

$EST METHOD=0 MAXEVAL=9999 PRINT=1
SIG=3 NOABORT SIGL=9
$TABLE ID TIME IPRED MDV EVID NOPRINT
ONEHEADER FILE=ABS_DDDE.tab
```

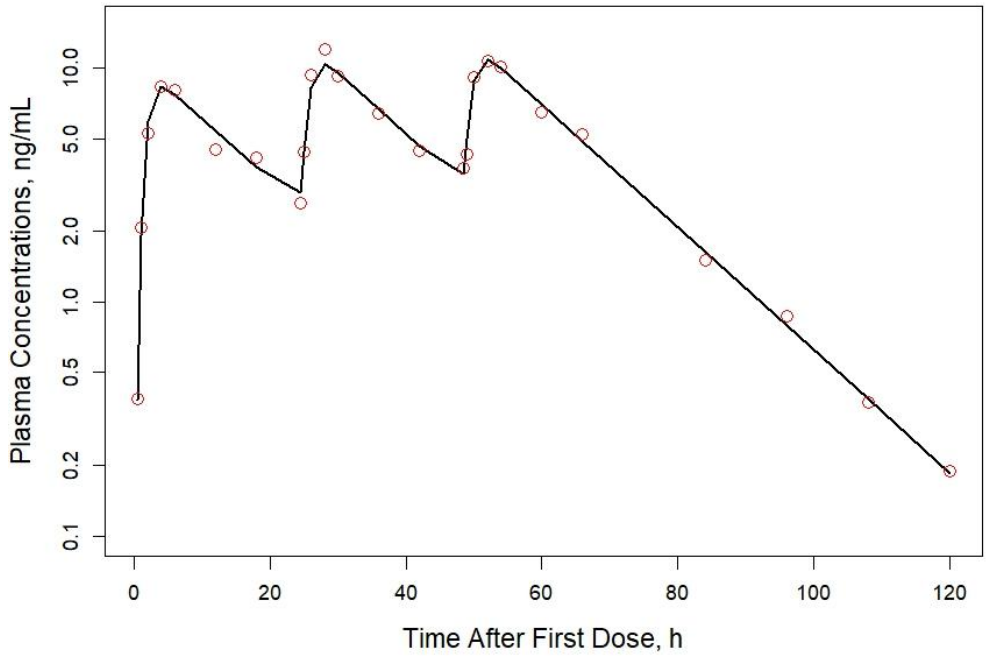
DDDE model

Comparison of Fittings Between ODE and DDDE Models

ODE model



DDDE model



Parameter	ODE (%RSE)	DDDE (%RSE)
CL/F, L/h	0.609 (1.9)	0.609 (1.9)
V/F, L	10.1 (2.7)	10.1 (2.8)
k_a, 1/h	1.0 (3.9)	1.0 (3.9)
ν	1.88 (6.3)	1.89 (6.8)
σ^2	0.007 (24.7)	0.007(27.6)

- First-order approximation estimation method

Negligible differences in fittings and parameter estimates between ODE and DDDE models

Pros and Cons of Using ODE vs DDDE Absorption Models

- **Both ODE and DDDE models identically describe the disposition states (central and peripheral compartments)**
- **ODE models require “manual” coding dose input functions**
- **DDDE models use built-in features for dosing events**
- **ODE models can be implemented in any software with ODE solvers**
- **DDDE models require software with integro-differential equation solvers**
- **ODE models run much faster than DDDE models**
- **Both ODE and DDDE are prone to introduce stiffness**

Questions?

For more information, including presented source codes and data, please email:

wk@buffalo.edu

University at Buffalo