PFIM 3.2

Caroline Bazzoli, Thu Thuy Nguyen, Anne Dubois, Sylvie Retout, Emanuelle Comets,France Mentré

INSERM, UMR738, Paris, France ; Université Paris 7, Paris, France

Janvier 2010

www.pfim.biostat.fr

Examples documentation

****

PFIM 3.2 is free library of functions.
The Université Paris Diderot and INSERM are the co-owners of this library of functions.

Disclaimer

We inform users that the PFIM 3.2 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 3.2 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

www.pfim.biostat.fr

CONTENT

These next examples are supplied in the folder called Examples in the tool PFIM 3.2. 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. Example 1: PK model using the library of PK models (ODE)

The purpose is to evaluate a design using a one compartment first order absorption model with a Michaelis-Menten elimination described by a differential equation system.
The design to be evaluated is composed of one group of 30 subjects with a dose of 13.8 and sampling times at 0.5, 2, 16, 30.

1.1. MODEL FILE

```
source(paste(directory.program,"\\","LibraryPK.r",sep=""))

formED<-oral1_1cpt_kaVVmkm(doseMM=13.8)
```

1.2. INPUT FILE

```
###########################################################################
##                                                                      ##
##                     INPUT FILE FOR PFIM 3.2                           ##
###########################################################################


#Name of the project
#--------------------
project<-"Example 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"

#Block diagonal Fisher information matrix (option<-1) or complete Information
matrix (option<-2)
#-----------------------------------------------------------
option<-1

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

nr<-1

################### 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(30)

#Vector of the times 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<-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(0)))


# Error tolerance for solving differential equations
#----------------------------------------------------
RtolEQ<-1e-08
AtolEQ<-1e-08
Hmax<-Inf# Default value


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


#Name of the fixed effects parameters
#-------------------------------------
parameters<-c("ka","V","Vm","km")


#Fixed effects parameters values
#--------------------------------
beta<-c(2.72,12.2,1.0004,0.37)


#Number of occasions
#--------------------------------------------------------------------------------
n_occ<-1


#Random effect model (1) = additive  (2) = exponential
#------------------------------------------------------------------
Trand<-2;


#Diagonal Matrix of variance for inter-subject random effects:
```

```
#---------------------------------------------------
omega<-diag(c(0.25,0.25,0.25,0.25))

#Diagonal Matrix of variance for inter-occasion random effects:
#---------------------------------------------------
gamma<-diag(c(0,0,0,0))

#Standard deviation of residual error (sig.inter+sig.slope*f)^2:
#------------------------------------------------------------------
sig.interA<-0
sig.slopeA<-0.2

#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, 2, 16, 30))

#Vector of initial proportions or numbers of subjects for each elementary design
#--------------------------------------------------------------
subjects<-c(30)

#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<-1000




###################################################################
#                                                                 #
#                        Covariate model                          #
#                                                                 #
###################################################################

##########################################
# Covariates not changing with occasion  #
##########################################

#Add covariate to the model  (Yes==T No==F)
#---------------------------------------------------------------------------
covariate.model<-F

#Vector of covariates
#---------------------------------------------------------------------
covariate.name<-list(c("Sex"))

#Categories for each covariate (the first category is the reference)
#-----------------------------------------------------------------------
covariate.category<-list(Sex=c("M","F"))

#Proportions of subjects in each category
#-------------------------------------------------------------------------
covariate.proportions<-list(Sex=c(0.5,0.5))

#Parameter(s) associated with each covariate
#-------------------------------------------------------------------------
parameter.associated<-list(Sex=c("ka"))

# Values of covariate parameters in covariate model
# (values of parameters for all other categories than the reference category (for
which beta=0)
# covariate is additive on parameter if additive random effect model (Trand=1)
# covariate is additive on log parameters if exponential random effect model
(Trand=2)
#-----------------------------------------------------------------------
beta.covariate<-list(Sex=list(c(log(1.2))))



#####################################
#Covariates changing with occasion  #
#####################################


#Add covariate to the model   (Yes==T No==F)
#---------------------------------------------------------------------------
covariate_occ.model<-F

#Vector of covariates depending on the occasion
#---------------------------------------------------------------------
covariate_occ.name<-list(c("Treat"))

#Categories for each covariate (the first category is the reference)
#-----------------------------------------------------------------------
covariate_occ.category<-list(
Treat=c("A","B"))

#Sequences of values of covariates at each occasion
#Specify as many values in each sequence as number of occasions (n_occ) for each
covariate
#----------------------------------------------------------------------------------
---------------------

covariate_occ.sequence<-list(
Treat=list(c("A","B"),c("B","A")))

#Proportions of elementary designs corresponding to each sequence of covariate
values
#Specify as many values of proportion as number of sequences defined in
covariate_occ.sequence for each covariate
#----------------------------------------------------------------------------------
-------------------------------
covariate_occ.proportions<-list(
Treat=c(0.5,0.5))

#Parameter(s) associated with each covariate
#-------------------------------------------------------------------------
parameter_occ.associated<-list(Treat=c("Cl"))


# Values of covariate parameters in covariate model
# (values of parameters for all other categories than the reference category (for
which beta=0)
# covariate is additive on parameter if additive random effect model (Trand=1)
# covariate is additive on log parameters if exponential random effect model
(Trand=2)
#-----------------------------------------------------------------------
beta.covariate_occ<-list(Treat=list(c(log(1.1))))


#############################################
# Power and number of subjects             #
#############################################

#Type one error alpha
#-----------------------------------------------------------------------------
alpha<-0.05

#Compute expected power for comparison test (Yes=T, No=F)
#---------------------------------------------------------------------------
```

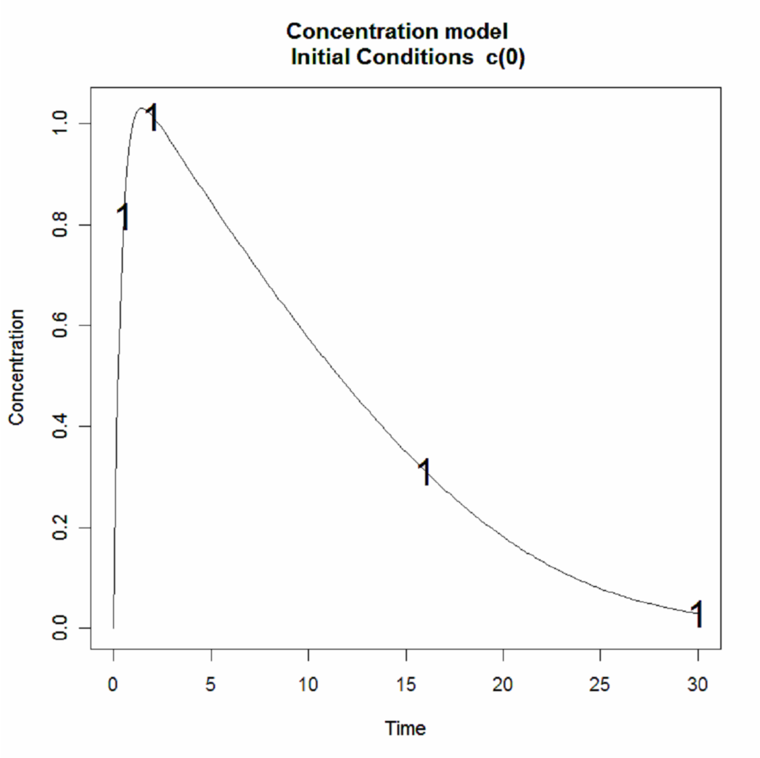
```
compute.power<-F

#Compute the number of subjects needed for a given power for comparison test(Yes=T, 
No=F)
#-----------------------------------------------------------------------------
compute.nni<-F

#Equivalence interval
interval_eq<-c(log(0.8),log(1.25))

#Compute expected power for equivalence test (Yes=T, No=F)
#----------------------------------------------------------------------------
compute.power_eq<-F

#Compute the number of subjects needed for a given power for equivalence test 
(Yes=T, No=F)
#-----------------------------------------------------------------------------
compute.nni_eq<-F

#Set value the given power
#----------------------------------------------------------------------------
given.power<-0.9



############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<-T

#Vector of lower and upper admissible sampling times
#----------------------------------------------------
#lowerA<-c(0)
#upperA<-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(0.5, 2, 5, 16, 18,30))
#sampwinB<-sampwinA

#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

#Vector of Names on Y axes for each response
#---------------------------------
names.datax<-c("Time")

#Vector of Names on Y axes for each response
#---------------------------------
names.datay<-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(30)
```

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

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

1.3. OUTPUT FILE

```
PFIM 3.2

Project:  Example 1

Date:  Fri Jan 08 09:28:41 2010

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

Differential Equations form of the model:

function(t,y,p){
    ka<-p[1]
    V<-p[2]
    Vm <-p[3]
    km<-p[4]
    yd1<-(-Vm/V)*y[1]/(km+y[1])+(doseMM*ka/V)*exp(-ka*t)
    return(list(c(yd1),c(y[1])))
  }


Population design:
Sample times for response: A                           Number of subjects per group
c(0.5, 2, 16, 30)        30


Variance error model response A : ( 0 + 0.2 *f)^2


Initial Conditions at time 0 :

0

Random effect model: Trand =  2

Variance error model response A : ( 0 + 0.2 *f)^2


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


Computation of the Fisher information matrix: option =  1

******************* POPULATION FISHER INFORMATION MATRIX ******************

            [,1]       [,2]        [,3]      [,4]        [,5]        [,6]
 [1,]  7.5443784 -0.3464365   1.4757860  13.94836   0.0000000   0.0000000
 [2,] -0.3464365  0.7444780   0.2402634   2.46935   0.0000000   0.0000000
 [3,]  1.4757860  0.2402634 100.9952291 -98.17955   0.0000000   0.0000000
 [4,] 13.9483580  2.4693500 -98.1795506 329.69735   0.0000000   0.0000000
 [5,]  0.0000000  0.0000000   0.0000000   0.00000  51.9243773   2.2026911
 [6,]  0.0000000  0.0000000   0.0000000   0.00000   2.2026911 204.6405919
 [7,]  0.0000000  0.0000000   0.0000000   0.00000   0.2687700   0.1433148
 [8,]  0.0000000  0.0000000   0.0000000   0.00000   3.2842497   2.0707953
 [9,]  0.0000000  0.0000000   0.0000000   0.00000 134.8814273  31.3970284
             [,7]      [,8]        [,9]
 [1,]   0.0000000  0.000000    0.00000
 [2,]   0.0000000  0.000000    0.00000
 [3,]   0.0000000  0.000000    0.00000
 [4,]   0.0000000  0.000000    0.00000
 [5,]   0.2687700  3.284250  134.88143
 [6,]   0.1433148  2.070795   31.39703
 [7,] 170.2727693 22.011095   23.64034
 [8,]  22.0110948 33.953657   72.37834
 [9,]  23.6403354 72.378344 1433.80666


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

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

      Beta   StdError      RSE
ka  2.7200 0.40414952 14.85844 %
V  12.2000 1.22169158 10.01387 %
Vm  1.0004 0.12439676 12.43470 %
km  0.3700 0.07317843 19.77796 %


------------------------- Variance of Inter-Subject Random Effects ----------------

   Omega   StdError      RSE
ka  0.25 0.16049473 64.19789 %
V   0.25 0.07002607 28.01043 %
Vm  0.25 0.08017411 32.06965 %
km  0.25 0.19078721 76.31489 %


------------------------ Standard deviation of residual error ---------------------

           Sigma  StdError     RSE
sig.slopeA   0.2 0.0323076 16.1538 %


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

5.524547e+15

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

56.12341
```


Concentration model
Initial Conditions c(0)
Concentration
0.0
0.2
0.4
0.6
0.8
1.0
Time
0
5
10
15
20
25
30
1
1
1
1

2. Example 2: PK model using the library of PK models (ODE) (Computation of the full Fisher information matrix)

This example is the same as the example see in the previous section. However, instead of the computation of a block diagonal Fisher information matrix, the full one is used for design evaluation.

2.1. MODEL FILE

```
source(paste(directory.program,"\\","LibraryPK.r",sep=""))

formED<-oral1_1cpt_kaVVmkm(doseMM=13.8)
```

2.2. INPUT FILE

```
#########################################################################
##                   INPUT FILE FOR PFIM 3.2                          ##
#########################################################################


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

project<-"Example 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"

#Block diagonal Fisher information matrix (option<-1) or complete Information
matrix (option<-2)
#----------------------------------------------------------
option<-2

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

nr<-1

################### 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(30)

#Vector of the times 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<-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(0)))


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

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


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


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

parameters<-c("ka","V","Vm","km")


#Fixed effects parameters values
#--------------------------------
beta<-c(2.72,12.2,1.0004,0.37)


#Number of occasions
#-------------------------------------------------------------------------------
n_occ<-1


#Random effect model (1) = additive  (2) = exponential
#------------------------------------------------------------------

Trand<-2;
```

```
#Diagonal Matrix of variance for inter-subject random effects:
#---------------------------------------------------

omega<-diag(c(0.25,0.25,0.25,0.25))

#Diagonal Matrix of variance for inter-occasion random effects:
#---------------------------------------------------

gamma<-diag(c(0,0,0,0))

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

sig.interA<-0
sig.slopeA<-0.2

#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, 2,16, 30))

#Vector of initial proportions or numbers of subjects for each elementary design
#--------------------------------------------------------------

subjects<-c(30)

#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<-1000




###################################################################
#                                                                 #
#                        Covariate model                          #
#                                                                 #
###################################################################

##########################################
# Covariates not changing with occasion  #
##########################################

#Add covariate to the model  (Yes==T No==F)
#---------------------------------------------------------------------------
covariate.model<-F

#Vector of covariates
#---------------------------------------------------------------------
covariate.name<-list(c("Sex"))

#Categories for each covariate (the first category is the reference)
#-----------------------------------------------------------------------
covariate.category<-list(Sex=c("M","F"))

#Proportions of subjects in each category
#-------------------------------------------------------------------------
covariate.proportions<-list(Sex=c(0.5,0.5))

#Parameter(s) associated with each covariate
#-------------------------------------------------------------------------
```

```
parameter.associated<-list(Sex=c("ka"))

# Values of covariate parameters in covariate model
# (values of parameters for all other categories than the reference category (for
which beta=0)
# covariate is additive on parameter if additive random effect model (Trand=1)
# covariate is additive on log parameters if exponential random effect model
(Trand=2)
#-------------------------------------------------------------------------
beta.covariate<-list(Sex=list(c(log(1.2))))



######################################
#Covariates changing with occasion  #
######################################


#Add covariate to the model   (Yes==T No==F)
#-----------------------------------------------------------------------------
covariate_occ.model<-F

#Vector of covariates depending on the occasion
#-----------------------------------------------------------------------
covariate_occ.name<-list(c("Treat"))

#Categories for each covariate (the first category is the reference)
#-------------------------------------------------------------------------
covariate_occ.category<-list(
Treat=c("A","B"))

#Sequences of values of covariates at each occasion
#Specify as many values in each sequence as number of occasions (n_occ) for each
covariate
#------------------------------------------------------------------------------------
---------------------

covariate_occ.sequence<-list(
Treat=list(c("A","B"),c("B","A")))

#Proportions of elementary designs corresponding to each sequence of covariate
values
#Specify as many values of proportion as number of sequences defined in
covariate_occ.sequence for each covariate
#------------------------------------------------------------------------------------
-------------------------------
covariate_occ.proportions<-list(
Treat=c(0.5,0.5))

#Parameter(s) associated with each covariate
#---------------------------------------------------------------------------
parameter_occ.associated<-list(Treat=c("Cl"))


# Values of covariate parameters in covariate model
# (values of parameters for all other categories than the reference category (for
which beta=0)
# covariate is additive on parameter if additive random effect model (Trand=1)
# covariate is additive on log parameters if exponential random effect model
(Trand=2)
#-------------------------------------------------------------------------
beta.covariate_occ<-list(Treat=list(c(log(1.1))))


##############################################
# Power and number of subjects              #
##############################################
```

```
#Type one error alpha
#-----------------------------------------------------------------------------
alpha<-0.05

#Compute expected power for comparison test (Yes=T, No=F)
#---------------------------------------------------------------------------
compute.power<-F

#Compute the number of subjects needed for a given power for comparison test(Yes=T,
No=F)
#----------------------------------------------------------------------------
compute.nni<-F

#Equivalence interval
interval_eq<-c(log(0.8),log(1.25))

#Compute expected power for equivalence test (Yes=T, No=F)
#---------------------------------------------------------------------------
compute.power_eq<-F

#Compute the number of subjects needed for a given power for equivalence test
(Yes=T, No=F)
#----------------------------------------------------------------------------
compute.nni_eq<-F

#Set value the given power
#---------------------------------------------------------------------------
given.power<-0.9



############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<-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(0.5, 2, 5, 16, 18,30))
#sampwinB<-sampwinA


#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

#Vector of Names on Y axes for each response
#---------------------------------
names.datax<-c("Time")

#Vector of Names on Y axes for each response
#---------------------------------
names.datay<-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(30)


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

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

2.3. OUTPUT FILE

```
PFIM 3.2 Option 2

Project:  Example 2

Date:  Wed Jan 13 10:17:11 2010

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

Differential Equations form of the model:

function(t,y,p){
    ka<-p[1]
    V<-p[2]
    Vm <-p[3]
    km<-p[4]
    yd1<-(-Vm/V)*y[1]/(km+y[1])+(doseMM*ka/V)*exp(-ka*t)
    return(list(c(yd1),c(y[1])))
  }


Population design:


Sample times for response: A
                  subjects
```

```
c(0.5, 2, 16, 30)       30
Variance error model response A : ( 0 + 0.2 *f)^2


Initial Conditions at time 0 :

0


Random effect model: Trand =  2

Variance error model response A : ( 0 + 0.2 *f)^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,]   9.3697342   -1.825867     36.088015    -28.124776   2.8882476
 [2,]  -1.8258669   14.920336   -327.510871    449.127394  -4.6292570
 [3,]  36.0880147 -327.510871   8882.366648 -11821.563142   2.6462288
 [4,] -28.1247764  449.127394 -11821.563142  16126.006894  -1.7689124
 [5,]   2.8882476   -4.629257      2.646229     -1.768912  51.9243773
 [6,]   3.4540280   -3.813145   -159.089031    173.718068   2.2026911
 [7,]  -0.1073093  -10.355497     26.206971    -68.270779   0.2687700
 [8,]  -1.7206236   15.037381   -461.387100    617.804716   3.2842497
 [9,]   1.1196142  -28.955935   -217.596541    174.923333 134.8814273
              [,6]        [,7]        [,8]        [,9]
 [1,]    3.4540280  -0.1073093   -1.720624    1.119614
 [2,]   -3.8131453 -10.3554967   15.037381  -28.955935
 [3,] -159.0890310  26.2069714 -461.387100 -217.596541
 [4,]  173.7180679 -68.2707793  617.804716  174.923333
 [5,]    2.2026911   0.2687700    3.284250  134.881427
 [6,]  204.6405919   0.1433148    2.070795   31.397028
 [7,]    0.1433148 170.2727693   22.011095   23.640335
 [8,]    2.0707953  22.0110948   33.953657   72.378344
 [9,]   31.3970284  23.6403354   72.378344 1433.806658


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

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

      Beta   StdError       RSE
ka  2.7200 0.36644273 13.472159 %
V  12.2000 1.11532825  9.142035 %
Vm  1.0004 0.07588594  7.585560 %
km  0.3700 0.06404214 17.308687 %


------------------------- Variance of Inter-Subject Random Effects ----------------
------

   Omega   StdError       RSE
ka  0.25 0.16832117  67.32847 %
V   0.25 0.08484267  33.93707 %
Vm  0.25 0.12020636  48.08254 %
km  0.25 0.52462594 209.85037 %


------------------------ Standard deviation of residual error ---------------------
---

           Sigma   StdError      RSE
sig.slopeA   0.2 0.04658515 23.29257 %
```

```
****************************** DETERMINANT ******************************

1.439948e+17

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

80.62746
```

3. Example 3: PK and immediate response PD model using the libraries of PK and PD models (ODE)

The purpose is to evaluate a design using a PK / PD model. The PK model is a one compartment model with an infusion of 1 hour and a Michaelis-Menten elimination. The PD model is an immediate response model with linear drug action and no baseline. The PK / PD model is described by a differential equation system thanks to the use of the function create_formED implemented in the file CreateModel_PKPDdesign.r. The design to be evaluated is composed of one group of 100 subjects with a dose of 100 and sampling times at 0.5, 2, 30, 49, 180 for the PK and 0.5, 2, 14, 110, 150 for the PD.

3.1. MODEL FILE

```
source(paste(directory.program,dirsep,"CreateModel_PKPDdesign.r",sep=""))

create_formED(infusion_1cpt_VVmkm,immed_lin_null,dose=100,TInf=1)
# The differential equation system is created in the file model_created.r
```

3.2. INPUT FILE

```
#########################################################################
##                                                                     ##
##                      INPUT FILE FOR PFIM 3.2                        ##
#########################################################################


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

project<-"Example 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"

#Block diagonal Fisher information matrix (option<-1) or complete Information matrix (option<-2)
#-----------------------------------------------------------
```

```
option<-1

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

nr<-2

################## 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(30)

#Vector of the times 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))


# Error tolerance for solving differential equations
#----------------------------------------------------
RtolEQ<-1e-08
AtolEQ<-1e-08
Hmax<-0.5# Default value


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


#Name of the fixed effects parameters
#-------------------------------------
parameters<-c("V","Vm","km","Alin")
```

```
#Fixed effects parameters values
#--------------------------------
beta<-c(12.2,0.082,0.37,0.1)


#Number of occasions
#--------------------------------------------------------------------------------
n_occ<-1


#Random effect model (1) = additive  (2) = exponential
#------------------------------------------------------------------
Trand<-2;


#Diagonal Matrix of variance for inter-subject random effects:
#---------------------------------------------------
omega<-diag(c(0.25,0.25,0,0.25))

#Diagonal Matrix of variance for inter-occasion random effects:
#---------------------------------------------------
gamma<-diag(c(0,0,0))

#Standard deviation of residual error (sig.inter+sig.slope*f)^2:
#------------------------------------------------------------------
sig.interA<-0
sig.slopeA<-0.2

sig.interB<-0.1
sig.slopeB<-0

#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, 2, 30, 49, 180))
protB<-list(c(0.5, 2, 14, 110, 150))

#Vector of initial proportions or numbers of subjects for each elementary design
#--------------------------------------------------------------
subjects<-c(100)

#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<-1000


###################################################################
#                                                                 #
#                        Covariate model                          #
#                                                                 #
###################################################################

##########################################
# Covariates not changing with occasion  #
##########################################

#Add covariate to the model  (Yes==T No==F)
#---------------------------------------------------------------------------
covariate.model<-F

#Vector of covariates
```

```
#---------------------------------------------------------------------
covariate.name<-list(c("Sex"))

#Categories for each covariate (the first category is the reference)
#-----------------------------------------------------------------------
covariate.category<-list(Sex=c("M","F"))

#Proportions of subjects in each category
#-------------------------------------------------------------------------
covariate.proportions<-list(Sex=c(0.5,0.5))

#Parameter(s) associated with each covariate
#-------------------------------------------------------------------------
parameter.associated<-list(Sex=c("V"))

# Values of covariate parameters in covariate model
# (values of parameters for all other categories than the reference category (for
which beta=0)
# covariate is additive on parameter if additive random effect model (Trand=1)
# covariate is additive on log parameters if exponential random effect model
(Trand=2)
#-----------------------------------------------------------------------
beta.covariate<-list(Sex=list(c(log(1.2))))



#####################################
#Covariates changing with occasion  #
#####################################


#Add covariate to the model   (Yes==T No==F)
#---------------------------------------------------------------------------
covariate_occ.model<-F

#Vector of covariates depending on the occasion
#---------------------------------------------------------------------
covariate_occ.name<-list(c("Treat"))

#Categories for each covariate (the first category is the reference)
#-----------------------------------------------------------------------
covariate_occ.category<-list(
Treat=c("A","B"))

#Sequences of values of covariates at each occasion
#Specify as many values in each sequence as number of occasions (n_occ) for each
covariate
#----------------------------------------------------------------------------------
---------------------

covariate_occ.sequence<-list(
Treat=list(c("A","B"),c("B","A")))

#Proportions of elementary designs corresponding to each sequence of covariate
#values
#Specify as many values of proportion as number of sequences defined in
#covariate_occ.sequence for each covariate
#----------------------------------------------------------------------------------
covariate_occ.proportions<-list(
Treat=c(0.5,0.5))

#Parameter(s) associated with each covariate
#-------------------------------------------------------------------------
parameter_occ.associated<-list(Treat=c("Cl"))


# Values of covariate parameters in covariate model
# (values of parameters for all other categories than the reference category (for
#which beta=0)
# covariate is additive on parameter if additive random effect model (Trand=1)
# covariate is additive on log parameters if exponential random effect model
#(Trand=2)
#-------------------------------------------------------------------------
beta.covariate_occ<-list(Treat=list(c(log(1.1))))


##############################################
# Power and number of subjects               #
##############################################
#Type one error alpha
#-------------------------------------------------------------------------------
alpha<-0.05

#Compute expected power for comparison test (Yes=T, No=F)
#-----------------------------------------------------------------------------
compute.power<-F

#Compute the number of subjects needed for a given power for comparison test(Yes=T,
No=F)
#------------------------------------------------------------------------------
compute.nni<-F

#Equivalence interval
interval_eq<-c(log(0.8),log(1.25))

#Compute expected power for equivalence test (Yes=T, No=F)
#-----------------------------------------------------------------------------
compute.power_eq<-F

#Compute the number of subjects needed for a given power for equivalence test
(Yes=T, No=F)
#------------------------------------------------------------------------------
compute.nni_eq<-F

#Set value the given power
#-----------------------------------------------------------------------------
given.power<-0.9


############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(150)

#lowerB<-c(0)
#upperB<-c(150)

#Minimum delay between two sampling times
#-----------------------------------------
#delta.time<-0.5

#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, 2, 30, 32,110, 58))
#sampwinB<-list(c(0.5, 2, 14, 50, 110, 150))


#List of vector of allowed number of points to be taken from each sampling window
#-----------------------------------------------------------------------------
#nsampA<-list(c(5))
#nsampB<-list(c(5))

#Maximum total number of sampling times per subject
#--------------------------------------------------
#nmaxptsA<-5
#nmaxptsB<-5

#Minimum total number of sampling times per subject
#--------------------------------------------------
#nminptsA<-5
#nminptsB<-5

############# END OF OPTIMISATION ALGORITHM OPTION ###############


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

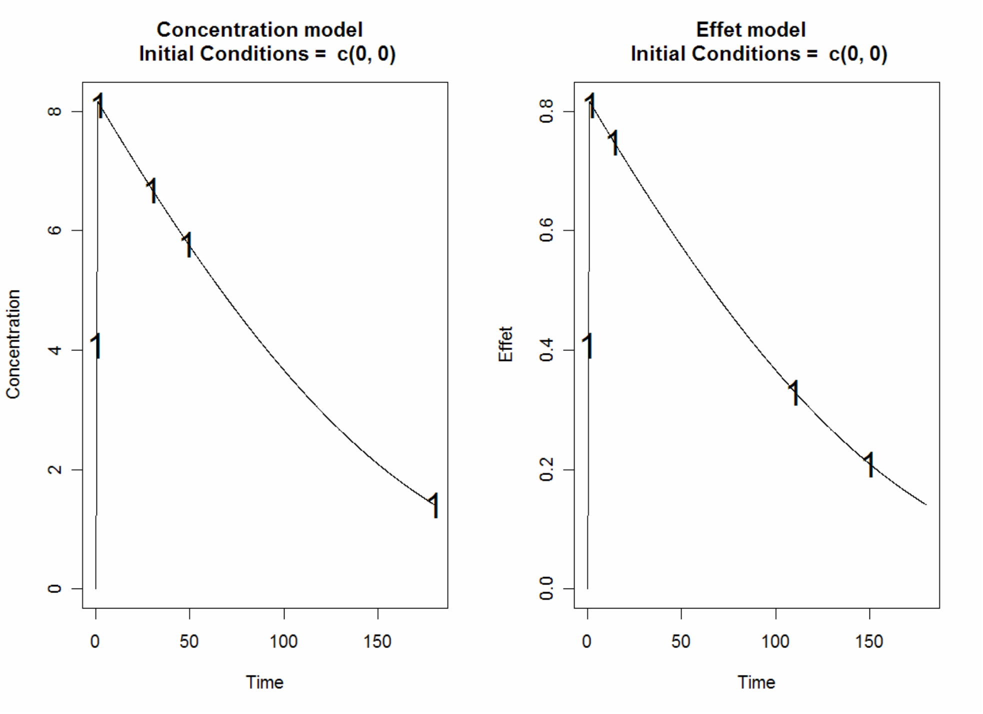
```
#graphical representation (Yes=T, No=F)
#--------------------------------------
graph.logical<-T

#Vector of Names on Y axes for each response
#----------------------------------
names.datax<-c("Time","Time")

#Vector of Names on Y axes for each response
#----------------------------------
names.datay<-c("Concentration","Effet")

#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(180)
graph.infB<-c(0)
graph.supB<-c(180)

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

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

3.3. OUTPUT FILE

```
PFIM 3.2 Option 1

Project:  Example 3

Date:  Fri Jan 08 17:19:21 2010

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

Differential Equations form of the model:

function(t,y,p){
V<-p[1]
Vm<-p[2]
km<-p[3]
Alin<-p[4]
pk<-y[1:1]
pd<-y[2:2]
conc<-y[1]
if(t<=1){
dpk1<-(100/(1*V))+(-Vm)*pk[1]/(km*V+pk[1])}
else{
dpk1<-(-Vm)*pk[1]/(km*V+pk[1])}
dpd1<-0
pdIm<-Alin*conc
return(list(c(dpk1,dpd1),c(pk[1],pdIm)))
}


Population design:
Sample times for response: A                           Number of subjects per group
c(0.5, 2, 30, 50, 180)        100
```

```
Sample times for response: B                         Number of subjects per group
c(0.5, 2, 14, 110, 150)        100


Variance error model response A : ( 0 + 0.2 *f)^2
Variance error model response B : ( 0.1 + 0 *f)^2


Initial Conditions at time 0 :

0 0


Random effect model: Trand =  2

Error tolerance for solving differential equations system: RtolEQ = 1e-08 , AtolEQ
= 1e-08 , Hmax =  0.5



Computation of the Fisher information matrix: option = 1

******************* POPULATION FISHER INFORMATION MATRIX ******************

             [,1]         [,2]          [,3]        [,4]        [,5]
 [1,]   2.5865260     5.374266     0.8128864   -11.44898   0.0000000
 [2,]   5.3742659 57945.602398 -6866.1966934   435.14502   0.0000000
 [3,]   0.8128864 -6866.196693   930.6251972   -26.96431   0.0000000
 [4,] -11.4489834   435.145019   -26.9643093 37662.89817   0.0000000
 [5,]   0.0000000     0.000000     0.0000000     0.00000 741.0423198
 [6,]   0.0000000     0.000000     0.0000000     0.00000   0.1445292
 [7,]   0.0000000     0.000000     0.0000000     0.00000   0.9754916
 [8,]   0.0000000     0.000000     0.0000000     0.00000  67.4492295
 [9,]   0.0000000     0.000000     0.0000000     0.00000   4.0750783
              [,6]         [,7]        [,8]         [,9]
 [1,]   0.00000000   0.00000000     0.00000     0.000000
 [2,]   0.00000000   0.00000000     0.00000     0.000000
 [3,]   0.00000000   0.00000000     0.00000     0.000000
 [4,]   0.00000000   0.00000000     0.00000     0.000000
 [5,]   0.14452922   0.97549156    67.44923     4.075078
 [6,] 759.04299755   0.06365987    41.68129    16.643793
 [7,]   0.06365987 709.24694938    60.48191    93.895495
 [8,]  41.68128936  60.48190935 15215.85874  1305.080765
 [9,]  16.64379328  93.89549451  1305.08077 74085.359061


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

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

       Beta    StdError       RSE
V    12.200 0.624466721  5.118580 %
Vm    0.082 0.011757152 14.337990 %
km    0.370 0.092773757 25.073988 %
Alin  0.100 0.005157086  5.157086 %


------------------------- Variance of Inter-Subject Random Effects ----------------

     Omega   StdError      RSE
V     0.25 0.03674230 14.69692 %
Vm    0.25 0.03629946 14.51978 %
Alin  0.25 0.03755845 15.02338 %


------------------------ Standard deviation of residual error --------------------
```

```
           Sigma    StdError      RSE
sig.slopeA   0.2 0.008116514 4.058257 %
sig.interB   0.1 0.003677015 3.677015 %


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

2.937072e+29

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

1880.238
```



4. Example 4: PK and turnover response PD model using the libraries of PK and PD models (ODE)

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 a turnover response model with *full Imax* for the effect. The model is described by a differential equation system obtained thanks to the use of the function create_formED implemented in the file CreateModel_PKPDdesign.r. The design to be evaluated is composed of one group of 100 subjects with a dose of 100 and sampling times at 0.5, 1, 2, 19, 38, 61, 160 for the PK and at 0, 0.7, 1.5, 23, 12, 44, 144 for the PD.

4.1. MODEL FILE

```
source(paste(directory.program,dirsep,"CreateModel_PKPDdesign.r",sep=""))

create_formED(infusion_1cpt_VVmkm,turn_input_Imaxfull,dose=100,Tinf=1)
# the differential equation system is created in the file model_created.r
```

4.2. INPUT FILE

```
#####################################################################
##                   INPUT FILE FOR PFIM 3.2                         ##
#########################################################################


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

project<-"Example 4"


#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"

#Block diagonal Fisher information matrix (option<-1) or complete Information
matrix (option<-2)
#----------------------------------------------------------
option<-1

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

nr<-2

################### 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(30)

#Vector of the times 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,Rin/kout))


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

RtolEQ<-1e-08
AtolEQ<-1e-08
Hmax<-0.5# Default value


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


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

parameters<-c("V","Vm","km","Rin","kout","C50")


#Fixed effects parameters values
#--------------------------------
beta<-c(12,0.1,0.5,6.4,1.2,1)


#Number of occasions
#--------------------------------------------------------------------------------
n_occ<-1


#Random effect model (1) = additive  (2) = exponential
#------------------------------------------------------------------

Trand<-2;


#Diagonal Matrix of variance for inter-subject random effects:
#---------------------------------------------------

omega<-diag(c(0.25,0.25,0,0.3,0.25,0))

#Diagonal Matrix of variance for inter-occasion random effects:
#---------------------------------------------------

gamma<-diag(c(0,0,0))

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

sig.interA<-0
sig.slopeA<-0.2
sig.interB<-3.8
sig.slopeB<-0

#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, 19, 38, 61, 160))
protB<-list(c(0, 0.7, 1.5, 23, 12, 44, 144))

#Vector of initial proportions or numbers of subjects for each elementary design
#--------------------------------------------------------------

subjects<-c(100)

#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<-1000




###################################################################
#                                                                 #
#                        Covariate model                          #
#                                                                 #
###################################################################

##########################################
# Covariates not changing with occasion  #
##########################################

#Add covariate to the model  (Yes==T No==F)
#---------------------------------------------------------------------------
covariate.model<-F

#Vector of covariates
#---------------------------------------------------------------------
covariate.name<-list(c("Sex"))

#Categories for each covariate (the first category is the reference)
#-----------------------------------------------------------------------
covariate.category<-list(Sex=c("M","F"))

#Proportions of subjects in each category
#-------------------------------------------------------------------------
covariate.proportions<-list(Sex=c(0.5,0.5))

#Parameter(s) associated with each covariate
#-------------------------------------------------------------------------
parameter.associated<-list(Sex=c("V"))

# Values of covariate parameters in covariate model
# (values of parameters for all other categories than the reference category (for
which beta=0)
# covariate is additive on parameter if additive random effect model (Trand=1)
# covariate is additive on log parameters if exponential random effect model
(Trand=2)
#-----------------------------------------------------------------------
beta.covariate<-list(Sex=list(c(log(1.2))))
```

```
#####################################
#Covariates changing with occasion  #
#####################################


#Add covariate to the model   (Yes==T No==F)
#---------------------------------------------------------------------------
covariate_occ.model<-F

#Vector of covariates depending on the occasion
#---------------------------------------------------------------------
covariate_occ.name<-list(c("Treat"))

#Categories for each covariate (the first category is the reference)
#-----------------------------------------------------------------------
covariate_occ.category<-list(
Treat=c("A","B"))

#Sequences of values of covariates at each occasion
#Specify as many values in each sequence as number of occasions (n_occ) for each
covariate
#----------------------------------------------------------------------------------
---------------------

covariate_occ.sequence<-list(
Treat=list(c("A","B"),c("B","A")))

#Proportions of elementary designs corresponding to each sequence of covariate
values
#Specify as many values of proportion as number of sequences defined in
covariate_occ.sequence for each covariate
#----------------------------------------------------------------------------------
-------------------------------
covariate_occ.proportions<-list(
Treat=c(0.5,0.5))

#Parameter(s) associated with each covariate
#-------------------------------------------------------------------------
parameter_occ.associated<-list(Treat=c("Cl"))


# Values of covariate parameters in covariate model
# (values of parameters for all other categories than the reference category (for
which beta=0)
# covariate is additive on parameter if additive random effect model (Trand=1)
# covariate is additive on log parameters if exponential random effect model
(Trand=2)
#-----------------------------------------------------------------------
beta.covariate_occ<-list(Treat=list(c(log(1.1))))


#############################################
# Power and number of subjects              #
#############################################

#Type one error alpha
#-----------------------------------------------------------------------------
alpha<-0.05

#Compute expected power for comparison test (Yes=T, No=F)
#---------------------------------------------------------------------------
compute.power<-F

#Compute the number of subjects needed for a given power for comparison test(Yes=T,
No=F)
#----------------------------------------------------------------------------
compute.nni<-F
```

```
#Equivalence interval
interval_eq<-c(log(0.8),log(1.25))

#Compute expected power for equivalence test (Yes=T, No=F)
#---------------------------------------------------------------------------
compute.power_eq<-F

#Compute the number of subjects needed for a given power for equivalence test
(Yes=T, No=F)
#----------------------------------------------------------------------------
compute.nni_eq<-F

#Set value the given power
#---------------------------------------------------------------------------
given.power<-0.9



############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<-T

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


lowerA<-c(0)
upperA<-c(150)

lowerB<-c(0)
upperB<-c(150)

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

delta.time<-0.5

#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(0.5,1,1.5,2,4,6,8))
#sampwinB<-sampwinA


#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

#Vector of Names on Y axes for each response
#---------------------------------
```

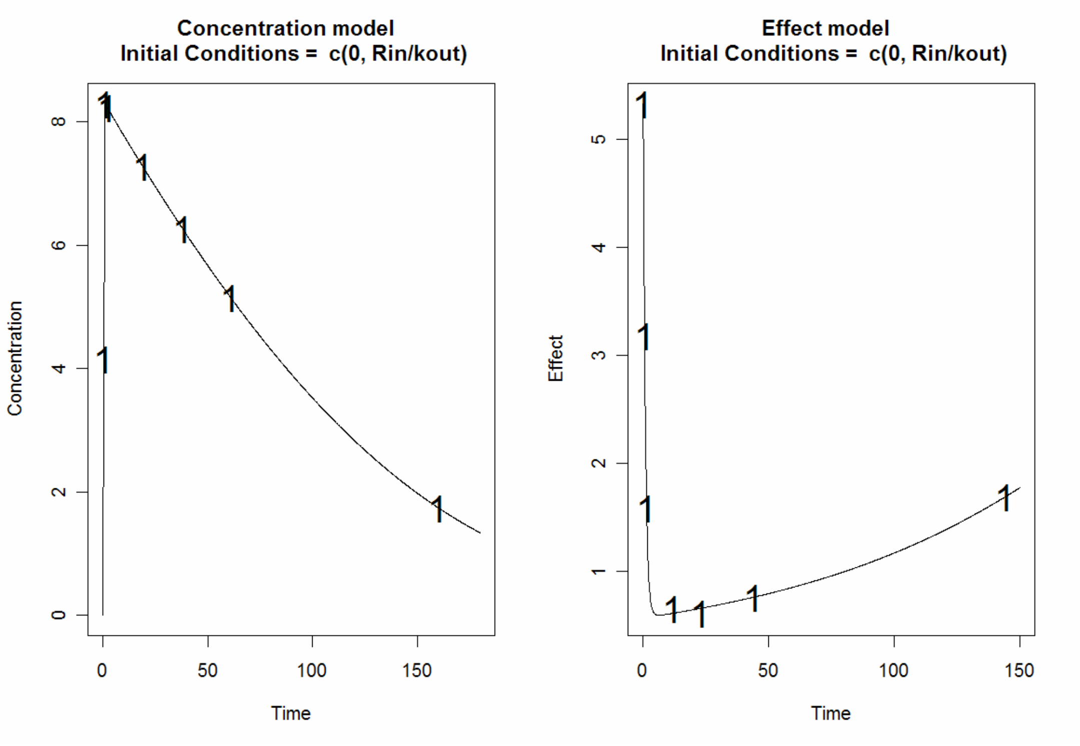
```
names.datax<-c("Time","Time")

#Vector of Names on Y axes for each response
#---------------------------------
names.datay<-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(180)
graph.infB<-c(0)
graph.supB<-c(150)

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

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

4.3. OUTPUT FILE

```
PFIM 3.2 Option 1

Project:  Example 4

Date:  Wed Jan 13 16:03:52 2010

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

Differential Equations form of the model:

function(t,y,p){
V<-p[1]
Vm<-p[2]
km<-p[3]
Rin<-p[4]
kout<-p[5]
C50<-p[6]
pk<-y[1:1]
pd<-y[2:2]
conc<-y[1]
if(t<=1){
dpk1<-(100/(1*V))+(-Vm)*pk[1]/(km*V+pk[1])}
else{
dpk1<-(-Vm)*pk[1]/(km*V+pk[1])}
dpd1<-Rin*(1-(conc)/(conc+C50))-kout*pd[1]
return(list(c(dpk1,dpd1),c(pk[1],pd[1])))
}


Population design:
Sample times for response: A                          Number of subjects per group
c(0.5, 1, 2, 19, 38, 61, 160)        100


Sample times for response: B                          Number of subjects per group
```

```
c(0, 0.7, 1.5, 23, 12, 44, 144)        100


Variance error model response A : ( 0 + 0.2 *f)^2
Variance error model response B : ( 3.8 + 0 *f)^2


Initial Conditions at time 0 :

0 Rin/kout


Random effect model: Trand =  2

Error tolerance for solving differential equations system: RtolEQ = 1e-08 , AtolEQ
= 1e-08 , Hmax =  0.5



Computation of the Fisher information matrix: option =  1

****************** POPULATION FISHER INFORMATION MATRIX ******************

             [,1]          [,2]          [,3]          [,4]         [,5]
 [1,]  2.697869739     4.6714671  6.797831e-01   0.004141561  -0.02677891
 [2,]  4.671467105 38402.2356550 -4.506943e+03   0.514677290   0.75514596
 [3,]  0.679783140 -4506.9434128  5.622686e+02  -0.060202359   0.12559365
 [4,]  0.004141561     0.5146773 -6.020236e-02   2.612395298 -15.21621758
 [5,] -0.026778907     0.7551460  1.255937e-01 -15.216217578 104.16358087
 [6,]  0.020411693     6.4764180 -1.716588e-01   1.880533487 -14.71377799
 [7,]  0.000000000     0.0000000  0.000000e+00   0.000000000   0.00000000
 [8,]  0.000000000     0.0000000  0.000000e+00   0.000000000   0.00000000
 [9,]  0.000000000     0.0000000  0.000000e+00   0.000000000   0.00000000
[10,]  0.000000000     0.0000000  0.000000e+00   0.000000000   0.00000000
[11,]  0.000000000     0.0000000  0.000000e+00   0.000000000   0.00000000
[12,]  0.000000000     0.0000000  0.000000e+00   0.000000000   0.00000000
             [,6]         [,7]         [,8]         [,9]        [,10]
 [1,]   0.02041169 0.000000e+00 0.000000e+00 0.000000e+00 0.000000e+00
 [2,]   6.47641804 0.000000e+00 0.000000e+00 0.000000e+00 0.000000e+00
 [3,]  -0.17165883 0.000000e+00 0.000000e+00 0.000000e+00 0.000000e+00
 [4,]   1.88053349 0.000000e+00 0.000000e+00 0.000000e+00 0.000000e+00
 [5,] -14.71377799 0.000000e+00 0.000000e+00 0.000000e+00 0.000000e+00
 [6,]  16.76347153 0.000000e+00 0.000000e+00 0.000000e+00 0.000000e+00
 [7,]   0.00000000 7.546350e+02 1.571228e-01 5.058487e-04 7.434995e-04
 [8,]   0.00000000 1.571228e-01 7.373659e+02 5.425003e-04 4.105767e-05
 [9,]   0.00000000 5.058487e-04 5.425003e-04 5.724897e+01 6.828194e+01
[10,]   0.00000000 7.434995e-04 4.105767e-05 6.828194e+01 1.124933e+02
[11,]   0.00000000 5.545026e+01 7.613839e+01 1.073289e-01 1.052174e-01
[12,]   0.00000000 1.699011e-03 8.643442e-03 1.012931e+01 1.388376e+01
            [,11]        [,12]
 [1,] 0.000000e+00  0.000000000
 [2,] 0.000000e+00  0.000000000
 [3,] 0.000000e+00  0.000000000
 [4,] 0.000000e+00  0.000000000
 [5,] 0.000000e+00  0.000000000
 [6,] 0.000000e+00  0.000000000
 [7,] 5.545026e+01  0.001699011
 [8,] 7.613839e+01  0.008643442
 [9,] 1.073289e-01 10.129307301
[10,] 1.052174e-01 13.883760044
[11,] 2.501455e+04  0.716061230
[12,] 7.160612e-01 83.808581021


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

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

```
     Beta   StdError       RSE
V    12.0 0.61405966  5.117164 %
Vm    0.1 0.02114825 21.148249 %
km    0.5 0.17477495 34.954990 %
Rin   6.4 1.61244665 25.194479 %
kout  1.2 0.26159928 21.799940 %
C50   1.0 0.26274049 26.274049 %


------------------------- Variance of Inter-Subject Random Effects ----------------

     Omega   StdError      RSE
V     0.25 0.03640549 14.56220 %
Vm    0.25 0.03683212 14.73285 %
Rin   0.30 0.25183675 83.94558 %
kout  0.25 0.17956896 71.82758 %


------------------------ Standard deviation of residual error ---------------------

           Sigma    StdError      RSE
sig.slopeA   0.2 0.006324225 3.162113 %
sig.interB   3.8 0.110491014 2.907658 %


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

4.041892e+24

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

112.3437
```



5. Example 5: PK model with inter-occasion variability

The purpose of this example is to optimise a design for a PK model including inter-occasion variability using the Fedorov-Wynn algorithm. The PK model is a one compartment oral model with first order absorption and first order elimination. The dose is fixed to 30 for the 40 subjects. We fix the inter-occasion variability to 15% for the three parameters. The design to be evaluated is composed of one group of 40 subjects with a dose of 30 and sampling times at 0.5, 2, 4, 8.

5.1. MODEL FILE

```
source(paste(directory.program,"\\","LibraryPK.r",sep=""))

formA<-oral1_1cpt_kaVCl()[[1]]

form<-c(formA)
```

5.2. INPUT FILE

```
#########################################################################
##
##
##        INPUT FILE FOR PFIM 3.2                          ##
#########################################################################


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

project<-"Example 5 "


#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"

#Block diagonal Fisher information matrix (option<-1) or complete Information
#matrix (option<-2)
#----------------------------------------------------------
option<-1

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

nr<-1

################### 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(30)

#Vector of the times 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(100))


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

RtolEQ<-1e-08
AtolEQ<-1e-08
Hmax<-0.01 # Default value
#Hmax<-Inf #<1.5/24

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


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

parameters<-c("ka","V","Cl")


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

beta<-c(1,3.5,2)


#Number of occasions
#----------------------------------------------------------------------------
n_occ<-2
```

```
#Random effect model (1) = additive  (2) = exponential
#------------------------------------------------------------------

Trand<-2;


#Diagonal Matrix of variance for inter-subject random effects:
#---------------------------------------------------

omega<-diag(c(0.09,0.09,0.09))

#Diagonal Matrix of variance for inter-occasion random effects:
#---------------------------------------------------

gamma<-diag(c(0.0225,0.0225,0.0225))

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

sig.interA<-0.1
sig.slopeA<-0


#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(0.5,2,4,8))

#Vector of initial proportions or numbers of subjects for each elementary design
#--------------------------------------------------------------

subjects<-c(40)

#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<-1000




###################################################################
#                                                                 #
#                        Covariate model                          #
#                                                                 #
###################################################################

##########################################
# Covariates not changing with occasion  #
##########################################

#Add covariate to the model  (Yes==T No==F)
#---------------------------------------------------------------------------
covariate.model<-F

#Vector of covariates
#---------------------------------------------------------------------
covariate.name<-list(c("Sex"))

#Categories for each covariate (the first category is the reference)
#-----------------------------------------------------------------------
```

```
covariate.category<-list(Sex=c("M","F"))

#Proportions of subjects in each category
#-------------------------------------------------------------------------
covariate.proportions<-list(Sex=c(0.5,0.5))

#Parameter(s) associated with each covariate
#-------------------------------------------------------------------------
parameter.associated<-list(Sex=c("V"))

# Values of covariate parameters in covariate model
# (values of parameters for all other categories than the reference category (for
which beta=0)
# covariate is additive on parameter if additive random effect model (Trand=1)
# covariate is additive on log parameters if exponential random effect model
(Trand=2)
#-----------------------------------------------------------------------
beta.covariate<-list(Sex=list(c(log(1.2))))



#####################################
#Covariates changing with occasion  #
#####################################


#Add covariate to the model   (Yes==T No==F)
#---------------------------------------------------------------------------
covariate_occ.model<-F

#Vector of covariates depending on the occasion
#---------------------------------------------------------------------
covariate_occ.name<-list(c("Treat"))

#Categories for each covariate (the first category is the reference)
#-----------------------------------------------------------------------
covariate_occ.category<-list(
Treat=c("A","B"))

#Sequences of values of covariates at each occasion
#Specify as many values in each sequence as number of occasions (n_occ) for each
covariate
#----------------------------------------------------------------------------------
---------------------

covariate_occ.sequence<-list(
Treat=list(c("A","B"),c("B","A")))

#Proportions of elementary designs corresponding to each sequence of covariate
values
#Specify as many values of proportion as number of sequences defined in
covariate_occ.sequence for each covariate
#----------------------------------------------------------------------------------
-------------------------------
covariate_occ.proportions<-list(
Treat=c(0.5,0.5))

#Parameter(s) associated with each covariate
#-------------------------------------------------------------------------
parameter_occ.associated<-list(Treat=c("Cl"))


# Values of covariate parameters in covariate model
# (values of parameters for all other categories than the reference category (for
which beta=0)
# covariate is additive on parameter if additive random effect model (Trand=1)
# covariate is additive on log parameters if exponential random effect model
(Trand=2)
```

```
#-----------------------------------------------------------------------
beta.covariate_occ<-list(Treat=list(c(log(1.1))))


#############################################
# Power and number of subjects              #
#############################################

#Type one error alpha
#-----------------------------------------------------------------------------
alpha<-0.05

#Compute expected power for comparison test (Yes=T, No=F)
#---------------------------------------------------------------------------
compute.power<-F

#Compute the number of subjects needed for a given power for comparison test(Yes=T,
No=F)
#----------------------------------------------------------------------------
compute.nni<-F

#Equivalence interval
interval_eq<-c(log(0.8),log(1.25))

#Compute expected power for equivalence test (Yes=T, No=F)
#---------------------------------------------------------------------------
compute.power_eq<-F

#Compute the number of subjects needed for a given power for equivalence test
(Yes=T, No=F)
#----------------------------------------------------------------------------
compute.nni_eq<-F

#Set value the given power
#---------------------------------------------------------------------------
given.power<-0.9



############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<-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


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

sampwinA<-list(c(0.5,1,1.5,2,4,6,8))



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

nsampA<-list(c(4))


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

nmaxptsA<-4


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

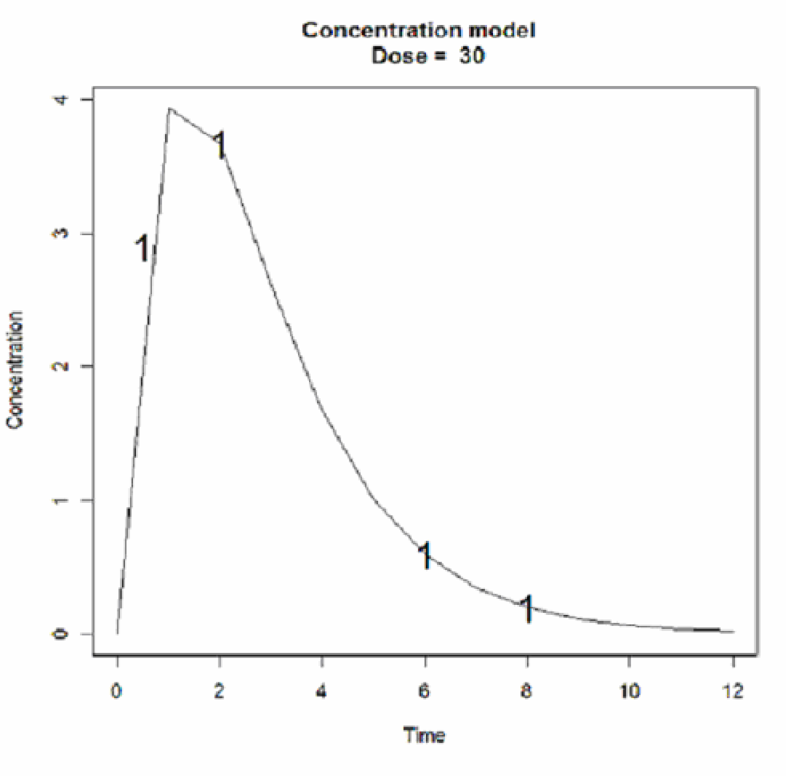
```
nminptsA<-4

############# END OF OPTIMISATION ALGORITHM OPTION ###############


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

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

graph.logical<-T

#Vector of Names on Y axes for each response
#---------------------------------
names.datax<-c("Time")

#Vector of Names on Y axes for each response
#---------------------------------
names.datay<-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.rangeA<-NULL # default range
#y.range<-c(0,10)

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

5.3. OUTPUT FILE

```
PFIM 3.2

Option: 1


Project: Example 5

Date: Thu Feb 10 14:53:33 2011


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

Analytical function model:

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


Initial population design:
```

```
Sample times for response: A
          Protocol subjects doses
1 c=(0.5, 2, 4, 8)       40    30


Total number of samples: 160

Associated criterion value: 1826.068

Identical sampling times for each response: TRUE

Number of occasions: 2

Random effect model: Trand =  2

Variance error model response A : ( 0.1 + 0 *f)^2


Optimization step:

Sampling windows for the response: A
Window 1 : t= 0.5 1 1.5 2 4 6 8
    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 35 protocols generated



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

Optimised population design:
Sample times for response: A
            times freq Subjects doses
1 c(0.5, 2, 6, 8)    1       40    30



Associated optimised criterion: 1913.953


******************* POPULATION FISHER INFORMATION MATRIX ******************

     [,1]       [,2]      [,3]         [,4]         [,5]         [,6]
[1,] 342.070959 -11.909950  2.294299    0.0000000    0.0000000    0.0000000
[2,] -11.909950  29.371134  0.624963    0.0000000    0.0000000    0.0000000
[3,]   2.294299   0.624963 98.030214    0.0000000    0.0000000    0.0000000
[4,]   0.000000   0.000000  0.000000 1462.6567613   21.7203074    0.2631903
[5,]   0.000000   0.000000  0.000000   21.7203074 1618.1679873    0.2392295
[6,]   0.000000   0.000000  0.000000    0.2631903    0.2392295 1921.9845543
[7,]   0.000000   0.000000  0.000000  731.3283806   10.8601537    0.1315952
[8,]   0.000000   0.000000  0.000000   10.8601537  809.0839937    0.1196147
[9,]   0.000000   0.000000  0.000000    0.1315952    0.1196147  960.9922772
[10,]   0.000000   0.000000  0.000000  414.3130643  276.7056180   28.1724200
      [,7]         [,8]         [,9]           [,10]
[1,]   0.000000e+00 0.000000e+00 0.000000e+00     0.00000
[2,]   0.000000e+00 0.000000e+00 0.000000e+00     0.00000
[3,]   0.000000e+00 0.000000e+00 0.000000e+00     0.00000
[4,]   7.313284e+02 1.086015e+01 1.315952e-01   414.31306
[5,]   1.086015e+01 8.090840e+02 1.196147e-01   276.70562
[6,]   1.315952e-01 1.196147e-01 9.609923e+02    28.17242
[7,]   1.252260e+04 4.388584e+03 3.991645e+01  2608.00290
[8,]   4.388584e+03 1.961824e+04 6.185025e+01  1889.93103
[9,]   3.991645e+01 6.185025e+01 3.560094e+04   926.11512
[10,]  2.608003e+03 1.889931e+03 9.261151e+02 20551.58326
************************* EXPECTED STANDARD ERRORS ************************
```

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

   Beta   StdError      RSE
ka  1.0 0.05445931 5.445931 %
V   3.5 0.18585115 5.310033 %
Cl  2.0 0.10101646 5.050823 %




------------------------- Variance of Inter-Subject Random Effects ----------------
------

   Omega   StdError      RSE
ka  0.09 0.02660961 29.56624 %
V   0.09 0.02516549 27.96165 %
Cl  0.09 0.02296550 25.51722 %


------------------------- Variance of Inter-Occasion Random Effects ---------------
-------

    Gamma    StdError      RSE
ka 0.0225 0.009552479 42.45546 %
V  0.0225 0.007539081 33.50703 %
Cl 0.0225 0.005339183 23.72970 %


------------------------ Standard deviation of residual error ---------------------
---

           Sigma    StdError      RSE
sig.interA   0.1 0.007098313 7.098313 %


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

6.596486e+32

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

1913.953
```



6. Example 6: PK model including a covariate effect

The purpose of this example is to evaluate a design for a PK model including a covariate effect. The PK model is a one compartment oral model with first order absorption and first order elimination. We add a gender effect on the volume of distribution (V). The dose is fixed to 30 for the 40 subjects with the same sampling times at 0.5, 2, 6 and 8. For alpha=0.05, we compute the predicted power and also the number of subjects needed to detect this gender effect for a given power= 0.9.

6.1. MODEL FILE

```
source(paste(directory.program,"\\","LibraryPK.r",sep=""))

formA<-oral1_1cpt_kaVCl()[[1]]

form<-c(formA)
```

6.2. INPUT FILE

```
#########################################################################
##                   INPUT FILE FOR PFIM 3.2                           ##
#########################################################################


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

project<-"Example 6 "
```

```
#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"

#Block diagonal Fisher information matrix (option<-1) or complete Information
matrix (option<-2)
#----------------------------------------------------------
option<-1

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

nr<-1

################### 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(30)

#Vector of the times 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(100))


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

RtolEQ<-1e-08
AtolEQ<-1e-08
Hmax<-0.01 # Default value
#Hmax<-Inf #<1.5/24

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


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

parameters<-c("ka","V","Cl")


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

beta<-c(1,3.5,2)


#Number of occasions
#--------------------------------------------------------------------------------
n_occ<-1


#Random effect model (1) = additive  (2) = exponential
#------------------------------------------------------------------

Trand<-2;


#Diagonal Matrix of variance for inter-subject random effects:
#---------------------------------------------------

omega<-diag(c(0.09,0.09,0.09))

#Diagonal Matrix of variance for inter-occasion random effects:
#---------------------------------------------------

gamma<-diag(c(0.09,0.09,0.09))

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

sig.interA<-0.1
sig.slopeA<-0


#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(0.5,2,6,8))

#Vector of initial proportions or numbers of subjects for each elementary design
#--------------------------------------------------------------

subjects<-c(40)
```

```
#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<-1000


###################################################################
#                                                                 #
#                        Covariate model                          #
#                                                                 #
###################################################################

##########################################
# Covariates not changing with occasion  #
##########################################

#Add covariate to the model  (Yes==T No==F)
#---------------------------------------------------------------------------
covariate.model<-T

#Vector of covariates
#---------------------------------------------------------------------
covariate.name<-list(c("Sex"))

#Categories for each covariate (the first category is the reference)
#-----------------------------------------------------------------------
covariate.category<-list(Sex=c("M","F"))

#Proportions of subjects in each category
#-------------------------------------------------------------------------
covariate.proportions<-list(Sex=c(0.5,0.5))

#Parameter(s) associated with each covariate
#-------------------------------------------------------------------------
parameter.associated<-list(Sex=c("V"))

# Values of covariate parameters in covariate model
# (values of parameters for all other categories than the reference category (for
which beta=0)
# covariate is additive on parameter if additive random effect model (Trand=1)
# covariate is additive on log parameters if exponential random effect model
(Trand=2)
#-----------------------------------------------------------------------
beta.covariate<-list(Sex=list(c(log(1.5))))



#####################################
#Covariates changing with occasion  #
#####################################


#Add covariate to the model   (Yes==T No==F)
#---------------------------------------------------------------------------
covariate_occ.model<-F

#Vector of covariates depending on the occasion
#---------------------------------------------------------------------
covariate_occ.name<-list(c("Treat"))

#Categories for each covariate (the first category is the reference)
#-----------------------------------------------------------------------
covariate_occ.category<-list(
Treat=c("A","B"))
```

```
#Sequences of values of covariates at each occasion
#Specify as many values in each sequence as number of occasions (n_occ) for each
covariate
#----------------------------------------------------------------------------------
---------------------

covariate_occ.sequence<-list(
Treat=list(c("A","B"),c("B","A")))

#Proportions of elementary designs corresponding to each sequence of covariate
values
#Specify as many values of proportion as number of sequences defined in
covariate_occ.sequence for each covariate
#----------------------------------------------------------------------------------
-------------------------------
covariate_occ.proportions<-list(
Treat=c(0.5,0.5))

#Parameter(s) associated with each covariate
#-------------------------------------------------------------------------
parameter_occ.associated<-list(Treat=c("Cl"))


# Values of covariate parameters in covariate model
# (values of parameters for all other categories than the reference category (for
which beta=0)
# covariate is additive on parameter if additive random effect model (Trand=1)
# covariate is additive on log parameters if exponential random effect model
(Trand=2)
#-----------------------------------------------------------------------
beta.covariate_occ<-list(Treat=list(c(log(1.1))))


#############################################
# Power and number of subjects              #
#############################################

#Type one error alpha
#-----------------------------------------------------------------------------
alpha<-0.05

#Compute expected power for comparison test (Yes=T, No=F)
#---------------------------------------------------------------------------
compute.power<-T

#Compute the number of subjects needed for a given power for comparison test(Yes=T,
No=F)
#----------------------------------------------------------------------------
compute.nni<-T

#Equivalence interval
interval_eq<-c(log(0.8),log(1.25))

#Compute expected power for equivalence test (Yes=T, No=F)
#---------------------------------------------------------------------------
compute.power_eq<-F

#Compute the number of subjects needed for a given power for equivalence test
(Yes=T, No=F)
#----------------------------------------------------------------------------
compute.nni_eq<-F

#Set value the given power
#---------------------------------------------------------------------------
given.power<-0.9
```

```
############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<-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(0.5,1,1.5,2,4,6,8))
#sampwinB<-sampwinA


#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

#Vector of Names on Y axes for each response
#----------------------------------
names.datax<-c("Time")

#Vector of Names on Y axes for each response
#----------------------------------
names.datay<-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.rangeA<-NULL # default range
#y.range<-c(0,10)

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

6.3. OUTPUT FILE

```
PFIM 3.2  Option 1

Project:  Example 6

Date:  Mon Jan 11 14:13:53 2010

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

Analytical function models :

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

Population design:
Sample times for response: A        Number of subjects per group  Doses
c(0.5, 2, 6, 8)                                      40      30


Random effect model: Trand =  2

Variance error model response A : ( 0.1 + 0 *f)^2


Covariate model :

         NB: Covariates are additive on log parameters

             Covariate 1 : Sex ( V )
    Categories References Proportions
(1)          M          *         0.5
(2)          F                    0.5


Computation of the Fisher information matrix: option =  1

******************* POPULATION FISHER INFORMATION MATRIX ******************

                                              s2
   342.150962 -20.4962991   3.7333850 -30.587703    0.0000000    0.0000000
   -20.496299  31.5727521   0.9836867  57.073989    0.0000000    0.0000000
     3.733385   0.9836867 109.0761038   1.237472    0.0000000    0.0000000
s2 -30.587703  57.0739890   1.2374720 199.758962    0.0000000    0.0000000
     0.000000   0.0000000   0.0000000   0.000000 1463.4668526   65.7219020
     0.000000   0.0000000   0.0000000   0.000000   65.7219020 1871.8838767
     0.000000   0.0000000   0.0000000   0.000000    0.7212424    0.6395265
     0.000000   0.0000000   0.0000000   0.000000  667.6716233  378.8205873

      0.0000000    0.00000
      0.0000000    0.00000
      0.0000000    0.00000
s2    0.0000000    0.00000
      0.7212424  667.67162
      0.6395265  378.82059
   2379.5321132   77.43697
     77.4369654 9002.93885
```

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

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

                  Beta   StdError       RSE
ka           1.0000000 0.05519603  5.519603 %
V            3.5000000 0.25949173  7.414049 %
Cl           2.0000000 0.09578998  4.789499 %
beta_V_Sex_F 0.4054651 0.10182064 25.112059 %


------------------------- Variance of Inter-Subject Random Effects ----------------
------

   Omega   StdError      RSE
ka  0.09 0.02660122 29.55691 %
V   0.09 0.02321855 25.79839 %
Cl  0.09 0.02050298 22.78109 %


------------------------ Standard deviation of residual error ---------------------
---

           Sigma   StdError      RSE
sig.interA   0.1 0.01076406 10.76406 %


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

6.130894e+21

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

528.9815

***************************** COMPARISON TEST ********************************

                  Beta       95 % CI   exp(Beta)       95 % CI
beta_V_Sex_F 0.4054651 [0.206;0.605]         1.5 [1.229;1.831]


Type I error = 0.05

             Expected_power Number_subjects_needed (for a given power=0.9)
beta_V_Sex_F       0.978421               26.50458
```

7. Example 7: PK model with two covariate effects

The purpose of this example is to evaluate a design for a PK model including two covariate effects. The PK model is a one compartment oral model with first order absorption and first order elimination. We add a gender effect and a genetic effect in three categories on the volume of distribution (V). The dose is fixed to 30 for the 40 subjects with the same sampling times at 0.5, 2, 6 and 8.

7.1. MODEL FILE

```
source(paste(directory.program,"\\","LibraryPK.r",sep=""))

formA<-oral1_1cpt_kaVCl()[[1]]

form<-c(formA)
```

7.2. INPUT FILE

```
#########################################################################
##                    INPUT FILE FOR PFIM 3.2                          ##
#########################################################################


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

project<-"Example 6 "


#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"

#Block  diagonal  Fisher  information  matrix  (option<-1)  or  complete  Information
#matrix (option<-2)
#----------------------------------------------------------
option<-1

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

nr<-1

################### 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(30)

#Vector of the times 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(100))


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

#RtolEQ<-1e-08
#AtolEQ<-1e-08
#Hmax<-0.01 # Default value
#Hmax<-Inf #<1.5/24

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


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

parameters<-c("ka","V","Cl")


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

beta<-c(1,3.5,2)


#Number of occasions
#--------------------------------------------------------------------------------
n_occ<-1


#Random effect model (1) = additive  (2) = exponential
#------------------------------------------------------------------

Trand<-2;


#Diagonal Matrix of variance for inter-subject random effects:
#---------------------------------------------------

omega<-diag(c(0.09,0.09,0.09))

#Diagonal Matrix of variance for inter-occasion random effects:
#---------------------------------------------------

gamma<-diag(c(0,0,0))

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

sig.interA<-0.1
sig.slopeA<-0
```

```
#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(0.5,2,6,8))

#Vector of initial proportions or numbers of subjects for each elementary design
#--------------------------------------------------------------

subjects<-c(40)

#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<-1000




###################################################################
#                                                                 #
#                        Covariate model                          #
#                                                                 #
###################################################################

##########################################
# Covariates not changing with occasion  #
##########################################

#Add covariate to the model  (Yes==T No==F)
#---------------------------------------------------------------------------
covariate.model<-T

#Vector of covariates
#---------------------------------------------------------------------
covariate.name<-list(c("Sex"),c("Genetics"))

#Categories for each covariate (the first category is the reference)
#-----------------------------------------------------------------------
covariate.category<-list(Sex=c("M","F"),Genetics=c("common_Hz","hz","rare_hz"))

#Proportions of subjects in each category
#-------------------------------------------------------------------------
covariate.proportions<-list(Sex=c(0.5,0.5),Genetics=c(0.5,0.25,0.25))

#Parameter(s) associated with each covariate
#-------------------------------------------------------------------------
parameter.associated<-list(Sex=c("V"),Genetics=c("V"))

# Values of covariate parameters in covariate model
# (values of parameters for all other categories than the reference category (for
which beta=0)
# covariate is additive on parameter if additive random effect model (Trand=1)
# covariate is additive on log parameters if exponential random effect model
(Trand=2)
#-----------------------------------------------------------------------
beta.covariate<-list(Sex=list(c(log(1.5))),Genetics=list(c(log(1.3),log(1.4))))



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

```
#Covariates changing with occasion  #
#####################################


#Add covariate to the model   (Yes==T No==F)
#---------------------------------------------------------------------------
covariate_occ.model<-F

#Vector of covariates depending on the occasion
#---------------------------------------------------------------------
covariate_occ.name<-list(c("Treat"))

#Categories for each covariate (the first category is the reference)
#-----------------------------------------------------------------------
covariate_occ.category<-list(
Treat=c("A","B"))

#Sequences of values of covariates at each occasion
#Specify as many values in each sequence as number of occasions (n_occ) for each
#covariate
#----------------------------------------------------------------------------------

covariate_occ.sequence<-list(
Treat=list(c("A","B"),c("B","A")))

#Proportions of elementary designs corresponding to each sequence of covariate
#values
#Specify as many values of proportion as number of sequences defined in
covariate_occ.sequence for each covariate
#----------------------------------------------------------------------------------
covariate_occ.proportions<-list(
Treat=c(0.5,0.5))

#Parameter(s) associated with each covariate
#-------------------------------------------------------------------------
parameter_occ.associated<-list(Treat=c("Cl"))


# Values of covariate parameters in covariate model
# (values of parameters for all other categories than the reference category (for
#which beta=0)
# covariate is additive on parameter if additive random effect model (Trand=1)
# covariate is additive on log parameters if exponential random effect model
#(Trand=2)
#-----------------------------------------------------------------------
beta.covariate_occ<-list(Treat=list(c(log(1.1))))


#############################################
# Power and number of subjects              #
#############################################

#Type one error alpha
#-----------------------------------------------------------------------------
alpha<-0.05

#Compute expected power for comparison test (Yes=T, No=F)
#---------------------------------------------------------------------------
compute.power<-F

#Compute the number of subjects needed for a given power for comparison test(Yes=T,
No=F)
#----------------------------------------------------------------------------
compute.nni<-F

#Equivalence interval
interval_eq<-c(log(0.8),log(1.25))
```

```
#Compute expected power for equivalence test (Yes=T, No=F)
#---------------------------------------------------------------------------
compute.power_eq<-F

#Compute the number of subjects needed for a given power for equivalence test
(Yes=T, No=F)
#----------------------------------------------------------------------------
compute.nni_eq<-F

#Set value the given power
#---------------------------------------------------------------------------
given.power<-0.9



############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<-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(0.5,1,1.5,2,4,6,8))
#sampwinB<-sampwinA


#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

#Vector of Names on Y axes for each response
#---------------------------------
names.datax<-c("Time")

#Vector of Names on Y axes for each response
#---------------------------------
names.datay<-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.rangeA<-NULL # default range
#y.range<-c(0,10)

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

7.3. OUTPUT FILE

```
PFIM 3.2  Option 1

Project:  Example 6

Date:  Mon Jan 11 14:21:07 2010

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

Analytical function models :

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

Population design:
Sample times for response: A        Number of subjects per group  Doses
c(0.5, 2, 6, 8)                                      40      30


Random effect model: Trand =  2

Variance error model response A : ( 0.1 + 0 *f)^2


Covariate model :

         NB: Covariates are additive on log parameters

             Covariate 1 : Sex ( V )
    Categories References Proportions
(1)          M          *         0.5
(2)          F                    0.5
         Covariate 2 : Genetics ( V )
    Categories References Proportions
(1)  common_Hz          *        0.50
```

```
(2)         hz                   0.25
(3)    rare_hz                   0.25


Computation of the Fisher information matrix: option =  1

******************* POPULATION FISHER INFORMATION MATRIX ******************

                                             s2          s2          s2
   341.583768 -21.126433   3.8060206 -37.902510 -18.9016959 -19.1722946
   -21.126433  31.054082   1.0187123  53.585645  26.7276369  26.7093331
     3.806021   1.018712 108.7748395   1.802899   0.8870889   0.9569525
s2 -37.902510  53.585645   1.8028994 187.549757  43.9992742  43.6710020
s2 -18.901696  26.727637   0.8870889  43.999274  93.5467291   0.0000000
s2 -19.172295  26.709333   0.9569525  43.671002   0.0000000  93.4826660
     0.000000   0.000000   0.0000000   0.000000   0.0000000   0.0000000
     0.000000   0.000000   0.0000000   0.000000   0.0000000   0.0000000
     0.000000   0.000000   0.0000000   0.000000   0.0000000   0.0000000
     0.000000   0.000000   0.0000000   0.000000   0.0000000   0.0000000

      0.0000000    0.000000    0.0000000    0.00000
      0.0000000    0.000000    0.0000000    0.00000
      0.0000000    0.000000    0.0000000    0.00000
s2    0.0000000    0.000000    0.0000000    0.00000
s2    0.0000000    0.000000    0.0000000    0.00000
s2    0.0000000    0.000000    0.0000000    0.00000
   1459.0294554   70.387310    0.7592298  661.52089
     70.3873104 1813.866409    0.6891870  411.22779
      0.7592298    0.689187 2366.4460382   88.78356
    661.5208877  411.227789   88.7835606 9081.52067


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

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

                            Beta   StdError       RSE
ka                     1.0000000 0.05530278  5.530278 %
V                      3.5000000 0.31461129  8.988894 %
Cl                     2.0000000 0.09592383  4.796191 %
beta_V_Sex_F           0.4054651 0.10267337 25.322370 %
beta_V_Genetics_hz     0.2623643 0.12603290 48.037372 %
beta_V_Genetics_rare_hz 0.3364722 0.12607617 37.470006 %


------------------------- Variance of Inter-Subject Random Effects ----------------
------

   Omega   StdError      RSE
ka  0.09 0.02663195 29.59106 %
V   0.09 0.02360907 26.23230 %
Cl  0.09 0.02056052 22.84502 %


------------------------ Standard deviation of residual error ---------------------
---

           Sigma   StdError      RSE
sig.interA   0.1 0.01072185 10.72185 %


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

2.532051e+25

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

347.0142
```

8. Example 8: PK model with inter-occasion variability and covariate effects (Equivalence test)

The purpose of this example is to evaluate a design for a crossover PK trial with two periods, two sequences: 20 subjects receive treatment A at period 1 then treatment B at period 2; 20 subjects receive treatment B at period 1 then treatment A at period 2. The PK model is a one compartment oral model with first order absorption and first order elimination. We add a gender effect which does not change with the occasion on the volume of distribution (V) and a treatment effect changing with the occasion on the clearance (Cl). The dose is fixed to 30 for the 40 subjects with the same sampling times at 0.5, 2, 6 and 8. With alpha=0.05, we then compute the expected power of the Wald test for equivalence on the interval [ln(0.8) and ln(1.25)] and the number of subjects needed for a given power of 0.9.

8.1. MODEL FILE

```
source(paste(directory.program,"\\","LibraryPK.r",sep=""))

formA<-oral1_1cpt_kaVCl()[[1]]

form<-c(formA)
```

8.2. INPUT FILE

```
###############################################################################
#                   INPUT FILE FOR PFIM 3.2                                ##
###############################################################################


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

project<-"Example 7"


#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"

#Block diagonal Fisher information matrix (option<-1) or complete Information matrix (option<-2)
#----------------------------------------------------------
option<-1

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

nr<-1

################### 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(30)

#Vector of the times 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(100))


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

RtolEQ<-1e-08
AtolEQ<-1e-08
Hmax<-0.01 # Default value
#Hmax<-Inf #<1.5/24

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


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

parameters<-c("ka","V","Cl")


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

```
beta<-c(1,3.5,2)


#Number of occasions
#--------------------------------------------------------------------------------
n_occ<-2


#Random effect model (1) = additive  (2) = exponential
#------------------------------------------------------------------

Trand<-2;


#Diagonal Matrix of variance for inter-subject random effects:
#---------------------------------------------------

omega<-diag(c(0.09,0.09,0.09))

#Diagonal Matrix of variance for inter-occasion random effects:
#---------------------------------------------------

gamma<-diag(c(0.0225,0.0225,0.0225))

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

sig.interA<-0.1
sig.slopeA<-0


#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(0.5,2,4,6,8))

#Vector of initial proportions or numbers of subjects for each elementary design
#--------------------------------------------------------------

subjects<-c(40)

#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<-1000




###################################################################
#                                                                 #
#                        Covariate model                          #
#                                                                 #
###################################################################

##########################################
# Covariates not changing with occasion  #
##########################################

#Add covariate to the model  (Yes==T No==F)
#---------------------------------------------------------------------------
covariate.model<-T
```

```
#Vector of covariates
#---------------------------------------------------------------------
covariate.name<-list(c("Sex"))

#Categories for each covariate (the first category is the reference)
#-----------------------------------------------------------------------
covariate.category<-list(Sex=c("M","F"))

#Proportions of subjects in each category
#-------------------------------------------------------------------------
covariate.proportions<-list(Sex=c(0.5,0.5))

#Parameter(s) associated with each covariate
#-------------------------------------------------------------------------
parameter.associated<-list(Sex=c("V"))

# Values of covariate parameters in covariate model
# (values of parameters for all other categories than the reference category (for
which beta=0)
# covariate is additive on parameter if additive random effect model (Trand=1)
# covariate is additive on log parameters if exponential random effect model
(Trand=2)
#-----------------------------------------------------------------------
beta.covariate<-list(Sex=list(c(log(1.2))))



#####################################
#Covariates changing with occasion  #
#####################################


#Add covariate to the model   (Yes==T No==F)
#---------------------------------------------------------------------------
covariate_occ.model<-T

#Vector of covariates depending on the occasion
#---------------------------------------------------------------------
covariate_occ.name<-list(c("Treat"))

#Categories for each covariate (the first category is the reference)
#-----------------------------------------------------------------------
covariate_occ.category<-list(
Treat=c("A","B"))

#Sequences of values of covariates at each occasion
#Specify as many values in each sequence as number of occasions (n_occc) for each
covariate
#----------------------------------------------------------------------------------
---------------------

covariate_occ.sequence<-list(
Treat=list(c("A","B"),c("B","A")))

#Proportions of elementary designs corresponding to each sequence of covariate
values
#Specify as many values of proportion as number of sequences defined in
covariate_occ.sequence for each covariate
#----------------------------------------------------------------------------------
-------------------------------
covariate_occ.proportions<-list(
Treat=c(0.5,0.5))

#Parameter(s) associated with each covariate
#-------------------------------------------------------------------------
parameter_occ.associated<-list(Treat=c("Cl"))
```

```
# Values of covariate parameters in covariate model
# (values of parameters for all other categories than the reference category (for
which beta=0)
# covariate is additive on parameter if additive random effect model (Trand=1)
# covariate is additive on log parameters if exponential random effect model
(Trand=2)
#-----------------------------------------------------------------------
beta.covariate_occ<-list(Treat=list(c(log(1.1))))


#############################################
# Power and number of subjects              #
#############################################

#Type one error alpha
#-----------------------------------------------------------------------------
alpha<-0.05

#Compute expected power for comparison test (Yes=T, No=F)
#---------------------------------------------------------------------------
compute.power<-F

#Compute the number of subjects needed for a given power for comparison test(Yes=T,
No=F)
#----------------------------------------------------------------------------
compute.nni<-F

#Equivalence interval
interval_eq<-c(log(0.8),log(1.25))

#Compute expected power for equivalence test (Yes=T, No=F)
#---------------------------------------------------------------------------
compute.power_eq<-T

#Compute the number of subjects needed for a given power for equivalence test
(Yes=T, No=F)
#----------------------------------------------------------------------------
compute.nni_eq<-T

#Set value the given power
#---------------------------------------------------------------------------
given.power<-0.9



############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<-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(0.5,1,1.5,2,4,6,8))
sampwinB<-sampwinA


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

nsampA<-list(c(5))
nsampB<-list(c(5))
```

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

nmaxptsA<-5
nmaxptsB<-5

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

nminptsA<-5
nminptsB<-5
############# END OF OPTIMISATION ALGORITHM OPTION ###############





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

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

graph.logical<-T

#Vector of Names on Y axes for each response
#---------------------------------
names.datax<-c("Time")

#Vector of Names on Y axes for each response
#---------------------------------
names.datay<-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.rangeA<-NULL # default range
#y.range<-c(0,10)

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

8.3. OUTPUT FILE

```
PFIM 3.2 Option 1

Project:  Example 8

Date:  Thu Feb 10 14:58:55 2011
```

```
**************************** INPUT SUMMARY ********************************

Analytical function models :

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

Population design:
Sample times for response: A
               times subjects doses
1 c(0.5, 2, 4, 6, 8)       40    30


Number of occasions: 2

Random effect model: Trand =  2

Variance error model response A : ( 0.1 + 0 *f)^2


Covariate model :

         NB: Covariates are additive on log parameters

         Covariates not changing with occasion

         Covariate 1 : Sex ( V )
    Categories References Proportions
(1)          M          *         0.5
(2)          F                    0.5


         Covariates changing with occasion

         Covariate  1 : Treat ( Cl )
    Categories References
(1)          A          *
(2)          B


    Sequences Proportions
(1)       A B         0.5
(2)       B A         0.5




Computation of the Fisher information matrix: option =  1


******************* POPULATION FISHER INFORMATION MATRIX ******************

                                              s2
   339.888866 -12.1244029  2.2403232 -17.9762700   1.8294873    0.0000000
   -12.124403  29.3831440  0.5085979  52.3319426   0.4561115    0.0000000
     2.240323   0.5085979 98.2790307   0.8587853  98.1435338    0.0000000
s2 -17.976270  52.3319426  0.8587853 183.1617992   0.8270805    0.0000000
     1.829487   0.4561115 98.1435338   0.8270805 953.4932901    0.0000000
     0.000000   0.0000000  0.0000000   0.0000000   0.0000000 1444.4800033
     0.000000   0.0000000  0.0000000   0.0000000   0.0000000   23.0348957
     0.000000   0.0000000  0.0000000   0.0000000   0.0000000   0.2509671
     0.000000   0.0000000  0.0000000   0.0000000   0.0000000  736.6832299
     0.000000   0.0000000  0.0000000   0.0000000   0.0000000   34.0127520
     0.000000   0.0000000  0.0000000   0.0000000   0.0000000   32.8694696
     0.000000   0.0000000  0.0000000   0.0000000   0.0000000  423.7018814

   0.0000000    0.0000000 0.000000e+00 0.000000e+00     0.00000     0.00000
   0.0000000    0.0000000 0.000000e+00 0.000000e+00     0.00000     0.00000
   0.0000000    0.0000000 0.000000e+00 0.000000e+00     0.00000     0.00000
```

```
s2 0.0000000    0.0000000 0.000000e+00 0.000000e+00     0.00000     0.00000
    0.0000000    0.0000000 0.000000e+00 0.000000e+00     0.00000     0.00000
   23.0348957    0.2509671 7.366832e+02 3.401275e+01    32.86947   423.70188
 1620.0004868    0.1586320 2.250149e+01 8.207294e+02    33.13093   269.48022
    0.1586320 1931.7539827 1.103168e-01 4.364955e-01   966.47638    18.52264
   22.5014905    0.1103168 1.214320e+04 4.704703e+03   189.35871  2594.79722
  820.7293932    0.4364955 4.704703e+03 2.268081e+04   281.84596  2441.16633
   33.1309310  966.4763757 1.893587e+02 2.818460e+02 39310.95582  1152.57041
  269.4802235   18.5226438 2.594797e+03 2.441166e+03  1152.57041 36667.05676


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

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

                      Beta   StdError       RSE
ka              1.00000000 0.05466202  5.466202 %
V               3.50000000 0.26457492  7.559283 %
Cl              2.00000000 0.10650676  5.325338 %
beta_V_Sex_F    0.18232156 0.10545703 57.841231 %
beta_Cl_Treat_B 0.09531018 0.03418960 35.871924 %


------------------------- Variance of Inter-Subject Random Effects ----------------
------

   Omega   StdError      RSE
ka  0.09 0.02678397 29.75997 %
V   0.09 0.02510621 27.89579 %
Cl  0.09 0.02289350 25.43722 %


------------------------- Variance of Inter-Occasion Random Effects ---------------
-------

    Gamma    StdError      RSE
ka 0.0225 0.009674645 42.99842 %
V  0.0225 0.007009914 31.15517 %
Cl 0.0225 0.005077443 22.56641 %


------------------------ Standard deviation of residual error ---------------------
---

         Sigma   StdError     RSE
sig.interA   0.1 0.00527779 5.27779 %


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

1.124604e+38

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

1482.234




**************************** EQUIVALENCE TEST ********************************

                      Beta       90 % CI   exp(Beta)       90 % CI
beta_V_Sex_F    0.18232156 [0.009;0.356]         1.2 [1.009;1.427]
beta_Cl_Treat_B 0.09531018 [0.039;0.152]         1.1  [1.04;1.164]
```

```
Type I error = 0.05
Equivalence interval = [log(0.8),log(1.25)]

                Expected_power Number_subjects_needed (for a given power=0.9)
beta_V_Sex_F         0.1042397             2286.08074
beta_Cl_Treat_B      0.9818745               24.50351
```