

MONOLIX 4.2

Model description with MLXTRAN

The following slidedeck provides a tutorial on how to easily describe simple and complex pharmacometric models, including PK, PK-PD and discrete data models, using the MLXTRAN language included in MONOLIX 4.2.

MONOLIX 4.2

Model description with MLXTRAN

1. Introduction to MLXTRAN [3](#)
2. MLXTRAN for PK models
 - 2.1 IV administration [12](#)
 - 2.2 Oral administration [43](#)
 - 2.3 Complex administration [67](#)
 - 2.4 PK parameters summary [77](#)
3. MLXTRAN for PKPD models [81](#)
4. MLXTRAN for discrete data models [95](#)
5. Symbols reference [127](#)

MONOLIX 4.2

MLXTRAN

1. Introduction to MLXTRAN

DESCRIPTION:

Example of model defined with a system of ODEs

INPUT:

parameter = {D, ka, kb}

regressor = {x1, x2}

EQUATION:
$$A_0 = D$$
$$t_0 = 0$$
$$\text{ddt_A} = -ka \cdot A + x1$$
$$\text{ddt_B} = ka \cdot A - kb \cdot B + x2$$
OUTPUT:

output = {A, B}

DESCRIPTION:

Exponential decay model

Analytical solution

$$\text{if } t \leq 0, \quad A = D$$

$$\text{if } t > 0, \quad A = De^{-k \times t}$$

Use the block **DESCRIPTION** to comment the model

DESCRIPTION:

Exponential decay model
Analytical solution

INPUT:

parameter = {D, k}

$$\text{if } t \leq 0, \quad A = D$$

$$\text{if } t > 0, \quad A = De^{-k \times t}$$

Block **INPUT** is used to define

- the list of parameters ψ . Here, $\psi = (D, k)$.
- the list of additional regression variables $x(t)$.
Here, there is no additional regression variables (t is the only regression variable).

DESCRIPTION:

Exponential decay model
Analytical solution

INPUT:

parameter = {D, k}

EQUATION:

```
if t <= 0
  A = D
else
  A = D*exp(-k*t)
end
```

$$\text{if } t \leq 0, \quad A = D$$

$$\text{if } t > 0, \quad A = D e^{-k \times t}$$

Block **EQUATION** is used to describe the mathematical model

By default, time **t** is the primary regressor,

t is a reserved keyword that can be used in the model.

DESCRIPTION:

Exponential decay model
Analytical solution

INPUT:

parameter = {D, k}

EQUATION:

```
if t <= 0
  A = D
else
  A = D*exp(-k*t)
end
```

OUTPUT:

output = A

$$\text{if } t \leq 0, \quad A = D$$

$$\text{if } t > 0, \quad A = D e^{-k \times t}$$

Block **OUTPUT** is used to define the output of the model.

DESCRIPTION:

Exponential decay model
ODE solution

INPUT:

parameter = {D, k}

EQUATION:

$t_0 = 0$

$A_0 = D$

$\text{ddt_A} = -k \times A$

OUTPUT:

output = A

if $t \leq 0$,

$$A = D$$

if $t > 0$,

$$\frac{dA(t)}{dt} = -k \times A(t)$$

Ordinary differential equations (ODEs) can be introduced in the block **EQUATION** :

ddt_A is the derivative of **A** with respect to time **t**.

A_0 is the initial value of the system, *i.e.* the value of **A** at time **t0** (**A** is constant before **t0**).

If **A_0** is not specified, then **A_0** = 0.

If **t0** is not specified, then **t0** is the first time **A** is computed.

DESCRIPTION:

ODEs defined model

INPUT:

parameter = {D, ka, kb}

EQUATION:

t0 = 0

A_0 = D

ddt_A = -ka*A

ddt_B = ka*A - kb*B

odeType = stiff

OUTPUT:

output = B

if $t \leq 0$,

$$A = D$$

$$B = 0$$

if $t \geq 0$,

$$\frac{dA(t)}{dt} = -k_a \times A(t)$$

$$\frac{dB(t)}{dt} = k_a \times A(t) - k_b \times B(t)$$

A system of several ODE's can be introduced in the block **EQUATION**.

The solver for the ODE system is set using **odeType**. If **odeType** is not specified, the solver for non-stiff ODE systems is set.

DESCRIPTION:

ODEs defined model

INPUT:

parameter = {D, ka, kb}

regressor = x

EQUATION:

$t_0 = 0$

$A_0 = D$

$\text{d}A/\text{d}t = -k_a \times A + x$

$\text{d}B/\text{d}t = k_a \times A - k_b \times B$

OUTPUT:

output = B

if $t \leq 0$,

$$A = D$$

$$B = 0$$

if $t \geq 0$,

$$\frac{dA(t)}{dt} = -k_a \times A(t) + x(t)$$

$$\frac{dB(t)}{dt} = k_a \times A(t) - k_b \times B(t)$$

Additional regression variables can be defined in the block **INPUT**,

By definition, a regressor $x=x(t)$ is a time-varying variable.

DESCRIPTION:

Example of model defined with a system of ODEs

INPUT:

parameter = {D, ka, kb}

regressor = {x1, x2}

EQUATION:

$t_0 = 0$

$A_0 = D$

$\text{ddt}_A = -k_a \times A + x_1$

$\text{ddt}_B = k_a \times A - k_b \times B + x_2$

OUTPUT:

output = {A, B}

if $t \leq 0$,

$$A = D$$

$$B = 0$$

if $t \geq 0$,

$$\frac{dA(t)}{dt} = -k_a \times A(t) + x_1(t)$$

$$\frac{dB(t)}{dt} = k_a \times A(t) - k_b \times B(t) + x_2(t)$$

Several regressors can be defined in the block **INPUT**.

Several outputs can be defined in the block **OUTPUT**.

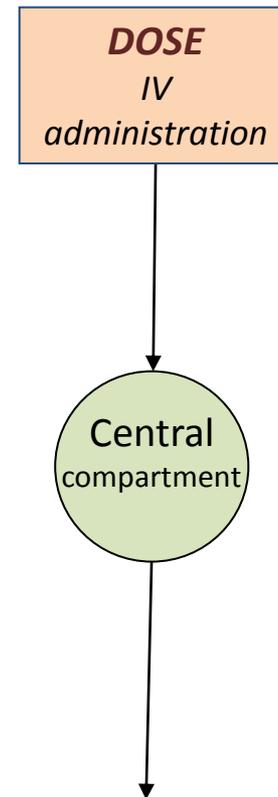
MONOLIX 4.2

MLXTRAN

2. MLXTRAN for PK models

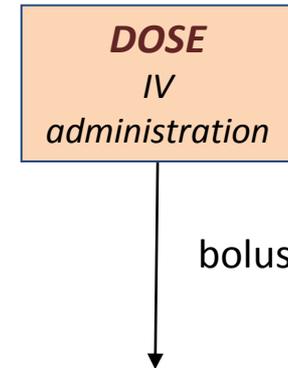
2.1 IV administration

- *The data* 13
- *The PK macros* 16
- *PK macros and equations* 25
- *The pkmodel function* 29
- *Non linear eliminations* 32
- *2 & 3 compartments models* 37



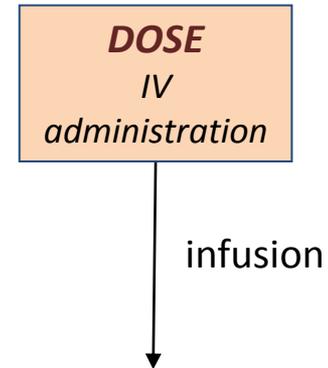
The data file

ID	TIME	AMT	Y
1	0	500	.
1	2	.	44.6
1	6	.	37
1	12	.	16.1
1	18	.	14.2
1	24	.	6.79
2	0	500	.
2	2	.	48.6
2	6	.	37.6
2	12	.	20.2
2	18	.	6.09
2	24	.	4.34
3	0	500	.
3	2	.	56.3
3	6	.	29.6
3	12	.	16.9
3	18	.	7.96
3	24	.	2.71



If there is no RATE (or TINF) column in the data, then administration is assumed to be an IV bolus

ID	TIME	AMT	RATE	Y
1	0	500	200	.
1	2	.	.	34
1	6	.	.	38.2
1	12	.	.	23.6
1	18	.	.	24.9
1	24	.	.	13.3
2	0	500	200	.
2	2	.	.	17
2	6	.	.	16.9
2	12	.	.	5.39
2	18	.	.	1.63
2	24	.	.	0.35
3	0	500	200	.
3	2	.	.	48.7
3	6	.	.	41.2
3	12	.	.	14.5
3	18	.	.	6.32
3	24	.	.	4.42



If there is a RATE (or TINF) column in the data, then administration is assumed to be an IV infusion

The PK macros

DESCRIPTION:

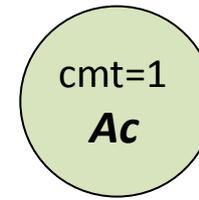
PK model, IV administration,
linear elimination, parameters (V, k)

INPUT:

parameter = {V, k}

PK:

compartment(cmt=1, amount=Ac)



Block **PK** is used to define the PK model, using macros with unordered named arguments

- **compartment(cmt=1, amount=Ac)** creates a compartment 1 which amount is **Ac**,

DESCRIPTION:

PK model, IV administration,
linear elimination, parameters (V, k)

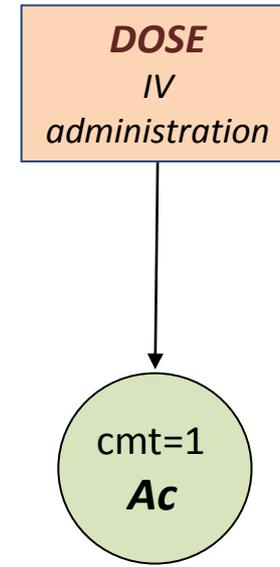
INPUT:

parameter = {V, k}

PK:

compartment(cmt=1, amount=Ac)

iv(cmt=1)



Block **PK** is used to define the PK model, using macros with unordered named arguments

- **compartment(cmt=1, amount=Ac)** creates a compartment 1 which amount is **Ac**,
- **iv(cmt=1)** means an IV (bolus or infusion, according to the data) administration in compartment 1

DESCRIPTION:

PK model, IV administration,
linear elimination, parameters (V, k)

INPUT:

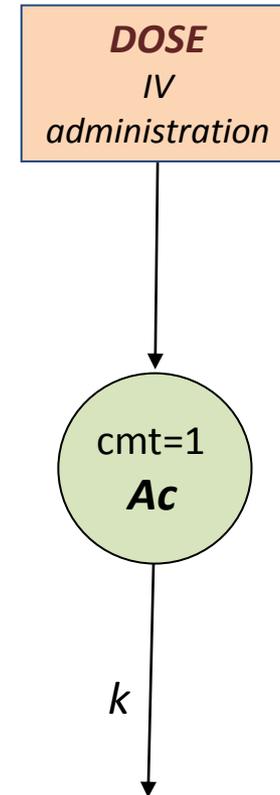
parameter = {V, k}

PK:

compartment(cmt=1, amount=Ac)

iv(cmt=1)

elimination(cmt=1, k)



Block **PK** is used to define the PK model, using macros with unordered named arguments

- **compartment(cmt=1, amount=Ac)** creates a compartment 1 which amount is **Ac**,
- **iv(cmt=1)** means an IV (bolus or infusion, according to the data) administration in compartment 1
- **elimination(cmt=1, k)** defines a linear elimination from compartment 1 with rate constant **k**

DESCRIPTION:

PK model, IV administration,
linear elimination, parameters (V, k)

INPUT:

parameter = {V, k}

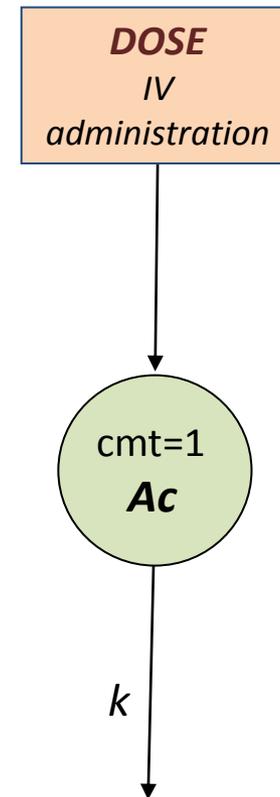
PK:

compartment(cmt=1, amount=Ac)

iv(cmt=1)

elimination(cmt=1, k)

$C_c = A_c/V$



$C_c = A_c/V$ computes the concentration C_c in compartment 1

DESCRIPTION:

PK model, IV administration,
linear elimination, parameters (V, k)

INPUT:

parameter = {V, k}

PK:

compartment(cmt=1, amount=Ac)

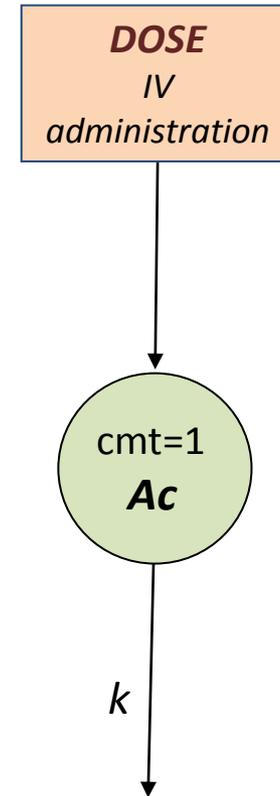
iv(cmt=1)

elimination(cmt=1, k)

$C_c = A_c/V$

OUTPUT:

output = Cc



Block **OUTPUT** is used to define the outputs of the model

Here, **output = Cc** defines the predicted concentration **Cc** as the only output of the model.

DESCRIPTION:

PK model, IV administration,
linear elimination, parameters (V, k)

INPUT:

parameter = {V, k}

PK:

compartment(amount=Ac)

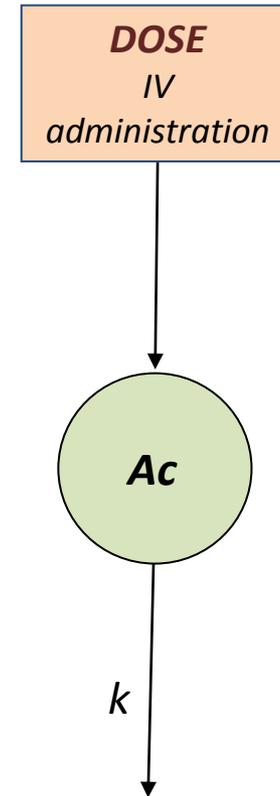
iv()

elimination(k)

$C_c = A_c/V$

OUTPUT:

output = C_c



When the number of the compartment is not specified, **cmt=1** is assumed by default.

There is only one compartment here. Then, the default can be used and the number of the compartment can be omitted.

DESCRIPTION:

PK model, IV administration,
linear elimination, parameters (V, k)

INPUT:

parameter = {V, k}

PK:

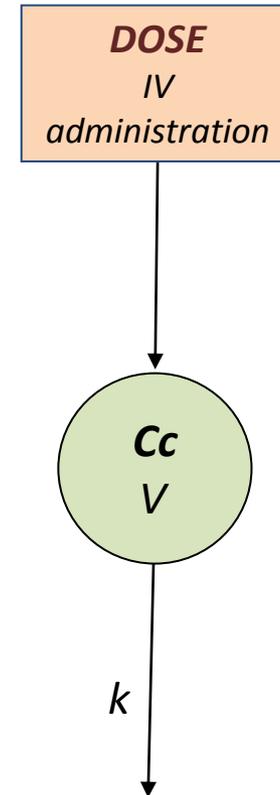
compartment(concentration=Cc, volume=V)

iv()

elimination(k)

OUTPUT:

output = Cc



Concentration **Cc** in compartment 1 can be defined in PK macro **compartment** instead of the amount **Ac**.

In this case, the volume of the compartment is required.

DESCRIPTION:

PK model, IV administration,
linear elimination, parameters (V, Cl)

INPUT:

parameter = {V, Cl}

PK:

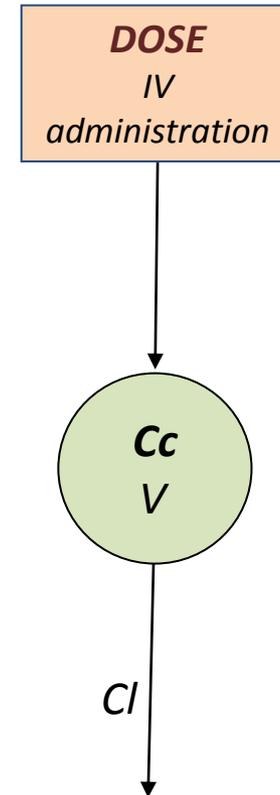
compartment(concentration=Cc, volume=V)

iv()

elimination(Cl)

OUTPUT:

output = Cc



Several parametrizations can be used.

Here, clearance Cl is used instead of the elimination rate constant k ,

PK macro **elimination** accepts the parametrization with Cl instead of k .

Combining PK macros & equations

DESCRIPTION:

PK model, IV administration,
linear elimination, parameters (V, k)

INPUT:

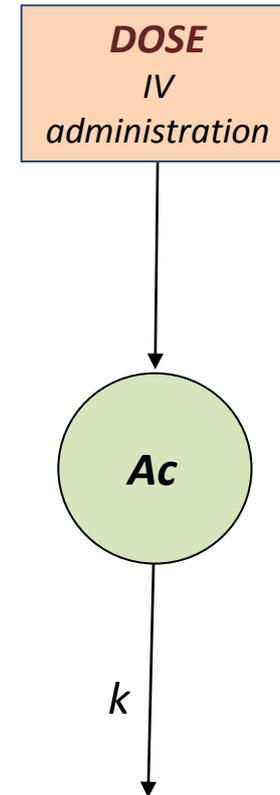
parameter = {V, k}

PK:

compartment(amount=Ac)
iv()

EQUATION:
$$\text{ddt_Ac} = -k * \text{Ac}$$
$$\text{Cc} = \text{Ac}/V$$
OUTPUT:

output = Cc



Some components of the PK model can be described using equations in a block **EQUATION**.

Here, **Ac** is the solution of an ODE defined by $\text{ddt_Ac} = -k * \text{Ac}$

It is equivalent to define ODEs in a block **EQUATION** or PK macros in a block **PK**

DESCRIPTION:

PK model, IV administration,
linear elimination, parameters (V, k)

INPUT:

parameter = {V, k}

PK:

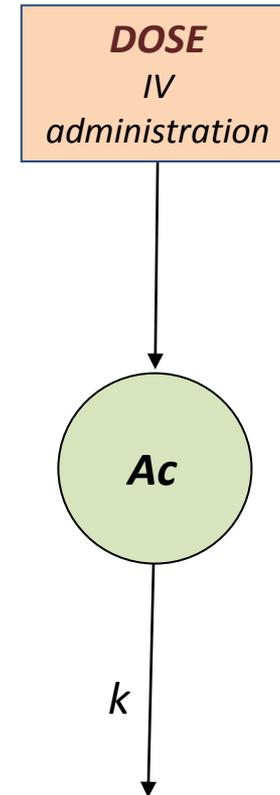
depot(target=Ac)

EQUATION:

$$\text{d}dt_Ac = -k * Ac$$

$$Cc = Ac/V$$
OUTPUT:

output = Cc



Instead of using the PK macros **compartment** & **iv**, it is possible to consider the doses as source terms of the autonomous system defined with ODEs.

depot(target=Ac) means that **Ac** is the target of the source term :

$$Ac(tDose^+) = Ac(tDose^-) + amtDose$$

DESCRIPTION:

PK model, IV administration,
linear elimination, parameters (V, k)

INPUT:

parameter = {V, k}

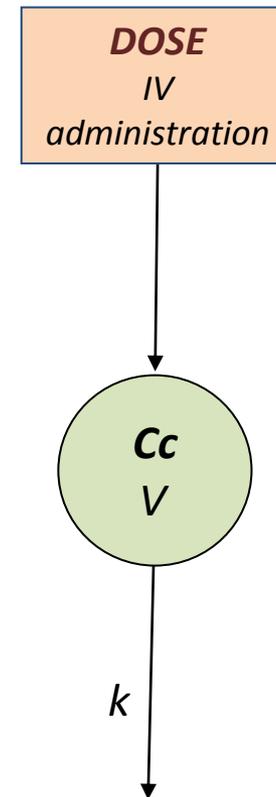
EQUATION:

$\text{deltat} = t - t\text{Dose}$

$Cc = \text{amtDose}/V * \exp(-k * \text{deltat})$

OUTPUT:

output = Cc



The analytical expression of **Cc** for a single dose administration can be used in the block **EQUATION**, **tDose** and **amtDose** are reserved keywords:

- **tDose** is the time of the last dose (if **tDose**=0 for all subjects, then it can be omitted).
- **amtDose** is the amount of drug administrated at time **tDose**.

The pkmodel function

DESCRIPTION:

PK model, IV administration,
linear elimination, parameters (V, k)

INPUT:

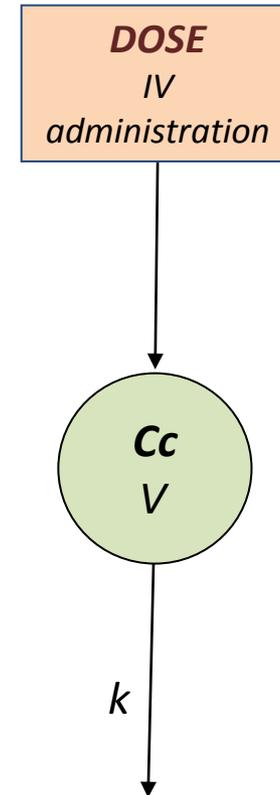
parameter = {V, k}

EQUATION:

$C_c = \text{pkmodel}(V, k)$

OUTPUT:

output = C_c



Instead of the PK macros, function **pkmodel** can be used to define most standard PK models.

The PK model is defined according to the unordered named arguments of function **pkmodel**.

Here, **pkmodel(V,k)** assumes a linear elimination with rate constant k .

There is no parameters defining the absorption, then, an IV administration is assumed.

DESCRIPTION:

PK model, IV administration,
linear elimination, parameters (V, Cl)

INPUT:

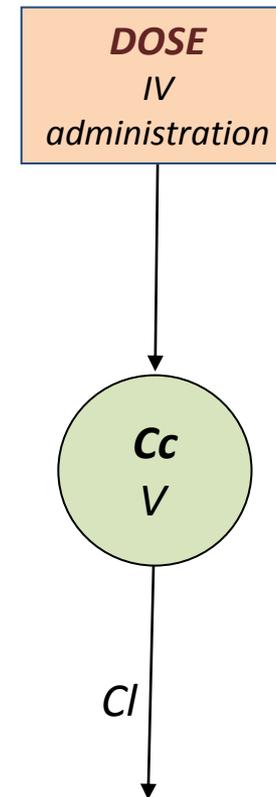
parameter = {V, Cl}

EQUATION:

$C_c = \text{pkmodel}(V, Cl)$

OUTPUT:

output = C_c



Function **pkmodel** also accepts the parametrization with **Cl** instead of **k**.

Non linear elimination

DESCRIPTION:

PK model, IV administration,
nonlinear elimination, parameters (V, Vm, Km)

INPUT:

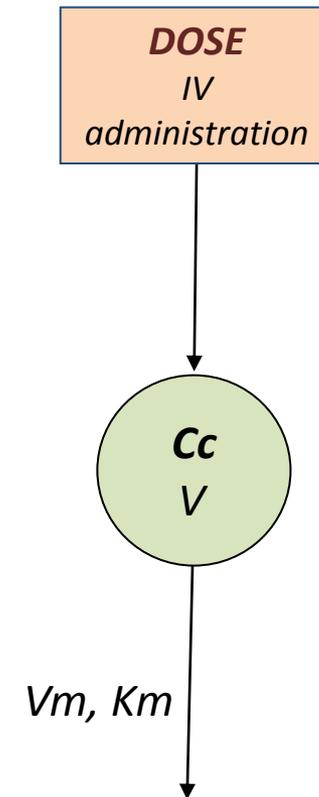
parameter = {V, Vm, Km}

EQUATION:

$C_c = \text{pkmodel}(V, V_m, K_m)$

OUTPUT:

output = C_c



Non linear elimination can be defined using the function **pkmodel** with parameters **Vm** and **Km**

DESCRIPTION:

PK model, IV administration,
nonlinear elimination, parameters (V , V_m , K_m)

INPUT:

parameter = $\{V, V_m, K_m\}$

PK:

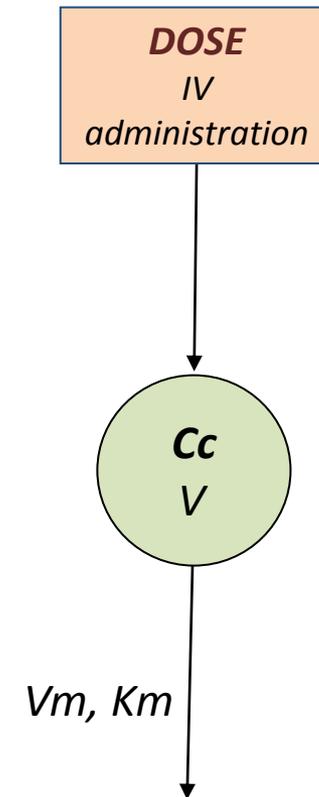
compartment(concentration= C_c , volume= V)

iv()

elimination(V_m, K_m)

OUTPUT:

output = C_c



Non linear elimination can also be defined with the PK macro **elimination** using parameters V_m and K_m

DESCRIPTION:

PK model, IV administration,
nonlinear elimination, parameters (V, Vm, Km)

INPUT:

parameter = {V, Vm, Km}

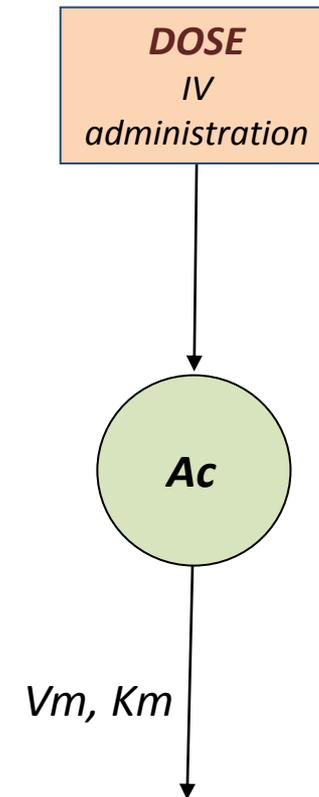
PK:

compartment(amount=Ac)

iv()

EQUATION:
$$\text{ddt_Ac} = -V_m * \text{Ac} / (V * K_m + \text{Ac})$$
$$C_c = \text{Ac} / V$$
OUTPUT:

output = Cc



Instead of using the PK macro **elimination**, the elimination can be defined in the block **EQUATION**.

DESCRIPTION:

PK model, IV administration,
Combination of linear and nonlinear eliminations,

INPUT:

parameter = {V, Vm, Km, k}

PK:

compartment(amount=Ac)

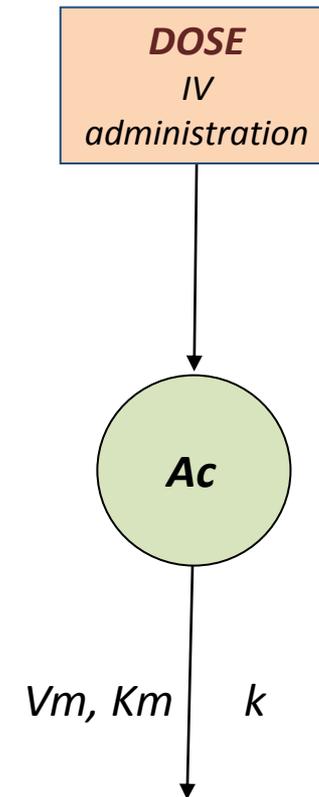
iv()

EQUATION:

$$\text{ddt_Ac} = -V_m * \text{Ac} / (V * K_m + \text{Ac}) - k * \text{Ac}$$

$$C_c = \text{Ac} / V$$
OUTPUT:

output = Cc



Complex user defined elimination processes that cannot be described with the PK macros should be described in the block **EQUATION**.

Here, the elimination process is a combination of linear and non linear eliminations: it is described with an ODE in the block **EQUATION**.

2 & 3 compartments models

DESCRIPTION:

PK model, IV administration,
2 compartments,
linear elimination,

INPUT:

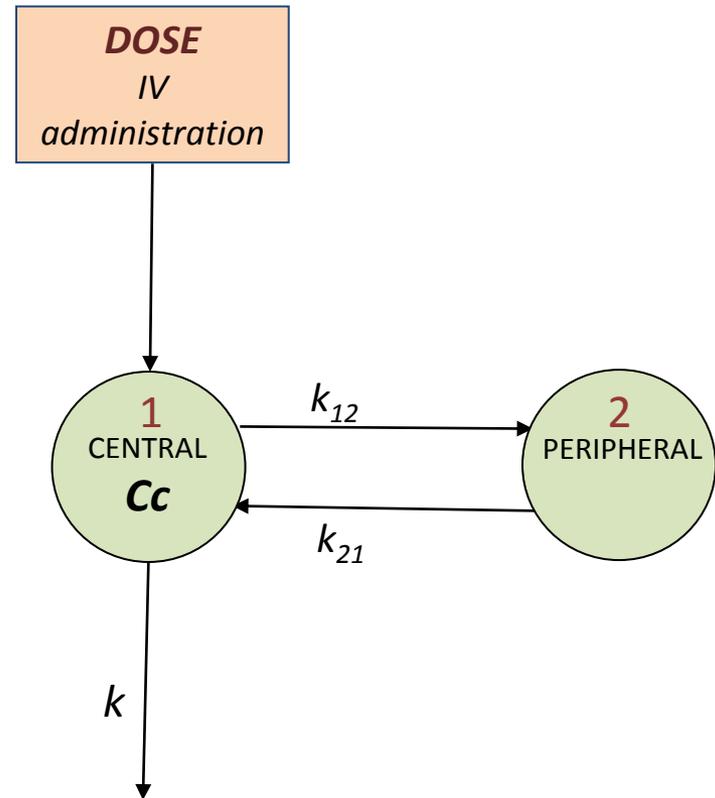
parameter = {V, k, k₁₂, k₂₁}

EQUATION:

$C_c = \text{pkmodel}(V, k, k_{12}, k_{21})$

OUTPUT:

output = C_c



Function **pkmodel** introduces a peripheral compartment in the model when parameters **k_{12}** and **k_{21}** are used.

DESCRIPTION:

PK model, IV administration,
2 compartments,
linear elimination,

INPUT:

parameter = {V, k, k₁₂, k₂₁}

PK:

compartment(concentration=C_c, volume=V)

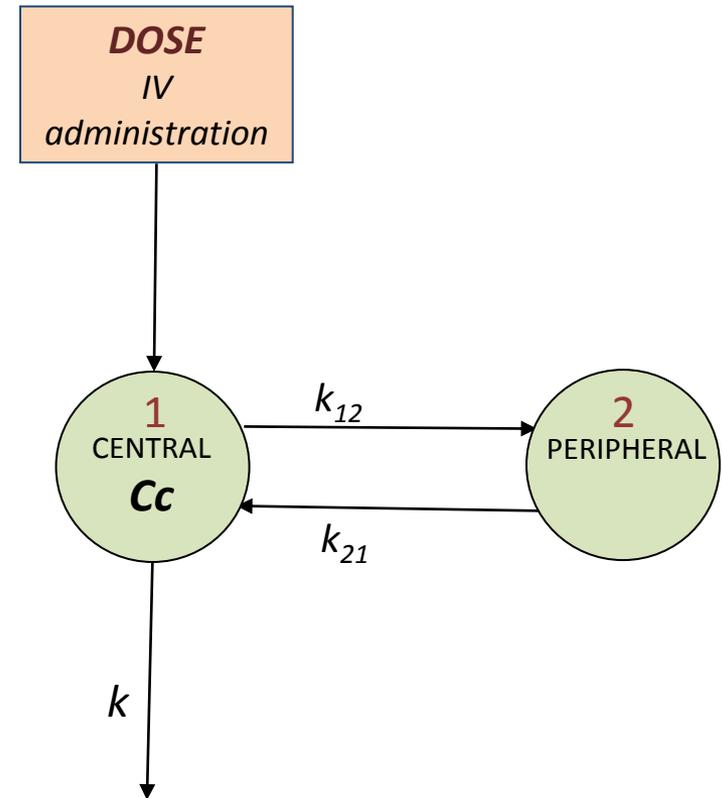
iv()

peripheral(k₁₂, k₂₁)

elimination(k)

OUTPUT:

output = C_c



PK macro **peripheral** can also be used to introduce a peripheral compartment in the model.

DESCRIPTION:

PK model, IV administration,
2 compartments,
linear elimination,

INPUT:

parameter = {V, k, k₁₂, k₂₁}

PK:

compartment(amount=Ac)
iv()

EQUATION:

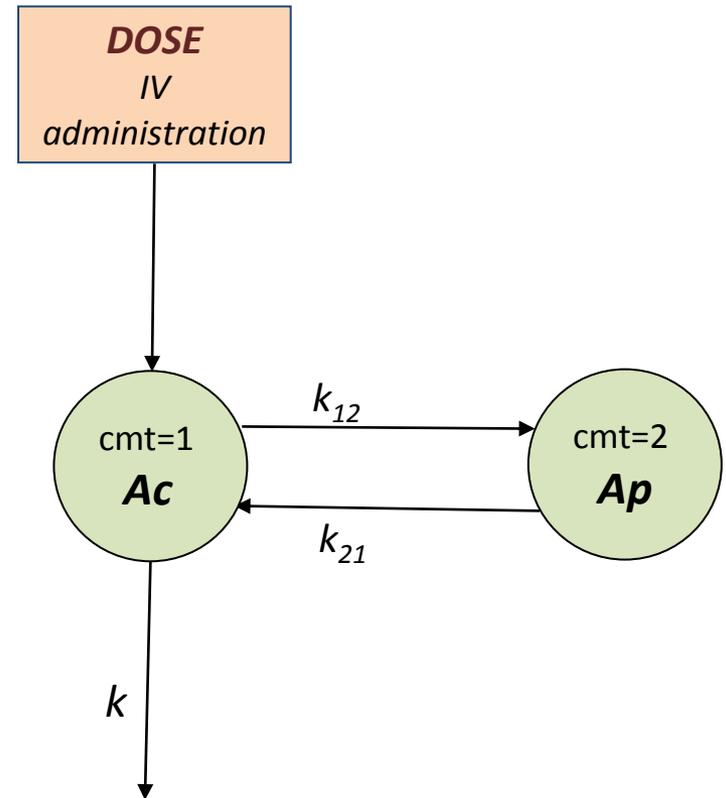
$$\text{ddt_Ac} = -k_{12} \cdot \text{Ac} + k_{21} \cdot \text{Ap} - k \cdot \text{Ac}$$

$$\text{ddt_Ap} = k_{12} \cdot \text{Ac} - k_{21} \cdot \text{Ap}$$

$$\text{Cc} = \text{Ac}/V$$

OUTPUT:

output = Cc



Transfers between the central and the peripheral compartments can be described with a system of (differential) equations in the block **EQUATION**.

DESCRIPTION:

PK model, IV administration,
3 compartments,
linear elimination,

INPUT:

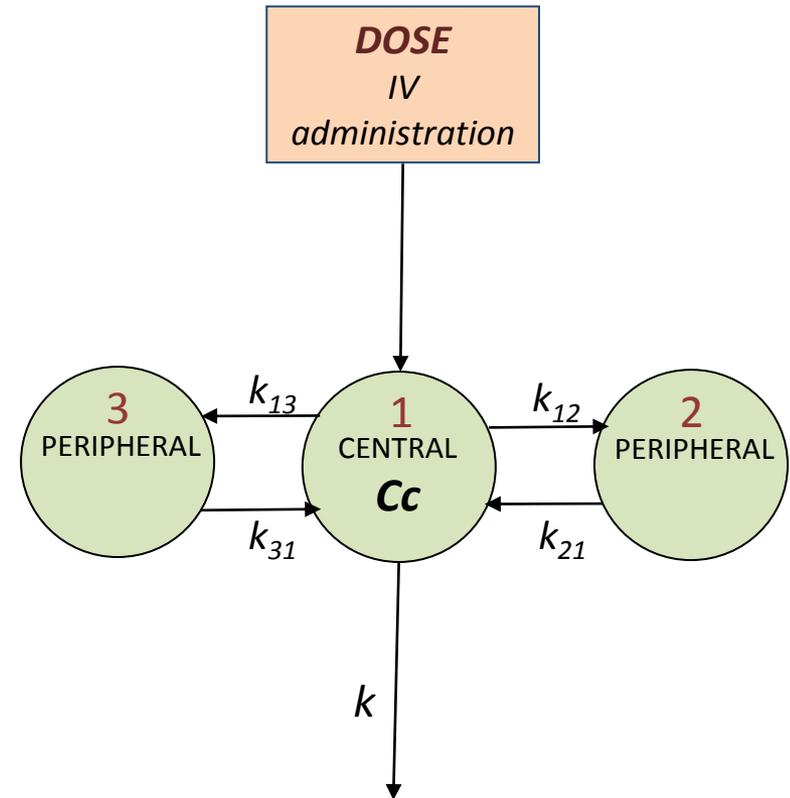
parameter = {V, k, k₁₂, k₂₁, k₁₃, k₃₁}

EQUATION:

$C_c = \text{pkmodel}(V, k, k_{12}, k_{21}, k_{13}, k_{31})$

OUTPUT:

output = C_c



Function **pkmodel** introduces two peripheral compartments in the model when parameters k_{12} , k_{21} , k_{13} and k_{31} are used.

DESCRIPTION:

PK model, IV administration,
3 compartments,
linear elimination,

INPUT:

parameter = {V, k, k₁₂, k₂₁, k₁₃, k₃₁}

PK:

compartment(concentration=C_c, volume=V)

iv()

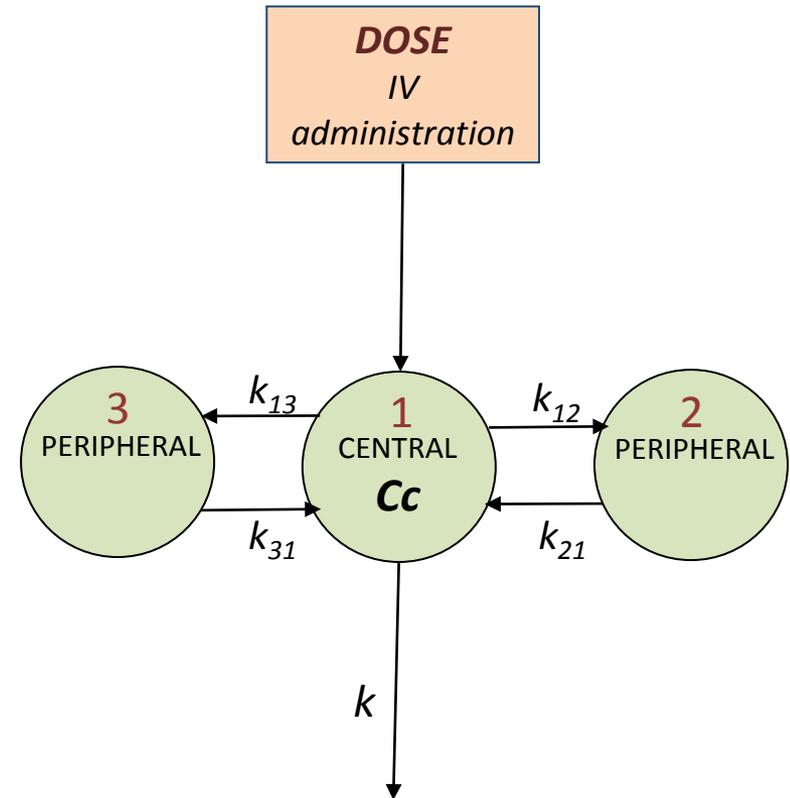
peripheral(k₁₂, k₂₁)

peripheral(k₁₃, k₃₁)

elimination(k)

OUTPUT:

output = C_c



PK macro **peripheral** can also be used to introduce these two peripheral compartments in the model.

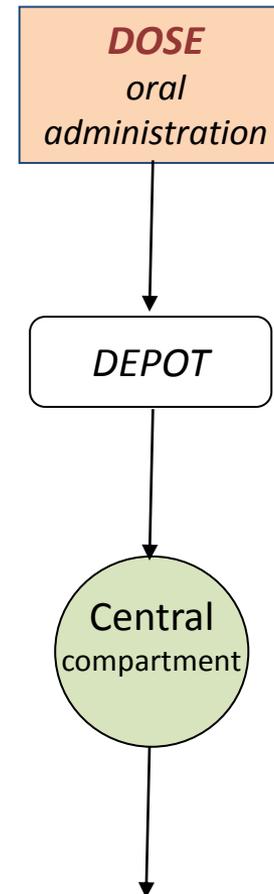
MONOLIX 4.2

MLXTRAN

2. MLXTRAN for PK models

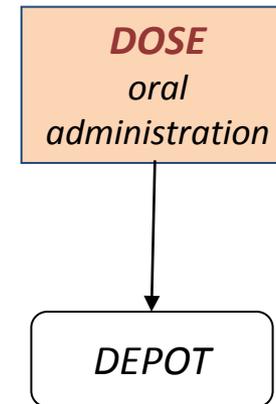
2.2 Oral administration

- *The data* 44
- *First order absorption* 46
- *Some extensions* 56
- *Zero-order absorption* 62
- *Sequential absorptions* 65



The data file

ID	TIME	AMT	Y
1	0	500	.
1	2	.	48
1	6	.	32.8
1	12	.	18.1
1	18	.	6.46
1	24	.	3.06
2	0	500	.
2	2	.	29.5
2	6	.	41
2	12	.	25
2	18	.	16
2	24	.	5.69
3	0	500	.
3	2	.	21.5
3	6	.	28.9
3	12	.	25.3
3	18	.	8.28
3	24	.	6.58



The datafile only contains information about the administration, not about the absorption process which will be described in the model.

Then, the same datafile can be used with different absorption processes (zero-order, first order, sequential zero-order & first order,...)

For a unique type of administration, a unique depot compartment is assumed. Then, only the amount and the time of administration are required in the datafile.

First order absorption process

6 different solutions for coding the same PK model

DESCRIPTION:

PK model, oral administration, first order absorption
linear elimination, parameters (ka, V, k)

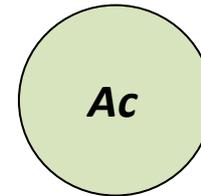
VERSION 1

INPUT:

parameter = {ka, V, k}

PK:

compartment(amount=Ac)



Block **PK** is used to define the PK model

- **compartment(amount=Ac)** creates a central compartment which amount is **Ac**,

DESCRIPTION:

PK model, oral administration, first order absorption
linear elimination, parameters (k_a , V , k)

VERSION 1

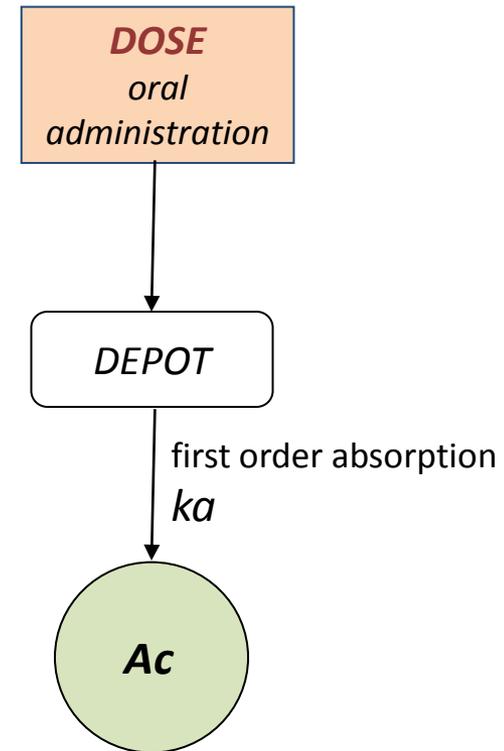
INPUT:

parameter = { k_a , V , k }

PK:

compartment(amount= A_c)

oral(k_a)



Block **PK** is used to define the PK model

- **compartment(amount= A_c)** creates a central compartment which amount is A_c ,
- **oral(k_a)** indicates an oral administration and defines a first order absorption in the central compartment,

DESCRIPTION:

PK model, oral administration, first order absorption
linear elimination, parameters (k_a , V , k)

VERSION 1

INPUT:

parameter = { k_a , V , k }

PK:

compartment(amount= A_c)

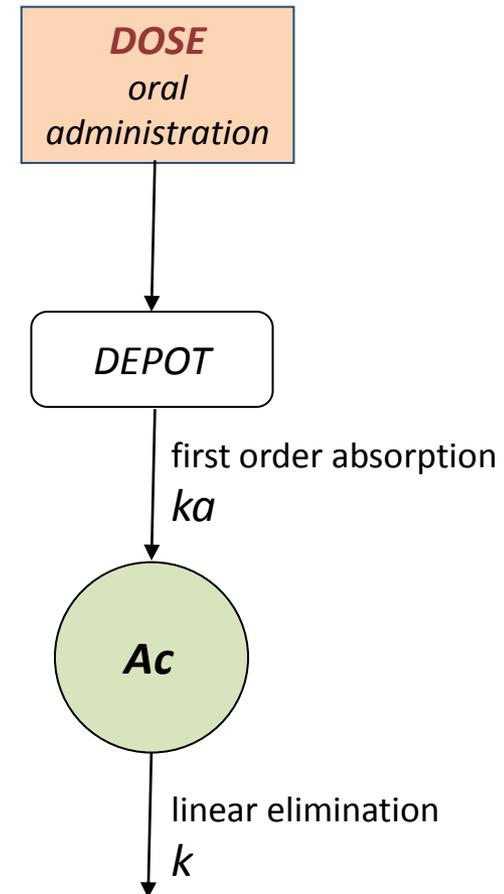
oral(k_a)

elimination(k)

$C_c = A_c/V$

OUTPUT:

output = C_c



Block **PK** is used to define the PK model

- **compartment(amount= A_c)** creates a central compartment which amount is A_c ,
- **oral(k_a)** indicates an oral administration and defines a first order absorption in the central compartment,
- **elimination(k)** defines a linear elimination from the central compartment with rate constant k .

DESCRIPTION:

PK model, oral administration, first order absorption
linear elimination, parameters (k_a , V , k)

VERSION 2

INPUT:

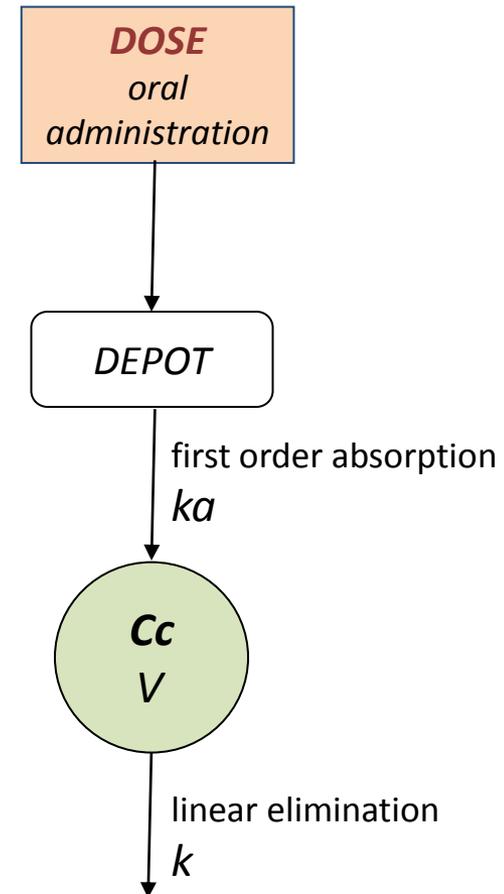
parameter = $\{k_a, V, k\}$

EQUATION:

$C_c = \text{pkmodel}(k_a, V, k)$

OUTPUT:

output = C_c



Function **pkmodel** can be used to define most standard PK models.

The PK model is defined according to the arguments of function **pkmodel**.

Here, **pkmodel(k_a , V , k)** assumes a first order absorption with constant rate k_a and a linear elimination with rate constant k .

DESCRIPTION:

PK model, oral administration, first order absorption
linear elimination, parameters (ka, V, k)

VERSION 3

INPUT:

parameter = {ka, V, k}

PK:

compartment(amount=Ac)

oral(ka)

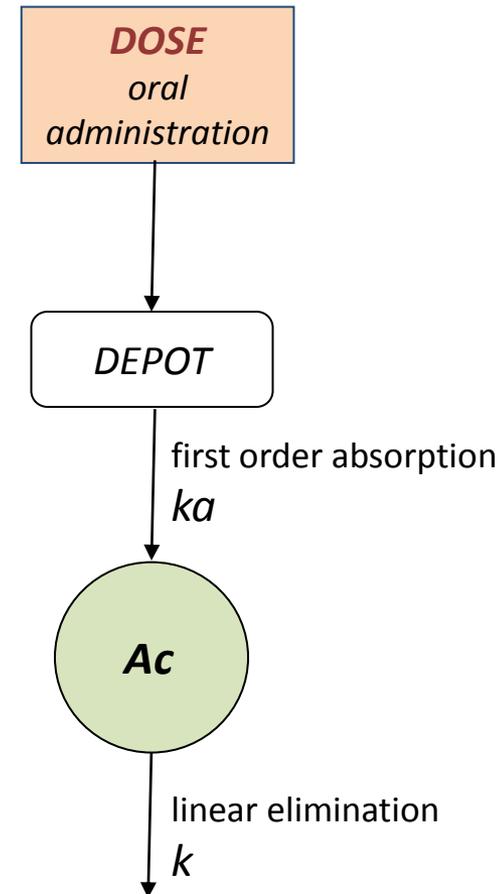
EQUATION:

$\text{ddt_Ac} = -k * \text{Ac}$

$C_c = \text{Ac} / V$

OUTPUT:

output = Cc



Some components of the PK model can be described using equations in a block **EQUATION**.

Here,

- the absorption rate is defined by the PK macro **oral(ka)**
- the elimination rate is defined by the ODE **ddt_Ac = -k*Ac**

DESCRIPTION:

PK model, oral administration, first order absorption
linear elimination, parameters (k_a , V , k)

VERSION 4

INPUT:

parameter = { k_a , V , k }

PK:

compartment(cmt=1, amount= A_d)

compartment(cmt=2, amount= A_c)

iv(cmt=1)

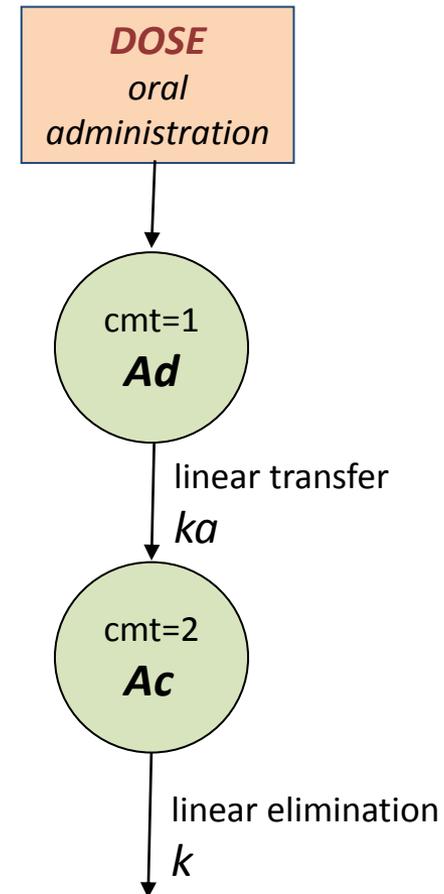
transfer(from=1, to=2, $k_t=k_a$)

elimination(cmt=2, k)

$C_c = A_c/V$

OUTPUT:

output = C_c



An oral administration can be described as an IV bolus in the depot compartment (cmt=1) and a linear transfer from the depot compartment (cmt=1) to the central compartment (cmt=2)

DESCRIPTION:

PK model, oral administration, first order absorption
linear elimination, parameters (ka, V, k)

VERSION 5

INPUT:

parameter = {ka, V, k}

PK:

compartment(cmt=1, amount=Ad)

iv(cmt=1)

EQUATION:

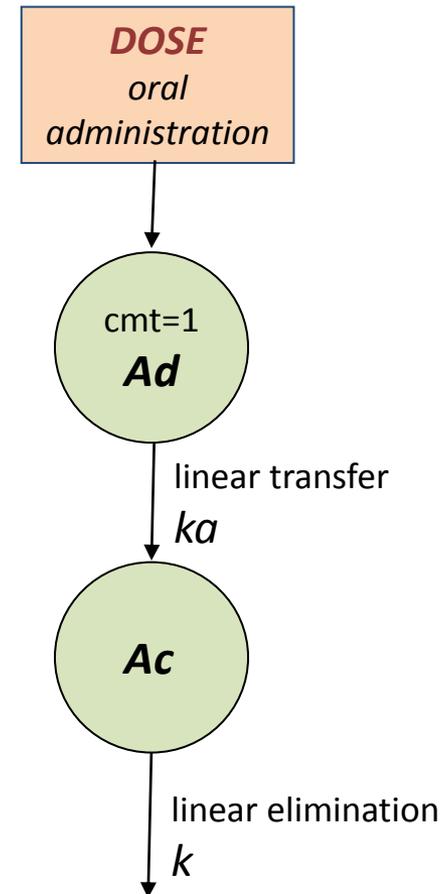
$$\text{ddt_Ad} = -ka * \text{Ad}$$

$$\text{ddt_Ac} = ka * \text{Ad} - k * \text{Ac}$$

$$\text{Cc} = \text{Ac} / V$$

OUTPUT:

output = Cc



Instead of using the PK macros, (differential) equations in the block **EQUATION** can be used to describe the transfer between compartments and the elimination from the central compartment.

Here, only the first compartment which receives the dose needs to be created with the PK macro **compartment**.

DESCRIPTION:

PK model, oral administration, first order absorption
linear elimination, parameters (ka, V, k)

VERSION 6

INPUT:

parameter = {ka, V, k}

PK:

depot(target=Ad)

EQUATION:

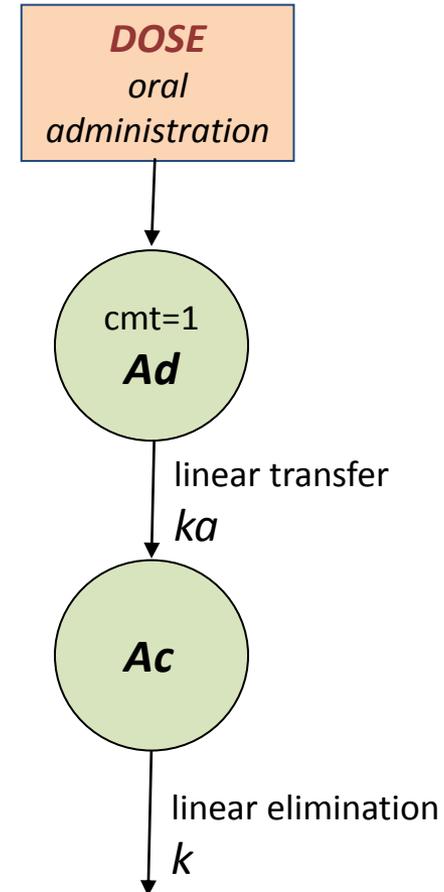
$$\text{d}d\text{t_Ad} = -k\text{a} \cdot \text{Ad}$$

$$\text{d}d\text{t_Ac} = k\text{a} \cdot \text{Ad} - k \cdot \text{Ac}$$

$$\text{Cc} = \text{Ac}/V$$

OUTPUT:

output = Cc



Instead of using the PK macros **compartment** & **iv**, it is possible to consider the doses as source terms for the autonomous system defined with ODEs.

depot(target=Ad) means that **Ad** is the target of the source term :

$$\text{Ad}(t\text{Dose}^+) = \text{Ad}(t\text{Dose}^-) + \text{amtDose}$$

DESCRIPTION:

PK model, oral administration, first order absorption
linear elimination, parameters (ka, V, k)

VERSION 7

INPUT:

parameter = {ka, V, k}

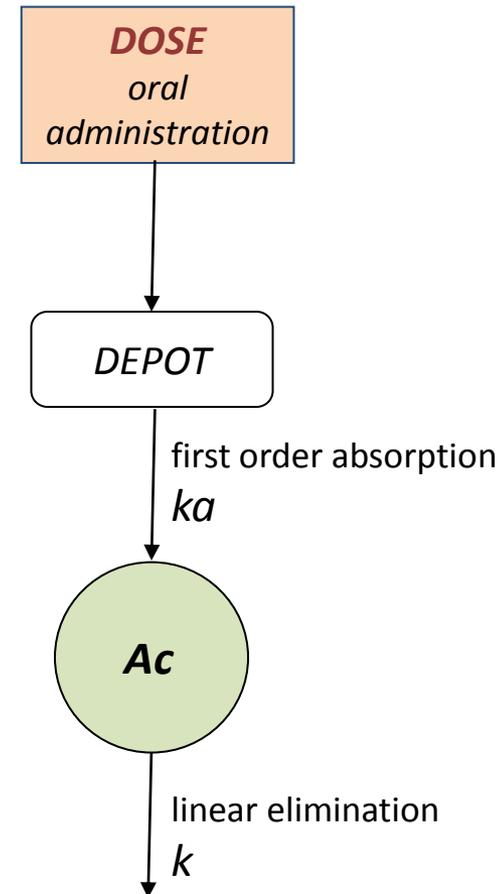
EQUATION:

$dt = t - tDose$

$Cc = amtDose * ka / (V * (ka - k)) * (exp(-k * dt) - exp(-ka * dt))$

OUTPUT:

output = Cc



The analytical expression of **Cc** for a single dose administration can be used in the block **EQUATION**, **tDose** and **amtDose** are reserved keywords:

- **tDose** is the time of the last dose (if **tDose**=0 for all subjects, then it can be omitted).
- **amtDose** is the amount of drug administrated at time **tDose**.

Some extensions:

- *bioavailability*
- *lag-time*
- *transit compartment*
- *Non linear elimination*
- *Multiple compartments*

DESCRIPTION:

PK model, oral administration, bioavailability F , first order absorption, linear elimination,

INPUT:

parameter = $\{F, ka, V, k\}$

PK:

compartment(amount= A_c)

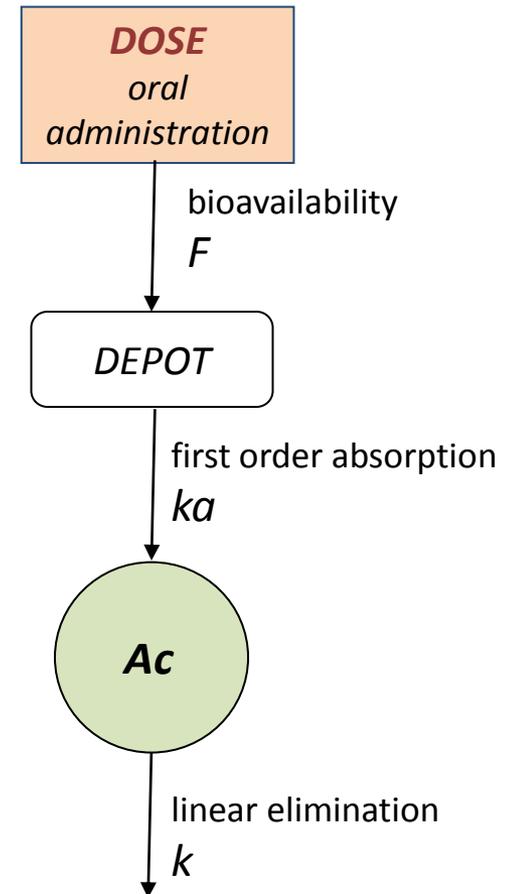
oral($ka, p=F$)

elimination(k)

$C_c = A_c/V$

OUTPUT:

output = C_c



p is a reserved keyword used by the PK macro **absorption** and the function **pkmodel** to specify which fraction of the dose is absorbed.

Thus, **oral($ka, p=F$)** can be used to specify the bioavailability F .

DESCRIPTION:

PK model, oral administration, lag time, first order absorption, linear elimination,

INPUT:

parameter = {Tlag, ka, V, k}

PK:

compartment(amount=Ac)

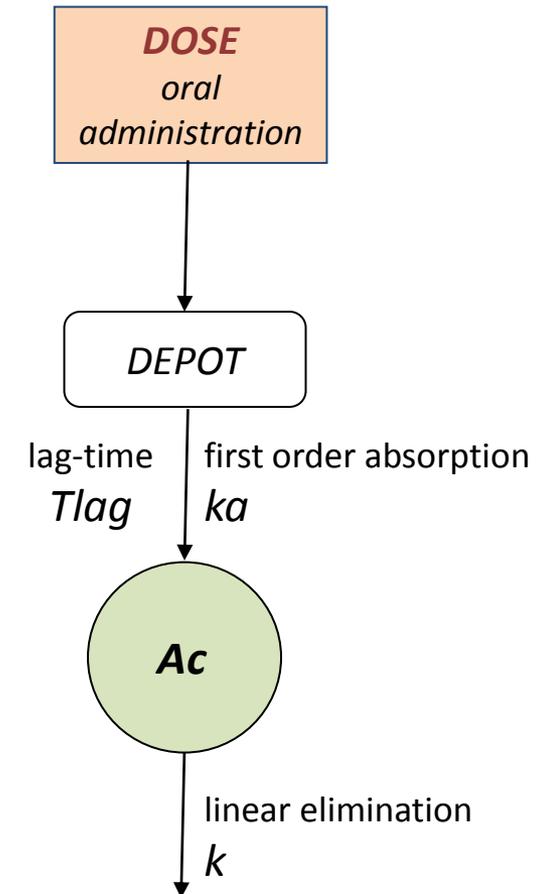
oral(Tlag, ka)

elimination(k)

$C_c = A_c/V$

OUTPUT:

output = C_c



Tlag is a reserved keyword used by the PK macro **oral** and the function **pkmodel** to introduce a lag-time in the absorption process.

DESCRIPTION:

PK model, oral administration, transit compartment, first order absorption, linear elimination,

INPUT:

parameter = {Mtt, Ktr, ka, V, k}

PK:

compartment(amount=Ac)

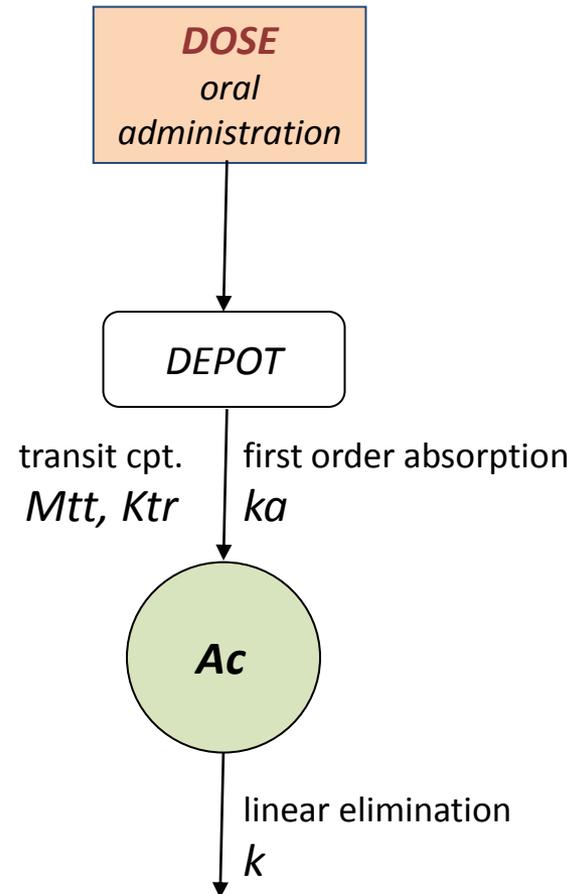
oral(Mtt, Ktr, ka)

elimination(k)

$C_c = A_c/V$

OUTPUT:

output = C_c



Mtt and **Ktr** are reserved keyword used by the PK macro **oral** and the function **pkmodel** to introduce a transit compartment in the absorption process.

Mtt is the mean transit time and **Ktr** the transit rate constant.

DESCRIPTION:

PK model, oral administration, bioavailability F ,
lag-time, first order absorption
non linear elimination,

INPUT:

parameter = $\{F, Tlag, ka, V, k_{12}, k_{21}, Vm, Km\}$

PK:

compartment(concentration= Cc , volume= V)

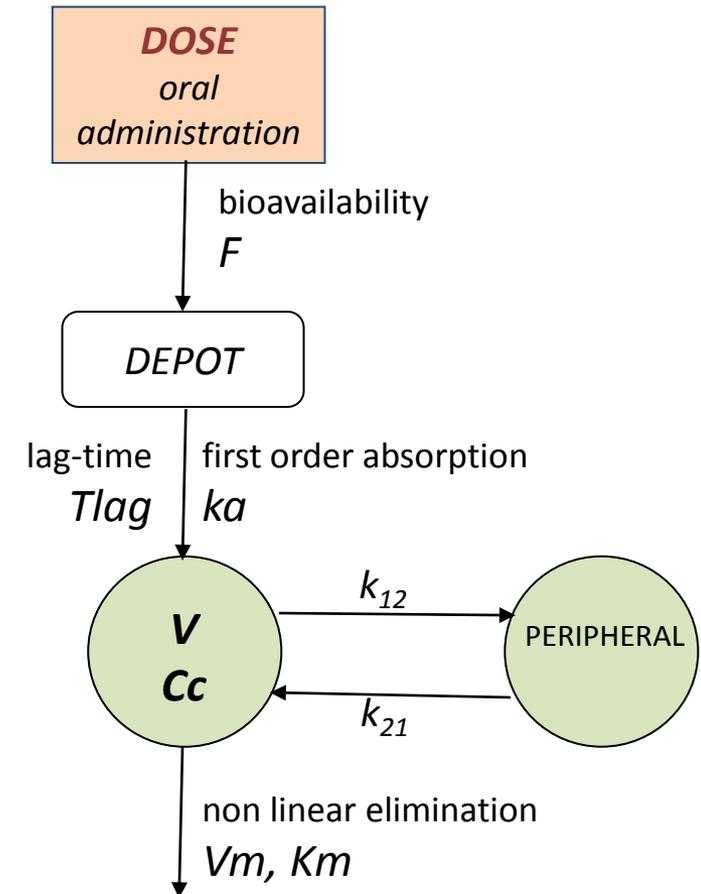
oral($Tlag, ka, p=F$)

peripheral(k_{12}, k_{21})

elimination(Vm, Km)

OUTPUT:

output = Cc



The final PK model combines any absorption, distribution and elimination processes defined by the PK macros **oral**, **peripheral** and **elimination**.

DESCRIPTION:

PK model, oral administration, bioavailability F ,
lag-time, first order absorption
non linear elimination,

INPUT:

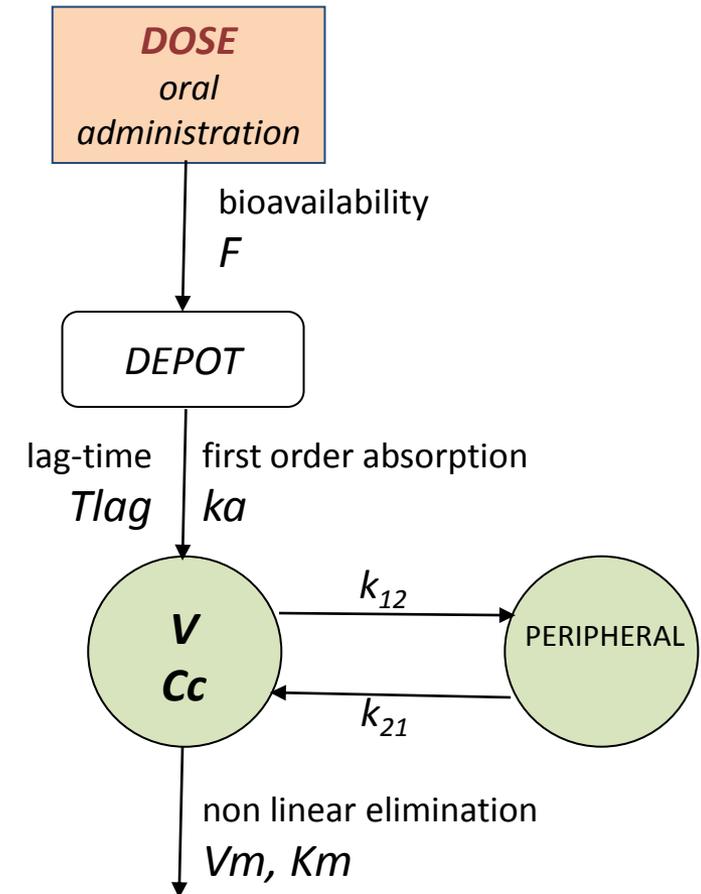
parameter = $\{F, Tlag, ka, V, k_{12}, k_{21}, Vm, Km\}$

EQUATION:

$Cc = \text{pkmodel}(Tlag, ka, p=F, V, k_{12}, k_{21}, Vm, Km)$

OUTPUT:

output = Cc



The final PK model can be equivalently defined using the **pkmodel** function.

Zero order absorption process

DESCRIPTION:

PK model, oral administration, zero order absorption
linear elimination, parameters (Tk_0 , V , k)

INPUT:

parameter = { Tk_0 , V , k }

PK:

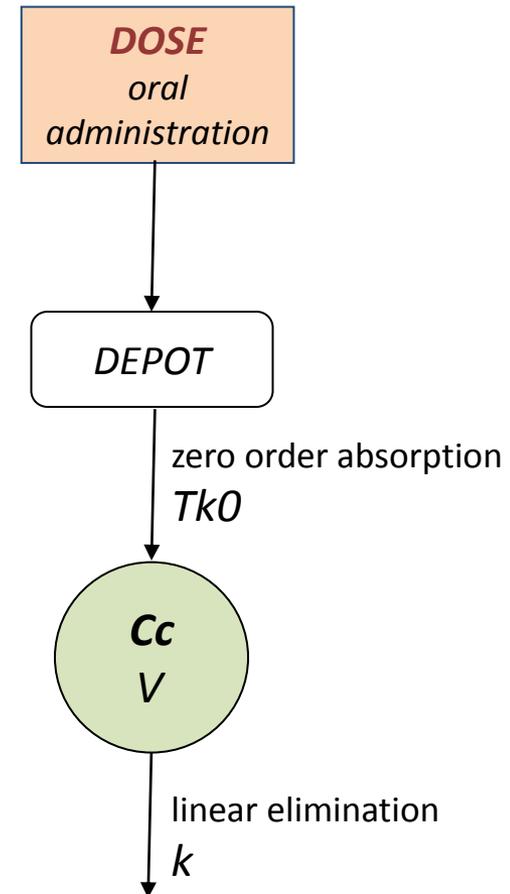
compartment(concentration= C_c , volume= V)

oral(Tk_0)

elimination(k)

OUTPUT:

output = C_c



Tk₀ is a reserved keyword used by the PK macro **oral** and the function **pkmodel** to specify the duration of a zero-order absorption process.

Remark: the structure of the datafile is the same for a first order absorption or a zero order absorption.

DESCRIPTION:

PK model, oral administration, zero order absorption
linear elimination, parameters ($Tk0$, V , k)

INPUT:

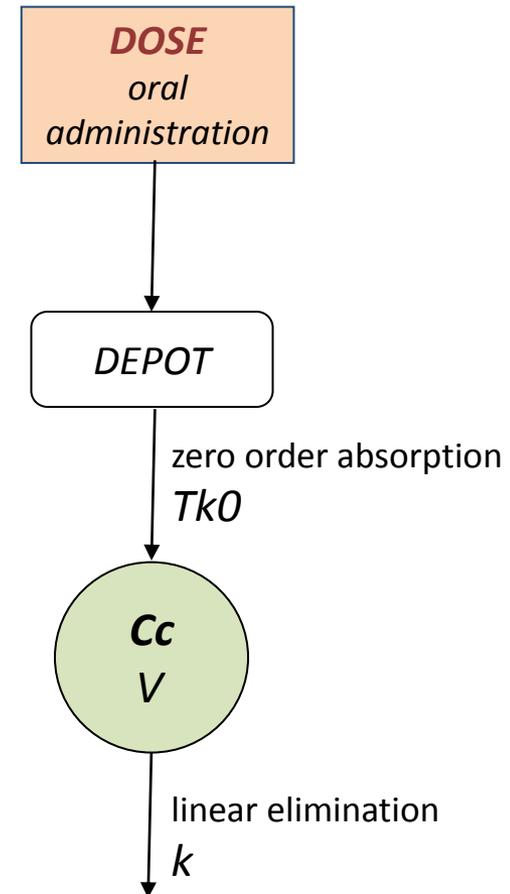
parameter = { $Tk0$, V , k }

EQUATION:

$Cc = \text{pkmodel}(Tk0, V, k)$

OUTPUT:

output = Cc



Tk0 is a reserved keyword used by the PK macro **oral** and the function **pkmodel** to specify the duration of a zero-order absorption process.

Remark: the structure of the datafile is the same for a first order absorption or a zero order absorption.

*Sequential zero order / first order
absorption processes*

DESCRIPTION:

PK model, oral administration,
sequential zero-order & first order absorptions
linear elimination,

INPUT:

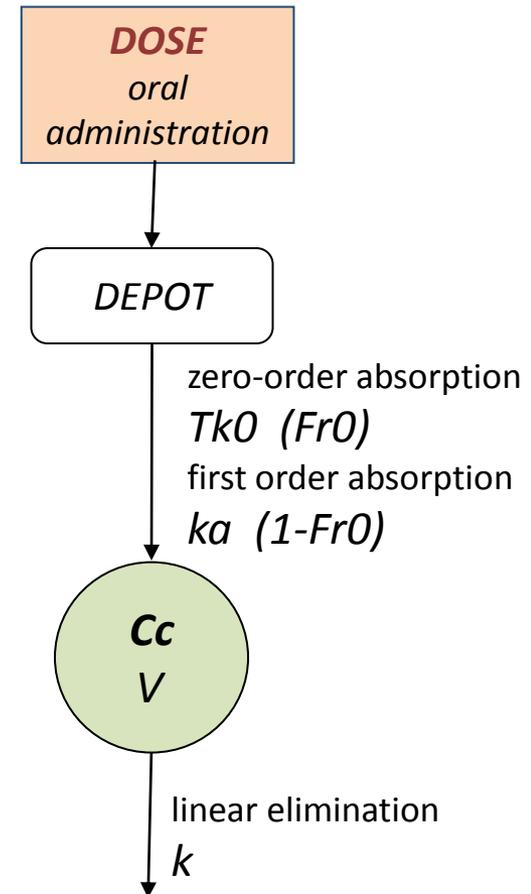
parameter = {Fr0, Tk0, ka, V, k}

PK:

compartment(concentration=Cc, volume=V)
oral(Tk0, p=Fr0)
oral(ka, Tlag=Tk0, p=1-Fr0)
elimination(k)

OUTPUT:

output = Cc



PK macro **absorption** allows to easily describe complex absorption processes. Here,

- **oral(Tk0,p=Fr)** means that a fraction **Fr** is first absorbed with a zero-order process,
- **oral(ka, Tlag=Tk0, p=1-Fr)** means that the remaining fraction **1-Fr** is absorbed with a first order process when the zero-order process ends.

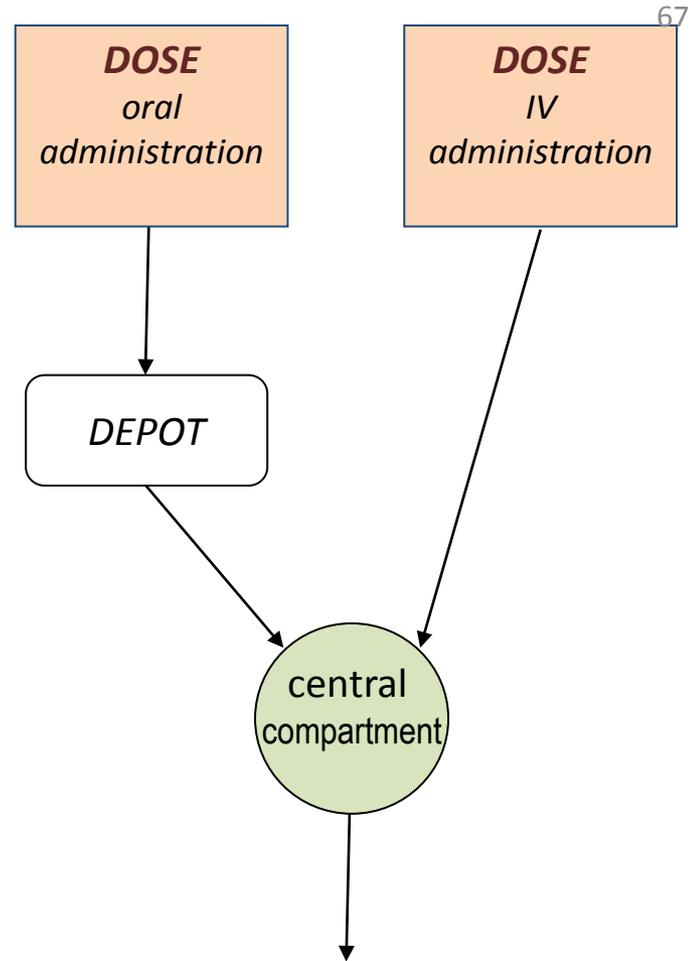
Remark: the structure of the datafile is the same for any absorption process.

MONOLIX 4.2

MLXTRAN

2. MLXTRAN for PK models

2.3 Complex administration



Id	time	adm	amt	Y
1	0	2	2.24	.
1	1	.	.	142
1	2	.	.	54.9
1	3	.	.	25.9
1	4	.	.	17.5
1	6	1	7	.
1	7	.	.	192
1	8	.	.	141
1	9	.	.	189
1	10	.	.	133
2	0	2	2.73	.
2	1	.	.	176
2	2	.	.	69.3
2	3	.	.	30.7
2	4	.	.	19.6
2	6	1	7	.
2	7	.	.	386
2	8	.	.	220
2	9	.	.	98.6
2	10	.	.	47.3

DOSE
oral
administration
adm = 1



DOSE
IV
administration
adm = 2



An additional column **adm** is used to specify the different types of administration.

Here, **adm=1** indicates an oral administration and **adm=2** an IV bolus administration.

The datafile only contains information about the administration, not about the absorption process which will be described in the model.

DESCRIPTION:

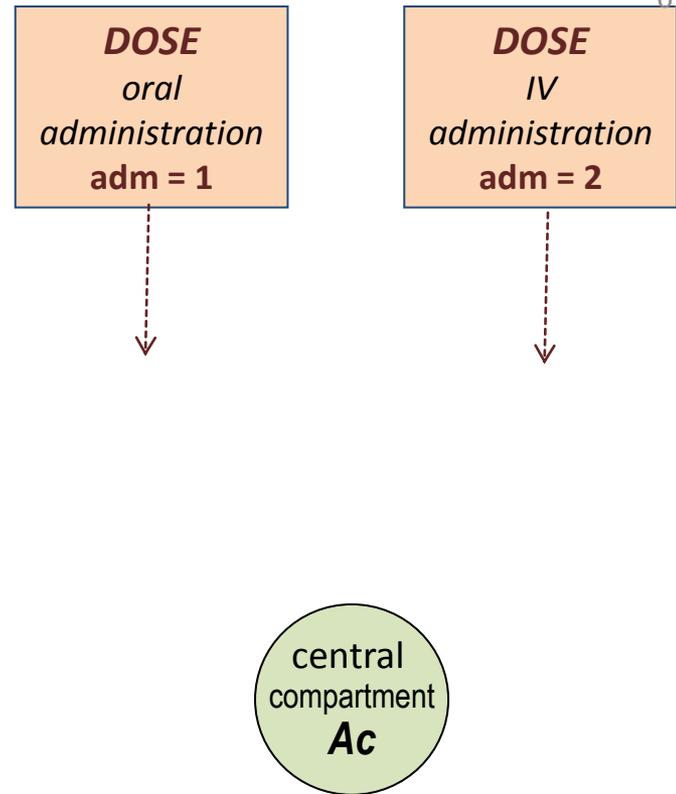
PK model, combination of oral and IV administrations, first order absorption, linear elimination,

INPUT:

parameter = {F, ka, V, k}

PK:

compartment(amount=Ac)



Block **PK** is used to define the PK model

- **compartment(amount=Ac)** creates a central compartment which amount is **Ac**,

DESCRIPTION:

PK model, combination of oral and IV administrations, first order absorption, linear elimination,

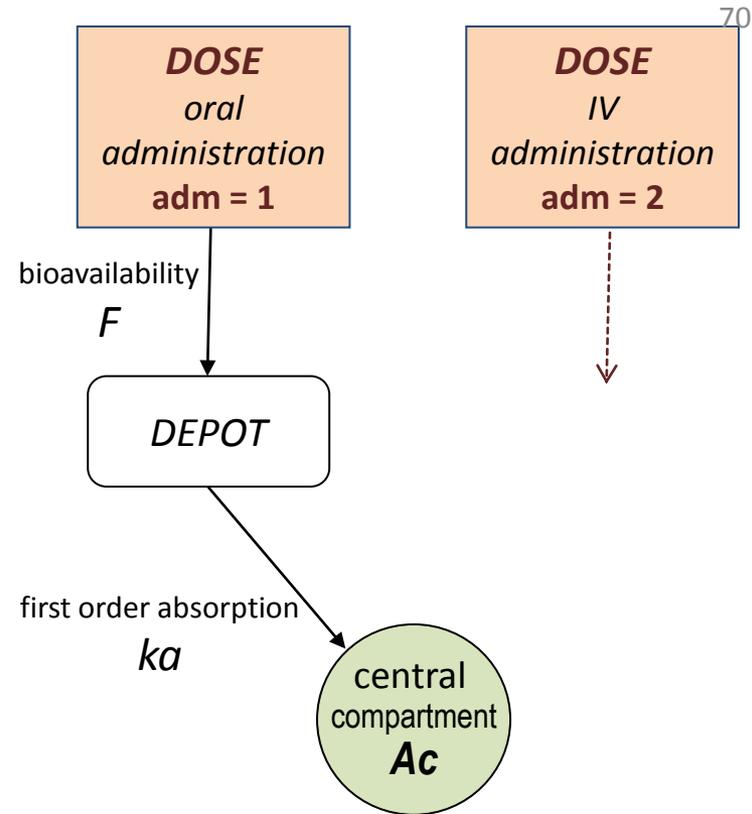
INPUT:

parameter = { F , ka , V , k }

PK:

compartment(amount= Ac)

oral(adm=1, ka , $p=F$)



Block **PK** is used to define the PK model

- **compartment(amount= Ac)** creates a central compartment which amount is Ac ,
- **oral(adm=1, ka , $p=F$)** indicates an oral administration from **adm=1** with a bioavailability F and defines a first order absorption in the central compartment.

DESCRIPTION:

PK model, combination of oral and IV administrations, first order absorption, linear elimination,

INPUT:

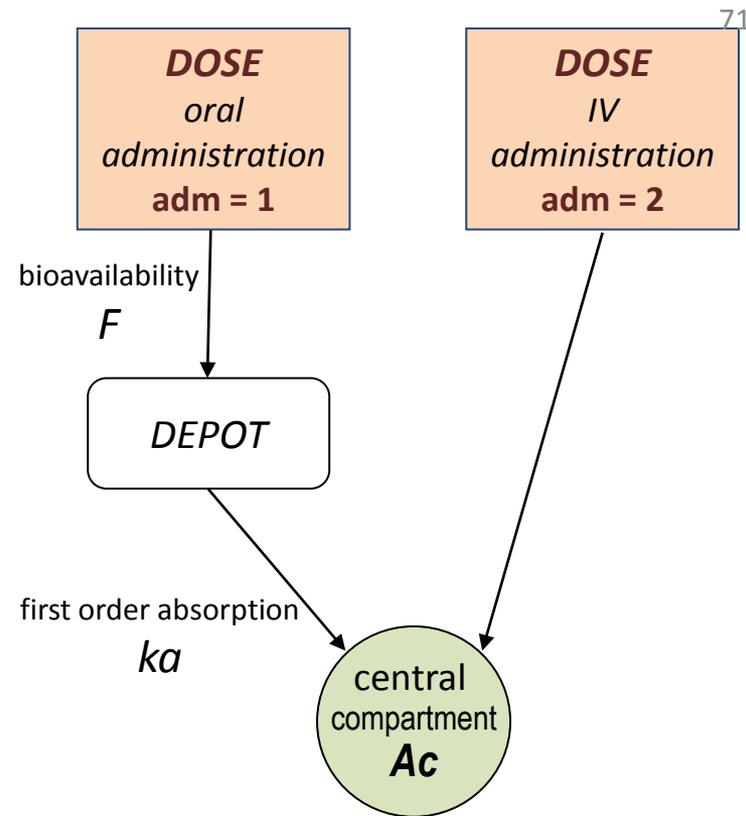
parameter = { F , k_a , V , k }

PK:

compartment(amount= A_c)

oral(adm=1, k_a , $p=F$)

iv(adm=2)



Block **PK** is used to define the PK model

- **compartment(amount= A_c)** creates a central compartment which amount is A_c ,
- **oral(adm=1, k_a , $p=F$)** indicates an oral administration from **adm=1** with a bioavailability F and defines a first order absorption in the central compartment.
- **IV(adm=2)** indicates a IV administration from **adm=2**,

DESCRIPTION:

PK model, combination of oral and IV administrations, first order absorption, linear elimination,

INPUT:

parameter = { F , ka , V , k }

PK:

compartment(amount= Ac)

oral(adm=1, ka , $p=F$)

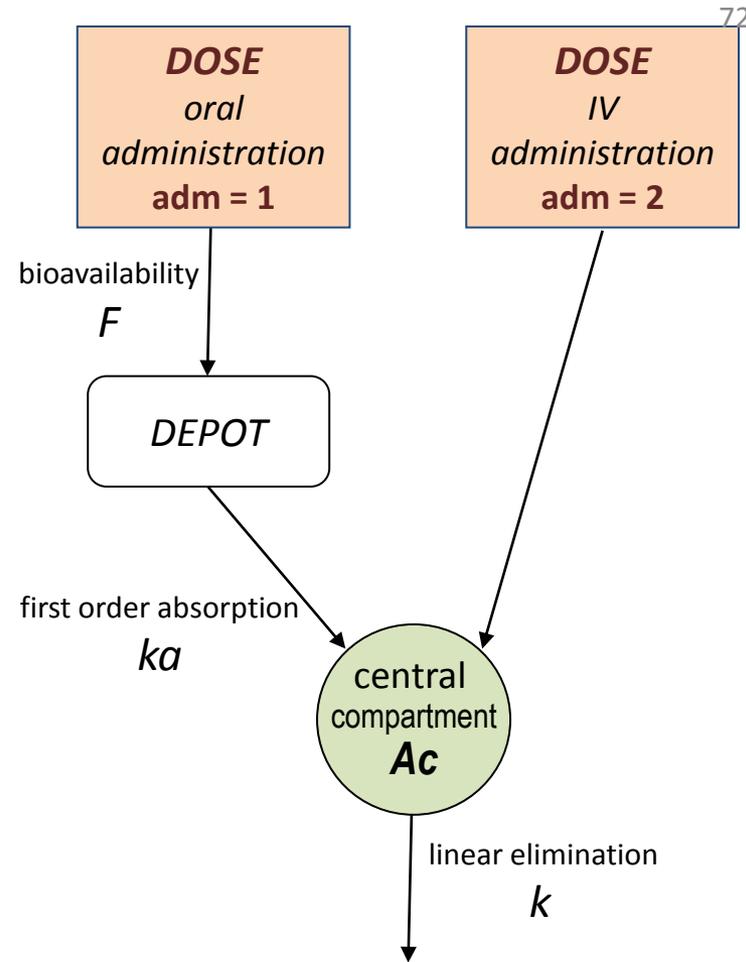
iv(adm=2)

elimination(k)

$C_c = Ac/V$

OUTPUT:

output = C_c



Block **PK** is used to define the PK model

- **compartment(amount= Ac)** creates a central compartment which amount is Ac ,
- **oral(adm=1, ka , $p=F$)** indicates an oral administration from **adm=1**, defines a first order absorption in the central compartment with a bioavailability F
- **IV(adm=2)** indicates a IV administration from **adm=2**,
- **elimination(k)** defines a linear elimination from the central compartment.

DESCRIPTION:

PK model, combination of oral and IV administrations, first order absorption, linear elimination,

INPUT:

parameter = {F, ka, V, k}

PK:

compartment(cmt=1, amount=Ad)

compartment(cmt=2, amount=Ac)

iv(adm=1, cmt=1, p=F)

iv(adm=2, cmt=2)

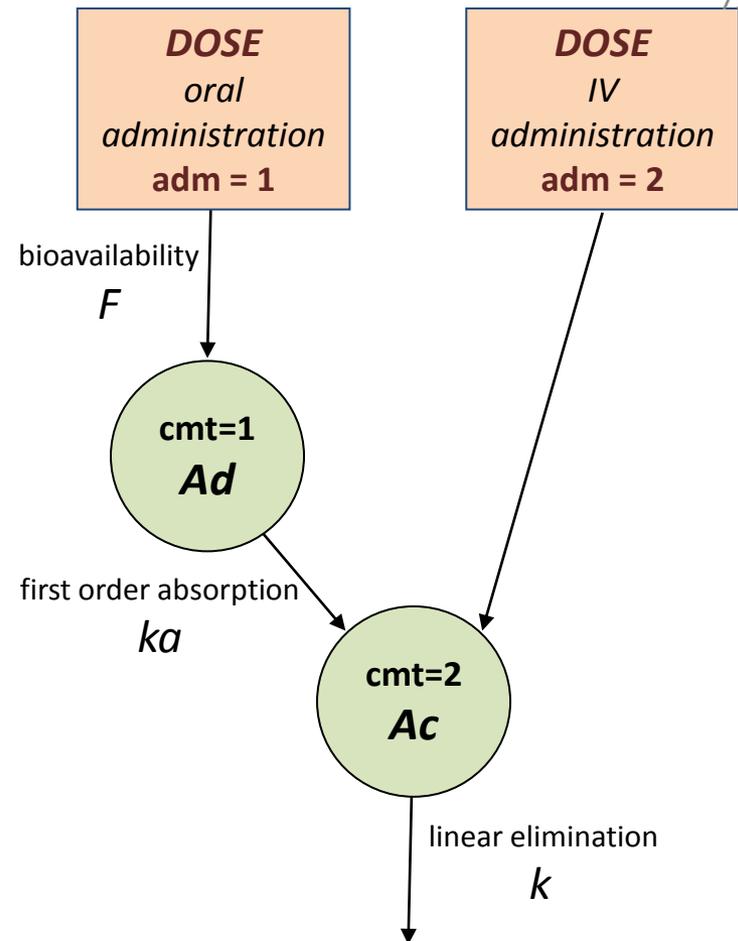
EQUATION:

$$\text{ddt_Ad} = -ka * \text{Ad}$$

$$\text{ddt_Ac} = ka * \text{Ad} - k * \text{Ac}$$

$$\text{Cc} = \text{Ac} / V$$
OUTPUT:

output = Cc



The same PK model can be described using a complete system of differential equations. Here block **PK** is used to create the compartments and associate the different administrations to the compartments.

- **compartment(cmt=1, amount=Ad)** creates a depot compartment which amount is **Ad**,
- **compartment(cmt=2, amount=Ac)** creates a central compartment which amount is **Ac**,
- **iv(adm=1, cmt=1, p=F)** indicates an IV bolus admin. from **adm=1** to **cmt=1** with a bioavailability **F**
- **iv(adm=2, cmt=2)** indicates an IV bolus administration from **adm=2** to **cmt=2**

DESCRIPTION:

PK model, combination of oral and IV administrations, first order absorption, linear elimination,

INPUT:

parameter = {F, ka, V, k}

PK:

depot(adm=1, target=Ad, p=F)

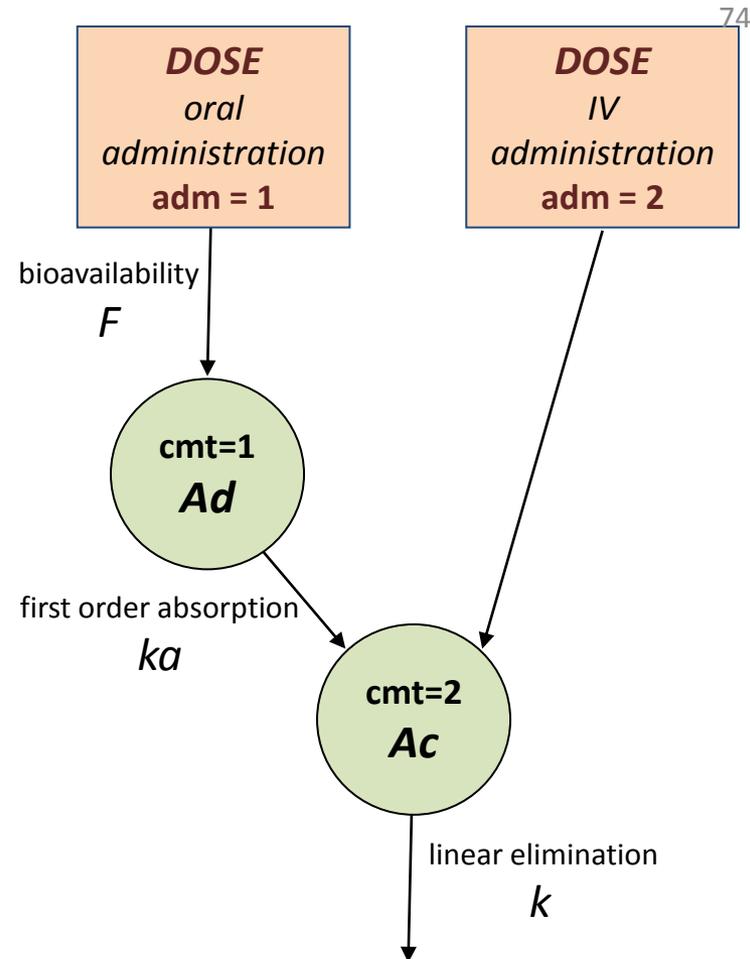
depot(adm=2, target=Ac)

EQUATION:

$$\text{d}dt_Ad = -ka*Ad$$
$$\text{d}dt_Ac = ka*Ad - k*Ac$$
$$Cc=Ac/V$$

OUTPUT:

output = Cc



Instead of using the PK macros **compartment** & **iv**, it is possible to consider the doses as source terms of the autonomous system defined with ODEs.

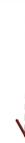
adm and **type** are synonymous : they can be used equivalently.

Id	time	amt	Y	adm
1	0	2	.	3
1	0.5	.	229	.
1	1	.	142	.
1	2	.	54.9	.
1	4	.	17.5	.
1	6	7	.	1
1	6.5	.	8.1	.
1	7	.	192	.
1	8	.	141	.
1	9	.	189	.
1	10	.	133	.
1	12	7	.	2
1	13	.	50	.
1	15	.	201	.
1	18	.	154	.

DOSE
oral
administration
adm = 1

DOSE
oral
administration
adm = 2

DOSE
IV
administration
adm = 3



Extension to any combination of multiple oral and IV administrations is straightforward using the **adm** column in the data file.

DESCRIPTION:

PK model, combin. of oral and IV admin.,
latent compartment,

INPUT:

parameter = {F1, F2, ka1, ka2, kl, V, k1, k2}

PK:

compartment(cmt=1, amount=A1)

compartment(cmt=2, amount=Ac)

oral(adm=1, cmt=1, ka=ka1, p=F1)

oral(adm=2, cmt=2, ka=ka2, p=F2)

iv(adm=3, cmt=2)

transfer(from=1, to=2, kt=kl)

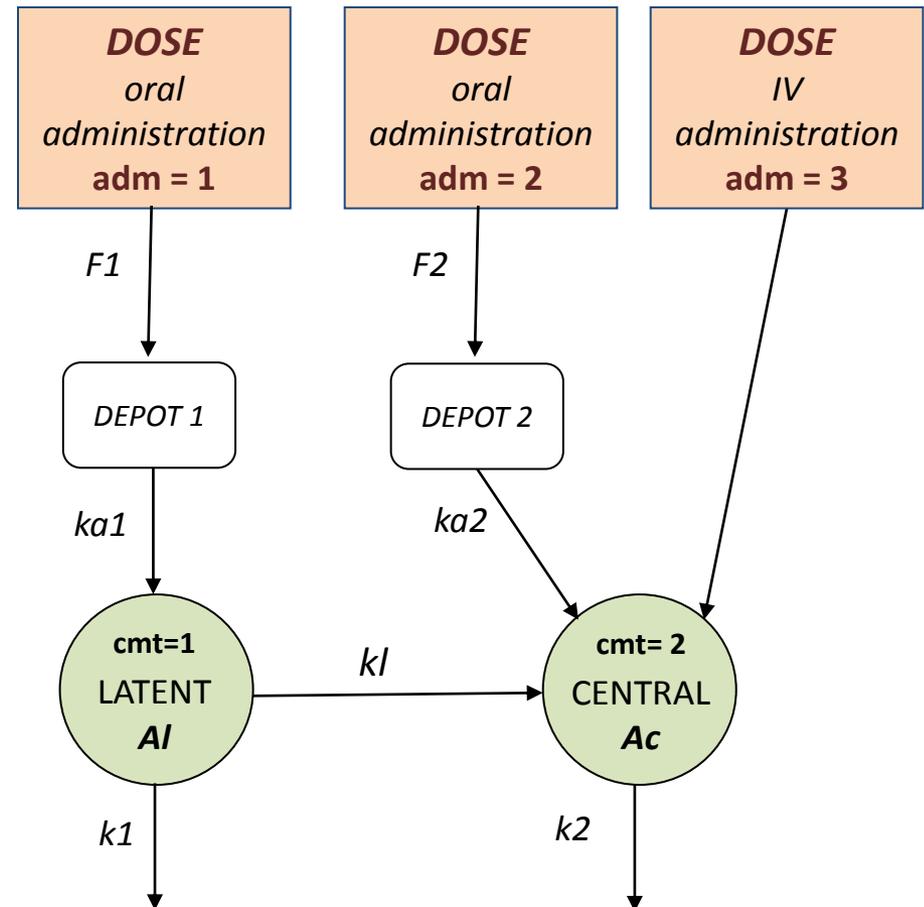
elimination(cmt=1, k=k1)

elimination(cmt=2, k=k2)

$C_c = A_c/V$

OUTPUT:

output = C_c



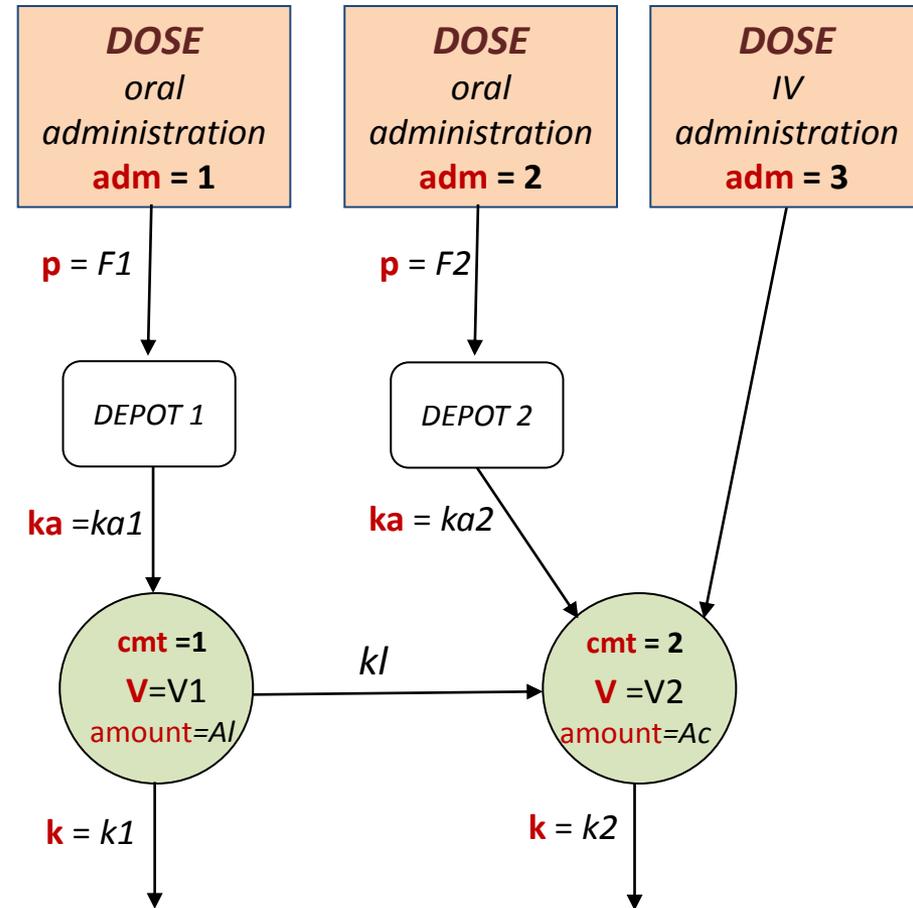
Description of complex PK models is then possible thanks to the use of the PK macros.

MONOLIX 4.2

MLXTRAN

2. MLXTRAN for PK models

2.4 PK parameters summary



List of PK parameters:

adm	Type of administration (link with the data file)
cmt	Compartment number (default =1)
ka	Absorption constant rate (first order absorption)
Tko	Absorption duration (zero order absorption)
Tlag	Lag time before absorption
Mtt, Ktr	Mean transit time & transit rate constant
p	Fraction of dose which is absorbed
V	Central compartment volume
Cl	Central compartment clearance
Vm, Km	Michaelis Menten elimination parameters
k12, k21	Transfer rate constants between compartments 1 (central) & 2 (peripheral)
k13, k31	Transfer rate constants between compartments 1 (central) & 3 (peripheral)
kt	Transfer rate constant between two compartments (used by transfer)
from, to	Source and target compartments (used by transfer)
keo	Effect compartment transfer rate constant
amount	Amount in a compartment (used by compartment)
concentration	concentration in a compartment (volume of compartment V required)

PK macros and their input parameters

Parameters for a call are matched by name, not by ordering

iv	oral	elimination	peripheral	transfer
adm (default = 1) cmt (default = 1)	adm (default = 1) cmt (default = 1)	cmt (default = 1)		from to
	ka or Tko	k or Cl or (Vm , Km) v	(k12 , k21) (k13 , k31)	kt
	Tlag (default = 0) or (Mtt , Ktr) (default = 0)			
	p (default = 1)			

Required parameters

Optional parameters

Remark: **oral** or **absorption** can be used indifferently.

Input parameters of the **pkmodel** function:

ka or **Tko**

Tlag or (**Mtt**, **Ktt**)

p

V

k or **Cl** or (**Km**, **Vm**)

(**k12**, **k21**)

(**k13**, **k31**)

keo

Parameters for a call are matched by name, not by ordering

Output of the **pkmodel** function:

Cc : concentration in the central compartment

Ce : concentration in the effect compartment

Required parameters

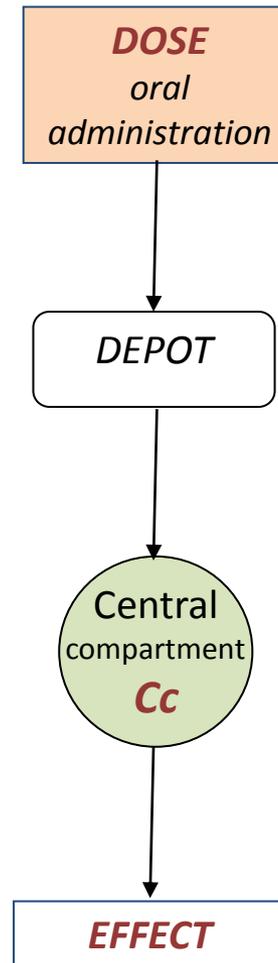
Optional parameters

MONOLIX 4.2

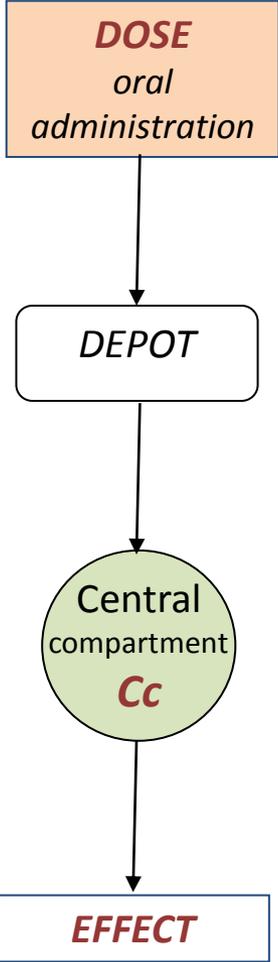
MLXTRAN

3. MLXTRAN for PKPD models

- *Immediate response model* 83
- *Effect compartment* 86
- *Indirect response model* 89
- *Error models* 93



Id	TIME	AMT	Y	DVID
1	0	100	.	.
1	0	.	100	2
1	24	.	9.2	1
1	24	.	49	2
1	36	.	8.5	1
1	36	.	32	2
1	48	.	6.4	1
1	48	.	26	2
1	72	.	4.8	1
1	72	.	22	2
1	96	.	3.1	1
1	96	.	28	2
1	120	.	2.5	1
1	120	.	33	2



The datafile contains an additional column **DVID** (or **YTYPE**) to specify if an observation is a PK data or a PD data

Immediate response models

DESCRIPTION:

PK model for oral administration
Imax model

INPUT:

parameter = {ka, V, Cl, I_{max}, IC₅₀, S₀}

EQUATION:

$C_c = \text{pkmodel}(ka, V, Cl)$

$E = S_0 * (1 - I_{max} * C_c / (C_c + IC_{50}))$

OUTPUT:

output = {C_c, E}

In this example, both PK and PD models are defined in the block **EQUATION**.

We can use the function **pkmodel** for computing the concentration **C_c**.

There are 2 outputs for a PK/PD model.

DESCRIPTION:

PK model for oral administration
Imax model

INPUT:

parameter = {ka, V, Cl, I_{max}, IC₅₀, S₀}

EQUATION:

$$C_c = \text{pkmodel}(ka, V, Cl)$$

$$E = S_0 * (1 - I_{\max} * C_c / (C_c + IC_{50}))$$
OUTPUT:

output = {C_c, E}

DESCRIPTION:

PK model for single oral administration
Imax model

INPUT:

parameter = {ka, V, Cl, I_{max}, IC₅₀, S₀}

EQUATION:

$$k = Cl/V$$

$$p_1 = \text{amtDose} * ka / (V * (ka - k))$$

$$C_c = p_1 * (\exp(-k * t) - \exp(-ka * t))$$

$$E = S_0 * (1 - I_{\max} * C_c / (C_c + IC_{50}))$$
OUTPUT:

output = {C_c, E}

In this example, both PK and PD models are defined in the block **EQUATION**.

We can use the function **pkmodel** for computing the concentration **C_c**.

There are 2 outputs for a PK/PD model.

For a single dose administration, the concentration **C_c** can be computed analytically instead of using the function **pkmodel**

Effect compartment

DESCRIPTION:

PK model for oral administration,
Effect compartment,
Imax model

INPUT:

parameter = {ka, V, Cl, ke0, Imax, IC50, S0}

EQUATION:

{Cc, Ce} = pkmodel(ka, V, Cl, ke0)
 $E = S0 * (1 - Imax * Ce / (Ce + IC50))$

OUTPUT:

output = {Cc, E}

It is easy to add an effect compartment in the model, just by adding a second output to the function **pkmodel**.

Here, **Ce** is the concentration in the effect compartment.

DESCRIPTION:

PK model for oral administration,
Effect compartment,
Imax model

INPUT:

parameter = {ka, V, Cl, ke0, Imax, C50, S0}

EQUATION:

{Cc, Ce} = pkmodel(ka, V, Cl, ke0)
E = S0*(1-Imax*Ce/(Ce+C50))

OUTPUT:

output = {Cc, E}

DESCRIPTION:

PK model for single oral administration,
Effect compartment,
Imax model

INPUT:

parameter = {ka, V, Cl, ke0, Imax, IC50, S0}

EQUATION:

$k = Cl/V$
 $p1 = amtDose*ka/(V*(ka-k))$
 $Cc = p1*(exp(-k*t) - exp(-ka*t))$
 $ddt_Ce = -ke0*(Ce - Cc)$
 $E = S0*(1-Imax*Ce/(Ce+IC50))$

OUTPUT:

output = {Cc, E}

It is easy to add an effect compartment in the model, just by adding a second output to the function **pkmodel**.

Here, **Ce** is the concentration in the effect compartment.

For a single dose administration, the concentration **Cc** can be computed analytically instead of using the function **pkmodel**. Then, the concentration in the effect compartment is computed as the solution of a differential equation. When it is not specified, the initial value of an ODE is 0.

Indirect response models

DESCRIPTION:

PK model for oral administration,
Use of pkmodel,
Indirect response model

INPUT:

parameter = {ka, V, Cl, I_{max}, IC₅₀, R_{in}, k_{out}}

EQUATION:

$C_c = \text{pkmodel}(ka, V, Cl)$

$E_0 = R_{in}/k_{out}$

$\text{ddt}_E = R_{in} * (1 - I_{max} * C_c / (C_c + IC_{50})) - k_{out} * E$

OUTPUT:

output = {C_c, E}

In a turnover model, the effect **E** is function of the concentration **C_c** and is defined as the solution of an ODE.

Here, the initial value of **E** is different from 0 and must be defined as **E₀**

The concentration **C_c** can be computed using the function **pkmodel**

DESCRIPTION:

PK model for oral administration,
Use of pkmodel,
Indirect response model

INPUT:

parameter = {ka, V, Cl, I_{max}, IC₅₀, R_{in}, k_{out}}

EQUATION:

$$C_c = \text{pkmodel}(k_a, V, Cl)$$

$$E_0 = R_{in}/k_{out}$$

$$\text{ddt_E} = R_{in} * (1 - I_{max} * C_c / (C_c + IC_{50})) - k_{out} * E$$
OUTPUT:

output = {C_c, E}

DESCRIPTION:

PK model for oral administration,
Use of ODEs,
Indirect response model

INPUT:

parameter = {ka, V, Cl, I_{max}, IC₅₀, R_{in}, k_{out}}

PK:

compartment(cmt=1, amount=Ad)

iv(cmt=1)

EQUATION:

$$C_c = A_c/V$$

$$E_0 = R_{in}/k_{out}$$

$$\text{ddt_Ad} = -k_a * A_d$$

$$\text{ddt_Ac} = k_a * A_d - Cl/V * A_c$$

$$\text{ddt_E} = R_{in} * (1 - I_{max} * C_c / (C_c + IC_{50})) - k_{out} * E$$
OUTPUT:

output = {C_c, E}

The concentration **C_c** and the effect **E** can also be defined together with a unique ODE system.

DESCRIPTION:

PK model for oral administration,
Use of pkmodel,
Indirect response model

INPUT:

parameter = {ka, V, Cl, I_{max}, IC₅₀, Rin, k_{out}}

EQUATION:

$C_c = \text{pkmodel}(ka, V, Cl)$

$E_0 = Rin/k_{out}$

$ddt_E = Rin \cdot (1 - I_{max} \cdot C_c / (C_c + IC_{50})) - k_{out} \cdot E$

OUTPUT:

output = {C_c, E}

DESCRIPTION:

PK model for single oral administration
Analytical solution of PK
Indirect response model

INPUT:

parameter = {ka, V, Cl, I_{max}, IC₅₀, Rin, k_{out}}

EQUATION:

$k = Cl/V$

$p1 = \text{amtDose} \cdot ka / (V \cdot (ka - k))$

$C_c = p1 \cdot (\exp(-k \cdot t) - \exp(-ka \cdot t))$

$E_0 = Rin/k_{out}$

$ddt_E = Rin \cdot (1 - I_{max} \cdot C_c / (C_c + IC_{50})) - k_{out} \cdot E$

OUTPUT:

output = {C_c, E}

For a single dose administration, the concentration **C_c** can be computed analytically instead of using an ODE or the function **pkmodel**

Error models

INPUT:

parameter = {ka, V, Cl, I_{max}, IC₅₀, R_{in}, k_{out}}

EQUATION:

$$C_c = \text{pkmodel}(ka, V, Cl)$$

$$E_0 = R_{in}/k_{out}$$

$$\text{ddt}_E = R_{in} * (1 - I_{max} * C_c / (C_c + IC_{50})) - k_{out} * E$$
OBSERVATION:

Concentration = {type=continuous,
prediction=C_c, errorModel=proportional}

Effect = {type=continuous, prediction=E,
errorModel=constant, autocorrelation=yes}

OUTPUT:

output = {Concentration, Effect}

INPUT:

parameter = {ka, V, Cl, I_{max}, IC₅₀, R_{in}, k_{out}}

EQUATION:

$$C_c = \text{pkmodel}(ka, V, Cl)$$

$$E_0 = R_{in}/k_{out}$$

$$\text{ddt}_E = R_{in} * (1 - I_{max} * C_c / (C_c + IC_{50})) - k_{out} * E$$
OUTPUT:

output = {C_c, E}

For each continuous observation, the model can define the distribution of the observation, including an error model, or define only its prediction. The autocorrelation is disabled by default.

When the distribution of the observation is available, it must be the output of the model.

MONOLIX 4.2

MLXTRAN

4. MLXTRAN for discrete data models

- *Count data model* 96
- *Categorical data model* 101
- *Categorical data model
with Markovian dependence* 109
- *Time-to-event data model* 113
- *Joint models* 120

Count data model

DESCRIPTION:

count data model - Poisson distribution
 $P(Y=k)$ is defined

INPUT:

parameter = lambda

OBSERVATION:

```
Y = {
type = count
P(Y=k) = exp(-lambda + k*log(lambda) - factln(k))
}
```

OUTPUT:

output = Y

$$P(Y = k) = \frac{e^{-\lambda} \times \lambda^k}{k!}$$

The probability distribution of the count data **Y** is defined in the block **OBSERVATION**, **output=Y** means that the model is used for describing the probability distribution of **Y**. Then, this model can be used for estimation or for simulation. Here, the probability distribution of the count data **Y** is defined by $P(Y=k)$, for any $k \geq 0$.

DESCRIPTION:

count data model - Poisson distribution
 $\log(P(Y=k))$ is defined

INPUT:

parameter = lambda

OBSERVATION:

$Y = \{$
type = count
 $\log(P(Y=k)) = -\lambda + k \log(\lambda) - \text{factln}(k)$
 $\}$

OUTPUT:

output = Y

$$\begin{aligned}\log(P(Y = k)) \\ = -\lambda + k \log(\lambda) - \log(k!)\end{aligned}$$

The probability distribution of the count data Y can equivalently be defined by $\log(P(Y=k))$, for any $k \geq 0$

DESCRIPTION:

count data model – Generalized Poisson distribution

INPUT:

parameter = {dlt, lbd}

OBSERVATION:

```
Y = {
type = count,
log(P(Y=k)) = log(lbd) + (k-1)*log(lbd+k*dlt)
              - lbd -k*dlt - factln(k)
}
```

OUTPUT:

output = Y

$$P(Y = k) = \frac{e^{-\lambda - k\delta} \lambda(\lambda + k\delta)^{k-1}}{k!}$$

More complex probability distributions can be defined, such as the Generalized Poisson distribution

DESCRIPTION:

count data model – zero-inflated Poisson distribution

INPUT:

parameter = {lambda, p0}

OBSERVATION:

```

Y = {
type = count,
if (k > 0)
  aux= log(1-p0) - lambda + k*log(lambda) - factln(k)
else
  aux= log(p0+(1-p0)*exp(-lambda))
end

log(P(Y=k)) = aux
}

```

OUTPUT:

output = Y

$$P(Y = 0) = p_0 + (1 - p_0)e^{-\lambda}$$

for $k \geq 1$,

$$P(Y = k) = (1 - p_0) \frac{e^{-\lambda} \lambda^k}{k!}$$

Or the zero-inflated Poisson distribution,

Categorical data model

DESCRIPTION:

Categorical data model – Bernoulli distribution

INPUT:

parameter = p

OBSERVATION:

$Y = \{$
 type = categorical
 categories = $\{0,1\}$
 $P(Y=1) = p$
 $\}$

OUTPUT:

output = Y

$$Y \in \{0,1\}$$

$$P(Y = 1) = p$$

$$P(Y = 0) = 1 - p$$

The probability distribution of the categorical data Y is defined in the block **OBSERVATION**,

Use **type=categorical** to specify that the data is of type « categorical »

Indicate with **categories=** the set of values taken by Y .

If Y takes K values, then $K-1$ probabilities are necessary to define the distribution of Y . Here, it is equivalent to define this binomial distribution with $P(Y=1)=p$ or $P(Y=0)=1-p$.

DESCRIPTION:

Categorical data model – 3 categories

INPUT:

parameter = {a1, a2, a3}

OBSERVATION:

Y = {
 type = categorical
 categories = {1, 2, 3}
 P(Y=1) = a1/(a1+a2+a3)
 P(Y=2) = a2/(a1+a2+a3)
 }

OUTPUT:

output = Y

$$Y \in \{1, 2, 3\}$$

$$P(Y = 1) = \frac{a_1}{a_1 + a_2 + a_3}$$

$$P(Y = 2) = \frac{a_2}{a_1 + a_2 + a_3}$$

$$P(Y = 3) = \frac{a_3}{a_1 + a_2 + a_3}$$

Here, Y takes 3 values, then two probabilities are needed to define its probability distribution.

In this example, P(Y=1) and P(Y=2) are defined.

DESCRIPTION:

Categorical data model – 3 categories

INPUT:

parameter = {a1, a2, a3}

OBSERVATION:

Y = {
 type = categorical
 categories = {1, 2, 3}
 P(Y=1) = a1/(a1+a2+a3)
 P(Y=3) = a3/(a1+a2+a3)
 }

OUTPUT:

output = Y

$$Y \in \{1, 2, 3\}$$

$$P(Y = 1) = \frac{a_1}{a_1 + a_2 + a_3}$$

$$P(Y = 2) = \frac{a_2}{a_1 + a_2 + a_3}$$

$$P(Y = 3) = \frac{a_3}{a_1 + a_2 + a_3}$$

It is equivalent to define for example $P(Y=1)$ and $P(Y=3)$,
 Obviously, the sum of the K probabilities must be equal to 1.

DESCRIPTION:

Ordered categorical data model – 3 categories
Cumulative probabilities

INPUT:

parameter = {a1, a2, a3}

OBSERVATION:

Y = {
type = categorical
categories = {1, 2, 3}
P(Y<=1) = a1/(a1+a2+a3)
P(Y<=2) = (a1+a2)/(a1+a2+a3)
}

OUTPUT:

output = Y

$$Y \in \{1, 2, 3\}$$

$$P(Y = 1) = \frac{a_1}{a_1 + a_2 + a_3}$$

$$P(Y = 2) = \frac{a_2}{a_1 + a_2 + a_3}$$

$$P(Y = 3) = \frac{a_3}{a_1 + a_2 + a_3}$$

If Y is an ordered categorical data, it is possible to define the cumulative distribution function

K - 1 probabilities are needed to define the distribution of Y.

In this example, we use P(Y<=1) and P(Y<=2) to define the distribution of Y.

DESCRIPTION:

Ordered categorical data model – 3 categories
Tail probabilities

INPUT:

parameter = {a1, a2, a3}

OBSERVATION:

Y = {
type = categorical
categories = {1, 2, 3}
P(Y>1) = (a2+a3)/(a1+a2+a3)
P(Y>2) = a3/(a1+a2+a3)
}

OUTPUT:

output = Y

$$Y \in \{1, 2, 3\}$$

$$P(Y = 1) = \frac{a_1}{a_1 + a_2 + a_3}$$

$$P(Y = 2) = \frac{a_2}{a_1 + a_2 + a_3}$$

$$P(Y = 3) = \frac{a_3}{a_1 + a_2 + a_3}$$

We can equivalently define the distribution of Y with the tail distribution, here P(Y>1) and P(Y>2).

DESCRIPTION:

Ordered categorical data model – 3 categories
Logit – probabilities

INPUT:

parameter = {theta1, theta2}

OBSERVATION:

Y = {
type = categorical
categories = {1, 2, 3}
logit(P(Y<=1)) = theta1
logit(P(Y<=2)) = theta1+theta2
}

OUTPUT:

output = Y

$$Y \in \{1, 2, 3\}$$

$$P(Y \leq 1) = \frac{1}{1 + e^{-\theta_1}}$$

$$P(Y \leq 2) = \frac{1}{1 + e^{-\theta_1 - \theta_2}}$$

$$P(Y \leq 3) = 1$$

According to the parametrization, it can be convenient to define the distribution of Y using the logit of K-1 probabilities.

In this example, we use the logit of P(Y<=1) and P(Y<=2) to define the distribution of Y.

DESCRIPTION:

Ordered categorical data model – 3 categories
 Logit – probabilities

INPUT:

parameter = {theta1, theta2}

OBSERVATION:

Y = {
 type = categorical
 categories = {1, 2, 3}
 logit(P(Y>1)) = -theta1
 logit(P(Y>2)) = -theta1 - theta2
 }

OUTPUT:

output = Y

$$Y \in \{1, 2, 3\}$$

$$P(Y \leq 1) = \frac{1}{1 + e^{-\theta_1}}$$

$$P(Y \leq 2) = \frac{1}{1 + e^{-\theta_1 - \theta_2}}$$

$$P(Y \leq 3) = 1$$

We can equivalently use the logit of $P(Y>1)$ and $P(Y>2)$ to define the distribution of Y.

*Categorical data model
with Markovian dependence*

DESCRIPTION:

Categorical data model – Markovian dependence

INPUT:

parameter = {p11, p21}

OBSERVATION:

Y = { type = categorical

categories = {1,2}

dependence = Markov

$P(Y=1 \mid Y_{-p}=1) = p_{11}$

$P(Y=1 \mid Y_{-p}=2) = p_{21}$

}

OUTPUT:

output = Y

$$Y_j \in \{1, 2\}$$

$$P(Y_j = 1 \mid Y_{j-1} = 1) = p_{11}$$

$$P(Y_j = 1 \mid Y_{j-1} = 2) = p_{21}$$

The conditional distributions $P(Y_j = k \mid Y_{j-1} = m)$ define the joint distribution of \mathbf{Y} given Y_1 .

Here, the « initial » distribution of Y_1 is not defined. It is assumed to be constant and does not contribute to the likelihood for estimating the population parameters.

For simulation, $P(Y_1 = 0) = P(Y_1 = 1) = 1/2$ is used.

Y_{-p} holds for « previous » value of Y (i.e. Y_{j-1})

DESCRIPTION:

Categorical data model – Markovian dependence

INPUT:

parameter = {p, p11, p21}

OBSERVATION:

Y = { type = categorical

categories = {1, 2}

dependence = Markov

$P(Y_1=1) = p$

$P(Y=1 \mid Y_{p=1}) = p_{11}$

$P(Y=1 \mid Y_{p=2}) = p_{21}$

OUTPUT:

output = Y

$$Y_j \in \{1, 2\}$$

$$P(Y_1 = 1) = p$$

$$P(Y_j = 1 \mid Y_{j-1} = 1) = p_{11}$$

$$P(Y_j = 1 \mid Y_{j-1} = 2) = p_{21}$$

The conditional distributions $P(Y_j = k \mid Y_{j-1} = m)$ and the « initial » distribution of Y_1 define the joint distribution of Y.

In this example, $p = P(Y_1 = 1)$ is a parameter of the model.

DESCRIPTION:

Categorical data model – Markovian dependence
Continuous time

INPUT:

parameter = {q12, q21}

OBSERVATION:

Y = { type = categorical
categories = {1, 2}
dependence = Markov
transitionRate(1,2) = q12
transitionRate(2,1) = q21
}

OUTPUT:

output = Y

for any $0 \leq s < t$, let

$$Q(s, t) = \left(q_{ij}(s, t) \right)_{1 \leq i, j \leq K}$$

$$P(s, t) = \left(p_{ij}(s, t) \right)_{1 \leq i, j \leq K}$$

$$p_{ij}(s, t) = P(Y(t) = j | Y(s) = i)$$

$$\frac{dP(s, t)}{dt} = Q(s, t)P(s, t)$$

$$\sum_{j=1}^K q_{ij}(s, t) = 0$$

The transition rates (q_{ij}) define the joint distribution of Y.

Time-to-event data model

Exact time of and right censored events

ID	TIME	Y
1	0	0
1	25	1
2	0	0
2	30	1
3	0	0
3	48	0
4	0	0
4	45	1
5	0	0
5	17	1
6	0	0
6	36	1
7	0	0
7	48	0

Single events

ID	TIME	Y
1	0	0
1	15	1
1	25	0
2	0	0
2	12	1
2	27	1
2	33	0
3	0	0
3	25	0
4	0	0
4	14	1
4	22	1
4	31	1
4	38	0

Repeated events

The same coding is used for single and repeated events:

- $Y=1$ means that the exact time of the event is known,
- $Y=0$ means that the event is right censored: the patient had not the event at that time,
- A record $Y=0$ at time t_0 is needed to define when we start observing. Here $t_0 = 0$.

DESCRIPTION:

Time-to-event data model
Constant hazard function

INPUT:

parameter = lambda

OBSERVATION:

```
Y = { type = event
      hazard = 1/lambda
    }
```

OUTPUT:

output = Y

$$h(t) = \frac{1}{\lambda}$$

$$P(T > t) = e^{-\frac{t}{\lambda}}$$

The probability distribution of the event data defined with the hazard function in the block **OBSERVATION**,
Use **type=event** to specify that the data is of type « event »,
hazard is a reserved keyword. Here, **hazard=lambda** means that the hazard is constant, *i.e.* the time to event has an exponential distribution.

DESCRIPTION:

Time-to-event data model
Weibull hazard function

INPUT:

parameter = {lambda, beta}

OBSERVATION:

```
Y = { type = event
      hazard = (beta/lambda)*(t/lambda)^(beta-1)
    }
```

OUTPUT:

output = Y

$$h(t) = \frac{\beta}{\lambda} \left(\frac{t}{\lambda} \right)^{\beta-1}$$

$$P(T > t) = e^{-\left(\frac{t}{\lambda} \right)^\beta}$$

hazard is a reserved keyword that can be used to define any complex hazard function.

Interval censored events

ID	TIME	Y
1	0	0
1	20	0
1	25	1
2	0	0
2	25	0
2	30	1
3	0	0
3	50	0
4	0	0
4	40	0
4	45	1
5	0	0
5	60	0

Single event
(interval length = 5)

ID	TIME	Y
1	0	0
1	5	2
1	10	0
1	15	1
1	20	3
1	25	0
1	30	2
2	0	0
2	5	0
2	10	1
2	15	3
2	20	0
2	25	1
2	30	3

Repeated events
(interval length = 5)

ID	Event
1	between 20 and 25
2	between 25 and 30
3	after 50
4	between 40 and 45

ID	Events
1	2 events between 0 and 5 0 event between 5 and 10 1 event between 10 and 15 3 events between 15 and 20 0 event between 20 and 25 2 events between 25 and 30

DESCRIPTION:

Time-to-event data model
Constant hazard function

INPUT:

parameter = lambda

OBSERVATION:

```
Y = { type = event
      eventType=intervalCensored,
      maxEventNumber=1,
      hazard = 1/lambda
    }
```

OUTPUT:

output = Y

$$h(t) = \frac{1}{\lambda}$$

$$P(T > t) = e^{-\frac{t}{\lambda}}$$

Use **eventType=intervalCensored** to specify that the events are interval censored.

maxEventNumber is the maximum number of events that can occur per individual. An infinite number of events is assumed if **maxEventNumber** is not specified.

This information is required in an estimation context for properly computing the likelihood of the observations. No additional information is required for estimation.

DESCRIPTION:

Time-to-event data model
Constant hazard function

INPUT:

parameter = lambda

OBSERVATION:

```
Y = { type = event
      eventType=intervalCensored,
      maxEventNumber=3,
      intervalLength=5,
      rightCensoringTime=200,
      hazard = 1/lambda
    }
```

OUTPUT:

output = Y

$$h(t) = \frac{1}{\lambda}$$

$$P(T > t) = e^{-\frac{t}{\lambda}}$$

Some additional parameters can be defined for simulation

intervalLength is the length of the intervals ; default = rightCensoringTime/10 (also used for VPCs).

rightCensoringTime is the length of the study ; default is the maximum of the times of observations available in the dataset (not used by Monolix, only for simulation).

Joint models

Id	TIME	AMT	Y	DVID
1	0	100	.	.
1	4	.	9.2	1
1	8	.	5	2
1	12	.	8.5	1
1	18	.	6.4	1
1	24	.	2	2
2	0	120	26	.
2	4	.	4.8	1
2	8	.	3	2
2	12	.	3.1	1
2	18	.	2.5	1
2	14	.	0	2
3	0	80	.	.
3	4	.	5.7	1
3	8	.	4	2
3	12	.	3.8	1

A joint model is used for modelling simultaneously some continuous data (a biomarker for instance) and another type of data (event, count, categorical...)

The datafile contains an additional column **DVID** (or **YTYPE**) to specify the type of data.

In this example, DVID =1 is used for a continuous response and DVID =2 for count data.

DESCRIPTION:

Joint PK and count data model

INPUT:

parameter = {ka, V, Cl, ke0, lambda0, I_{max}, IC₅₀}

EQUATION:

{C_c, C_e} = pkmodel(ka, V, Cl, ke0)

lambda = lambda0 * (1 - I_{max} * C_e / (IC₅₀ + C_e))

OBSERVATION:

seizure = {

type = count

log(P(seizure=k)) = -lambda + k*log(lambda) - factln(k)

}

OUTPUT:

output = {C_c, seizure}

In this example, an effect compartment and an I_{max} model are used for modelling the Poisson intensity of the count data.

DESCRIPTION:

Joint PK and categorical data model

INPUT:

parameter = {ka, V, Cl, ke0, theta1, theta2, alpha, beta}

EQUATION:

{Cc, Ce} = pkmodel(ka, V, Cl, ke0)

OBSERVATION:

level = {

type = categorical

categories = {0, 1, 2}

logit(P(level<=0)) = theta1 + alpha*t + beta*Ce

logit(P(level<=1)) = theta1 + alpha*t + beta*Ce + theta2

}

OUTPUT:

output = {Cc , level}

In this example, the probability distribution of the categorical data is function of the time and the concentration in the effect compartment.

DESCRIPTION:

Joint PK and time to event data model

INPUT:

parameter = {ka, V, Cl, ke0, lambda0, I_{max}, IC50}

EQUATION:

{C_c, C_e} = pkmodel(ka, V, Cl, ke0)

lambda = lambda0 * (1 - I_{max} * C_e / (IC50 + C_e))

OBSERVATION:

hemorrhaging = { type = event, hazard = 1/lambda }

OUTPUT:

output = { C_c , hemorrhaging }

In this example, an effect compartment and an I_{max} model are used for modelling the risk of hemorrhaging.

DESCRIPTION:

Joint PK and time to event data model

INPUT:

parameter = {ka, V, Cl, ke0, lambda0, I_{max}, IC₅₀}

EQUATION:

{C_c, C_e} = pkmodel(ka, V, Cl, ke0)

lambda=lambda0*(1 - I_{max}*C_e/(IC₅₀+C_e))

OBSERVATION:

Concentration = { type = continuous, prediction = C_c,
errorModel = band(0, 10) }

hemorrhaging = { type = event, hazard = 1/lambda }

OUTPUT:

output = {Concentration, hemorrhaging}

In addition to its prediction, an error model is fixed for the concentration here, allowing us to define its whole distribution, as we did for the risk of hemorrhaging.

DESCRIPTION:

Joint PK, count data and time to event data model

INPUT:

parameter = {ka, V, Cl, ke0, lambda0, lmax, IC50, beta, theta1, theta2}

EQUATION:

{Cc, Ce} = pkmodel(ka, V, Cl, ke0)

lambda=lambda0*(1 - lmax*Ce/(IC50+Ce))

OBSERVATION:

hemorrhaging = { type = event, hazard = 1/lambda }

level = { type = categorical, categories = {0, 1, 2}

logit(P(level<=0)) = theta1 + beta*Ce

logit(P(level<=1)) = theta1 + beta*Ce + theta2 }

OUTPUT:

output = {Cc , hemorrhaging, level}

A joint model can address simultaneously several types of data.

In this example, we model jointly PK data, time to event data and categorical data.

The distribution of several discrete data can be described in a same block **OBSERVATION**.

MONOLIX 4.2

MLXTRAN

5. Symbols reference

- *Blocks* 128
- *Interface* 129
- *PK* 130
- *ODE* 132
- *Observations* 133
- *Operators* 135
- *Conditionals* 136
- *Comments* 137
- *Usual functions* 138

List of symbols for blocks:

DESCRIPTION	Block for title and free comments - example
INPUT	Block for inputs from the data and statistical model - example
PK	Block for PK function and macros - example
EQUATION	Block for explicit mathematical model - example
OBSERVATION	Block for random variables of observations - example
OUTPUT	Block for outputs - example

List of symbols for interface:

parameter	List of individual parameters - example
regressor	List of regression values - example
output	List of outputs - example
table	List of additional outputs for tables

List of symbols for PK:

compartment	Macro for compartment - example
depot	Macro for a deposit, where the doses are the source terms - example
iv or input	Macro for IV administration - example
absorption or oral	Macro for oral administration - example
elimination	Macro for elimination - example
peripheral	Macro for peripheral compartment - example
transfer	Macro for transfer - example
cmt	Compartment number - example
target	Target ODE component of the source term - example
ka	Absorption constant rate - example
Tko	Absorption duration - example
Tlag	Lag time before absorption - example
Mtt, Ktr	Mean transit time & transit rate constant - example
p	Fraction of dose which is absorbed - example
V	Central compartment volume for PK function - example
Cl	Central compartment clearance - example
Vm, Km	Michaelis Menten elimination parameters - example

kij, kji	Transfer rate constants between compartments i & j - example
kt	Transfer rate constant between two compartments - example
from, to	Source and target compartments - example
keo	Effect compartment transfer rate constant - example
amount	Amount in a compartment - example
concentration	Concentration in a compartment - example
volume	Central compartment volume - example
pkmodel	Function for standard PK models - example
tDose	Time of the last administered dose - example
amtDose	Amount of the last administered dose - example
inftDose	Infusion time of the last administered dose

List of symbols for ODE:

t	First regression variable, denoted as continuous time
ddt_<component>	Derivative with respect to time t - example
<component>_o	Initial value - example
to or t_o	Initial time - example
odeType	ODE solver to use - example
nonstiff	Non stiff ODE solver
stiff	Stiff ODE solver - example
linear	Linear ODE solver

List of symbols for observations:

type	Type - example
count	Count data - example
categorical	Categorical data - example
event	Time-to-event data - example
P	Probability - example
k	Count value - example
dependence	dependence = Markov for Markovian dependence - example
Y_p , Y₁	previous (Y _p) and first (Y ₁) observations example
transitionRate	rate of transition for Markov process - example
categories	List of categories - example
intervalCensored	interval censored events - example
maxEventNumber	maximum number of events - example
intervalLength	length of censoring interval - example
rightCensoringTime	duration of the study - example
hazard	Hazard function - example
wsmm	Within subject mixture model
bsmm	Between subject mixture model

continuous	Continuous data - example
prediction	Prediction - example
errorModel	Error model - example
autocorrelation	Autocorrelation of observations - example
constant	Constant error model $y = f + a e$
proportional	Proportional error model $y = f + b f e$
combined1	Combined error model $y = f + (a + b f)e$
combined2	Combined error model $y = f + a e_1 + b f e_2$
proportionalc	Proportional error model with power $y = f + b f^c e$
combined1c	Combined error model with power $y = f + (a + b f^c)e$
combined2c	Combined error model with power $y = f + a e_1 + b f^c e_2$
exponential	Exponential error model $t(y) = \log(y)$ and $y = f e^{a e}$
logit	Logit error model $t(y) = \log(y/(1 - y))$
band(0, 10)	Extended logit error model $t(y) = \log(y/(10 - y))$ - example
band(0, 100)	Extended logit error model $t(y) = \log(y/(100 - y))$

List of symbols for operators:

+	Addition or unary plus
-	Substraction or unary minus
*	Multiplication
/	Division
^	Power
~ or !	Logical <i>NOT</i>
&& or &	Logical <i>AND</i>
 or 	Logical <i>OR</i>
==	Equal to
~= or !=	Not equal to
<	Less than
<=	Less than or equal to
>	Greater than
>=	Greater than or equal to

List of symbols for conditionals:

if <i><condition></i>	Weight following definitions with the indicator function of <i><condition></i> - example
else	Weight following definitions with the product of indicator functions of complements of all previous conditions - example
elseif <i><condition></i>	Weight following definitions with the product of indicator functions of <i><condition></i> and complements of all previous conditions
end	Terminate a conditional definitions - example

List of symbols for comments:

<code>;<line remainder></code>	Comment. Any following text on the line is ignored
--------------------------------------	--

List of symbols for usual functions:

abs, sqrt, exp, log, log10, logit, sin, cos, tan, asin, acos, atan, sinh, cosh, tanh, floor, ceil, gammaln, factln, min, max, atan2, rem