****

PFIM 3.0

Caroline Bazzoli[(1)], Sylvie Retout[(1)] and France Mentré[(1)]

(1) INSERM, U738, Paris, France ; Université Paris 7, Paris, France;

Examples documentation

****

PFIM 3.0 is free library of functions.
The University Paris Diderot and INSERM are the co-owners of this library of functions.

Disclaimer

We inform users that the PFIM 3.0 is a tool developed by the Laboratory « Models and methods of the therapeutic evaluation of the chronic diseases "- UMR-S 738, under R and GCC.

PFIM 3.0 is a library of functions. The functions are published after a scientific validation.

However, it may be that only extracts are published.

By using this library of functions, the user accepts all the conditions of use set forth hereinafter.

Licence
1.1.1
This program is free software: you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation, either version 3 of the License, or (at your option) any later version.

You should have received a copy of the GNU General Public License along with this program. If not, see
<*http://www.gnu.org/licenses/*>.

THIS SOFTWARE IS PROVIDED "AS IS" AND ANY EXPRESSED OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE UNIVERSITE PARIS DIDEROT OR INSERM OR ITS CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

www.pfim.biostat.fr

CONTENTS

These next examples are supplied in the folder called Examples in the tool PFIM 3.0. For each example, the model file model.r, the input file stdin.r and the output file stdout.r, present in the folder Examples, are showed below. When the graph has been specified in an example section, it is presented at the end of the section.

1. Evaluation

1.1 Example 1: Single response

1.1.1 Analytical form

The purpose is to evaluate a design using a one compartment first order absorption model described by an analytical form.
The design to be evaluated is composed of two groups: one group of 200 subjects with a dose of 100 and sampling times at (0.33, 1.5, 5, 12) and one group of 200 subjects with a dose of 100 with sampling times at (1, 3, 8).

MODEL FILE

```
formA<-expression(dose/v*ka/(ka-ke)*(exp(-ke*t)-exp(-ka*t)))
form<-c(formA)
```

INPUT FILE

```
##############################################################################
##                                                                          ##
##                          INPUT FILE FOR PFIM 3.0                          ##
##############################################################################


#Name of the project
#--------------------

project<-"Doc_example1.1.1"


#Name of the file containing the PK or PD model
#-----------------------------------------------

file.model<-"model.r";


#Name of the output file for the results
#----------------------------------------

output<-"Stdout.r";

#RUN:  Evaluation (EVAL) or Optimisation (OPT)
#----------------------------------------------------
run<-"EVAL"

#Number of responses
#-------------------------------------------------------------------

nr<-1

#Name of the fixed effects parameters
#-------------------------------------

parameters<-c('ka','ke','v')

#Fixed effects parameters values
#--------------------------------
```

```
beta<-c(2,0.25,15)

#Diagonal Matrix of variance for the random effects:
#---------------------------------------------------

omega<-diag(c(1,0.25,0.1));


#Standard deviation of the residual error (sig.inter+sig.slope*f)^2:
#-------------------------------------------------------------------

sig.interA<-0.5
sig.slopeA<-0.15

#Between-subject variance model (1) = additive  (2) = exponential
#-----------------------------------------------------------------

Trand<-2;

#List of the vectors of sampling times for each elementary design
#you can specify any sampling times for a group by writing NULL
#ONLY if you have several responses
#-----------------------------------------------------------------

protA<-list(c(0.333,1.5,5,12),c(1,3,8))

#Vector of initial proportions or numbers of subjects for each elementary design
#-------------------------------------------------------------
subjects<-c(200,200)


#Subjects input: (1) for number of subjects (2) for proportions of subjects
#-------------------------------------------------------------------------
subjects.input<-2

#If 'proportions of subjects' give the total number of samples
#-------------------------------------------------------------
Ntot<-250

################### MODEL OPTION ############################

#Model form: Differential equations (DE) or analytical form (AF)
#---------------------------------------------------------------

modelform<-"AF"

###### ANALYTICAL MODEL OPTION ############################
############################################################

#Identical dose in each elementary design (Yes=T, No=F)
#-----------------------------------------------------------
dose.identical<-T

# If 'Yes', enter the value of the dose,
# else, enter the vector of the dose values for each elementary design
#-------------------------------------------------------------------
dose<-c(100)

#Vector of last times for the intervals of each expression
#---------------------------------------------------------

boundA<-list(c(0,Inf))


###### END ANALYTICAL MODEL OPTION ########################
```

```
###### DIFFERENTIAL EQUATION OPTION ##################################
#####################################################################

#Initial time for which initial conditions are given
#---------------------------------------------------

#time.condinit<-0

#Identical initial conditions in each elementary design (Yes=T, No=F)
#-------------------------------------------------------------
#condinit.identical<-T

# If 'Yes', enter once the expression of the initial values of the system at the
initial time
# else, enter the vectors of the initial conditions for each elementary design
# If initial values depend on parameters to be estimated,
# enter this parameter into the expression without any quotation marks
#-------------------------------------------------------

#condinit<-expression(c(0,0,0,0))


# Error tolerance for solving differential equations
#---------------------------------------------------

RtolEQ<-1e-08
AtolEQ<-1e-08
Hmax<-Inf # Default value

###### END DIFFERENTIAL EQUATION OPTION ##################################



############ONLY FOR OPTIMISATION ###############################

#Identical sampling times for each response
# (only if you do not have sampling times==NULL)
#-------------------------------------------------------------------------------------
#identical.times<-F

######## OPTIMISATION ALGORITHM OPTION ###############

#Character string for thoice of the optimisation algorithm:
#      "FW" for the Fedorov-Wynn algorithm
#      "SIMP" for the Simplex algorithm
#----------------------------------------

#algo.option<-"FW"


########################
#SIMPLEX SPECIFICATION #
########################

#Optimisation of the proportions of subjects: (Yes=T, No=F)
#-----------------------------------------------------------

#subjects.opt<-T

#Vector of lower and upper admissible sampling times
#---------------------------------------------------


#lowerA<-c(0)
#upperA<-c(24)
```

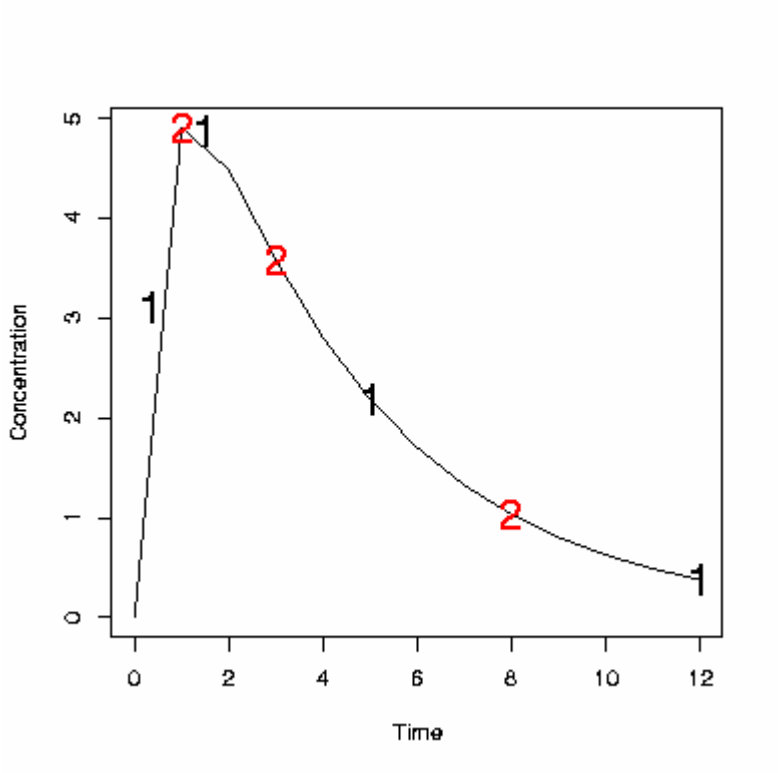
```
#lowerB<-c(0)
#upperB<-c(24)

#Minimum delay between two sampling times
#-----------------------------------------

#delta.time<-0

#Print iteration step (Yes=T, No=F)
#---------------------------------

#iter.print<-T


#Parameter for initial simplex building (%)
#-----------------------------------------

#simplex.parameter<-20


#Maximum iteration number
#------------------------

#Max.iter<-5000


#Relative convergence tolerance
#------------------------------
#Rctol<-1e-6



##############################
#FEDOROV-WYNN SPECIFICATION #
##############################


#Number of sampling windows
#--------------------------
#nwindA<-1
#nwindB<-1

#List of vector of the allowed sampling times for each sampling window
#---------------------------------------------------------------------

#sampwinA<-list(c(0.166,1,2,3,6,9,12,20))
#sampwinB<-list(c(0.166,1,2,3,6,9,12,20,36))


#List of vector of allowed number of points to be taken from each sampling window
#--------------------------------------------------------------------------------

#nsampA<-list(c(4))
#nsampB<-list(c(4))

#Maximum total number of sampling times per subject
#--------------------------------------------------

#nmaxptsA<-4
#nmaxptsB<-4

#Minimum total number of sampling times per subject
#--------------------------------------------------

#nminptsA<-4
#nminptsB<-4
############# END OF OPTIMISATION ALGORITHM OPTION ###############
```

```
############## GRAPH SPECIFICATION OPTION ###############

#graphical representation (Yes=T, No=F)
#-------------------------------------

graph.logical<-T

#Names on Y axes for each response
#--------------------------------
names.data<-c("Concentration")

#Controls logarithmic axes for the graphical representation.
#Values "xy", "x", or "y" produce log-log or log-x or log-y axes.
#(For standard graphic, log.logical<-F)
#-------------------------------------------------------------

#log.logical<-'y'
log.logical<-F

#Vector of lower and upper sampling times for the graphical representation
#------------------------------------------------------------------------

graph.infA<-c(0)
graph.supA<-c(12)

#Vector of lower and upper concentration for the graphical representation
#-----------------------------------------------------------------------
y.range<-NULL # default range
#y.range<-c(0,10)
```

OUTPUT FILE:

```
PFIM 3.0

Project:  Doc_example1.1.1

Date:  Mon Sep 10 09:35:27 2007


*************************** INPUT SUMMARY *******************************

Analytical function models :

dose/v * ka/(ka - ke) * (exp(-ke * t) - exp(-ka * t))


Population design:


Sample times for response: A
                     subjects Dose
c(0.333, 1.5, 5, 12)      200  100
c(1, 3, 8)                200  100


Variance error model response A : ( 0.5 + 0.15 *f)^2


Between-subject variance model: Trand =  2

****************** POPULATION FISHER INFORMATION matrix ******************

          [,1]       [,2]      [,3]      [,4]       [,5]       [,6]       [,7]
[1,]  63.26917   197.2991  -7.74132   0.00000    0.00000    0.00000    0.00000
[2,] 197.29913 16408.6040 158.51127   0.00000    0.00000    0.00000    0.00000
[3,]  -7.74132   158.5113  10.05526   0.00000    0.00000    0.00000    0.00000
[4,]   0.00000     0.0000   0.00000  83.77010   12.47033   67.45659   61.45098
[5,]   0.00000     0.0000   0.00000  12.47033 1314.88024  441.77864  380.33087
[6,]   0.00000     0.0000   0.00000  67.45659  441.77864 6400.24096  579.51626
[7,]   0.00000     0.0000   0.00000  61.45098  380.33087  579.51626 1663.93555
[8,]   0.00000     0.0000   0.00000 233.89675  430.90540 2245.49184 2516.58831
          [,8]
[1,]    0.0000
[2,]    0.0000
[3,]    0.0000
[4,]  233.8967
[5,]  430.9054
[6,] 2245.4918
[7,] 2516.5883
[8,] 6701.4272


************************* EXPECTED STANDARD ERRORS ************************

------------------------ Fixed Effects Parameters -------------------------

    Beta    StdError       CV .
ka  2.00 0.141452152 7.072608 %
ke  0.25 0.009079485 3.631794 %
v  15.00 0.378081321 2.520542 %


------------------------ Variance of Random Effects ----------------------

   Omega   StdError       CV .
```

```
ka  1.00 0.11583943 11.58394 %
ke  0.25 0.02893458 11.57383 %
v   0.10 0.01357357 13.57357 %


------------------------ Variance of residual error ------------------------

            SIG   StdError       CV .
sig.interA 0.50 0.03907990  7.81598 %
sig.slopeA 0.15 0.02064161 13.76107 %


****************************** DETERMINANT ********************************

1.664863e+22

****************************** CRITERION **********************************

599.3387
```



1.1.2 Several analytical form (library of models)

The purpose is to evaluate a design described by a two compartment model after repeated infusions of 550 mg during 1.5 hours at day 0 and after 3 weeks with a dose of 550 mg during also 1.5 hours.
The design to be evaluated is composed of one group of 90 subjects with sampling times after each of two doses: end of infusion, day 7, day 14 and 10 min before the next infusion.
In this example time in hours is divided by 24 in order to put sampling times in day unity.

MODEL FILE

```
#######################################################################
##Model form with library of models                                  ##
#######################################################################

source(paste(directory.program,dirsep,»libraryModels.r»,sep=««))

form1<-infusion_2cpt_Vkk12k21(Inf=1.5/24)[[1]]

form4<-infusion_2cpt_Vkk12k21_md(TInf=1.5/24,N=4,tau=21)[[1]]

formA<-c(form1,form4)

form<-c(formA)
```

INPUT FILE

```
#############################################################################
##                                                                         ##
##                          INPUT FILE FOR PFIM 3.0                          ##
#############################################################################


#Name of the project
#--------------------

project<-"Doc_example1.1.2"


#Name of the file containing the PK or PD model
#-----------------------------------------------

file.model<-"model.r";


#Name of the output file for the results
#----------------------------------------

output<-"Stdout.r";

#RUN:  Evaluation (EVAL) or Optimisation (OPT)
#------------------------------------------------------
run<-"EVAL"

#Number of responses
#-------------------------------------------------------------------

nr<-1

#Name of the fixed effects parameters
#-------------------------------------

parameters<-c('k','V','k12','k21')
```

```
#Fixed effects parameters values
#-------------------------------

beta<-c(0.1,3.08,0.175,0.116)

#Diagonal Matrix of variance for the random effects:
#--------------------------------------------------

omega<-diag(c(0.1,0.1,0.3,0.5));


#Standard deviation of the residual error (sig.inter+sig.slope*f)^2:
#------------------------------------------------------------------

sig.interA<-0
sig.slopeA<-0.25

#Between-subject variance model (1) = additive  (2) = exponential
#----------------------------------------------------------------

Trand<-2;

#List of the vectors of sampling times for each elementary design
#you can specify any sampling times for a group by writing NULL
#ONLY if you have several responses
#----------------------------------------------------------------

protA<-list(c(1.5/24,7,14,21-10/24,c(1.5/24,7,14,21-10/24)+3*21))

#Vector of initial proportions or numbers of subjects for each elementary design
#-------------------------------------------------------------
subjects<-c(90)

#Subjects input: (1) for number of subjects (2) for proportions of subjects
#-------------------------------------------------------------------------
subjects.input<-2

#If 'proportions of subjects' give the total number of samples
#------------------------------------------------------------
Ntot<-250

################### MODEL OPTION ############################

#Model form: Differential equations (DE) or analytical form (AF)
#---------------------------------------------------------------

modelform<-"AF"

###### ANALYTICAL MODEL OPTION ############################
#############################################################

#Identical dose in each elementary design (Yes=T, No=F)
#-----------------------------------------------------------
dose.identical<-T

# If 'Yes', enter the value of the dose,
# else, enter the vector of the dose values for each elementary design
#--------------------------------------------------------------------
dose<-c(1)

#Vector of last times for the intervals of each expression
#--------------------------------------------------------
boundA<-list(c(0,1.5/24),c(1.5/24,21),c(63,(1.5/24)+3*21),c((1.5/24)+3*21,Inf))

###### END ANALYTICAL MODEL OPTION ########################
```

```
###### DIFFERENTIAL EQUATION OPTION ##################################
#####################################################################

#Initial time for which initial conditions are given
#---------------------------------------------------

#time.condinit<-0

#Identical initial conditions in each elementary design (Yes=T, No=F)
#-------------------------------------------------------------
#condinit.identical<-T

# If 'Yes', enter once the expression of the initial values of the system at the
initial time
# else, enter the vectors of the initial conditions for each elementary design
# If initial values depend on parameters to be estimated,
# enter this parameter into the expression without any quotation marks
#-------------------------------------------------------

#condinit<-expression(c(0,0,0,0))


# Error tolerance for solving differential equations
#---------------------------------------------------

#RtolEQ<-1e-08
#AtolEQ<-1e-08
#Hmax<-Inf # Default value
###### END DIFFERENTIAL EQUATION OPTION ##################################



############ONLY FOR OPTIMISATION ###############################

#Identical sampling times for each response
# (only if you do not have sampling times==NULL)
#-----------------------------------------------------------------------------------------
#identical.times<-F

######## OPTIMISATION ALGORITHM OPTION ###############

#Character string for thoice of the optimisation algorithm:
#      "FW" for the Fedorov-Wynn algorithm
#      "SIMP" for the Simplex algorithm
#----------------------------------------

#algo.option<-"FW"


########################
#SIMPLEX SPECIFICATION #
########################

#Optimisation of the proportions of subjects: (Yes=T, No=F)
#-------------------------------------------------------------

#subjects.opt<-T

#Vector of lower and upper admissible sampling times
#---------------------------------------------------


#lowerA<-c(0)
#upperA<-c(24)

#lowerB<-c(0)
#upperB<-c(24)
```

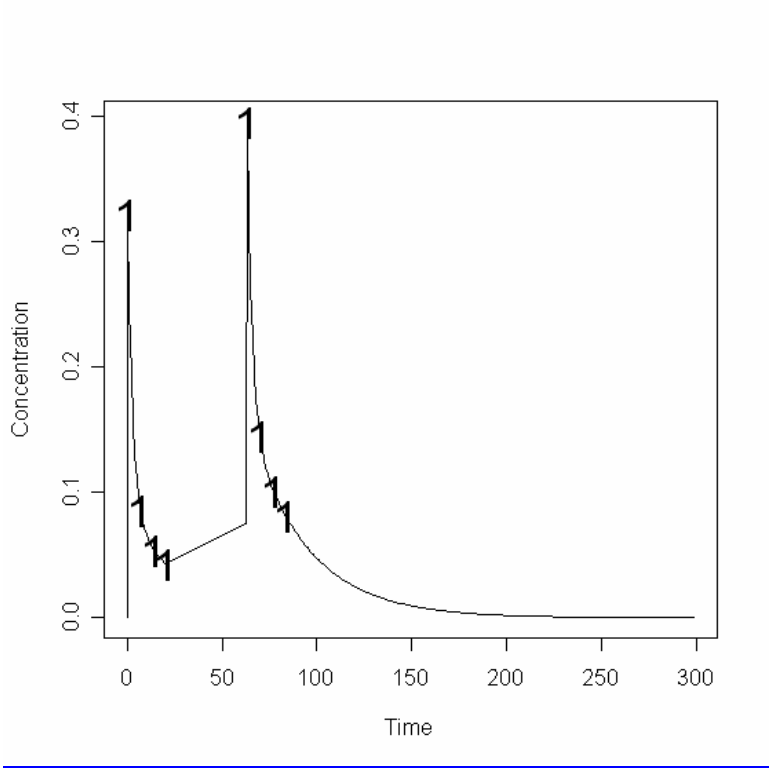
```
#Minimum delay between two sampling times
#-----------------------------------------

#delta.time<-0

#Print iteration step (Yes=T, No=F)
#---------------------------------

#iter.print<-T


#Parameter for initial simplex building (%)
#-----------------------------------------

#simplex.parameter<-20


#Maximum iteration number
#-----------------------

#Max.iter<-5000


#Relative convergence tolerance
#-----------------------------
#Rctol<-1e-6



##############################
#FEDOROV-WYNN SPECIFICATION #
##############################


#Number of sampling windows
#-------------------------
#nwindA<-1
#nwindB<-1

#List of vector of the allowed sampling times for each sampling window
#---------------------------------------------------------------------

#sampwinA<-list(c(0.166,1,2,3,6,9,12,20))
#sampwinB<-list(c(0.166,1,2,3,6,9,12,20,36))


#List of vector of allowed number of points to be taken from each sampling window
#--------------------------------------------------------------------------------

#nsampA<-list(c(4))
#nsampB<-list(c(4))

#Maximum total number of sampling times per subject
#-------------------------------------------------

#nmaxptsA<-4
#nmaxptsB<-4

#Minimum total number of sampling times per subject
#-------------------------------------------------

#nminptsA<-4
#nminptsB<-4
############# END OF OPTIMISATION ALGORITHM OPTION ###############
```

```
############## GRAPH SPECIFICATION OPTION ###############

#graphical representation (Yes=T, No=F)
#-------------------------------------

graph.logical<-T

#Names on Y axes for each response
#--------------------------------
names.data<-c("Concentration")

#Controls logarithmic axes for the graphical representation.
#Values "xy", "x", or "y" produce log-log or log-x or log-y axes.
#(For standard graphic, log.logical<-F)
#-----------------------------------------------------------

#log.logical<-'y'
log.logical<-F

#Vector of lower and upper sampling times for the graphical representation
#------------------------------------------------------------------------

graph.infA<-c(0)
graph.supA<-c(300)
#graph.infB<-c(0)
#graph.supB<-c(24)

#Vector of lower and upper concentration for the graphical representation
#-----------------------------------------------------------------------
y.range<-NULL # default range
#y.range<-c(0,10)
```

OUTPUT FILE

```
PFIM 3.0

Project:  Doc_example1.1.2

Date:  Mon Nov 12 16:10:53 2007


*************************** INPUT SUMMARY *******************************

Analytical function models :

16 * (1 * ((((k + k21 + k12) + sqrt((k + k21 + k12)^2 - 4 * k * k21))/2) - k21)/(V *
((((k + k21 + k12) + sqrt((k + k21 + k12)^2 - 4 * k * k21))/2) - (((k + k21 + k12) -
sqrt((k + k21 + k12)^2 - 4 * k * k21))/2)))/(((k + k21 + k12) + sqrt((k + k21 + k12)^2
- 4 * k * k21))/2) * (1 - exp(-(((k + k21 + k12) + sqrt((k + k21 + k12)^2 - 4 * k *
k21))/2) * t)) + 1 * (k21 - (((k + k21 + k12) - sqrt((k + k21 + k12)^2 - 4 * k *
k21))/2))/(V * ((((k + k21 + k12) + sqrt((k + k21 + k12)^2 - 4 * k * k21))/2) -

-16 * (1 * ((((k + k21 + k12) + sqrt((k + k21 + k12)^2 - 4 * k * k21))/2) - k21)/(V *
((((k + k21 + k12) + sqrt((k + k21 + k12)^2 - 4 * k * k21))/2) - (((k + k21 + k12) -
sqrt((k + k21 + k12)^2 - 4 * k * k21))/2)))/(((k + k21 + k12) + sqrt((k + k21 + k12)^2
- 4 * k * k21))/2) * (exp(-(((k + k21 + k12) + sqrt((k + k21 + k12)^2 - 4 * k *
k21))/2) * 0.0625) - 1) * exp(-(((k + k21 + k12) + sqrt((k + k21 + k12)^2 - 4 * k *
k21))/2) * (t - 0.0625)) + 1 * (k21 - (((k + k21 + k12) - sqrt((k + k21 + k12)^2 -

16 * (1 * ((((k + k21 + k12) + sqrt((k + k21 + k12)^2 - 4 * k * k21))/2) - k21)/(V *
((((k + k21 + k12) + sqrt((k + k21 + k12)^2 - 4 * k * k21))/2) - (((k + k21 + k12) -
sqrt((k + k21 + k12)^2 - 4 * k * k21))/2)))/(((k + k21 + k12) + sqrt((k + k21 + k12)^2
- 4 * k * k21))/2) * (1 - exp(-(((k + k21 + k12) + sqrt((k + k21 + k12)^2 - 4 * k *
k21))/2) * (t - (4 - 1) * 21))) + 1 * (k21 - (((k + k21 + k12) - sqrt((k + k21 + k12)^2
- 4 * k * k21))/2))/(V * ((((k + k21 + k12) + sqrt((k + k21 + k12)^2 - 4 *

-16 * (1 * ((((k + k21 + k12) + sqrt((k + k21 + k12)^2 - 4 * k * k21))/2) - k21)/(V *
((((k + k21 + k12) + sqrt((k + k21 + k12)^2 - 4 * k * k21))/2) - (((k + k21 + k12) -
sqrt((k + k21 + k12)^2 - 4 * k * k21))/2)))/(((k + k21 + k12) + sqrt((k + k21 + k12)^2
- 4 * k * k21))/2) * (exp(-(((k + k21 + k12) + sqrt((k + k21 + k12)^2 - 4 * k *
k21))/2) * 0.0625) - 1) * ((1 - exp(-4 * (((k + k21 + k12) + sqrt((k + k21 + k12)^2 - 4
* k * k21))/2) * 21))/(1 - exp(-(((k + k21 + k12) + sqrt((k + k21 + k12)^2 -


Population design:


Sample times for response: A
                                                                          subjects
c(0.0625, 7, 14, 20.5833333333333, 63.0625, 70, 77, 83.5833333333333)       90
                                                                              Dose
c(0.0625, 7, 14, 20.5833333333333, 63.0625, 70, 77, 83.5833333333333)    1


Variance error model response A : ( 0 + 0.25 *f)^2


Between-subject variance model: Trand =  2

****************** POPULATION FISHER INFORMATION matrix ******************

            [,1]      [,2]        [,3]        [,4]        [,5]        [,6]
 [1,] 65195.3443 492.44370 -2534.98111 -3625.39861    0.000000    0.000000
 [2,]   492.4437  73.26596    49.35633   -72.83411    0.000000    0.000000
 [3,] -2534.9811  49.35633  4825.91663 -2653.85113    0.000000    0.000000
 [4,] -3625.3986 -72.83411 -2653.85113  7535.99167    0.000000    0.000000
```

```
 [5,]     0.0000   0.00000     0.00000     0.00000 2361.351623  127.803308
 [6,]     0.0000   0.00000     0.00000     0.00000  127.803308 2683.705458
 [7,]     0.0000   0.00000     0.00000     0.00000   10.933345    3.931794
 [8,]     0.0000   0.00000     0.00000     0.00000    9.825508    3.761963
 [9,]     0.0000   0.00000     0.00000     0.00000  550.947770  506.429639
            [,7]       [,8]       [,9]
 [1,]   0.000000   0.000000     0.0000
 [2,]   0.000000   0.000000     0.0000
 [3,]   0.000000   0.000000     0.0000
 [4,]   0.000000   0.000000     0.0000
 [5,]  10.933345   9.825508   550.9478
 [6,]   3.931794   3.761963   506.4296
 [7,] 121.349871  16.123996   223.5470
 [8,]  16.123996  57.126849   127.5423
 [9,] 223.547001 127.542252 13795.7802

************************* EXPECTED STANDARD ERRORS ************************

------------------------ Fixed Effects Parameters ------------------------

     Beta   StdError        CV .
k   0.100 0.00420600  4.206000 %
V   3.080 0.12078092  3.921458 %
k12 0.175 0.01661015  9.491512 %
k21 0.116 0.01326660 11.436724 %


------------------------- Variance of Random Effects ----------------------

    Omega   StdError       CV .
k     0.1 0.02069639 20.69639 %
V     0.1 0.01938917 19.38917 %
k12   0.3 0.09358106 31.19369 %
k21   0.5 0.13575580 27.15116 %


------------------------ Variance of residual error -----------------------

            SIG    StdError      CV .
sig.slopeA 0.25 0.008765175 3.50607 %


****************************** DETERMINANT ********************************

6.583008e+28

****************************** CRITERION **********************************

1592.382
```


Concentration
0.0
0.1
0.2
0.3
0.4
0
50
100
150
200
250
300
Time

1.1.3 Differential equation system (ODE)

The purpose is to evaluate a design using a one compartment with a first order absorption described by a differential equation form.
The design to be evaluated is composed of two groups of 200 subjects: one group with sampling times at (0.133, 1.5, 5, 12) and one with sampling times at (1, 3, 8) with a dose of 100.

MODEL FILE:

```
formED<-function(t,y,p)
{

      ka<-p[1]
      ke<-p[2]
      v<-p[3]

      yd1<-ka*y[2]-ke*y[1]
      yd2<--ka*y[2]
      list(c(yd1,yd2),c(y[1]/v))
}
```

INPUT FILE:

```
##############################################################################
##                                                                          ##
##                          INPUT FILE FOR PFIM 3.0                           ##
##############################################################################


#Name of the project
#--------------------

project<-"Doc_example1.1.3"


#Name of the file containing the PK or PD model
#-----------------------------------------------

file.model<-"model.r";


#Name of the output file for the results
#----------------------------------------

output<-"Stdout.r";

#RUN:  Evaluation (EVAL) or Optimisation (OPT)
#-----------------------------------------------------
run<-"EVAL"

#Number of responses
#-------------------------------------------------------------------

nr<-1

#Name of the fixed effects parameters
#-------------------------------------

parameters<-c('ka','ke','v')

#Fixed effects parameters values
#--------------------------------
```

```
beta<-c(2,0.25,15)

#Diagonal Matrix of variance for the random effects:
#---------------------------------------------------

omega<-diag(c(1,0.25,0.1));


#Standard deviation of the residual error (sig.inter+sig.slope*f)^2:
#-------------------------------------------------------------------

sig.interA<-0.5
sig.slopeA<-0.15

#Between-subject variance model (1) = additive  (2) = exponential
#-----------------------------------------------------------------

Trand<-2;

#List of the vectors of sampling times for each elementary design
#you can specify any sampling times for a group by writing NULL
#ONLY if you have several responses
#-----------------------------------------------------------------

protA<-list(c(0.333,1.5,5,12),c(1,3,8))

#Vector of initial proportions or numbers of subjects for each elementary design
#-------------------------------------------------------------
subjects<-c(200,200)

#Subjects input: (1) for number of subjects (2) for proportions of subjects
#-------------------------------------------------------------------------
subjects.input<-2

#If 'proportions of subjects' give the total number of samples
#-------------------------------------------------------------
Ntot<-250

################### MODEL OPTION ############################

#Model form: Differential equations (DE) or analytical form (AF)
#---------------------------------------------------------------

modelform<-"DE"

###### ANALYTICAL MODEL OPTION ############################
#############################################################

#Identical dose in each elementary design (Yes=T, No=F)
#-----------------------------------------------------------
#dose.identical<-T

# If 'Yes', enter the value of the dose,
# else, enter the vector of the dose values for each elementary design
#-------------------------------------------------------------------
#dose<-c(100)

#Vector of last times for the intervals of each expression
#---------------------------------------------------------
#boundA<-list(c(0,Inf))


###### END ANALYTICAL MODEL OPTION ########################
```

```
###### DIFFERENTIAL EQUATION OPTION ##################################
#####################################################################

#Initial time for which initial conditions are given
#---------------------------------------------------

time.condinit<-0

#Identical initial conditions in each elementary design (Yes=T, No=F)
#-------------------------------------------------------------
condinit.identical<-T

# If 'Yes', enter once the expression of the initial values of the system at the
initial time
# else, enter the vectors of the initial conditions for each elementary design
# If initial values depend on parameters to be estimated,
# enter this parameter into the expression without any quotation marks
#-------------------------------------------------------

condinit<-expression(c(0,100))


# Error tolerance for solving differential equations
#---------------------------------------------------

RtolEQ<-1e-08
AtolEQ<-1e-08
Hmax<-Inf # Default value
###### END DIFFERENTIAL EQUATION OPTION ##############################



############ONLY FOR OPTIMISATION ###############################

#Identical sampling times for each response
# (only if you do not have sampling times==NULL)
#-------------------------------------------------------------------------------------------
#identical.times<-F

######## OPTIMISATION ALGORITHM OPTION ###############

#Character string for thoice of the optimisation algorithm:
#      "FW" for the Fedorov-Wynn algorithm
#      "SIMP" for the Simplex algorithm
#----------------------------------------

#algo.option<-"FW"


########################
#SIMPLEX SPECIFICATION #
########################

#Optimisation of the proportions of subjects: (Yes=T, No=F)
#-----------------------------------------------------------------

#subjects.opt<-T

#Vector of lower and upper admissible sampling times
#---------------------------------------------------


#lowerA<-c(0)
#upperA<-c(24)

#lowerB<-c(0)
#upperB<-c(24)
```

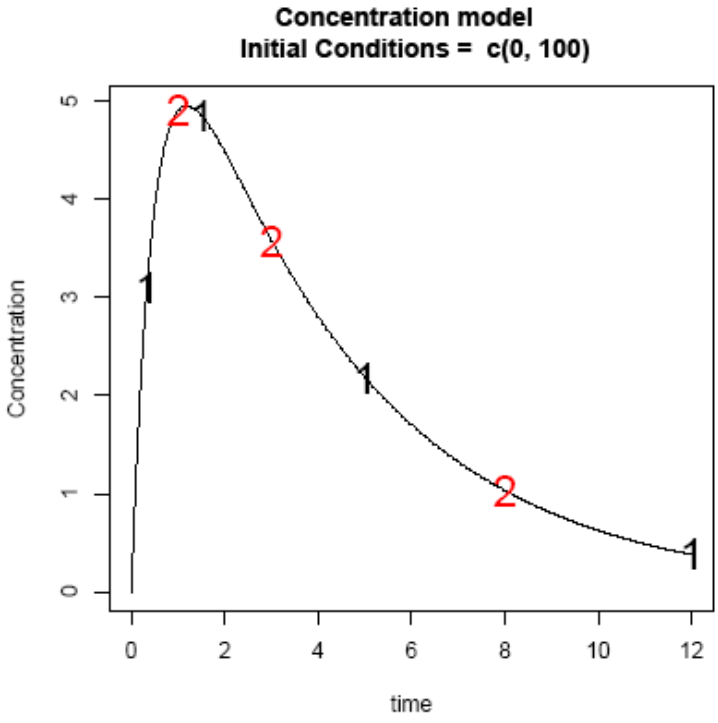
```
#Minimum delay between two sampling times
#-----------------------------------------

#delta.time<-0

#Print iteration step (Yes=T, No=F)
#---------------------------------

#iter.print<-T


#Parameter for initial simplex building (%)
#-----------------------------------------

#simplex.parameter<-20


#Maximum iteration number
#-----------------------

#Max.iter<-5000


#Relative convergence tolerance
#-----------------------------
#Rctol<-1e-6



############################
#FEDOROV-WYNN SPECIFICATION #
############################


#Number of sampling windows
#-------------------------
#nwindA<-1
#nwindB<-1

#List of vector of the allowed sampling times for each sampling window
#---------------------------------------------------------------------

#sampwinA<-list(c(0.166,1,2,3,6,9,12,20))
#sampwinB<-list(c(0.166,1,2,3,6,9,12,20,36))


#List of vector of allowed number of points to be taken from each sampling window
#--------------------------------------------------------------------------------

#nsampA<-list(c(4))
#nsampB<-list(c(4))

#Maximum total number of sampling times per subject
#-------------------------------------------------

#nmaxptsA<-4
#nmaxptsB<-4

#Minimum total number of sampling times per subject
#-------------------------------------------------

#nminptsA<-4
#nminptsB<-4
############# END OF OPTIMISATION ALGORITHM OPTION ###############
```

```
############## GRAPH SPECIFICATION OPTION ###############

#graphical representation (Yes=T, No=F)
#-------------------------------------

graph.logical<-T

#Names on Y axes for each response
#--------------------------------
names.data<-c("Concentration")

#Controls logarithmic axes for the graphical representation.
#Values "xy", "x", or "y" produce log-log or log-x or log-y axes.
#(For standard graphic, log.logical<-F)
#-------------------------------------------------------------

#log.logical<-'y'
log.logical<-F

#Vector of lower and upper sampling times for the graphical representation
#------------------------------------------------------------------------

graph.infA<-c(0)
graph.supA<-c(12)
#graph.infB<-c(0)
#graph.supB<-c(24)

#Vector of lower and upper concentration for the graphical representation
#-----------------------------------------------------------------------
y.range<-NULL # default range
#y.range<-c(0,10)
```

OUTPUT FILE:

```
PFIM 3.0

Project:  Doc_example1.1.3

Date:  Mon Sep 10 09:55:39 2007


*************************** INPUT SUMMARY *******************************

Differential Equations form of the model:

function(t,y,p)
{

      ka<-p[1]
      ke<-p[2]
      v<-p[3]

      yd1<-ka*y[2]-ke*y[1]
      yd2<--ka*y[2]
      list(c(yd1,yd2),c(y[1]/v))
}


Population design:


Sample times for response: A
                    subjects
c(0.333, 1.5, 5, 12)      200
c(1, 3, 8)                200


Variance error model response A : ( 0.5 + 0.15 *f)^2



Initial Conditions at time 0 :

0 100
0 100




Between-subject variance model: Trand =  2

Error tolerance for solving differential equations system: RtolEQ = 1e-08 , AtolEQ =
1e-08 , Hmax =  Inf

****************** POPULATION FISHER INFORMATION matrix ******************

          [,1]       [,2]      [,3]      [,4]       [,5]       [,6]       [,7]
[1,]  63.26917   197.2991  -7.74132   0.00000    0.00000    0.00000    0.00000
[2,] 197.29912 16408.6040 158.51127   0.00000    0.00000    0.00000    0.00000
[3,]  -7.74132   158.5113  10.05526   0.00000    0.00000    0.00000    0.00000
[4,]   0.00000     0.0000   0.00000  83.77010   12.47033   67.45659   61.45098
[5,]   0.00000     0.0000   0.00000  12.47033 1314.88025  441.77862  380.33088
[6,]   0.00000     0.0000   0.00000  67.45659  441.77862 6400.24113  579.51627
[7,]   0.00000     0.0000   0.00000  61.45098  380.33088  579.51627 1663.93553
[8,]   0.00000     0.0000   0.00000 233.89675  430.90541 2245.49187 2516.58821
          [,8]
[1,]    0.0000
[2,]    0.0000
[3,]    0.0000
```

```
[4,]  233.8968
[5,]  430.9054
[6,] 2245.4919
[7,] 2516.5882
[8,] 6701.4268


************************* EXPECTED STANDARD ERRORS ************************

------------------------ Fixed Effects Parameters ------------------------

    Beta    StdError       CV .
ka  2.00 0.141452149 7.072607 %
ke  0.25 0.009079485 3.631794 %
v  15.00 0.378081312 2.520542 %


------------------------- Variance of Random Effects ----------------------

   Omega   StdError       CV .
ka  1.00 0.11583943 11.58394 %
ke  0.25 0.02893458 11.57383 %
v   0.10 0.01357357 13.57357 %


------------------------ Variance of residual error -----------------------

            SIG   StdError       CV .
sig.interA 0.50 0.03907990  7.81598 %
sig.slopeA 0.15 0.02064161 13.76107 %


******************************* DETERMINANT *******************************

1.664864e+22

******************************** CRITERION ********************************

599.3387
```



1.2 Example 2: Two responses

1.2.1 Analytical form

The model is a joint model for a drug and its metabolite at steady-state (two responses). The first response is described by a one compartmental model with first order absorption and the second response is described by a one compartment with a first order metabolic rate constant.
The design to be evaluated is composed of 1 group of 50 subjects with sampling times at (1, 3, 6, 12) for the first response and sampling times at (1, 6, 11, 12) for the second response.

MODEL FILE

```
form1<-expression((dose*Ka/(V*(Ka-(Cl/V+R)))*((exp(-(Cl/V+R)*t)/(1-exp(-(Cl/V+R)*12)))-
(exp(-Ka*t)/(1-exp(-Ka*12))))))

formA<-c(form1)

form1<-expression((dose*Ka/(1/R)* ((exp(-(Cl/V+R)*t)/((Ka-(Cl/V+R))*((Clm/R)/(1/R)-
(Cl/V+R))*(1-exp(-(Cl/V+R)*12))))+ exp(-Ka*t)/(((Cl/V+R)-Ka)*((Clm/R)/(1/R)-Ka)*(1-
exp(-Ka*12)))+ exp(-((Clm/R)/(1/R))*t)/((Ka-(Clm/R)/(1/R))*((Cl/V+R)-(Clm/R)/(1/R))*(1-
exp(-(Clm/R)/(1/R) *12))))))

formB<-c(form1)

form<-c(formA,formB)
```

INPUT FILE

```
###############################################################################
##                                                                           ##
##                          INPUT FILE FOR PFIM 3.0                            ##
###############################################################################


#Name of the project
#--------------------

project<-"Doc_example1.2.1"


#Name of the file containing the PK or PD model
#-----------------------------------------------

file.model<-"model.r";


#Name of the output file for the results
#----------------------------------------

output<-"Stdout.r";

#RUN:  Evaluation (EVAL) or Optimisation (OPT)
#-----------------------------------------------------
run<-"EVAL"

#Number of responses
#-------------------------------------------------------------------

nr<-2

#Name of the fixed effects parameters
#-------------------------------------
```

```
parameters<-c('Ka','Cl','V','Clm','R')

#Fixed effects parameters values
#-------------------------------

beta<-c(2.8,160,300,0.16,0.003)

#Diagonal Matrix of variance for the random effects:
#---------------------------------------------------

omega<-diag(c(0,0.03,0,0.17,0));


#Standard deviation of the residual error (sig.inter+sig.slope*f)^2:
#-------------------------------------------------------------------

sig.interA<-0.000239
sig.slopeA<-0.276

sig.interB<-0
sig.slopeB<-0.135

#Between-subject variance model (1) = additive  (2) = exponential
#-----------------------------------------------------------------

Trand<-2;

#List of the vectors of sampling times for each elementary design
#you can specify any sampling times for a group by writing NULL
#ONLY if you have several responses
#-----------------------------------------------------------------

protA<-list(c(1,3,6,12))
protB<-list(c(1,6,11,12))

#Vector of initial proportions or numbers of subjects for each elementary design
#-------------------------------------------------------------
subjects<-c(50)

#Subjects input: (1) for number of subjects (2) for proportions of subjects
#--------------------------------------------------------------------------
subjects.input<-2

#If 'proportions of subjects' give the total number of samples
#-------------------------------------------------------------
Ntot<-250

################### MODEL OPTION ############################

#Model form: Differential equations (DE) or analytical form (AF)
#---------------------------------------------------------------

modelform<-"AF"

###### ANALYTICAL MODEL OPTION ############################
#########################################################

#Identical dose in each elementary design (Yes=T, No=F)
#-----------------------------------------------------------
dose.identical<-T

# If 'Yes', enter the value of the dose,
# else, enter the vector of the dose values for each elementary design
#-----------------------------------------------------------------
dose<-c(300)

#Vector of last times for the intervals of each expression
#---------------------------------------------------------
```

```
boundA<-list(c(0,Inf))
boundB<-list(c(0,Inf))

###### END ANALYTICAL MODEL OPTION ########################



###### DIFFERENTIAL EQUATION OPTION ###################################
#######################################################################

#Initial time for which initial conditions are given
#---------------------------------------------------

#time.condinit<-0

#Identical initial conditions in each elementary design (Yes=T, No=F)
#-------------------------------------------------------------
#condinit.identical<-T

# If 'Yes', enter once the expression of the initial values of the system at the
initial time
# else, enter the vectors of the initial conditions for each elementary design
# If initial values depend on parameters to be estimated,
# enter this parameter into the expression without any quotation marks
#-------------------------------------------------------

#condinit<-expression(c(0,100))


# Error tolerance for solving differential equations
#---------------------------------------------------

#RtolEQ<-1e-08
#AtolEQ<-1e-08
Hmax<-Inf # Default value
#Hmax<-Inf #<1.5/24
###### END DIFFERENTIAL EQUATION OPTION ###################################



############ONLY FOR OPTIMISATION ################################

#Identical sampling times for each response
# (only if you do not have sampling times==NULL)
#-----------------------------------------------------------------------------------
#identical.times<-F

######## OPTIMISATION ALGORITHM OPTION ###############

#Character string for thoice of the optimisation algorithm:
#      "FW" for the Fedorov-Wynn algorithm
#      "SIMP" for the Simplex algorithm
#-----------------------------------------

#algo.option<-"FW"


########################
#SIMPLEX SPECIFICATION #
########################

#Optimisation of the proportions of subjects: (Yes=T, No=F)
#-----------------------------------------------------------

#subjects.opt<-T

#Vector of lower and upper admissible sampling times
```

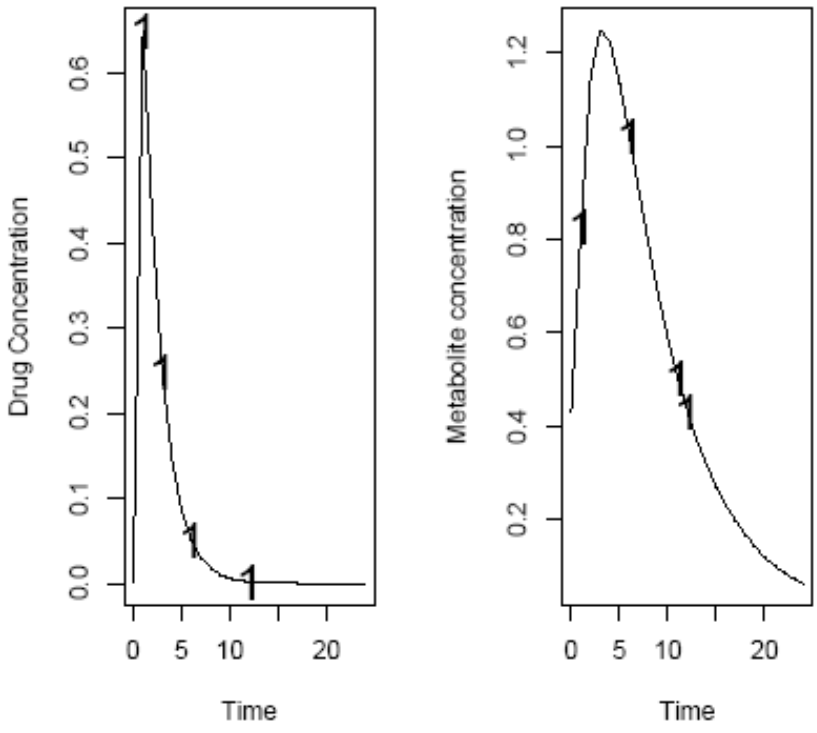
```
#-------------------------------------------------


#lowerA<-c(0)
#upperA<-c(24)

#lowerB<-c(0)
#upperB<-c(24)

#Minimum delay between two sampling times
#-----------------------------------------

#delta.time<-0

#Print iteration step (Yes=T, No=F)
#---------------------------------

#iter.print<-T


#Parameter for initial simplex building (%)
#-----------------------------------------

#simplex.parameter<-20


#Maximum iteration number
#------------------------

#Max.iter<-5000


#Relative convergence tolerance
#------------------------------
#Rctol<-1e-6



##############################
#FEDOROV-WYNN SPECIFICATION #
##############################


#Number of sampling windows
#--------------------------
#nwindA<-1
#nwindB<-1

#List of vector of the allowed sampling times for each sampling window
#---------------------------------------------------------------------

#sampwinA<-list(c(0.166,1,2,3,6,9,12,20))
#sampwinB<-list(c(0.166,1,2,3,6,9,12,20,36))


#List of vector of allowed number of points to be taken from each sampling window
#--------------------------------------------------------------------------------

#nsampA<-list(c(4))
#nsampB<-list(c(4))

#Maximum total number of sampling times per subject
#--------------------------------------------------

#nmaxptsA<-4
#nmaxptsB<-4

#Minimum total number of sampling times per subject
```

```
#-------------------------------------------------

#nminptsA<-4
#nminptsB<-4
############# END OF OPTIMISATION ALGORITHM OPTION ###############

############## GRAPH SPECIFICATION OPTION ###############

#graphical representation (Yes=T, No=F)
#-----------------------------------

graph.logical<-T

#Names on Y axes for each response
#--------------------------------
names.data<-c("Drug Concentration","Metabolite concentration")

#Controls logarithmic axes for the graphical representation.
#Values "xy", "x", or "y" produce log-log or log-x or log-y axes.
#(For standard graphic, log.logical<-F)
#-----------------------------------------------------------------

#log.logical<-'y'
log.logical<-F

#Vector of lower and upper sampling times for the graphical representation
#-------------------------------------------------------------------------

graph.infA<-c(0)
graph.supA<-c(24)
graph.infB<-c(0)
graph.supB<-c(24)

#Vector of lower and upper concentration for the graphical representation
#------------------------------------------------------------------------
y.range<-NULL # default range
#y.range<-c(0,10)
```

OUTPUT FILE:

```
PFIM 3.0

Project:  Doc_example1.2.1

Date:  Mon Sep 10 10:03:25 2007


*************************** INPUT SUMMARY *******************************

Analytical function models :

(dose * Ka/(V * (Ka - (Cl/V + R))) * ((exp(-(Cl/V + R) * t)/(1 - exp(-(Cl/V + R) *
12))) - (exp(-Ka * t)/(1 - exp(-Ka * 12)))))

(dose * Ka/(1/R) * ((exp(-(Cl/V + R) * t)/((Ka - (Cl/V + R)) * ((Clm/R)/(1/R) - (Cl/V +
R)) * (1 - exp(-(Cl/V + R) * 12)))) + exp(-Ka * t)/(((Cl/V + R) - Ka) * ((Clm/R)/(1/R)
- Ka) * (1 - exp(-Ka * 12))) + exp(-((Clm/R)/(1/R)) * t)/((Ka - (Clm/R)/(1/R)) * ((Cl/V
+ R) - (Clm/R)/(1/R)) * (1 - exp(-(Clm/R)/(1/R) * 12)))))



Population design:


Sample times for response: A
               subjects Dose
c(1, 3, 6, 12)       50  300
Sample times for response: B
                subjects Dose
c(1, 6, 11, 12)       50  300


Variance error model response A : ( 0.000239 + 0.276 *f)^2

Variance error model response B : ( 0 + 0.135 *f)^2



Between-subject variance model: Trand =  2

****************** POPULATION FISHER INFORMATION matrix ******************

               [,1]         [,2]         [,3]          [,4]          [,5]
 [1,]  1.880536e+01   0.19623258  -0.04105795 -2.204578e+01  2.816581e+04
 [2,]  1.962326e-01   0.05998843  -0.02382285  7.327439e-01 -6.453491e+01
 [3,] -4.105795e-02  -0.02382285   0.02039400 -1.561151e+00  1.686910e+02
 [4,] -2.204578e+01   0.73274389  -1.56115149  1.122399e+04 -4.079265e+05
 [5,]  2.816581e+04 -64.53491322 168.69095495 -4.079265e+05  1.157767e+08
 [6,]  0.000000e+00   0.00000000   0.00000000  0.000000e+00  0.000000e+00
 [7,]  0.000000e+00   0.00000000   0.00000000  0.000000e+00  0.000000e+00
 [8,]  0.000000e+00   0.00000000   0.00000000  0.000000e+00  0.000000e+00
 [9,]  0.000000e+00   0.00000000   0.00000000  0.000000e+00  0.000000e+00
[10,]  0.000000e+00   0.00000000   0.00000000  0.000000e+00  0.000000e+00
              [,6]        [,7]         [,8]         [,9]       [,10]
 [1,]     0.000000    0.000000        0.000 0.000000e+00     0.00000
 [2,]     0.000000    0.000000        0.000 0.000000e+00     0.00000
 [3,]     0.000000    0.000000        0.000 0.000000e+00     0.00000
 [4,]     0.000000    0.000000        0.000 0.000000e+00     0.00000
 [5,]     0.000000    0.000000        0.000 0.000000e+00     0.00000
 [6,] 23583.858950    3.518717    97795.648 3.419345e+02    12.80540
 [7,]     3.518717  825.609796     2006.351 7.015043e+00    29.63067
 [8,] 97795.647887 2006.351184 32748752.464 1.674357e+05  7301.56128
 [9,]   341.934506    7.015043   167435.739 3.637080e+03    25.52931
[10,]    12.805399   29.630675     7301.561 2.552931e+01 16319.46471
```

```
************************** EXPECTED STANDARD ERRORS ************************

------------------------ Fixed Effects Parameters ------------------------

       Beta     StdError        CV .
Ka    2.800 0.3068974696 10.960624 %
Cl  160.000 5.7106249862  3.569141 %
V   300.000 9.6715183829  3.223839 %
Clm   0.160 0.0104152237  6.509515 %
R     0.003 0.0001301779  4.339262 %


------------------------- Variance of Random Effects ----------------------

    Omega    StdError       CV .
Cl   0.03 0.006553635 21.84545 %
Clm  0.17 0.034806448 20.47438 %


------------------------ Variance of residual error ------------------------

                SIG     StdError        CV .
sig.interA 0.000239 0.0002010107 84.104903 %
sig.slopeA 0.276000 0.0189663144  6.871853 %
sig.slopeB 0.135000 0.0078285827  5.798950 %


******************************** DETERMINANT ********************************

2.202438e+32

******************************** CRITERION **********************************

1715.104
```



1.2.2 Differential equation system (ODE)

This is the same example as the previous example in the section 1.2.1 with differential equation option

The model is a joint model for a drug and its metabolite at steady-state (two responses). The first response is described by a one compartmental model with first order absorption and the second response is described by a one compartment with a first order metabolic rate constant.
The model is described using a differential equation system.
The design to be evaluated is composed of 1 group of 50 subjects with sampling times at (1, 3, 6, 12) for the first response and sampling times at (1, 6, 11, 12) for the second response.

MODE FILE:

```
formED<-function(t,y,p)
{
ka<-p[1]
cl<-p[2]
V<-p[3]
clm<-p[4]
R<-p[5]

yd1<--ka*y[1]
yd2<-ka*y[1]-cl/V*y[2]-R*y[2]
yd3<-R*y[2]-clm*y[3]

list(c(yd1,yd2,yd3),c(y[2]/V,y[3]))
}
```

INPUT FILE:

```
##############################################################################
##                                                                          ##
##                          INPUT FILE FOR PFIM 3.0                           ##
##############################################################################


#Name of the project
#--------------------

project<-"Doc_example1.2.2"


#Name of the file containing the PK or PD model
#-----------------------------------------------

file.model<-"model.r";


#Name of the output file for the results
#----------------------------------------

output<-"Stdout.r";

#RUN:  Evaluation (EVAL) or Optimisation (OPT)
#----------------------------------------------------------
run<-"EVAL"

#Number of responses
#-------------------------------------------------------------------------

nr<-2
```

```
#Name of the fixed effects parameters
#-------------------------------------

parameters<-c('Ka','Cl','V','Clm','R')

#Fixed effects parameters values
#-------------------------------

beta<-c(2.8,160,300,0.16,0.003)

#Diagonal Matrix of variance for the random effects:
#---------------------------------------------------

omega<-diag(c(0,0.03,0,0.17,0));


#Standard deviation of the residual error (sig.inter+sig.slope*f)^2:
#-------------------------------------------------------------------

sig.interA<-0.000239
sig.slopeA<-0.276

sig.interB<-0
sig.slopeB<-0.135

#Between-subject variance model (1) = additive  (2) = exponential
#-----------------------------------------------------------------

Trand<-2;

#List of the vectors of sampling times for each elementary design
#you can specify any sampling times for a group by writing NULL
#ONLY if you have several responses
#-----------------------------------------------------------------

protA<-list(c(1,3,6,12))
protB<-list(c(1,6,11,12))

#Vector of initial proportions or numbers of subjects for each elementary design
#-------------------------------------------------------------
subjects<-c(80)

#Subjects input: (1) for number of subjects (2) for proportions of subjects
#--------------------------------------------------------------------------
#subjects.input<-2

#If 'proportions of subjects' give the total number of samples
#-------------------------------------------------------------
#Ntot<-250

################### MODEL OPTION ############################

#Model form: Differential equations (DE) or analytical form (AF)
#---------------------------------------------------------------

modelform<-"DE"

###### ANALYTICAL MODEL OPTION ############################
############################################################

#Identical dose in each elementary design (Yes=T, No=F)
#------------------------------------------------------
#dose.identical<-T

# If 'Yes', enter the value of the dose,
# else, enter the vector of the dose values for each elementary design
#---------------------------------------------------------------------
#dose<-c(300)
```

```
#Vector of last times for the intervals of each expression
#-------------------------------------------------------
#boundA<-list(c(0,Inf))
#boundB<-list(c(0,Inf))

###### END ANALYTICAL MODEL OPTION ########################



###### DIFFERENTIAL EQUATION OPTION ###################################
#########################################################################

#Initial time for which initial conditions are given
#---------------------------------------------------

time.condinit<-0

#Identical initial conditions in each elementary design (Yes=T, No=F)
#-------------------------------------------------------------
condinit.identical<-T

# If 'Yes', enter once the expression of the initial values of the system at the
initial time
# else, enter the vectors of the initial conditions for each elementary design
# If initial values depend on parameters to be estimated,
# enter this parameter into the expression without any quotation marks
#-----------------------------------------------------

condinit<-expression(c(300,0,0))

# Error tolerance for solving differential equations
#---------------------------------------------------

RtolEQ<-1e-08
AtolEQ<-1e-08
Hmax<-Inf # Default value
#Hmax<-Inf #<1.5/24
###### END DIFFERENTIAL EQUATION OPTION ###################################



############ONLY FOR OPTIMISATION ################################

#Identical sampling times for each response
# (only if you do not have sampling times==NULL)
#-------------------------------------------------------------------------------------
#identical.times<-F

######## OPTIMISATION ALGORITHM OPTION ###############

#Character string for thoice of the optimisation algorithm:
#      "FW" for the Fedorov-Wynn algorithm
#      "SIMP" for the Simplex algorithm
#-----------------------------------------

#algo.option<-"FW"


########################
#SIMPLEX SPECIFICATION #
########################

#Optimisation of the proportions of subjects: (Yes=T, No=F)
#-----------------------------------------------------------

#subjects.opt<-T
```

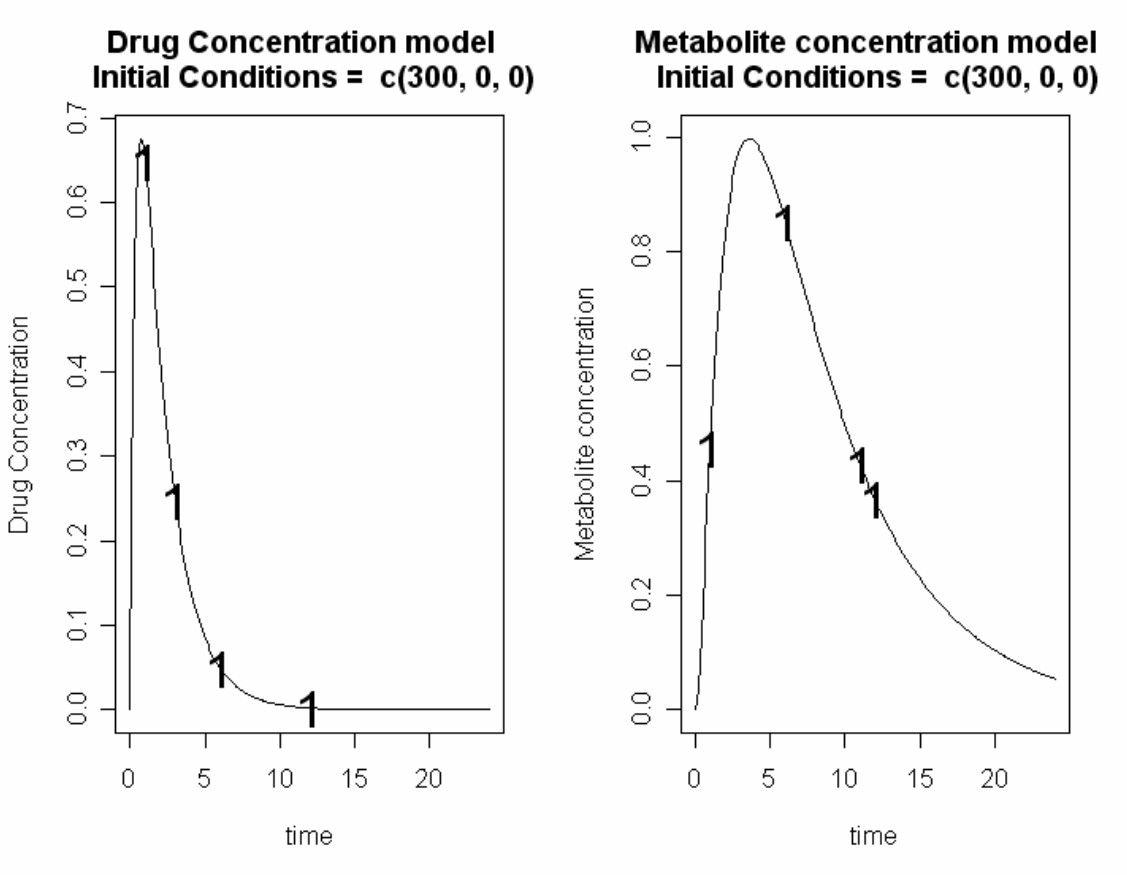
```
#Vector of lower and upper admissible sampling times
#---------------------------------------------------


#lowerA<-c(0)
#upperA<-c(24)

#lowerB<-c(0)
#upperB<-c(24)

#Minimum delay between two sampling times
#----------------------------------------

#delta.time<-0

#Print iteration step (Yes=T, No=F)
#---------------------------------

#iter.print<-T


#Parameter for initial simplex building (%)
#-----------------------------------------

#simplex.parameter<-20


#Maximum iteration number
#------------------------

#Max.iter<-5000


#Relative convergence tolerance
#------------------------------
#Rctol<-1e-6



##############################
#FEDOROV-WYNN SPECIFICATION #
##############################


#Number of sampling windows
#--------------------------
#nwindA<-1
#nwindB<-1

#List of vector of the allowed sampling times for each sampling window
#---------------------------------------------------------------------

#sampwinA<-list(c(0.166,1,2,3,6,9,12,20))
#sampwinB<-list(c(0.166,1,2,3,6,9,12,20,36))


#List of vector of allowed number of points to be taken from each sampling window
#--------------------------------------------------------------------------------

#nsampA<-list(c(4))
#nsampB<-list(c(4))

#Maximum total number of sampling times per subject
#--------------------------------------------------

#nmaxptsA<-4
#nmaxptsB<-4
```

```
#Minimum total number of sampling times per subject
#-------------------------------------------------

#nminptsA<-4
#nminptsB<-4
############# END OF OPTIMISATION ALGORITHM OPTION ###############




############## GRAPH SPECIFICATION OPTION ###############

#graphical representation (Yes=T, No=F)
#-------------------------------------

graph.logical<-T

#Names on Y axes for each response
#--------------------------------
names.data<-c("Drug Concentration","Metabolite concentration")

#Controls logarithmic axes for the graphical representation.
#Values "xy", "x", or "y" produce log-log or log-x or log-y axes.
#(For standard graphic, log.logical<-F)
#-------------------------------------------------------------

#log.logical<-'y'
log.logical<-F

#Vector of lower and upper sampling times for the graphical representation
#------------------------------------------------------------------------

graph.infA<-c(0)
graph.supA<-c(24)
graph.infB<-c(0)
graph.supB<-c(24)

#Vector of lower and upper concentration for the graphical representation
#-----------------------------------------------------------------------
y.range<-NULL # default range
#y.range<-c(0,10)

############# END OF GRAPH SPECIFICATION OPTION ###############
```

OUTPUT FILE:

```
PFIM 3.0

Project:  Doc_example1.2.2

Date:  Mon Sep 10 10:11:18 2007



*************************** INPUT SUMMARY *******************************

Differential Equations form of the model:

function(t,y,p)
{
ka<-p[1]
cl<-p[2]
V<-p[3]
clm<-p[4]
R<-p[5]

yd1<--ka*y[1]
yd2<-ka*y[1]-cl/V*y[2]-R*y[2]
yd3<-R*y[2]-clm*y[3]

list(c(yd1,yd2,yd3),c(y[2]/V,y[3]))
}


Population design:


Sample times for response: A
               subjects
c(1, 3, 6, 12)       80
Sample times for response: B
                subjects
c(1, 6, 11, 12)       80


Variance error model response A : ( 0.000239 + 0.276 *f)^2

Variance error model response B : ( 0 + 0.135 *f)^2



Initial Conditions at time 0 :

300 0 0




Between-subject variance model: Trand =  2

Error tolerance for solving differential equations system: RtolEQ = 1e-08 , AtolEQ =
1e-08 , Hmax =  Inf

****************** POPULATION FISHER INFORMATION matrix ******************

               [,1]          [,2]         [,3]          [,4]          [,5]
 [1,]  9.208030e+01    0.27145053   0.02622077 -9.399841e+00  1.953868e+05
 [2,]  2.714505e-01    0.09607511  -0.03837654  1.419415e+00 -2.843070e+02
 [3,]  2.622077e-02   -0.03837654   0.03332167 -3.013591e+00  6.540011e+02
 [4,] -9.399841e+00    1.41941542  -3.01359075  1.767545e+04 -7.258633e+05
 [5,]  1.953868e+05 -284.30697369 654.00110232 -7.258633e+05  6.058467e+08
 [6,]  0.000000e+00    0.00000000   0.00000000  0.000000e+00  0.000000e+00
```

```
 [7,]  0.000000e+00    0.00000000   0.00000000  0.000000e+00  0.000000e+00
 [8,]  0.000000e+00    0.00000000   0.00000000  0.000000e+00  0.000000e+00
 [9,]  0.000000e+00    0.00000000   0.00000000  0.000000e+00  0.000000e+00
[10,]  0.000000e+00    0.00000000   0.00000000  0.000000e+00  0.000000e+00
              [,6]        [,7]        [,8]         [,9]       [,10]
 [1,] 0.000000e+00    0.000000        0.00      0.00000     0.00000
 [2,] 0.000000e+00    0.000000        0.00      0.00000     0.00000
 [3,] 0.000000e+00    0.000000        0.00      0.00000     0.00000
 [4,] 0.000000e+00    0.000000        0.00      0.00000     0.00000
 [5,] 0.000000e+00    0.000000        0.00      0.00000     0.00000
 [6,] 3.780783e+04    8.252376   152416.32    531.06071    38.86665
 [7,] 8.252376e+00 1279.678447     4690.15     16.34178    83.51408
 [8,] 1.524163e+05 4690.149761 54374422.93 267405.91647 22089.44650
 [9,] 5.310607e+02   16.341782   267405.92   5825.84197    76.96576
[10,] 3.886665e+01   83.514082    22089.45     76.96576 25909.69795


************************* EXPECTED STANDARD ERRORS ************************

------------------------ Fixed Effects Parameters -------------------------

       Beta     StdError       CV .
Ka    2.800 2.008917e-01 7.174704 %
Cl  160.000 4.460304e+00 2.787690 %
V   300.000 7.601815e+00 2.533938 %
Clm   0.160 8.189305e-03 5.118316 %
R     0.003 8.034842e-05 2.678281 %


------------------------- Variance of Random Effects ----------------------

    Omega    StdError       CV .
Cl   0.03 0.005172954 17.24318 %
Clm  0.17 0.027961705 16.44806 %


------------------------ Variance of residual error -----------------------

                SIG     StdError        CV .
sig.interA 0.000239 0.0001549660 64.839349 %
sig.slopeA 0.276000 0.0148915322  5.395483 %
sig.slopeB 0.135000 0.0062142632  4.603158 %


****************************** DETERMINANT ********************************

1.271735e+35

****************************** CRITERION **********************************

3239.214
```


Drug Concentration model
Initial Conditions = c(300, 0, 0)
Drug Concentration
0.0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0
5
10
15
20
time
Metabolite concentration model
Initial Conditions = c(300, 0, 0)
Metabolite concentration
0.0
0.2
0.4
0.6
0.8
1.0
0
5
10
15
20
time

1.2.3 Analytical form (library of model and a group with NULL)

This example deals with the evaluation of a joint modelling of a drug concentration and its effect: (two responses): a one compartment model with a first order absorption and elimination for the drug concentration is used and an Imax with constant baseline model for the effect. The model is described using an analytical form.
The design to be evaluated is composed of 2 groups:

- ***one*** *group of 32 with sampling times at (.5,1,2,3,6,9,12,24,36,48,72,96,120) for the first response and (0.5,1,2,3,6,9,12) for the second response.*

- ***one*** *group of 16 subjects with sampling times at (0,24,36,48,72,96,120,144) for the first response and any sampling times for the second response.*

The second group has only samples for the first response.

MODEL FILE

```
source(paste(directory.program,dirsep,»libraryModels.r»,sep=««))

formA<-oral1_1cpt_kaVCl()[[1]]

formB<-paste("-Imax*",formA,"/(C50+",formA,")+S0")
formB<-parse(text=formB)

form<-c(formA,formB)
```

INPUT FILE

```
##############################################################################
##                                                                          ##
##                          INPUT FILE FOR PFIM 3.0                           ##
##############################################################################


#Name of the project
#--------------------

project<-"Doc_example1.2.3"


#Name of the file containing the PK or PD model
#-----------------------------------------------

file.model<-"model.r";


#Name of the output file for the results
#----------------------------------------

output<-"Stdout.r";

#RUN:  Evaluation (EVAL) or Optimisation (OPT)
#------------------------------------------------------
run<-"EVAL"

#Number of responses
#-------------------------------------------------------------------

nr<-2

#Name of the fixed effects parameters
```

```
#-----------------------------------

parameters<-c('ka','Cl','V','Imax','C50','S0')

#Fixed effects parameters values
#-------------------------------

beta<-c(0.591,0.134,7.74,72.9,0.169,95.8)

#Diagonal Matrix of variance for the random effects:
#--------------------------------------------------

omega<-diag(c(0.515,0.0817,0.0505,0.0012,0.688,0.000541));


#Standard deviation of the residual error (sig.inter+sig.slope*f)^2:
#-------------------------------------------------------------------

sig.interA<-0.0596
sig.slopeA<-0.0737

sig.interB<-8.28
sig.slopeB<-0

#Between-subject variance model (1) = additive  (2) = exponential
#----------------------------------------------------------------

Trand<-2;

#List of the vectors of sampling times for each elementary design
#you can specify any sampling times for a group by writing NULL
#ONLY if you have several responses
#----------------------------------------------------------------

protA<-list(c(0.5,1,2,3,6,9,12,24,36,48,72,96,120),c(0.5,1,2,3,6,9,12))
protB<-list(c(0,24,36,48,72,96,120,144),c(NULL))

#Vector of initial proportions or numbers of subjects for each elementary design
#-------------------------------------------------------------
subjects<-c(32,16)

#Subjects input: (1) for number of subjects (2) for proportions of subjects
#--------------------------------------------------------------------------
#subjects.input<-2

#If 'proportions of subjects' give the total number of samples
#------------------------------------------------------------
#Ntot<-250

################### MODEL OPTION ############################

#Model form: Differential equations (DE) or analytical form (AF)
#---------------------------------------------------------------

modelform<-"AF"

###### ANALYTICAL MODEL OPTION ############################
#########################################################

#Identical dose in each elementary design (Yes=T, No=F)
#-----------------------------------------------------
dose.identical<-T

# If 'Yes', enter the value of the dose,
# else, enter the vector of the dose values for each elementary design
#--------------------------------------------------------------------
dose<-c(200)
```

```
#Vector of last times for the intervals of each expression
#-------------------------------------------------------
boundA<-list(c(0,Inf))
boundB<-list(c(0,Inf))

###### END ANALYTICAL MODEL OPTION ########################



###### DIFFERENTIAL EQUATION OPTION ###################################
#######################################################################

#Initial time for which initial conditions are given
#---------------------------------------------------

#time.condinit<-0

#Identical initial conditions in each elementary design (Yes=T, No=F)
#-------------------------------------------------------------
#condinit.identical<-T

# If 'Yes', enter once the expression of the initial values of the system at the
initial time
# else, enter the vectors of the initial conditions for each elementary design
# If initial values depend on parameters to be estimated,
# enter this parameter into the expression without any quotation marks
#-------------------------------------------------------

#condinit<-expression(c(300,0,0))

# Error tolerance for solving differential equations
#---------------------------------------------------

RtolEQ<-1e-08
AtolEQ<-1e-08
Hmax<-Inf # Default value
#Hmax<-Inf #<1.5/24
###### END DIFFERENTIAL EQUATION OPTION ###################################



############ONLY FOR OPTIMISATION ################################

#Identical sampling times for each response
# (only if you do not have sampling times==NULL)
#-----------------------------------------------------------------------------------------
#identical.times<-F

######## OPTIMISATION ALGORITHM OPTION ###############

#Character string for thoice of the optimisation algorithm:
#      "FW" for the Fedorov-Wynn algorithm
#      "SIMP" for the Simplex algorithm
#-----------------------------------------

#algo.option<-"FW"


########################
#SIMPLEX SPECIFICATION #
########################

#Optimisation of the proportions of subjects: (Yes=T, No=F)
#-----------------------------------------------------------

#subjects.opt<-T
```

```
#Vector of lower and upper admissible sampling times
#---------------------------------------------------


#lowerA<-c(0)
#upperA<-c(24)

#lowerB<-c(0)
#upperB<-c(24)

#Minimum delay between two sampling times
#----------------------------------------

#delta.time<-0

#Print iteration step (Yes=T, No=F)
#---------------------------------

#iter.print<-T


#Parameter for initial simplex building (%)
#-----------------------------------------

#simplex.parameter<-20


#Maximum iteration number
#------------------------

#Max.iter<-5000


#Relative convergence tolerance
#------------------------------
#Rctol<-1e-6



##############################
#FEDOROV-WYNN SPECIFICATION #
##############################


#Number of sampling windows
#--------------------------
#nwindA<-1
#nwindB<-1

#List of vector of the allowed sampling times for each sampling window
#---------------------------------------------------------------------

#sampwinA<-list(c(0.166,1,2,3,6,9,12,20))
#sampwinB<-list(c(0.166,1,2,3,6,9,12,20,36))


#List of vector of allowed number of points to be taken from each sampling window
#--------------------------------------------------------------------------------

#nsampA<-list(c(4))
#nsampB<-list(c(4))

#Maximum total number of sampling times per subject
#--------------------------------------------------

#nmaxptsA<-4
#nmaxptsB<-4
```

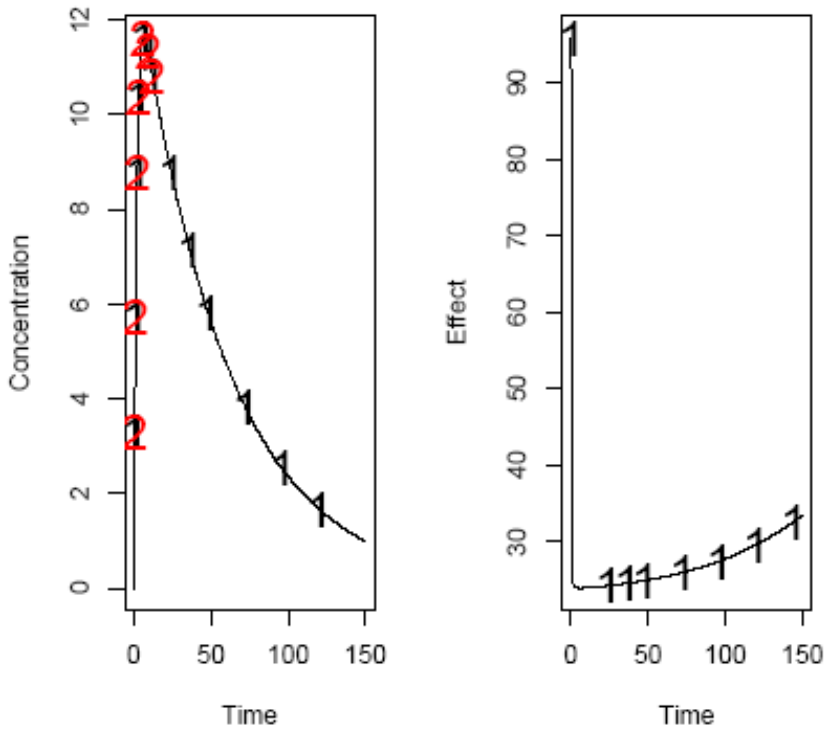
```
#Minimum total number of sampling times per subject
#-------------------------------------------------

#nminptsA<-4
#nminptsB<-4
############# END OF OPTIMISATION ALGORITHM OPTION ###############




############## GRAPH SPECIFICATION OPTION ###############

#graphical representation (Yes=T, No=F)
#-------------------------------------

graph.logical<-T

#Names on Y axes for each response
#--------------------------------
names.data<-c("Concentration","Effect")

#Controls logarithmic axes for the graphical representation.
#Values "xy", "x", or "y" produce log-log or log-x or log-y axes.
#(For standard graphic, log.logical<-F)
#-----------------------------------------------------------------

#log.logical<-'y'
log.logical<-F

#Vector of lower and upper sampling times for the graphical representation
#------------------------------------------------------------------------

graph.infA<-c(0)
graph.supA<-c(150)
graph.infB<-c(0)
graph.supB<-c(150)

#Vector of lower and upper concentration for the graphical representation
#-----------------------------------------------------------------------
y.range<-NULL # default range
#y.range<-c(0,10)

############# END OF GRAPH SPECIFICATION OPTION ###############
```

OUTPUT FILE:

```
PFIM 3.0

Project:  Doc_example1.2.3

Date:  Wed Apr 16 11:01:30 2008


*************************** INPUT SUMMARY *******************************

Analytical function models :

dose/V * ka/(ka - (Cl/V)) * (exp(-(Cl/V) * t) - exp(-ka * t))

-Imax * dose/V * ka/(ka - (Cl/V)) * (exp(-(Cl/V) * t) - exp(-ka * t))/(C50 + dose/V *
ka/(ka - (Cl/V)) * (exp(-(Cl/V) * t) - exp(-ka * t))) + S0



Population design:


Sample times for response: A
[[1]]
 [1]   0.5   1.0   2.0   3.0   6.0   9.0  12.0  24.0  36.0  48.0  72.0  96.0
[13] 120.0

[[2]]
[1]  0.5  1.0  2.0  3.0  6.0  9.0 12.0

               Number of subjects per group
 32 16


Sample times for response: B
[[1]]
[1]   0  24  36  48  72  96 120 144

[[2]]
NULL

               Number of subjects per group
 32 16




Variance error model response A : ( 0.0596 + 0.0737 *f)^2

Variance error model response B : ( 8.28 + 0 *f)^2



Between-subject variance model: Trand =  2

****************** POPULATION FISHER INFORMATION matrix ******************

               [,1]          [,2]         [,3]          [,4]        [,5]
 [1,] 262.518980869    57.6979316 -1.261249762   0.004618294  -0.3689308
 [2,]  57.697931627 22871.3972560 33.670143483  -0.255938346   8.7117868
 [3,]  -1.261249762    33.6701435 15.192991991   0.002458308  -0.1897739
 [4,]   0.004618294    -0.2559383  0.002458308   1.315321009 -14.1758690
 [5,]  -0.368930786     8.7117868 -0.189773852 -14.175868997 365.0503118
 [6,]  -0.005168854     0.2588288 -0.002735817  -1.258375515  14.0594326
 [7,]   0.000000000     0.0000000  0.000000000   0.000000000   0.0000000
 [8,]   0.000000000     0.0000000  0.000000000   0.000000000   0.0000000
```

```
 [9,]   0.000000000     0.0000000  0.000000000   0.000000000   0.0000000
[10,]   0.000000000     0.0000000  0.000000000   0.000000000   0.0000000
[11,]   0.000000000     0.0000000  0.000000000   0.000000000   0.0000000
[12,]   0.000000000     0.0000000  0.000000000   0.000000000   0.0000000
[13,]   0.000000000     0.0000000  0.000000000   0.000000000   0.0000000
[14,]   0.000000000     0.0000000  0.000000000   0.000000000   0.0000000
[15,]   0.000000000     0.0000000  0.000000000   0.000000000   0.0000000
              [,6]         [,7]         [,8]         [,9]         [,10]
 [1,] -0.005168854 0.000000e+00 0.000000e+00 0.000000e+00 0.000000e+00
 [2,]  0.258828829 0.000000e+00 0.000000e+00 0.000000e+00 0.000000e+00
 [3,] -0.002735817 0.000000e+00 0.000000e+00 0.000000e+00 0.000000e+00
 [4,] -1.258375515 0.000000e+00 0.000000e+00 0.000000e+00 0.000000e+00
 [5,] 14.059432559 0.000000e+00 0.000000e+00 0.000000e+00 0.000000e+00
 [6,]  1.640295860 0.000000e+00 0.000000e+00 0.000000e+00 0.000000e+00
 [7,]  0.000000000 8.758031e+01 7.245942e-01 4.135635e-01 6.186045e-04
 [8,]  0.000000000 7.245942e-01 2.365669e+03 4.268813e+01 9.766870e-02
 [9,]  0.000000000 4.135635e-01 4.268813e+01 8.632887e+03 3.006282e-02
[10,]  0.000000000 6.186045e-04 9.766870e-02 3.006282e-02 7.634729e+05
[11,]  0.000000000 2.121574e-05 6.081595e-04 9.628267e-04 4.765932e+02
[12,]  0.000000000 1.338180e-03 1.724987e-01 6.429943e-02 1.206776e+06
[13,]  0.000000000 1.593732e+00 1.849638e+01 2.463419e+01 4.988302e+00
[14,]  0.000000000 1.668706e+01 2.437493e+02 3.985970e+02 2.292600e+01
[15,]  0.000000000 1.778469e-05 3.201969e-04 7.971768e-04 3.857838e+02
              [,11]        [,12]        [,13]        [,14]        [,15]
 [1,] 0.000000e+00 0.000000e+00 0.000000e+00 0.000000e+00 0.000000e+00
 [2,] 0.000000e+00 0.000000e+00 0.000000e+00 0.000000e+00 0.000000e+00
 [3,] 0.000000e+00 0.000000e+00 0.000000e+00 0.000000e+00 0.000000e+00
 [4,] 0.000000e+00 0.000000e+00 0.000000e+00 0.000000e+00 0.000000e+00
 [5,] 0.000000e+00 0.000000e+00 0.000000e+00 0.000000e+00 0.000000e+00
 [6,] 0.000000e+00 0.000000e+00 0.000000e+00 0.000000e+00 0.000000e+00
 [7,] 2.121574e-05 1.338180e-03 1.593732e+00 1.668706e+01 1.778469e-05
 [8,] 6.081595e-04 1.724987e-01 1.849638e+01 2.437493e+02 3.201969e-04
 [9,] 9.628267e-04 6.429943e-02 2.463419e+01 3.985970e+02 7.971768e-04
[10,] 4.765932e+02 1.206776e+06 4.988302e+00 2.292600e+01 3.857838e+02
[11,] 1.698526e+00 8.095805e+02 4.163839e-02 1.939721e-01 7.309473e-01
[12,] 8.095805e+02 3.541004e+06 9.020200e+00 4.136031e+01 8.706294e+02
[13,] 4.163839e-02 9.020200e+00 1.233417e+03 1.066845e+04 2.459059e-02
[14,] 1.939721e-01 4.136031e+01 1.066845e+04 1.282890e+05 1.183754e-01
[15,] 7.309473e-01 8.706294e+02 2.459059e-02 1.183754e-01 6.426506e+00


************************* EXPECTED STANDARD ERRORS ************************

------------------------ Fixed Effects Parameters -------------------------

       Beta    StdError        CV .
ka    0.591 0.061750381 10.448457 %
Cl    0.134 0.006625192  4.944173 %
V     7.740 0.257033155  3.320842 %
Imax 72.900 1.816840342  2.492236 %
C50   0.169 0.068729044 40.668073 %
S0   95.800 1.515785991  1.582240 %


------------------------- Variance of Random Effects ----------------------

        Omega     StdError        CV .
ka   0.515000 0.1068570013  20.74893 %
Cl   0.081700 0.0205629539  25.16885 %
V    0.050500 0.0107640841  21.31502 %
Imax 0.001200 0.0017526767 146.05640 %
C50  0.688000 0.8569644713 124.55879 %
S0   0.000541 0.0007852696 145.15150 %


------------------------ Variance of residual error -----------------------

              SIG    StdError        CV .
```

```
sig.interA 0.0596 0.053742954 90.172742 %
sig.slopeA 0.0737 0.005270083  7.150723 %
sig.interB 8.2800 0.407418889  4.920518 %


****************************** DETERMINANT ********************************

9.213388e+39

****************************** CRITERION **********************************

461.6306
```



1.3 Example 3: Three responses with ODE

This example deals with the PK of one drug and two metabolites. The model is described by a differential equation system. The first two compartments correspond to the drug. Compartments 3 and 4 correspond to metabolites.
The design to be evaluated is composed of one group of 8 subjects with 16 samples for the first response (drug), 14 samples for the first metabolite and 13 samples for the second metabolite.

Reference: *Gueorguieva, Aarons & al. Optimal design for multivariate response pharmacokinetics models. Journal of Pharmacokinetics and Pharmacodynamics. 2006 33(2): 97-123.*

MODEL FILE

```
formED<-function(t,y,p)
{

      CLD<-p[1]
      V1<-p[2]
      V2<-p[3]
      CL<-p[4]
      FM<-p[5]
      RM<-p[6]
      FG1<-p[7]
      RG1<-p[8]
      TInf<-0.1666
      R<-(50)/TInf


      if (t<=TInf) yd1<--CLD/V1*y[1]+CLD/V2*y[2]-CL/V1*y[1]+R
      else yd1<--CLD/V1*y[1]+CLD/V2*y[2]-CL/V1*y[1]
      yd2<-CLD/V1*y[1]-CLD/V2*y[2]
      yd3<-FG1*CL/V1*y[1]-y[3]*RG1
      yd4<-FM*CL/V1*y[1]-RM*y[4]

      list(c(yd1,yd2,yd3,yd4),c(y[2]/V1,y[3]/5.92,y[4]/5.92))
}
```

INPUT FILE

```
##############################################################################
##                                                                          ##
##                        INPUT FILE FOR PFIM 3.0                           ##
##############################################################################


#Name of the project
#--------------------

project<-"Doc_example1.3"


#Name of the file containing the PK or PD model
#-----------------------------------------------

file.model<-"model.r";




#Name of the output file for the results
#----------------------------------------
```

```
output<-"Stdout.r";

#RUN:  Evaluation (EVAL) or Optimisation (OPT)
#-----------------------------------------------------
run<-"EVAL"

#Number of responses
#-------------------------------------------------------------------

nr<-3

#Name of the fixed effects parameters
#-------------------------------------

parameters<-c('CLD','V1','V2','CL','FM','RM','FG1','RG1')

#Fixed effects parameters values
#-------------------------------

beta<-c(3.99,5.92,13.6,6.8,0.003,0.0051,0.54,2.415)

#Diagonal Matrix of variance for the random effects:
#---------------------------------------------------

omega<-diag(c(0,0.0625,0.0625,0.0625,0,0,0.0625,0.0625));


#Standard deviation of the residual error (sig.inter+sig.slope*f)^2:
#-------------------------------------------------------------------

sig.interA<-0
sig.slopeA<-0.223

sig.interB<-0
sig.slopeB<-0.3

sig.interC<-0
sig.slopeC<-0.141

#Between-subject variance model (1) = additive  (2) = exponential
#-----------------------------------------------------------------

Trand<-2;

#List of the vectors of sampling times for each elementary design
#you can specify any sampling times for a group by writing NULL
#ONLY if you have several responses
#-----------------------------------------------------------------

protA<-list(c(0.0831,0.166,0.25,0.5,0.75,1,1.5,2,3,4,6,8,12,16,24,48))
protB<-list(c(0.25,0.5,0.75,1,1.5,2,3,4,6,8,12,16,24,48))
protC<-list(c(0.5,0.75,1,1.5,2,3,4,6,8,12,16,24,48))

#Vector of initial proportions or numbers of subjects for each elementary design
#-------------------------------------------------------------
subjects<-c(8)

#Subjects input: (1) for number of subjects (2) for proportions of subjects
#--------------------------------------------------------------------------
#subjects.input<-2

#If 'proportions of subjects' give the total number of samples
#-------------------------------------------------------------
#Ntot<-250

################### MODEL OPTION ############################

#Model form: Differential equations (DE) or analytical form (AF)
```

```
#-------------------------------------------------------------

modelform<-"DE"

###### ANALYTICAL MODEL OPTION ############################
##########################################################

#Identical dose in each elementary design (Yes=T, No=F)
#---------------------------------------------------------
#dose.identical<-T

# If 'Yes', enter the value of the dose,
# else, enter the vector of the dose values for each elementary design
#-----------------------------------------------------------------
#dose<-c(300)

#Vector of last times for the intervals of each expression
#---------------------------------------------------------
#boundA<-list(c(0,Inf))
#boundB<-list(c(0,Inf))

###### END ANALYTICAL MODEL OPTION ########################



###### DIFFERENTIAL EQUATION OPTION ###################################
#######################################################################

#Initial time for which initial conditions are given
#---------------------------------------------------

time.condinit<-0

#Identical initial conditions in each elementary design (Yes=T, No=F)
#-------------------------------------------------------------
condinit.identical<-T

# If 'Yes', enter once the expression of the initial values of the system at the
initial time
# else, enter the vectors of the initial conditions for each elementary design
# If initial values depend on parameters to be estimated,
# enter this parameter into the expression without any quotation marks
#-----------------------------------------------------

condinit<-expression(c(0,0,0,0))

# Error tolerance for solving differential equations
#---------------------------------------------------

RtolEQ<-1e-08
AtolEQ<-1e-08
Hmax<-Inf # Default value
#Hmax<-Inf #<1.5/24
###### END DIFFERENTIAL EQUATION OPTION ###################################



############ONLY FOR OPTIMISATION ################################
```

```
#Identical sampling times for each response
# (only if you do not have sampling times==NULL)
#-------------------------------------------------------------------------------------
#identical.times<-F

######## OPTIMISATION ALGORITHM OPTION ###############

#Character string for thoice of the optimisation algorithm:
#      "FW" for the Fedorov-Wynn algorithm
#      "SIMP" for the Simplex algorithm
#-----------------------------------------

#algo.option<-"FW"


########################
#SIMPLEX SPECIFICATION #
########################

#Optimisation of the proportions of subjects: (Yes=T, No=F)
#-------------------------------------------------------------

#subjects.opt<-T

#Vector of lower and upper admissible sampling times
#-------------------------------------------------


#lowerA<-c(0)
#upperA<-c(24)

#lowerB<-c(0)
#upperB<-c(24)

#Minimum delay between two sampling times
#-----------------------------------------

#delta.time<-0

#Print iteration step (Yes=T, No=F)
#---------------------------------

#iter.print<-T


#Parameter for initial simplex building (%)
#-----------------------------------------

#simplex.parameter<-20


#Maximum iteration number
#------------------------

#Max.iter<-5000


#Relative convergence tolerance
#-----------------------------
#Rctol<-1e-6




############################
#FEDOROV-WYNN SPECIFICATION #
```

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


#Number of sampling windows
#--------------------------
#nwindA<-1
#nwindB<-1

#List of vector of the allowed sampling times for each sampling window
#---------------------------------------------------------------------

#sampwinA<-list(c(0.166,1,2,3,6,9,12,20))
#sampwinB<-list(c(0.166,1,2,3,6,9,12,20,36))


#List of vector of allowed number of points to be taken from each sampling window
#--------------------------------------------------------------------------------

#nsampA<-list(c(4))
#nsampB<-list(c(4))

#Maximum total number of sampling times per subject
#--------------------------------------------------

#nmaxptsA<-4
#nmaxptsB<-4

#Minimum total number of sampling times per subject
#--------------------------------------------------

#nminptsA<-4
#nminptsB<-4
############# END OF OPTIMISATION ALGORITHM OPTION ###############




############## GRAPH SPECIFICATION OPTION ###############

#graphical representation (Yes=T, No=F)
#-------------------------------------

graph.logical<-T

#Names on Y axes for each response
#--------------------------------
names.data<-c("Tolcapone concentration","Metabolite_G1","Metabolite_M")

#Controls logarithmic axes for the graphical representation.
#Values "xy", "x", or "y" produce log-log or log-x or log-y axes.
#(For standard graphic, log.logical<-F)
#-------------------------------------------------------------

#log.logical<-'y'
log.logical<-F

#Vector of lower and upper sampling times for the graphical representation
#-------------------------------------------------------------------------

graph.infA<-c(0.000000001)
graph.supA<-c(48)
graph.infB<-c(0.000001)
graph.supB<-c(48)
graph.infC<-c(0.000001)
graph.supC<-c(48)
```

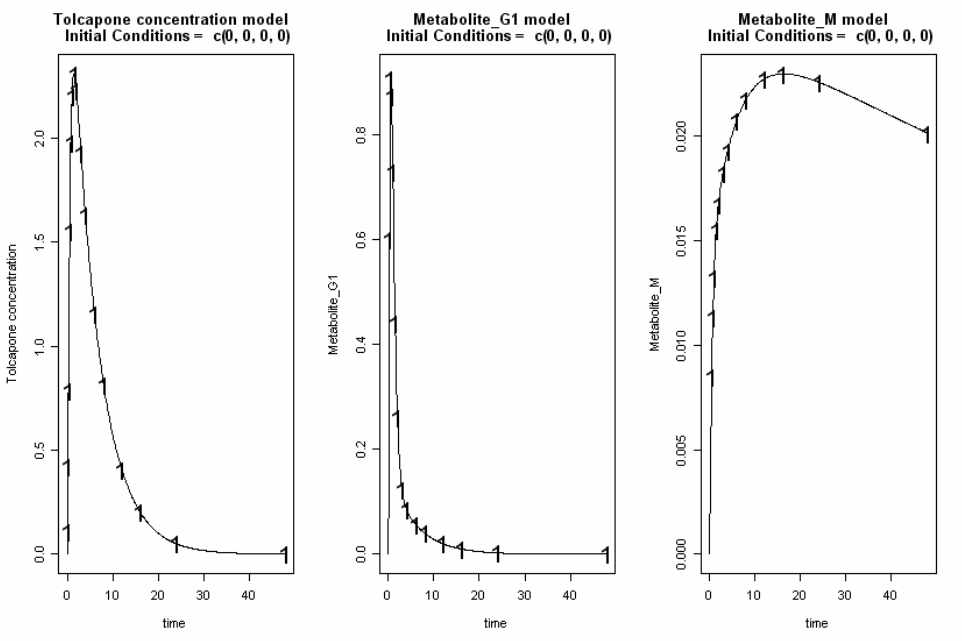
```
#Vector of lower and upper concentration for the graphical representation
#-------------------------------------------------------------------------
y.range<-NULL # default range
#y.range<-c(0,10)
```

OUTPUT FILE:

```
PFIM 3.0

Project:  Doc_example1.3

Date:  Mon Sep 10 11:09:53 2007


*************************** INPUT SUMMARY *******************************

Differential Equations form of the model:

function(t,y,p)
{

      CLD<-p[1]
      V1<-p[2]
      V2<-p[3]
      CL<-p[4]
      FM<-p[5]
      RM<-p[6]
      FG1<-p[7]
      RG1<-p[8]
      TInf<-0.1666
      R<-(50)/TInf


      if (t<=TInf) yd1<--CLD/V1*y[1]+CLD/V2*y[2]-CL/V1*y[1]+R
      else yd1<--CLD/V1*y[1]+CLD/V2*y[2]-CL/V1*y[1]
      yd2<-CLD/V1*y[1]-CLD/V2*y[2]
      yd3<-FG1*CL/V1*y[1]-y[3]*RG1
      yd4<-FM*CL/V1*y[1]-RM*y[4]

      list(c(yd1,yd2,yd3,yd4), c(y[2]/V1,y[3]/5.92,y[4]/5.92))
}


Population design:


Sample times for response: A
                                                                        subjects
c(0.0831, 0.166, 0.25, 0.5, 0.75, 1, 1.5, 2, 3, 4, 6, 8, 12, 16, 24, 48)        8
Sample times for response: B
                                                          subjects
c(0.25, 0.5, 0.75, 1, 1.5, 2, 3, 4, 6, 8, 12, 16, 24, 48)        8
Sample times for response: C
                                                       subjects
c(0.5, 0.75, 1, 1.5, 2, 3, 4, 6, 8, 12, 16, 24, 48)        8


Variance error model response A : ( 0 + 0.223 *f)^2

Variance error model response B : ( 0 + 0.3 *f)^2

Variance error model response C : ( 0 + 0.141 *f)^2



Initial Conditions at time 0 :

0 0 0 0
```

```
Between-subject variance model: Trand =  2

Error tolerance for solving differential equations system: RtolEQ = 1e-08 , AtolEQ
= 1e-08 , Hmax =  Inf

****************** POPULATION FISHER INFORMATION matrix ******************

              [,1]          [,2]          [,3]          [,4]          [,5]
 [1,]     15.602142 -2.004450e+00 -1.768904e+00   -2.13480215    -20342.626
 [2,]     -2.004450  3.555666e+00 -4.960337e-04    0.00640370     -1562.062
 [3,]     -1.768904 -4.960337e-04  6.683636e-01   -0.06437158      1590.274
 [4,]     -2.134802  6.403700e-03 -6.437158e-02    2.54865900      5743.871
 [5,] -20342.626127 -1.562062e+03  1.590274e+03 5743.87059834 410015616.728
 [6,]    130.302266  4.637291e+00 -4.479792e+00  -18.75962528 -14118846.997
 [7,]     -5.522343 -7.211581e-01  1.019857e-01    0.60223743    -27198.984
 [8,]     -2.000095 -1.916056e-01  5.711367e-02    0.27222787    -10122.880
 [9,]      0.000000  0.000000e+00  0.000000e+00    0.00000000         0.000
[10,]      0.000000  0.000000e+00  0.000000e+00    0.00000000         0.000
[11,]      0.000000  0.000000e+00  0.000000e+00    0.00000000         0.000
[12,]      0.000000  0.000000e+00  0.000000e+00    0.00000000         0.000
[13,]      0.000000  0.000000e+00  0.000000e+00    0.00000000         0.000
[14,]      0.000000  0.000000e+00  0.000000e+00    0.00000000         0.000
[15,]      0.000000  0.000000e+00  0.000000e+00    0.00000000         0.000
[16,]      0.000000  0.000000e+00  0.000000e+00    0.00000000         0.000
              [,6]          [,7]          [,8]          [,9]         [,10]
 [1,]  1.303023e+02 -5.522343e+00 -2.000095e+00 0.000000e+00 0.000000e+00
 [2,]  4.637291e+00 -7.211581e-01 -1.916056e-01 0.000000e+00 0.000000e+00
 [3,] -4.479792e+00  1.019857e-01  5.711367e-02 0.000000e+00 0.000000e+00
 [4,] -1.875963e+01  6.022374e-01  2.722279e-01 0.000000e+00 0.000000e+00
 [5,] -1.411885e+07 -2.719898e+04 -1.012288e+04 0.000000e+00 0.000000e+00
 [6,]  1.136410e+06  1.171089e+02  3.991061e+01 0.000000e+00 0.000000e+00
 [7,]  1.171089e+02  3.132631e+02 -2.160562e+01 0.000000e+00 0.000000e+00
 [8,]  3.991061e+01 -2.160562e+01  1.624380e+01 0.000000e+00 0.000000e+00
 [9,]  0.000000e+00  0.000000e+00  0.000000e+00 9.705297e+02 9.968358e-05
[10,]  0.000000e+00  0.000000e+00  0.000000e+00 9.968358e-05 9.551273e+02
[11,]  0.000000e+00  0.000000e+00  0.000000e+00 4.153395e-03 2.214950e+00
[12,]  0.000000e+00  0.000000e+00  0.000000e+00 3.321788e-01 3.506094e-02
[13,]  0.000000e+00  0.000000e+00  0.000000e+00 4.690016e-01 2.199238e-01
[14,]  0.000000e+00  0.000000e+00  0.000000e+00 1.214577e+01 5.488007e+00
[15,]  0.000000e+00  0.000000e+00  0.000000e+00 4.743043e-01 3.540587e+00
[16,]  0.000000e+00  0.000000e+00  0.000000e+00 2.451151e+00 1.159478e+01
             [,11]         [,12]         [,13]         [,14]         [,15]
 [1,] 0.000000e+00   0.00000000   0.0000000     0.000000     0.0000000
 [2,] 0.000000e+00   0.00000000   0.0000000     0.000000     0.0000000
 [3,] 0.000000e+00   0.00000000   0.0000000     0.000000     0.0000000
 [4,] 0.000000e+00   0.00000000   0.0000000     0.000000     0.0000000
 [5,] 0.000000e+00   0.00000000   0.0000000     0.000000     0.0000000
 [6,] 0.000000e+00   0.00000000   0.0000000     0.000000     0.0000000
 [7,] 0.000000e+00   0.00000000   0.0000000     0.000000     0.0000000
 [8,] 0.000000e+00   0.00000000   0.0000000     0.000000     0.0000000
 [9,] 4.153395e-03   0.33217881   0.4690016    12.145771     0.4743043
[10,] 2.214950e+00   0.03506094   0.2199238     5.488007     3.5405868
[11,] 8.680388e+02   0.30564750   1.2491002     7.482452     7.0349621
[12,] 3.056475e-01 521.52298129  49.6175599    11.267669    55.4588332
[13,] 1.249100e+00  49.61755992 560.9497674    12.667793    44.5186338
[14,] 7.482452e+00  11.26766892  12.6677928 4468.070324    40.8858855
[15,] 7.034962e+00  55.45883325  44.5186338    40.885885 2049.3472480
[16,] 3.613004e+01   5.10987194  14.1406771    28.931455    43.4287009
             [,16]
 [1,]    0.000000
 [2,]    0.000000
 [3,]    0.000000
 [4,]    0.000000
 [5,]    0.000000
 [6,]    0.000000
 [7,]    0.000000
 [8,]    0.000000
 [9,]    2.451151
```

```
[10,]   11.594784
[11,]   36.130044
[12,]    5.109872
[13,]   14.140677
[14,]   28.931455
[15,]   43.428701
[16,] 9789.573229


************************* EXPECTED STANDARD ERRORS ************************

------------------------ Fixed Effects Parameters ------------------------

       Beta     StdError        CV .
CLD  3.9900 3.907344e-01  9.792841 %
V1   5.9200 5.822866e-01  9.835922 %
V2  13.6000 1.590804e+00 11.697091 %
CL   6.8000 7.105785e-01 10.449685 %
FM   0.0030 7.385455e-05  2.461818 %
RM   0.0051 1.299158e-03 25.473682 %
FG1  0.5400 6.178332e-02 11.441355 %
RG1  2.4150 2.778339e-01 11.504511 %


------------------------ Variance of Random Effects ----------------------

     Omega   StdError       CV .
V1  0.0625 0.03209987 51.35978 %
V2  0.0625 0.03235762 51.77219 %
CL  0.0625 0.03394487 54.31180 %
FG1 0.0625 0.04403035 70.44856 %
RG1 0.0625 0.04243115 67.88985 %


------------------------ Variance of residual error ----------------------

             SIG   StdError       CV .
sig.slopeA 0.223 0.01496288 6.709811 %
sig.slopeB 0.300 0.02213982 7.379940 %
sig.slopeC 0.141 0.01010848 7.169132 %


******************************* DETERMINANT *******************************

8.820274e+44

******************************** CRITERION ********************************

644.3067
```


Tolcapone concentration model
Initial Conditions = c(0, 0, 0, 0)
Tolcapone concentration
time
Metabolite_G1 model
Initial Conditions = c(0, 0, 0, 0)
Metabolite_G1
time
Metabolite_M model
Initial Conditions = c(0, 0, 0, 0)
Metabolite_M
time

2 Optimisation

2.1 Example 1: Single response and analytical form

2.1.1 Simplex algorithm

The purpose is to optimise a design using a one compartment first order absorption model (analytical form). The design to be evaluated is composed of 4 sampling times for total number 360 samples. In this example, the Simplex is used.

MODEL INPUT

```
source(paste(directory.program,dirsep,»libraryModels.r»,sep=««))

formA<-infusion_2cpt_Vkk12k21(TInf=1.5/24) [[1]]

form<-c(formA)
```

INPUT FILE

```
#############################################################################
##                                                                         ##
##                         INPUT FILE FOR PFIM 3.0                           ##
#############################################################################


#Name of the project
#--------------------

project<-"Doc_example2.1.1"


#Name of the file containing the PK or PD model
#-----------------------------------------------

file.model<-"model.r";


#Name of the output file for the results
#----------------------------------------

output<-"Stdout.r";

#RUN:  Evaluation (EVAL) or Optimisation (OPT)
#-----------------------------------------------------
run<-"OPT"

#Number of responses
#-------------------------------------------------------------------

nr<-1

#Name of the fixed effects parameters
#-------------------------------------

parameters<-c('k','V','k12','k21')

#Fixed effects parameters values
#--------------------------------

beta<-c(0.08,3.08,0.175,0.116)
```

```
#Diagonal Matrix of variance for the random effects:
#-----------------------------------------------

omega<-diag(c(0.25,0.1,0.3,0.5));


#Standard deviation of the residual error (sig.inter+sig.slope*f)^2:
#-------------------------------------------------------------------

sig.interA<-0
sig.slopeA<-0.25

#Between-subject variance model (1) = additive  (2) = exponential
#-----------------------------------------------------------------

Trand<-2;

#List of the vectors of sampling times for each elementary design
#you can specify any sampling times for a group by writing NULL
#ONLY if you have several responses
#-----------------------------------------------------------------

protA<-list(c(0.0625,7,14,20.58))


#Vector of proportions or numbers of subjects for each elementary design
#----------------------------------------------------------------
subjects<-c(1)

#Subjects input: (1) for number of subjects (2) for proportions of subjects
#-------------------------------------------------------------------------
subjects.input<-2

#If 'proportions of subjects' give the total number of samples
#------------------------------------------------------------
Ntot<-360

################### MODEL OPTION ############################

#Model form: Differential equations (DE) or analytical form (AF)
#---------------------------------------------------------------

modelform<-"AF"

###### ANALYTICAL MODEL OPTION ############################
############################################################

#Identical dose in each elementary design (Yes=T, No=F)
#-----------------------------------------------------------
dose.identical<-T

# If 'Yes', enter the value of the dose,
# else, enter the vector of the dose values for each elementary design
#-------------------------------------------------------------------
dose<-c(400)

#Vector of last times for the intervals of each expression
#---------------------------------------------------------
boundA<-list(c(0,1.5/24),c(1.5/24,Inf))

###### END ANALYTICAL MODEL OPTION ######################




###### DIFFERENTIAL EQUATION OPTION ###################################
#######################################################################
```

```
#Initial time for which initial conditions are given
#-------------------------------------------------

#time.condinit<-0

#Identical initial conditions in each elementary design (Yes=T, No=F)
#-----------------------------------------------------------
#condinit.identical<-T

# If 'Yes', enter once the expression of the initial values of the system at the
initial time
# else, enter the vectors of the initial conditions for each elementary design
# If initial values depend on parameters to be estimated,
# enter this parameter into the expression without any quotation marks
#-----------------------------------------------------

#condinit<-expression(c(0,0,0,0))

# Error tolerance for solving differential equations
#-------------------------------------------------

#RtolEQ<-1e-08
#AtolEQ<-1e-08
#Hmax<-Inf # Default value
#Hmax<-Inf #<1.5/24
###### END DIFFERENTIAL EQUATION OPTION ##################################



############ONLY FOR OPTIMISATION ################################

#Identical sampling times for each response
# (only if you do not have sampling times==NULL)
#-------------------------------------------------------------------------------------
identical.times<-F

######## OPTIMISATION ALGORITHM OPTION ###############

#Character string for thoice of the optimisation algorithm:
#      "FW" for the Fedorov-Wynn algorithm
#      "SIMP" for the Simplex algorithm
#-----------------------------------------

algo.option<-"SIMP"


########################
#SIMPLEX SPECIFICATION #
########################

#Optimisation of the proportions of subjects: (Yes=T, No=F)
#-------------------------------------------------------------

subjects.opt<-F

#Vector of lower and upper admissible sampling times
#-------------------------------------------------


lowerA<-c(0)
upperA<-c(21)

#Minimum delay between two sampling times
#-----------------------------------------

delta.time<-0

#Print iteration step (Yes=T, No=F)
```

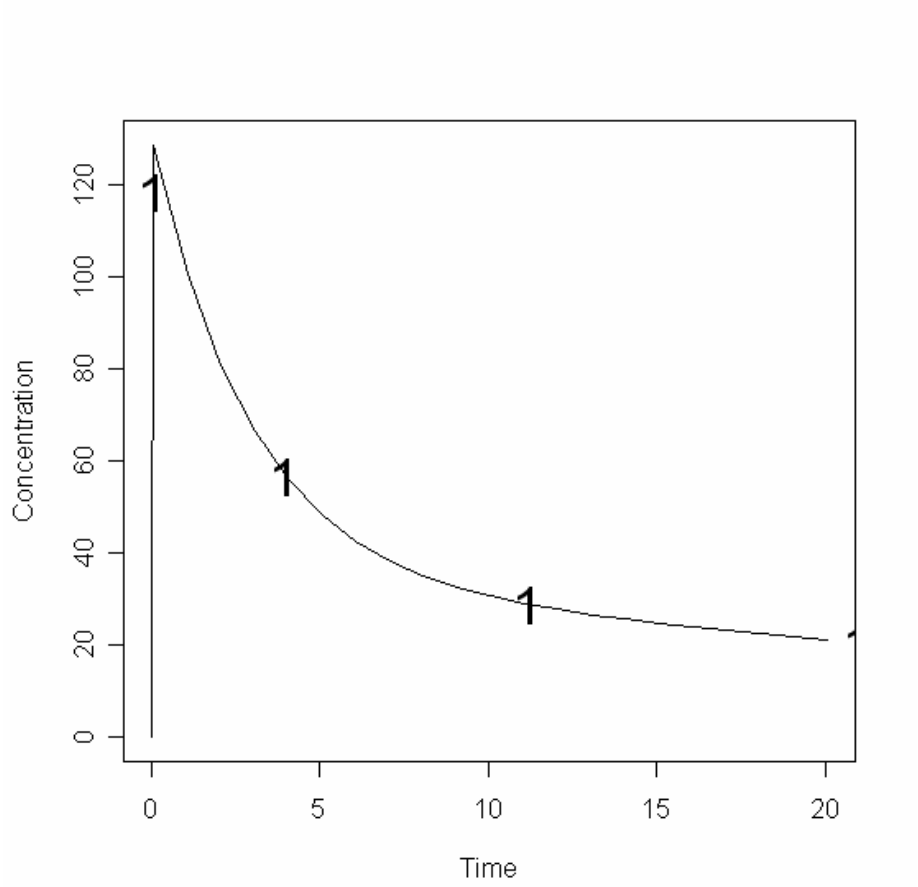
```
#--------------------------------

iter.print<-T


#Parameter for initial simplex building (%)
#----------------------------------------

simplex.parameter<-20


#Maximum iteration number
#------------------------

Max.iter<-5000


#Relative convergence tolerance
#------------------------------
Rctol<-1e-6



##############################
#FEDOROV-WYNN SPECIFICATION #
##############################


#Number of sampling windows
#--------------------------
#nwindA<-1
#nwindB<-1

#List of vector of the allowed sampling times for each sampling window
#---------------------------------------------------------------------

#sampwinA<-list(c(0.166,1,2,3,6,9,12,20))
#sampwinB<-list(c(0.166,1,2,3,6,9,12,20,36))


#List of vector of allowed number of points to be taken from each sampling window
#--------------------------------------------------------------------------------

#nsampA<-list(c(4))
#nsampB<-list(c(4))

#Maximum total number of sampling times per subject
#--------------------------------------------------

#nmaxptsA<-4
#nmaxptsB<-4

#Minimum total number of sampling times per subject
#--------------------------------------------------

#nminptsA<-4
#nminptsB<-4
############# END OF OPTIMISATION ALGORITHM OPTION ###############




############## GRAPH SPECIFICATION OPTION ###############

#graphical representation (Yes=T, No=F)
#--------------------------------------
```

```
graph.logical<-T

#Names on Y axes for each response
#---------------------------------
names.data<-c("Concentration")

#Controls logarithmic axes for the graphical representation.
#Values "xy", "x", or "y" produce log-log or log-x or log-y axes.
#(For standard graphic, log.logical<-F)
#-----------------------------------------------------------------

#log.logical<-'y'
log.logical<-F

#Vector of lower and upper sampling times for the graphical representation
#-------------------------------------------------------------------------

graph.infA<-c(0)
graph.supA<-c(21)


#Vector of lower and upper concentration for the graphical representation
#-----------------------------------------------------------------------
y.range<-NULL # default range
#y.range<-c(0,10)

############## END OF GRAPH SPECIFICATION OPTION ###############
```

OUTPUT FILE:

```
PFIM 3.0

Project: Doc_example2.1.1

Date: Tue Nov 13 09:18:47 2007


*************************** INPUT SUMMARY *******************************

Analytical function model:

6400 * (1 * ((((k + k21 + k12) + sqrt((k + k21 + k12)^2 - 4 * k * k21))/2) - k21)/(V *
((((k + k21 + k12) + sqrt((k + k21 + k12)^2 - 4 * k * k21))/2) - (((k + k21 + k12) -
sqrt((k + k21 + k12)^2 - 4 * k * k21))/2)))/(((k + k21 + k12) + sqrt((k + k21 + k12)^2
- 4 * k * k21))/2) * (1 - exp(-(((k + k21 + k12) + sqrt((k + k21 + k12)^2 - 4 * k *
k21))/2) * t)) + 1 * (k21 - (((k + k21 + k12) - sqrt((k + k21 + k12)^2 - 4 * k *
k21))/2))/(V * ((((k + k21 + k12) + sqrt((k + k21 + k12)^2 - 4 * k * k21))/2) -

-6400 * (1 * ((((k + k21 + k12) + sqrt((k + k21 + k12)^2 - 4 * k * k21))/2) - k21)/(V *
((((k + k21 + k12) + sqrt((k + k21 + k12)^2 - 4 * k * k21))/2) - (((k + k21 + k12) -
sqrt((k + k21 + k12)^2 - 4 * k * k21))/2)))/(((k + k21 + k12) + sqrt((k + k21 + k12)^2
- 4 * k * k21))/2) * (exp(-(((k + k21 + k12) + sqrt((k + k21 + k12)^2 - 4 * k *
k21))/2) * 0.0625) - 1) * exp(-(((k + k21 + k12) + sqrt((k + k21 + k12)^2 - 4 * k *
k21))/2) * (t - 0.0625)) + 1 * (k21 - (((k + k21 + k12) - sqrt((k + k21 + k12)^2 -



Variance error model response A : ( 0 + 0.25 *f)^2

Between-subject variance model: Trand = 2

Initial population design:


Sample times for response: A
                        subjects.prop doses
c(0.0625, 7, 14, 20.58)             1   400



Total number of samples (nr responses): 360

Associated criterion value: 530.8472

Window of the allowed optimised sampling times:

Upper and lower admissible samples times for the response A : [ 0 : 21 ]


Minimum delay between two sampling times: 0

Optimisation of the proportions of subjects: FALSE


*************************** OPTIMISED DESIGN *****************************

Number of iterations: 108
Number of function evaluations: 141
Convergence Achieved


       Optimised population design :
```

```
Sample times for response: A
                          subjects.prop subjects
c(0.058, 3.95, 11.183, 21)             1       90
Associated optimised criterion: 627.9753


****************** POPULATION FISHER INFORMATION matrix ******************

            [,1]      [,2]       [,3]        [,4]       [,5]        [,6]
 [1,] 26391.0294 284.92066  1432.8498 -9036.62024   0.000000    0.000000
 [2,]   284.9207  65.49098   143.1435   -90.74009   0.000000    0.000000
 [3,]  1432.8498 143.14352  4090.8920 -1180.73265   0.000000    0.000000
 [4,] -9036.6202 -90.74009 -1180.7326  4985.85564   0.000000    0.000000
 [5,]     0.0000   0.00000     0.0000     0.00000 158.489357   27.381472
 [6,]     0.0000   0.00000     0.0000     0.00000  27.381472 2144.338643
 [7,]     0.0000   0.00000     0.0000     0.00000   2.235553   33.071084
 [8,]     0.0000   0.00000     0.0000     0.00000  39.069289    5.839065
 [9,]     0.0000   0.00000     0.0000     0.00000 175.083977  612.133903
            [,7]      [,8]       [,9]
 [1,]   0.000000  0.000000    0.00000
 [2,]   0.000000  0.000000    0.00000
 [3,]   0.000000  0.000000    0.00000
 [4,]   0.000000  0.000000    0.00000
 [5,]   2.235553 39.069289  175.08398
 [6,]  33.071084  5.839065  612.13390
 [7,]  87.199845  3.191705  248.15985
 [8,]   3.191705 25.005691   77.86601
 [9,] 248.159847 77.866010 4157.62383



************************** EXPECTED STANDARD ERRORS ************************

------------------------ Fixed Effects Parameters -------------------------

     Beta   StdError        CV .
k   0.080 0.01023816 12.797702 %
V   3.080 0.13143564  4.267391 %
k12 0.175 0.01692783  9.673044 %
k21 0.116 0.02384510 20.556123 %


------------------------- Variance of Random Effects ----------------------

    Omega   StdError       CV .
k    0.25 0.10197052 40.78821 %
V    0.10 0.02207736 22.07736 %
k12  0.30 0.11791642 39.30547 %
k21  0.50 0.25776876 51.55375 %


------------------------ Variance of residual error -----------------------

            SIG   StdError       CV .
sig.slopeA 0.25 0.01793514 7.174056 %


****************************** DETERMINANT *******************************

1.518738e+25

****************************** CRITERION *********************************

627.9753
```


Concentration
0
20
40
60
80
100
120
0
5
10
15
20
Time
1
1
1
1

2.1.2 Federov-Wynn algorithm

This example is the same as the example see in section 2.1.1. The Federov-Wynn algorithm is used instead of the Simplex algorithm to optimise the design. The design can be composed of several groups with 1 and 4 samples. It is an option of the Federov-Wynn algorithm.

INPUT FILE

```
##############################################################################
##                                                                          ##
##                          INPUT FILE FOR PFIM 3.0                           ##
##############################################################################


#Name of the project
#--------------------

project<-"Doc_example2.1.2"


#Name of the file containing the PK or PD model
#-----------------------------------------------

file.model<-"model.r";


#Name of the output file for the results
#----------------------------------------

output<-"Stdout.r";

#RUN:  Evaluation (EVAL) or Optimisation (OPT)
#----------------------------------------------------------
run<-"OPT"

#Number of responses
#-------------------------------------------------------------------------

nr<-1

#Name of the fixed effects parameters
#-------------------------------------

parameters<-c('k','V','k12','k21')

#Fixed effects parameters values
#--------------------------------

beta<-c(0.08,3.08,0.175,0.116)

#Diagonal Matrix of variance for the random effects:
#---------------------------------------------------

omega<-diag(c(0.25,0.1,0.3,0.5));


#Standard deviation of the residual error (sig.inter+sig.slope*f)^2:
#--------------------------------------------------------------------

sig.interA<-0
sig.slopeA<-0.25

#Between-subject variance model (1) = additive  (2) = exponential
#-----------------------------------------------------------------

Trand<-2;
```

```
#List of the vectors of sampling times for each elementary design
#you can specify any sampling times for a group by writing NULL
#ONLY if you have several responses
#-----------------------------------------------------------------

protA<-list(c(0.0625,7,14,20.58))


#Vector of proportions or numbers of subjects for each elementary design
#-----------------------------------------------------------------
subjects<-c(1)


#Subjects input: (1) for number of subjects (2) for proportions of subjects
#-------------------------------------------------------------------------
subjects.input<-2

#If 'proportions of subjects' give the total number of samples
#-----------------------------------------------------------
Ntot<-360

################### MODEL OPTION ############################

#Model form: Differential equations (DE) or analytical form (AF)
#-------------------------------------------------------------

modelform<-"AF"

###### ANALYTICAL MODEL OPTION ############################
############################################################

#Identical dose in each elementary design (Yes=T, No=F)
#-----------------------------------------------------------
dose.identical<-T

# If 'Yes', enter the value of the dose,
# else, enter the vector of the dose values for each elementary design
#-------------------------------------------------------------------
dose<-c(400)

#Vector of last times for the intervals of each expression
#--------------------------------------------------------
boundA<-list(c(0,1.5/24),c(1.5/24,Inf))

###### END ANALYTICAL MODEL OPTION ########################




###### DIFFERENTIAL EQUATION OPTION ###################################
#######################################################################

#Initial time for which initial conditions are given
#--------------------------------------------------

#time.condinit<-0

#Identical initial conditions in each elementary design (Yes=T, No=F)
#-----------------------------------------------------------
#condinit.identical<-T

# If 'Yes', enter once the expression of the initial values of the system at the
initial time
# else, enter the vectors of the initial conditions for each elementary design
# If initial values depend on parameters to be estimated,
# enter this parameter into the expression without any quotation marks
#-----------------------------------------------------------
```

```
#condinit<-expression(c(0,0,0,0))


# Error tolerance for solving differential equations
#-----------------------------------------------------

#RtolEQ<-1e-08
#AtolEQ<-1e-08
#Hmax<-Inf # Default value
#Hmax<-Inf #<1.5/24
###### END DIFFERENTIAL EQUATION OPTION ###################################



############ONLY FOR OPTIMISATION ##############################

#Identical sampling times for each response
# (only if you do not have sampling times==NULL)
#-------------------------------------------------------------------------------------
identical.times<-T

######## OPTIMISATION ALGORITHM OPTION ###############

#Character string for thoice of the optimisation algorithm:
#      "FW" for the Fedorov-Wynn algorithm
#      "SIMP" for the Simplex algorithm
#-----------------------------------------

algo.option<-"FW"


########################
#SIMPLEX SPECIFICATION #
########################

#Optimisation of the proportions of subjects: (Yes=T, No=F)
#---------------------------------------------------------------

#subjects.opt<-F

#Vector of lower and upper admissible sampling times
#--------------------------------------------------


#lowerA<-c(0)
#upperA<-c(21)

#Minimum delay between two sampling times
#-----------------------------------------

#delta.time<-0

#Print iteration step (Yes=T, No=F)
#---------------------------------

#iter.print<-T


#Parameter for initial simplex building (%)
#-----------------------------------------

#simplex.parameter<-20


#Maximum iteration number
#-----------------------
```

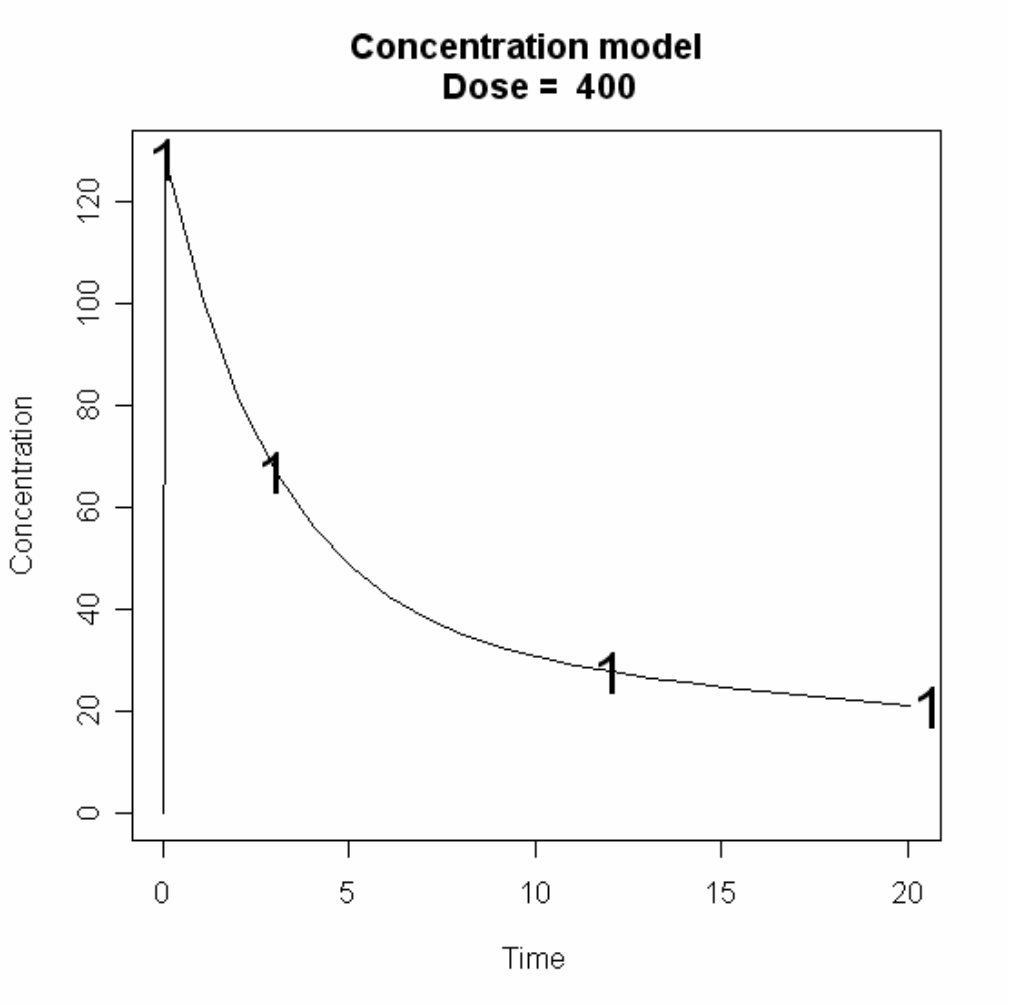
```
#Max.iter<-5000


#Relative convergence tolerance
#-----------------------------
#Rctol<-1e-6



############################
#FEDOROV-WYNN SPECIFICATION #
############################


#Number of sampling windows
#-------------------------
nwindA<-1
#nwindB<-1

#List of vector of the allowed sampling times for each sampling window
#---------------------------------------------------------------------

sampwinA<-list(c(0.0625,1,3,7,12,14,15,20.58))
#sampwinB<-list(c(0.166,1,2,3,6,9,12,20,36))


#List of vector of allowed number of points to be taken from each sampling window
#--------------------------------------------------------------------------------

nsampA<-list(c(4,1))
#nsampB<-list(c(4))

#Maximum total number of sampling times per subject
#-------------------------------------------------

nmaxptsA<-4
#nmaxptsB<-4

#Minimum total number of sampling times per subject
#-------------------------------------------------

nminptsA<-1
#nminptsB<-4
############# END OF OPTIMISATION ALGORITHM OPTION ###############




############## GRAPH SPECIFICATION OPTION ###############

#graphical representation (Yes=T, No=F)
#-------------------------------------

graph.logical<-T

#Names on Y axes for each response
#--------------------------------
names.data<-c("Concentration")

#Controls logarithmic axes for the graphical representation.
#Values "xy", "x", or "y" produce log-log or log-x or log-y axes.
#(For standard graphic, log.logical<-F)
#-------------------------------------------------------------

#log.logical<-'y'
log.logical<-F
```

```
#Vector of lower and upper sampling times for the graphical representation
#------------------------------------------------------------------------

graph.infA<-c(0)
graph.supA<-c(21)


#Vector of lower and upper concentration for the graphical representation
#-----------------------------------------------------------------------
y.range<-NULL # default range
#y.range<-c(0,10)

############## END OF GRAPH SPECIFICATION OPTION ###############
```

OUTPUT FILE:

```
PFIM 3.0

Project: Doc_example2.1.2

Date: Tue Nov 13 09:29:48 2007



*************************** INPUT SUMMARY *******************************

Analytical function model:

6400 * (1 * ((((k + k21 + k12) + sqrt((k + k21 + k12)^2 - 4 * k * k21))/2) - k21)/(V *
((((k + k21 + k12) + sqrt((k + k21 + k12)^2 - 4 * k * k21))/2) - (((k + k21 + k12) -
sqrt((k + k21 + k12)^2 - 4 * k * k21))/2)))/(((k + k21 + k12) + sqrt((k + k21 + k12)^2
- 4 * k * k21))/2) * (1 - exp(-(((k + k21 + k12) + sqrt((k + k21 + k12)^2 - 4 * k *
k21))/2) * t)) + 1 * (k21 - (((k + k21 + k12) - sqrt((k + k21 + k12)^2 - 4 * k *
k21))/2))/(V * ((((k + k21 + k12) + sqrt((k + k21 + k12)^2 - 4 * k * k21))/2) -

-6400 * (1 * ((((k + k21 + k12) + sqrt((k + k21 + k12)^2 - 4 * k * k21))/2) - k21)/(V *
((((k + k21 + k12) + sqrt((k + k21 + k12)^2 - 4 * k * k21))/2) - (((k + k21 + k12) -
sqrt((k + k21 + k12)^2 - 4 * k * k21))/2)))/(((k + k21 + k12) + sqrt((k + k21 + k12)^2
- 4 * k * k21))/2) * (exp(-(((k + k21 + k12) + sqrt((k + k21 + k12)^2 - 4 * k *
k21))/2) * 0.0625) - 1) * exp(-(((k + k21 + k12) + sqrt((k + k21 + k12)^2 - 4 * k *
k21))/2) * (t - 0.0625)) + 1 * (k21 - (((k + k21 + k12) - sqrt((k + k21 + k12)^2 -



Variance error model response A : ( 0 + 0.25 *f)^2

Between-subject variance model: Trand = 2

Initial population design:


Sample times for response: A
                 Protocol subjects doses
1 c(0.0625, 7, 14, 20.58)       90   400




Between-subject variance model: Trand = 2

Total number of samples: 360

Associated criterion value: 530.8472



Sampling windows for the response: A
Window 1 : t= 0.0625 1 3 7 12 14 15 20.58
    Nb of sampling points to be taken in this window, n[ 1 ]= 4 1
Maximum total number of points in one elementary protocol : 4
Minimum total number of points in one elementary protocol : 1




*************************** OPTIMISED DESIGN ****************************



Optimised population design:
Sample times for response: A
                prot.opti subjects.opti Subjects doses
```

```
1 c(0.0625, 3, 12, 20.58)             1       90   400


Associated optimised criterion: 597.7232


****************** POPULATION FISHER INFORMATION matrix ******************

            [,1]      [,2]       [,3]       [,4]       [,5]        [,6]
 [1,] 25788.2415 274.75431  1655.9057 -9338.5902   0.000000    0.000000
 [2,]   274.7543  66.91207   150.6082   -75.6858   0.000000    0.000000
 [3,]  1655.9057 150.60821  3444.4739 -1045.7213   0.000000    0.000000
 [4,] -9338.5902 -75.68580 -1045.7213  5252.9566   0.000000    0.000000
 [5,]     0.0000   0.00000     0.0000     0.0000 151.332045   25.462318
 [6,]     0.0000   0.00000     0.0000     0.0000  25.462318 2238.408302
 [7,]     0.0000   0.00000     0.0000     0.0000   2.985759   36.610219
 [8,]     0.0000   0.00000     0.0000     0.0000  41.724015    4.062316
 [9,]     0.0000   0.00000     0.0000     0.0000 163.083159  593.253623
            [,7]      [,8]      [,9]
 [1,]   0.000000  0.000000    0.0000
 [2,]   0.000000  0.000000    0.0000
 [3,]   0.000000  0.000000    0.0000
 [4,]   0.000000  0.000000    0.0000
 [5,]   2.985759 41.724015  163.0832
 [6,]  36.610219  4.062316  593.2536
 [7,]  61.819500  2.503522  228.3075
 [8,]   2.503522 27.756651   79.0022
 [9,] 228.307471 79.002193 4360.0953



************************** EXPECTED STANDARD ERRORS ************************

------------------------ Fixed Effects Parameters -------------------------

     Beta   StdError        CV .
k   0.080 0.01065649 13.320612 %
V   3.080 0.13160835  4.272998 %
k12 0.175 0.01847432 10.556755 %
k21 0.116 0.02368800 20.420687 %


------------------------- Variance of Random Effects ----------------------

    Omega  StdError       CV .
k    0.25 0.1066889 42.67557 %
V    0.10 0.0215495 21.54950 %
k12  0.30 0.1419385 47.31285 %
k21  0.50 0.2503984 50.07968 %


------------------------ Variance of residual error -----------------------

            SIG   StdError       CV .
sig.slopeA 0.25 0.01760084 7.040336 %


****************************** DETERMINANT *******************************

9.7387e+24

****************************** CRITERION *********************************

597.7232
```


Concentration model
Dose = 400
Concentration
0
20
40
60
80
100
120
0
5
10
15
20
Time
1
1
1
1

2.2 Example 2: Two responses with ODE

2.2.1 Simplex algorithm (a group with NULL)

This example deals with the optimisation of a design using the Simplex algorithm and with for the joint modelling for the time course of a drug concentration and metabolite concentration. This model has been used for design evaluation in the section 1.2.2.
The design to be evaluated is composed of two groups with the proportions of subjects equal to 0.5 and 0.5. The total number of sample is equal to 400.

The second group has only samples for the first response.
An option has been added for optimisation, the optimised design has to have the same sampling times for the two responses (identical.times=T) between 0 and 15 hrs.

INPUT FILE

```
##############################################################################
##                                                                          ##
##                          INPUT FILE FOR PFIM 3.0                           ##
##############################################################################


#Name of the project
#--------------------

project<-"Doc_example2.2.1"


#Name of the file containing the PK or PD model
#-----------------------------------------------

file.model<-"model.r";


#Name of the output file for the results
#----------------------------------------

output<-"Stdout.r";

#RUN:  Evaluation (EVAL) or Optimisation (OPT)
#----------------------------------------------------------
run<-"OPT"

#Number of responses
#-------------------------------------------------------------------------

nr<-2

#Name of the fixed effects parameters
#-------------------------------------

parameters<-c('Ka','CL','V','Clm','R')

#Fixed effects parameters values
#--------------------------------

beta<-c(2.86,183,256,0.158,0.0012)

#Diagonal Matrix of variance for the random effects:
#---------------------------------------------------

omega<-diag(c(0,0.0316,0,0.1640,0));
```

```
#Standard deviation of the residual error (sig.inter+sig.slope*f)^2:
#-------------------------------------------------------------------
#Drug model
sig.interA<-0.000239
sig.slopeA<-0.276

#Metabolite model
sig.interB<-0
sig.slopeB<-0.135

#Between-subject variance model (1) = additive  (2) = exponential
#-------------------------------------------------------------------

Trand<-2;

#List of the vectors of sampling times for each elementary design
#you can specify any sampling times for a group by writing NULL
#ONLY if you have several responses
#-------------------------------------------------------------------

protA<-list(c(1,3,6,12),c(1,3,6,12))
protB<-list(c(1,3,6,12),c(NULL))

#Vector of initial proportions or numbers of subjects for each elementary design
#-------------------------------------------------------------
subjects<-c(0.5,0.5)

#Subjects input: (1) for number of subjects (2) for proportions of subjects
#-----------------------------------------------------------------------------
subjects.input<-2

#If 'proportions of subjects' give the total number of samples
#-------------------------------------------------------------
Ntot<-400

################### MODEL OPTION ############################

#Model form: Differential equations (DE) or analytical form (AF)
#-------------------------------------------------------------

modelform<-"DE"

###### ANALYTICAL MODEL OPTION ############################
###########################################################

#Identical dose in each elementary design (Yes=T, No=F)
#-----------------------------------------------------------
#dose.identical<-T

# If 'Yes', enter the value of the dose,
# else, enter the vector of the dose values for each elementary design
#-------------------------------------------------------------------
#dose<-c(1)

#Vector of last times for the intervals of each expression
#-----------------------------------------------------------
#boundA<-list(c(0,1.5/24),c(1.5/24,Inf))

###### END ANALYTICAL MODEL OPTION ########################




###### DIFFERENTIAL EQUATION OPTION ###################################
#######################################################################
```

```
#Initial time for which initial conditions are given
#-------------------------------------------------

time.condinit<-0

#Identical initial conditions in each elementary design (Yes=T, No=F)
#-----------------------------------------------------------
condinit.identical<-T

# If 'Yes', enter once the expression of the initial values of the system at the
initial time
# else, enter the vectors of the initial conditions for each elementary design
# If initial values depend on parameters to be estimated,
# enter this parameter into the expression without any quotation marks
#-----------------------------------------------------

condinit<-expression(c(300,0,0))


# Error tolerance for solving differential equations
#-----------------------------------------------------

RtolEQ<-1e-08
AtolEQ<-1e-08
Hmax<-Inf # Default value
#Hmax<-Inf #<1.5/24
###### END DIFFERENTIAL EQUATION OPTION ##################################



############ONLY FOR OPTIMISATION ################################

#Identical sampling times for each response
# (only if you do not have sampling times==NULL)
#-------------------------------------------------------------------------------------------
identical.times<-T

######## OPTIMISATION ALGORITHM OPTION ###############

#Character string for thoice of the optimisation algorithm:
#      "FW" for the Fedorov-Wynn algorithm
#      "SIMP" for the Simplex algorithm
#-----------------------------------------

algo.option<-"SIMP"


########################
#SIMPLEX SPECIFICATION #
########################

#Optimisation of the proportions of subjects: (Yes=T, No=F)
#-------------------------------------------------------------

subjects.opt<-F

#Vector of lower and upper admissible sampling times
#-----------------------------------------------------
lowerA<-c(0)
upperA<-c(15)

lowerB<-c(0)
upperB<-c(15)

#Minimum delay between two sampling times
#-----------------------------------------

delta.time<-0
```

```
#Print iteration step (Yes=T, No=F)
#---------------------------------

iter.print<-T


#Parameter for initial simplex building (%)
#-----------------------------------------

simplex.parameter<-20


#Maximum iteration number
#------------------------

Max.iter<-5000


#Relative convergence tolerance
#------------------------------
Rctol<-1e-6



############################
#FEDOROV-WYNN SPECIFICATION #
############################
#Number of sampling windows
#--------------------------
nwindA<-1
#nwindB<-1

#List of vector of the allowed sampling times for each sampling window
#---------------------------------------------------------------------

sampwinA<-list(c(0.0625,1,3,7,12,14,15,20.58))
#sampwinB<-list(c(0.166,1,2,3,6,9,12,20,36))


#List of vector of allowed number of points to be taken from each sampling window
#--------------------------------------------------------------------------------

nsampA<-list(c(4,1))
#nsampB<-list(c(4))

#Maximum total number of sampling times per subject
#--------------------------------------------------

nmaxptsA<-4
#nmaxptsB<-4

#Minimum total number of sampling times per subject
#--------------------------------------------------

nminptsA<-1
#nminptsB<-4
############# END OF OPTIMISATION ALGORITHM OPTION ###############




############## GRAPH SPECIFICATION OPTION ###############

#graphical representation (Yes=T, No=F)
#--------------------------------------
```

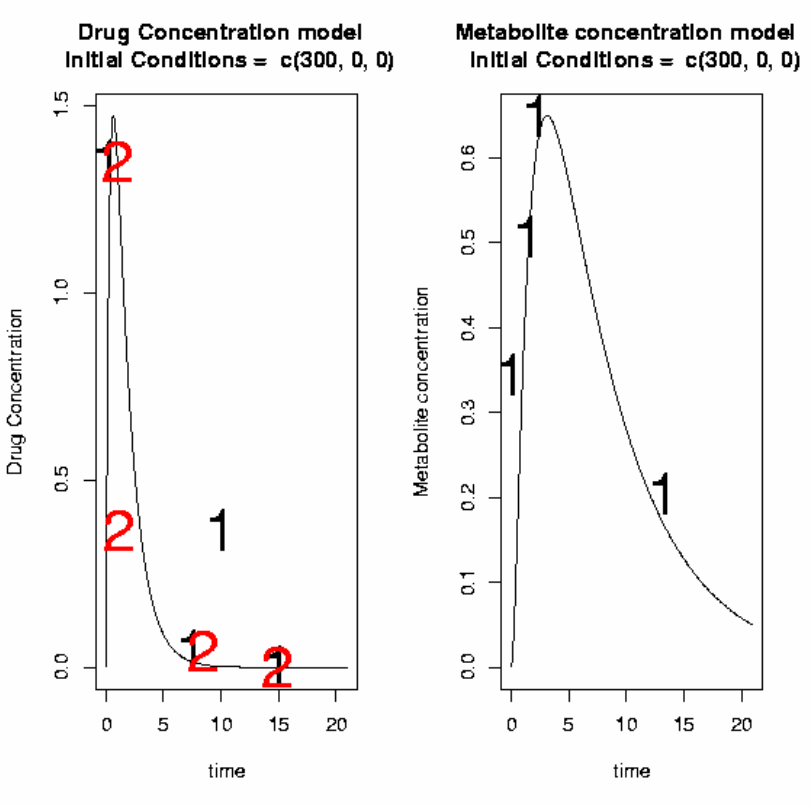
```
graph.logical<-T

#Names on Y axes for each response
#---------------------------------
names.data<-c("Drug Concentration","Metabolite concentration")

#Controls logarithmic axes for the graphical representation.
#Values "xy", "x", or "y" produce log-log or log-x or log-y axes.
#(For standard graphic, log.logical<-F)
#-----------------------------------------------------------------

#log.logical<-'y'
log.logical<-F

#Vector of lower and upper sampling times for the graphical representation
#------------------------------------------------------------------------

graph.infA<-c(0)
graph.supA<-c(21)
graph.infB<-c(0)
graph.supB<-c(21)

#Vector of lower and upper concentration for the graphical representation
#-----------------------------------------------------------------------
y.range<-NULL # default range
#y.range<-c(0,10)
```

OUTPUT FILE:

```
PFIM 3.0 Optimisation

Project: Doc_example2.2.1

Date: Tue Sep 11 14:20:28 2007

*************************** INPUT SUMMARY *******************************

Differential Equations form of the model:

function(t,y,p)
{
ka<-p[1]
cl<-p[2]
V<-p[3]
clm<-p[4]
R<-p[5]
Tinf<-0.01    #temps d'infusion
R1<-300/0.01


if (t<=Tinf & t%%12<=Tinf) yd1<--ka*y[1]+R1
else
yd1<--ka*y[1]
yd2<-ka*y[1]-cl/V*y[2]-R*y[2]
yd3<-R*y[2]-clm*y[3]

list(c(yd1,yd2,yd3),c(y[2]/V,y[3]))
}


Initial Population design:


Sample times for response: A
[[1]]
[1]  1  3  6 12

[[2]]
[1]  1  3  6 12

               Number of subjects per group
 0.5 0.5




Sample times for response: B
[[1]]
[1]  1  3  6 12

[[2]]
NULL

               Number of subjects per group
 0.5 0.5


Variance error model response A : ( 0.000239 + 0.276 *f)^2

Variance error model response B : ( 0 + 0.135 *f)^2



Initial Conditions at time 0 :
```

```
300 0 0
300 0 0

Between-subject variance model: Trand =  2

Total number of samples: 400

Associated criterion value: 3028.316

Window of the allowed optimised sampling times for the  2 responses:
Upper and lower admissible samples times for the response 1 :  [  0 : 15 ]
Upper and lower admissible samples times for the response 2 :  [  0 : 15 ]


Minimum delay between two sampling times: 0

Optimisation of the proportions of subjects: FALSE

Error tolerance for solving differential equations system: RtolEQ = 1e-08 , AtolEQ =
1e-08 , Hmax =  Inf

*************************** OPTIMISED DESIGN ****************************

Number of iterations: 569
Number of function evaluations: 1305
Convergence Achieved



       Optimised population design :


Sample times for response: A
[[1]]
[1]  0.101  9.937  7.479 14.895

[[2]]
[1]  0.995  1.077  8.398 14.825

      Number of subjects per group
     Proportions                 Numbers
      0.5 0.5                     25 50


Sample times for response: B
[[1]]
[1]  0.003  2.254  1.501 13.215

[[2]]
NULL

      Number of subjects per group
     Proportions                 Numbers
       0.5 0.5                    25 50


Associated optimised criterion: 4499.323



****************** POPULATION FISHER INFORMATION matrix ******************

               [,1]          [,2]         [,3]          [,4]          [,5]         [,6]
 [1,]  2.790166e+02    0.19272329  -0.62904761 -6.560117e+00  6.984355e+05 0.000000e+00
 [2,]  1.927233e-01    0.05897688  -0.03428538  2.391391e-01 -2.635866e+02 0.000000e+00
 [3,] -6.290476e-01   -0.03428538   0.03799973 -7.564594e-01  8.670542e+02 0.000000e+00
 [4,] -6.560117e+00    0.23913906  -0.75645942  7.820456e+03 -6.341369e+05 0.000000e+00
```

```
 [5,]  6.984355e+05 -263.58657989 867.05419262 -6.341369e+05  3.400003e+09 0.000000e+00
 [6,]  0.000000e+00    0.00000000   0.00000000  0.000000e+00  0.000000e+00 2.926750e+04
 [7,]  0.000000e+00    0.00000000   0.00000000  0.000000e+00  0.000000e+00 7.171478e-01
 [8,]  0.000000e+00    0.00000000   0.00000000  0.000000e+00  0.000000e+00 7.215792e+04
 [9,]  0.000000e+00    0.00000000   0.00000000  0.000000e+00  0.000000e+00 3.869330e+02
[10,]  0.000000e+00    0.00000000   0.00000000  0.000000e+00  0.000000e+00 8.123205e+00
            [,7]         [,8]         [,9]        [,10]
 [1,]   0.0000000 0.000000e+00 0.000000e+00     0.000000
 [2,]   0.0000000 0.000000e+00 0.000000e+00     0.000000
 [3,]   0.0000000 0.000000e+00 0.000000e+00     0.000000
 [4,]   0.0000000 0.000000e+00 0.000000e+00     0.000000
 [5,]   0.0000000 0.000000e+00 0.000000e+00     0.000000
 [6,]   0.7171478 7.215792e+04 3.869330e+02     8.123205
 [7,] 571.7205457 6.527614e+02 2.738706e+00    49.987532
 [8,] 652.7614278 1.943103e+09 3.023784e+05  7393.894114
 [9,]   2.7387058 3.023784e+05 3.280815e+03    31.021595
[10,]  49.9875321 7.393894e+03 3.102159e+01 10816.256150



************************* EXPECTED STANDARD ERRORS ************************

------------------------ Fixed Effects Parameters ------------------------

        Beta     StdError       CV .
Ka    2.8600 9.601219e-02 3.357070 %
CL  183.0000 6.204161e+00 3.390252 %
V   256.0000 8.264770e+00 3.228426 %
Clm   0.1580 1.147967e-02 7.265615 %
R     0.0012 2.703675e-05 2.253062 %


------------------------- Variance of Random Effects ----------------------

     Omega    StdError       CV .
CL  0.0316 0.005849938 18.51246 %
Clm 0.1640 0.041830848 25.50661 %


------------------------ Variance of residual error -----------------------

                SIG     StdError       CV .
sig.interA 0.000239 2.285044e-05 9.560852 %
sig.slopeA 0.276000 1.759853e-02 6.376281 %
sig.slopeB 0.135000 9.617347e-03 7.123961 %


****************************** DETERMINANT ********************************

3.399943e+36

******************************** CRITERION ********************************

4499.323
```


Drug Concentration model
Initial Conditions = c(300, 0, 0)
Drug Concentration
time
Metabolite concentration model
Initial Conditions = c(300, 0, 0)
Metabolite concentration
time

2.2.2 Federov-Wynn algorithm

This example deal with the optimisation of a design using the previous model described in the section 2.2.1 (defined by a differential equation system). The design is composed of two groups with the sampling at (1, 3, 6, 12) and (1, 3, 6, 15). For each group the initial conditions are different.

INPUT FILE

```
##############################################################################
##                                                                          ##
##                          INPUT FILE FOR PFIM 3.0                           ##
##############################################################################


#Name of the project
#--------------------

project<-"Doc_example2.2.2"


#Name of the file containing the PK or PD model
#-----------------------------------------------

file.model<-"model.r";


#Name of the output file for the results
#----------------------------------------

output<-"Stdout.r";

#RUN:  Evaluation (EVAL) or Optimisation (OPT)
#----------------------------------------------------------
run<-"OPT"

#Number of responses
#-------------------------------------------------------------------------

nr<-2

#Name of the fixed effects parameters
#-------------------------------------

parameters<-c('Ka','CL','V','Clm','R')

#Fixed effects parameters values
#--------------------------------

beta<-c(2.86,183,256,0.158,0.0012)

#Diagonal Matrix of variance for the random effects:
#----------------------------------------------------

omega<-diag(c(0,0.0316,0,0.1640,0));


#Standard deviation of the residual error (sig.inter+sig.slope*f)^2:
#--------------------------------------------------------------------
#Drug model
sig.interA<-0.000239
sig.slopeA<-0.276

#Metabolite model
sig.interB<-0
sig.slopeB<-0.135
```

```
#Between-subject variance model (1) = additive  (2) = exponential
#-------------------------------------------------------------------

Trand<-2;

#List of the vectors of sampling times for each elementary design
#you can specify any sampling times for a group by writing NULL
#ONLY if you have several responses
#-------------------------------------------------------------------

protA<-list(c(1,3,6,12),c(1,3,6,15))
protB<-list(c(1,3,6,12),c(1,3,6,15))

#Vector of initial proportions or numbers of subjects for each elementary design
#-------------------------------------------------------------
subjects<-c(0.5,0.5)

#Subjects input: (1) for number of subjects (2) for proportions of subjects
#-------------------------------------------------------------------------
subjects.input<-2

#If 'proportions of subjects' give the total number of samples
#-------------------------------------------------------------
Ntot<-400

################### MODEL OPTION ############################

#Model form: Differential equations (DE) or analytical form (AF)
#-------------------------------------------------------------

modelform<-"DE"

###### ANALYTICAL MODEL OPTION ############################
############################################################

#Identical dose in each elementary design (Yes=T, No=F)
#-------------------------------------------------------------
#dose.identical<-T

# If 'Yes', enter the value of the dose,
# else, enter the vector of the dose values for each elementary design
#-------------------------------------------------------------------
#dose<-c(1)

#Vector of the times intervals of each expression
#-------------------------------------------------------------
#boundA<-list(c(0,1.5/24),c(1.5/24,Inf))

###### END ANALYTICAL MODEL OPTION ########################




###### DIFFERENTIAL EQUATION OPTION ###################################
#######################################################################

#Initial time for which initial conditions are given
#--------------------------------------------------

time.condinit<-0

#Identical initial conditions in each elementary design (Yes=T, No=F)
#-------------------------------------------------------------
condinit.identical<-F

# If 'Yes', enter once the expression of the initial values of the system at the
initial time
```

```
# else, enter the vectors of the initial conditions for each elementary design
# If initial values depend on parameters to be estimated,
# enter this parameter into the expression without any quotation marks
#-------------------------------------------------------

condinit<-c(expression(c(300,0,0)),expression(c(200,0,0)))

# Error tolerance for solving differential equations
#-------------------------------------------------

RtolEQ<-1e-08
AtolEQ<-1e-08
Hmax<-Inf # Default value
#Hmax<-Inf #<1.5/24
###### END DIFFERENTIAL EQUATION OPTION ##################################



############ONLY FOR OPTIMISATION ################################

#Identical sampling times for each response
# (only if you do not have sampling times==NULL)
#-----------------------------------------------------------------------------------
identical.times<-T

######## OPTIMISATION ALGORITHM OPTION ###############

#Character string for thoice of the optimisation algorithm:
#      "FW" for the Fedorov-Wynn algorithm
#      "SIMP" for the Simplex algorithm
#-----------------------------------------

algo.option<-"FW"


########################
#SIMPLEX SPECIFICATION #
########################

#Optimisation of the proportions of subjects: (Yes=T, No=F)
#---------------------------------------------------------------

subjects.opt<-F

#Vector of lower and upper admissible sampling times
#--------------------------------------------------
lowerA<-c(0)
upperA<-c(15)

lowerB<-c(0)
upperB<-c(15)

#Minimum delay between two sampling times
#---------------------------------------

delta.time<-0

#Print iteration step (Yes=T, No=F)
#---------------------------------

iter.print<-T


#Parameter for initial simplex building (%)
#-----------------------------------------

simplex.parameter<-20
```

```
#Maximum iteration number
#------------------------

Max.iter<-5000


#Relative convergence tolerance
#-----------------------------
Rctol<-1e-6



##############################
#FEDOROV-WYNN SPECIFICATION #
##############################


#Number of sampling windows
#-------------------------
nwindA<-1
nwindB<-1

#List of vector of the allowed sampling times for each sampling window
#---------------------------------------------------------------------

sampwinA<-list(c(0.0625,1,3,6,11,12,14,15))
sampwinB<-list(c(0.0625,1,3,6,11,12,14,15))


#List of vector of allowed number of points to be taken from each sampling window
#--------------------------------------------------------------------------------

nsampA<-list(c(4))
nsampB<-list(c(4))

#Maximum total number of sampling times per subject
#-------------------------------------------------

nmaxptsA<-4
nmaxptsB<-4

#Minimum total number of sampling times per subject
#-------------------------------------------------

nminptsA<-4
nminptsB<-4
############# END OF OPTIMISATION ALGORITHM OPTION ###############




############## GRAPH SPECIFICATION OPTION ###############

#graphical representation (Yes=T, No=F)
#-------------------------------------

graph.logical<-T

#Names on Y axes for each response
#--------------------------------
names.data<-c("Drug Concentration","Metabolite concentration")

#Controls logarithmic axes for the graphical representation.
#Values "xy", "x", or "y" produce log-log or log-x or log-y axes.
#(For standard graphic, log.logical<-F)
```

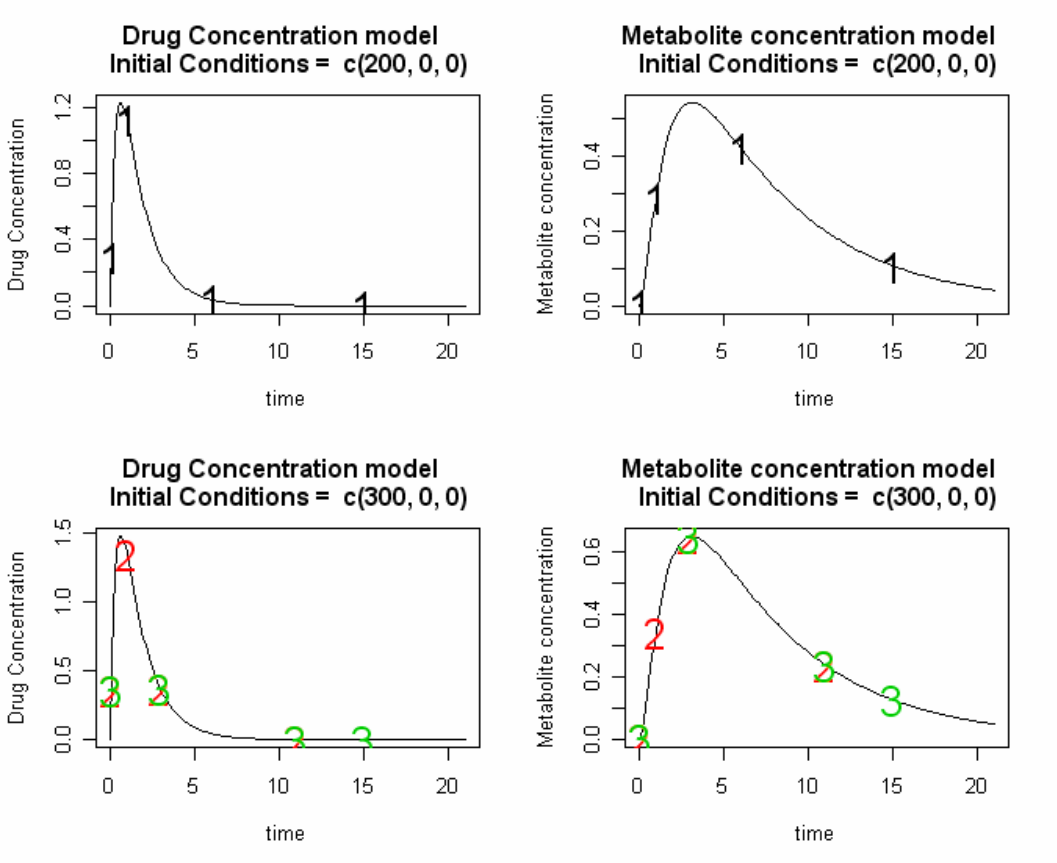
```
#------------------------------------------------------------

#log.logical<-'y'
log.logical<-F

#Vector of lower and upper sampling times for the graphical representation
#------------------------------------------------------------------------

graph.infA<-c(0)
graph.supA<-c(21)
graph.infB<-c(0)
graph.supB<-c(21)

#Vector of lower and upper concentration for the graphical representation
#-----------------------------------------------------------------------
y.range<-NULL # default range
#y.range<-c(0,10)

############## END OF GRAPH SPECIFICATION OPTION ###############
```

OUTPUT FILE:

```
PFIM 3.0

Project: Doc_example2.2.2

Date: Thu Feb 14 09:31:33 2008

*************************** INPUT SUMMARY *******************************

Differential Equations form of the model:

function(t,y,p)
{
ka<-p[1]
cl<-p[2]
V<-p[3]
clm<-p[4]
R<-p[5]
Tinf<-0.01    #temps d'infusion
R1<-300/0.01


if (t<=Tinf & t%%12<=Tinf) yd1<--ka*y[1]+R1
else
yd1<--ka*y[1]
yd2<-ka*y[1]-cl/V*y[2]-R*y[2]
yd3<-R*y[2]-clm*y[3]

list(c(yd1,yd2,yd3),c(y[2]/V,y[3]))
}


Initial Population design:


Sample times for response: A
         Protocol subjects     condinit
1 c=(1, 3, 6, 12)       25 c(300, 0, 0)
2 c=(1, 3, 6, 15)       25 c(200, 0, 0)


Sample times for response: B
         Protocol subjects     condinit
1 c=(1, 3, 6, 12)       25 c(300, 0, 0)
2 c=(1, 3, 6, 15)       25 c(200, 0, 0)


Initial Conditions at time 0 :

300 0 0
200 0 0

Error tolerance for solving differential equations system: RtolEQ = 1e-08 , AtolEQ =
1e-08 , Hmax =  Inf

Variance error model response A : ( 0.000239 + 0.276 *f)^2

Variance error model response B : ( 0 + 0.135 *f)^2



Between-subject variance model: Trand =  2

Total number of samples: 400

Associated criterion value: 3516.596
```

```
Sampling windows for the response: A
Window 1 : t= 0.0625 1 3 6 11 12 14 15
    Nb of sampling points to be taken in this window, n[ 1 ]= 4
Maximum total number of points in one elementary protocol : 4
Minimum total number of points in one elementary protocol : 4


Sampling windows for the response: B
Window 1 : t= 0.0625 1 3 6 11 12 14 15
    Nb of sampling points to be taken in this window, n[ 1 ]= 4
Maximum total number of points in one elementary protocol : 4
Minimum total number of points in one elementary protocol : 4



Now evaluating the Fisher Information Matrix for the 140 protocols generated


*************************** OPTIMISED DESIGN ****************************



Optimised population design:
Sample times for response: A
                 times      freq Subjects     condinit
1  c(0.0625, 1, 6, 15) 0.5346307 26.73153 c(200, 0, 0)
2  c(0.0625, 1, 3, 11) 0.2144796 10.72398 c(300, 0, 0)
3 c(0.0625, 3, 11, 15) 0.2508898 12.54449 c(300, 0, 0)


Sample times for response: B
                 times      freq Subjects     condinit
1  c(0.0625, 1, 6, 15) 0.5346307 26.73153 c(200, 0, 0)
2  c(0.0625, 1, 3, 11) 0.2144796 10.72398 c(300, 0, 0)
3 c(0.0625, 3, 11, 15) 0.2508898 12.54449 c(300, 0, 0)

Associated optimised criterion: 4564.192



****************** POPULATION FISHER INFORMATION matrix ******************

               [,1]          [,2]          [,3]          [,4]          [,5]
 [1,]  4.062368e+02    0.13803387   -0.78846896 -4.858967e+00  9.869045e+05
 [2,]  1.380339e-01    0.04224601   -0.02362077  8.425567e-01 -8.636369e+02
 [3,] -7.884690e-01   -0.02362077    0.03234629 -1.711913e+00  1.759011e+03
 [4,] -4.858967e+00    0.84255673   -1.71191347  1.172308e+04 -8.761633e+05
 [5,]  9.869045e+05 -863.63690155 1759.01126606 -8.761633e+05  4.197394e+09
 [6,]  0.000000e+00    0.00000000    0.00000000  0.000000e+00  0.000000e+00
 [7,]  0.000000e+00    0.00000000    0.00000000  0.000000e+00  0.000000e+00
 [8,]  0.000000e+00    0.00000000    0.00000000  0.000000e+00  0.000000e+00
 [9,]  0.000000e+00    0.00000000    0.00000000  0.000000e+00  0.000000e+00
[10,]  0.000000e+00    0.00000000    0.00000000  0.000000e+00  0.000000e+00
               [,6]        [,7]         [,8]         [,9]       [,10]
 [1,] 0.000000e+00    0.000000 0.000000e+00 0.000000e+00     0.00000
 [2,] 0.000000e+00    0.000000 0.000000e+00 0.000000e+00     0.00000
 [3,] 0.000000e+00    0.000000 0.000000e+00 0.000000e+00     0.00000
 [4,] 0.000000e+00    0.000000 0.000000e+00 0.000000e+00     0.00000
 [5,] 0.000000e+00    0.000000 0.000000e+00 0.000000e+00     0.00000
 [6,] 2.002130e+04    6.006498 1.014628e+05 3.979513e+02    99.08100
 [7,] 6.006498e+00  856.660003 2.439244e+03 8.347964e+00    62.24991
 [8,] 1.014628e+05 2439.243923 1.172122e+09 1.333233e+05 28872.96343
 [9,] 3.979513e+02    8.347964 1.333233e+05 2.879889e+03   138.17546
[10,] 9.908100e+01   62.249910 2.887296e+04 1.381755e+02 15664.86750
```

```
************************* EXPECTED STANDARD ERRORS ************************

------------------------ Fixed Effects Parameters ------------------------

        Beta     StdError       CV .
Ka    2.8600 9.330398e-02 3.262377 %
CL  183.0000 6.644973e+00 3.631133 %
V   256.0000 8.854297e+00 3.458710 %
Clm   0.1580 9.405684e-03 5.952964 %
R     0.0012 2.857172e-05 2.380976 %


------------------------- Variance of Random Effects ----------------------

     Omega    StdError       CV .
CL  0.0316 0.007078152 22.39922 %
Clm 0.1640 0.034171568 20.83632 %


------------------------ Variance of residual error -----------------------

                SIG     StdError        CV .
sig.interA 0.000239 2.929079e-05 12.255562 %
sig.slopeA 0.276000 1.871180e-02  6.779639 %
sig.slopeB 0.135000 7.992832e-03  5.920616 %


******************************* DETERMINANT *******************************

3.923186e+36

******************************** CRITERION ********************************

4564.192
```


Drug Concentration model
Initial Conditions = c(200, 0, 0)
Drug Concentration
time
Metabolite concentration model
Initial Conditions = c(200, 0, 0)
Metabolite concentration
time
Drug Concentration model
Initial Conditions = c(300, 0, 0)
Drug Concentration
time
Metabolite concentration model
Initial Conditions = c(300, 0, 0)
Metabolite concentration
time

2.3 Examples 3: Two responses with ODE and with Tlag using the Federov-Wynn algorithm

This example deals with the optimisation of a design using the Federov-Wynn algorithm. The model used is described in the section 1.2.3. It is a joint modelling of a drug concentration and its effect.
The optimal population design has to be composed of groups with 4, 3, 2 or 1 sampling times for a total of 32 subjects for each response (identical.times=T).

INPUT FILE

```
##############################################################################
##                                                                          ##
##                          INPUT FILE FOR PFIM 3.0                           ##
##############################################################################


#Name of the project
#--------------------

project<-"Doc_example2.3"


#Name of the file containing the PK or PD model
#-----------------------------------------------

file.model<-"model.r";


#Name of the output file for the results
#----------------------------------------

output<-"Stdout.r";

#RUN:  Evaluation (EVAL) or Optimisation (OPT)
#----------------------------------------------------------
run<-"OPT"

#Number of responses
#-------------------------------------------------------------------------

nr<-2

#Name of the fixed effects parameters
#-------------------------------------

parameters<-c('ka','cl','V','kin','c50','kout')

#Fixed effects parameters values
#--------------------------------

beta<-c(1.6,0.133,7.95,5.41,1.2,0.056)

#Diagonal Matrix of variance for the random effects:
#----------------------------------------------------

omega<-diag(c(0.171,0.0634,0.0206,0.19,0.0167,0.0129));


#Standard deviation of the residual error (sig.inter+sig.slope*f)^2:
#-------------------------------------------------------------------
#Drug model
sig.interA<-0.248
sig.slopeA<-0.0625

#Metabolite model
```

```
sig.interB<-3.84
sig.slopeB<-0

#Between-subject variance model (1) = additive  (2) = exponential
#-------------------------------------------------------------------

Trand<-2;

#List of the vectors of sampling times for each elementary design
#you can specify any sampling times for a group by writing NULL
#ONLY if you have several responses
#-------------------------------------------------------------

protA<-list(c(0.5,1,12,24,36))
protB<-list(c(0.5,1,2,96,120))

#Vector of initial proportions or numbers of subjects for each elementary design
#-----------------------------------------------------------
subjects<-c(32)

#Subjects input: (1) for number of subjects (2) for proportions of subjects
#-------------------------------------------------------------------------
subjects.input<-1

#If 'proportions of subjects' give the total number of samples
#-----------------------------------------------------------
Ntot<-250

################### MODEL OPTION ############################

#Model form: Differential equations (DE) or analytical form (AF)
#-----------------------------------------------------------

modelform<-"DE"

###### ANALYTICAL MODEL OPTION ############################
#############################################################

#Identical dose in each elementary design (Yes=T, No=F)
#-----------------------------------------------------------
#dose.identical<-T

# If 'Yes', enter the value of the dose,
# else, enter the vector of the dose values for each elementary design
#-------------------------------------------------------------------
#dose<-c(1)

#Vector of last times for the intervals of each expression
#-----------------------------------------------------------
#boundA<-list(c(0,1.5/24),c(1.5/24,Inf))

###### END ANALYTICAL MODEL OPTION ########################




###### DIFFERENTIAL EQUATION OPTION ###################################
#######################################################################

#Initial time for which initial conditions are given
#--------------------------------------------------

time.condinit<-0

#Identical initial conditions in each elementary design (Yes=T, No=F)
#-----------------------------------------------------------
condinit.identical<-T
```

```
# If 'Yes', enter once the expression of the initial values of the system at the
initial time
# else, enter the vectors of the initial conditions for each elementary design
# If initial values depend on parameters to be estimated,
# enter this parameter into the expression without any quotation marks
#-----------------------------------------------------------

condinit<-expression(c(100,0,kin/kout))


# Error tolerance for solving differential equations
#-----------------------------------------------------

RtolEQ<-1e-08
AtolEQ<-1e-08
Hmax<-Inf # Default value
#Hmax<-Inf #<1.5/24
###### END DIFFERENTIAL EQUATION OPTION ###################################



############ONLY FOR OPTIMISATION ################################

#Identical sampling times for each response
# (only if you do not have sampling times==NULL)
#-------------------------------------------------------------------------------------
identical.times<-T

######## OPTIMISATION ALGORITHM OPTION ###############

#Character string for thoice of the optimisation algorithm:
#      "FW" for the Fedorov-Wynn algorithm
#      "SIMP" for the Simplex algorithm
#-----------------------------------------

algo.option<-"FW"


########################
#SIMPLEX SPECIFICATION #
########################

#Optimisation of the proportions of subjects: (Yes=T, No=F)
#-------------------------------------------------------------

subjects.opt<-F

#Vector of lower and upper admissible sampling times
#-----------------------------------------------------
lowerA<-c(0)
upperA<-c(15)

lowerB<-c(0)
upperB<-c(15)

#Minimum delay between two sampling times
#-----------------------------------------

delta.time<-0

#Print iteration step (Yes=T, No=F)
#---------------------------------

iter.print<-T


#Parameter for initial simplex building (%)
#-----------------------------------------
```

```
simplex.parameter<-20


#Maximum iteration number
#------------------------

Max.iter<-5000


#Relative convergence tolerance
#------------------------------
Rctol<-1e-6



##############################
#FEDOROV-WYNN SPECIFICATION #
##############################


#Number of sampling windows
#--------------------------
nwindA<-1
nwindB<-1

#List of vector of the allowed sampling times for each sampling window
#---------------------------------------------------------------------

sampwinA<-list(c(0.5,1,2,3,6,9,12,24,36,48,72,96,120,144))
sampwinB<-list(c(0.5,1,2,3,6,9,12,24,36,48,72,96,120,144))


#List of vector of allowed number of points to be taken from each sampling window
#--------------------------------------------------------------------------------

nsampA<-list(c(5,4,1,2,3))
nsampB<-list(c(5,4,1,2,3))

#Maximum total number of sampling times per subject
#--------------------------------------------------

nmaxptsA<-5
nmaxptsB<-5

#Minimum total number of sampling times per subject
#--------------------------------------------------

nminptsA<-1
nminptsB<-1
############# END OF OPTIMISATION ALGORITHM OPTION ###############




############## GRAPH SPECIFICATION OPTION ###############

#graphical representation (Yes=T, No=F)
#--------------------------------------

graph.logical<-F

#Names on Y axes for each response
#---------------------------------
names.data<-c("Drug Concentration","Effect")
```

```
#Controls logarithmic axes for the graphical representation.
#Values "xy", "x", or "y" produce log-log or log-x or log-y axes.
#(For standard graphic, log.logical<-F)
#-------------------------------------------------------------

#log.logical<-'y'
log.logical<-F

#Vector of lower and upper sampling times for the graphical representation
#-------------------------------------------------------------------------

graph.infA<-c(0)
graph.supA<-c(144)
graph.infB<-c(0)
graph.supB<-c(144)

#Vector of lower and upper concentration for the graphical representation
#-------------------------------------------------------------------------
y.range<-NULL # default range
#y.range<-c(0,10)

############## END OF GRAPH SPECIFICATION OPTION ################
```

OUTPUT FILE

```
PFIM 3.0

Project: Doc_example2.3

Date: Wed Apr 16 14:20:58 2008



*************************** INPUT SUMMARY *********************************

Differential Equations form of the model:

function(t,y,p)
{
ka<-p[1]
cl<-p[2]
V<-p[3]
kin<-p[4]
C50<-p[5]
kout<-p[6]
tlag<-1.07    #tlag



if (t<=tlag) {
yd1<-0
yd2<-0
}
else {
yd1<--ka*y[1]
yd2<-ka*y[1]-(cl/V)*y[2]
}
yd3<-kin*(1-((y[2]/V)/((y[2]/V)+C50)))-kout*y[3]

list(c(yd1,yd2,yd3),c(y[2]/V,y[3]))
}


Initial Population design:


Sample times for response: A
                Protocol subjects            condinit
1 c=(0.5, 1, 12, 24, 36)       32 c(100, 0, kin/kout)


Sample times for response: B
                Protocol subjects            condinit
1 c=(0.5, 1, 12, 24, 36)       32 c(100, 0, kin/kout)




Initial Conditions at time 0 :

100 0 kin/kout

Error tolerance for solving differential equations system: RtolEQ = 1e-08 , AtolEQ
= 1e-08 , Hmax =  Inf

Variance error model response A : ( 0.248 + 0.0625 *f)^2

Variance error model response B : ( 3.84 + 0 *f)^2
```

```
Between-subject variance model: Trand =  2

Total number of samples: 320

Associated criterion value: 163.2488


Sampling windows for the response: A
Window 1 : t= 0.5 1 2 3 6 9 12 24 36 48 72 96 120 144
    Nb of sampling points to be taken in this window, n[ 1 ]= 5 4 1 2 3
Maximum total number of points in one elementary protocol : 5
Minimum total number of points in one elementary protocol : 1


Sampling windows for the response: B
Window 1 : t= 0.5 1 2 3 6 9 12 24 36 48 72 96 120 144
    Nb of sampling points to be taken in this window, n[ 1 ]= 5 4 1 2 3
Maximum total number of points in one elementary protocol : 5
Minimum total number of points in one elementary protocol : 1



Now evaluating the Fisher Information Matrix for the 3472 protocols generated


*************************** OPTIMISED DESIGN ****************************



Optimised population design:
Sample times for response: A
                   times        freq    Subjects            condinit
1                    0.5 0.215376423 10.80873594 c(100, 0, kin/kout)
2  c(1, 12, 24, 96, 120) 0.062161472  3.11959371 c(100, 0, kin/kout)
3      c(2, 12, 24, 120) 0.348010109 17.46500067 c(100, 0, kin/kout)
4          c(2, 24, 120) 0.298421888 14.97639965 c(100, 0, kin/kout)
5 c(0.5, 2, 24, 96, 120) 0.001470490  0.07379701 c(100, 0, kin/kout)
6  c(2, 12, 24, 96, 120) 0.040080847  2.01147034 c(100, 0, kin/kout)
7 c(0.5, 2, 12, 24, 120) 0.025017656  1.25551923 c(100, 0, kin/kout)
8   c(1, 2, 12, 24, 120) 0.008096748  0.40633792 c(100, 0, kin/kout)
9                     1 0.001364369  0.06847129 c(100, 0, kin/kout)


Sample times for response: B
                   times        freq    Subjects            condinit
1                    0.5 0.215376423 10.80873594 c(100, 0, kin/kout)
2  c(1, 12, 24, 96, 120) 0.062161472  3.11959371 c(100, 0, kin/kout)
3      c(2, 12, 24, 120) 0.348010109 17.46500067 c(100, 0, kin/kout)
4          c(2, 24, 120) 0.298421888 14.97639965 c(100, 0, kin/kout)
5 c(0.5, 2, 24, 96, 120) 0.001470490  0.07379701 c(100, 0, kin/kout)
6  c(2, 12, 24, 96, 120) 0.040080847  2.01147034 c(100, 0, kin/kout)
7 c(0.5, 2, 12, 24, 120) 0.025017656  1.25551923 c(100, 0, kin/kout)
8   c(1, 2, 12, 24, 120) 0.008096748  0.40633792 c(100, 0, kin/kout)
9                     1 0.001364369  0.06847129 c(100, 0, kin/kout)



Associated optimised criterion: 1080.091



****************** POPULATION FISHER INFORMATION matrix *****************

              [,1]         [,2]         [,3]         [,4]         [,5]
 [1,]   58.4257441   -19.003370   -7.3994590   -0.1482382   -6.1613660
 [2,]  -19.0033697 32089.618184  -24.5612501    4.6505213  628.5392791
 [3,]   -7.3994590   -24.561250   22.7372679   -0.1397231   -6.5121073
```

```
 [4,]   -0.1482382     4.650521   -0.1397231    8.5141001    0.8404392
 [5,]   -6.1613660   628.539279   -6.5121073    0.8404392  392.7788144
 [6,] -159.1752694  4113.170152 -181.8136826 -575.6189311 -976.5856854
 [7,]    0.0000000     0.000000    0.0000000    0.0000000    0.0000000
 [8,]    0.0000000     0.000000    0.0000000    0.0000000    0.0000000
 [9,]    0.0000000     0.000000    0.0000000    0.0000000    0.0000000
[10,]    0.0000000     0.000000    0.0000000    0.0000000    0.0000000
[11,]    0.0000000     0.000000    0.0000000    0.0000000    0.0000000
[12,]    0.0000000     0.000000    0.0000000    0.0000000    0.0000000
[13,]    0.0000000     0.000000    0.0000000    0.0000000    0.0000000
[14,]    0.0000000     0.000000    0.0000000    0.0000000    0.0000000
[15,]    0.0000000     0.000000    0.0000000    0.0000000    0.0000000
             [,6]         [,7]         [,8]         [,9]        [,10]
 [1,]   -159.1753   0.00000000    0.0000000 0.000000e+00   0.00000000
 [2,]   4113.1702   0.00000000    0.0000000 0.000000e+00   0.00000000
 [3,]   -181.8137   0.00000000    0.0000000 0.000000e+00   0.00000000
 [4,]   -575.6189   0.00000000    0.0000000 0.000000e+00   0.00000000
 [5,]   -976.5857   0.00000000    0.0000000 0.000000e+00   0.00000000
 [6,] 497131.2620   0.00000000    0.0000000 0.000000e+00   0.00000000
 [7,]      0.0000 309.44429713    0.4517802 1.354932e+02   0.02657866
 [8,]      0.0000   0.45178016 4099.1038725 1.124380e+01   0.14512924
 [9,]      0.0000 135.49319466   11.2438009 2.646292e+04   0.47175914
[10,]      0.0000   0.02657866    0.1451292 4.717591e-01 618.71670108
[11,]      0.0000   1.92567743  128.8163194 4.989366e+01   0.48202482
[12,]      0.0000   3.72139625   12.2735484 8.553246e+01 317.77148975
[13,]      0.0000  46.35947493   78.3656990 3.838059e+02   0.53775131
[14,]      0.0000 418.41148471  222.9089053 3.088959e+03   1.03419442
[15,]      0.0000   0.07345128    2.3980104 1.749796e+00   1.46345188
            [,11]        [,12]        [,13]       [,14]        [,15]
 [1,]    0.0000000     0.000000    0.0000000    0.000000   0.00000000
 [2,]    0.0000000     0.000000    0.0000000    0.000000   0.00000000
 [3,]    0.0000000     0.000000    0.0000000    0.000000   0.00000000
 [4,]    0.0000000     0.000000    0.0000000    0.000000   0.00000000
 [5,]    0.0000000     0.000000    0.0000000    0.000000   0.00000000
 [6,]    0.0000000     0.000000    0.0000000    0.000000   0.00000000
 [7,]    1.9256774     3.721396   46.3594749  418.411485   0.07345128
 [8,]  128.8163194    12.273548   78.3656990  222.908905   2.39801040
 [9,]   49.8936584    85.532459  383.8059238 3088.958625   1.74979557
[10,]    0.4820248   317.771490    0.5377513    1.034194   1.46345188
[11,] 4182.2651431    63.975839  359.4615753  712.352901  70.66756621
[12,]   63.9758387 29244.083319   57.1956241  127.150040 170.08645724
[13,]  359.4615753    57.195624  787.2819212 1198.095153   6.75577145
[14,]  712.3529011   127.150040 1198.0951527 8449.161485  13.47260211
[15,]   70.6675662   170.086457    6.7557714   13.472602   6.93591789
```

```
************************* EXPECTED STANDARD ERRORS ************************

------------------------ Fixed Effects Parameters ------------------------

      Beta    StdError       CV .
ka   1.600 0.133995845 8.374740 %
cl   0.133 0.005679761 4.270497 %
V    7.950 0.215460180 2.710191 %
kin  5.410 0.357255256 6.603609 %
c50  1.200 0.051630359 4.302530 %
kout 0.056 0.001487427 2.656119 %


------------------------- Variance of Random Effects ----------------------

      Omega    StdError        CV .
ka   0.1710 0.058902902  34.44614 %
cl   0.0634 0.015641801  24.67161 %
V    0.0206 0.006284753  30.50851 %
kin  0.1900 0.040316248  21.21908 %
```

```
c50  0.0167 0.017563677 105.17172 %
kout 0.0129 0.006435573  49.88817 %


------------------------ Variance of residual error -----------------------

              SIG   StdError       CV .
sig.interA 0.2480 0.04081567 16.45793 %
sig.slopeA 0.0625 0.01290621 20.64994 %
sig.interB 3.8400 0.45812447 11.93032 %


****************************** DETERMINANT *********************************

3.176164e+45

****************************** CRITERION ***********************************

1080.091

Time difference of 3.074217 mins
sys.self
    7.44
```