

Introduction

Background

- Nonlinear mixed effect models are increasingly used for the analysis of longitudinal studies in drug development. Using the Fisher Information Matrix (FIM) to optimize the design of these studies is an efficient alternative to clinical trial simulation
- PFIM 4.0 is a R program devoted to the design evaluation and optimisation [1]
- S4 is an object-oriented programming language which can offer several advantages compared to the traditional programming [2]

Objectives

- To increase the clarity of the different defined elements in PFIM
- To increase the modularity of PFIM, resulting in simplifying the use (manipulation of the different elements) and in facilitating implementation of new functions

The new PFIM

Methods

- As in PFIM 4.0., the model and its parameter values should be given to evaluate the FIM for a given design, or to optimise the design (Figure 1)
- The FIM is evaluated by first order linearisation of the model [3]
- Following an object-oriented approach, PFIM is implemented in R S4 language. The new PFIM can be represented as a class diagram, with PFIM as a central main object (Figure 2)

Implemented features (New features)

- Model
 - Analytic and Ordinary Differential Equation (ODE) form with one or several responses (e.g. PKPD model)
 - Error models: Constant, Proportional, Combined
 - Model parameters: possible different distributions (normal and log-normal) in the same project
- Design evaluation: Expected standard errors of parameters, FIM determinant, eigenvalues
- Design optimisation given constraints: Multiplicative algorithm [4], with possibility to optimise both sampling times and doses based on D-criterion

PFIM use

- The following objects have to be defined in a PFIM project:
 - The project name
 - The model
 - Equations of each compartment/response
 - Parameters: fixed effects (μ), standard deviation of random effects (ω) and distribution
 - The design: number of subjects, sampling times, doses, constraints in case of optimisation
- FIM computation & design evaluation/optimisation are performed
- Results can be presented in a summary
- All the elements (model, design and results) can be saved/reloaded/manipulated in R.

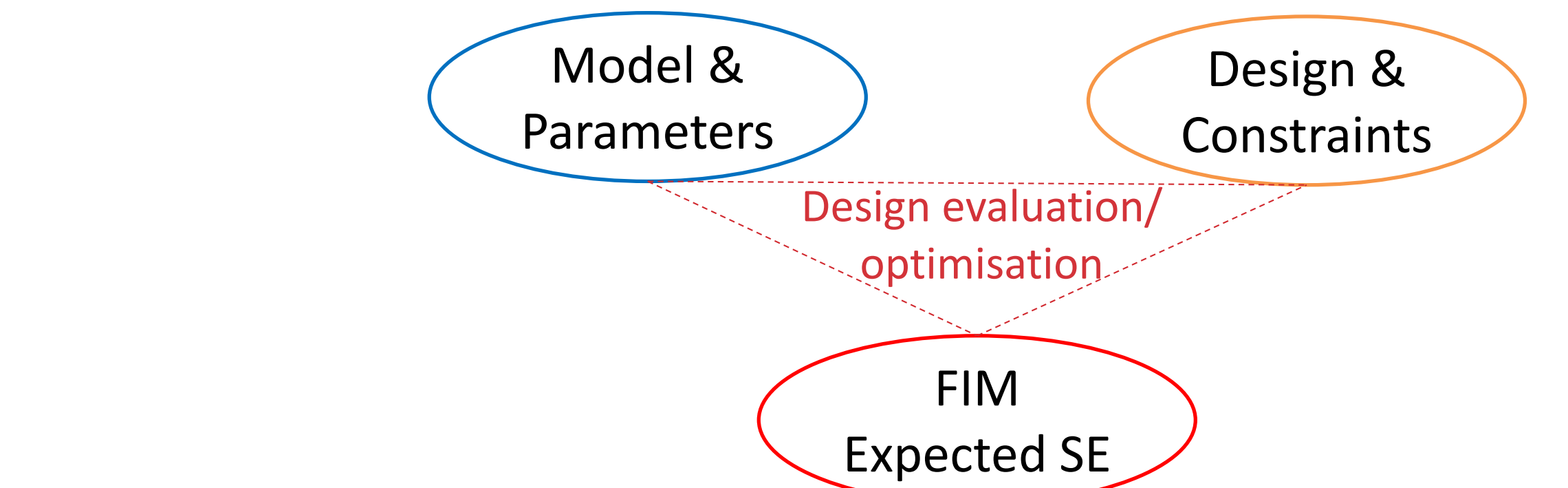


Figure 1: FIM-based optimal design

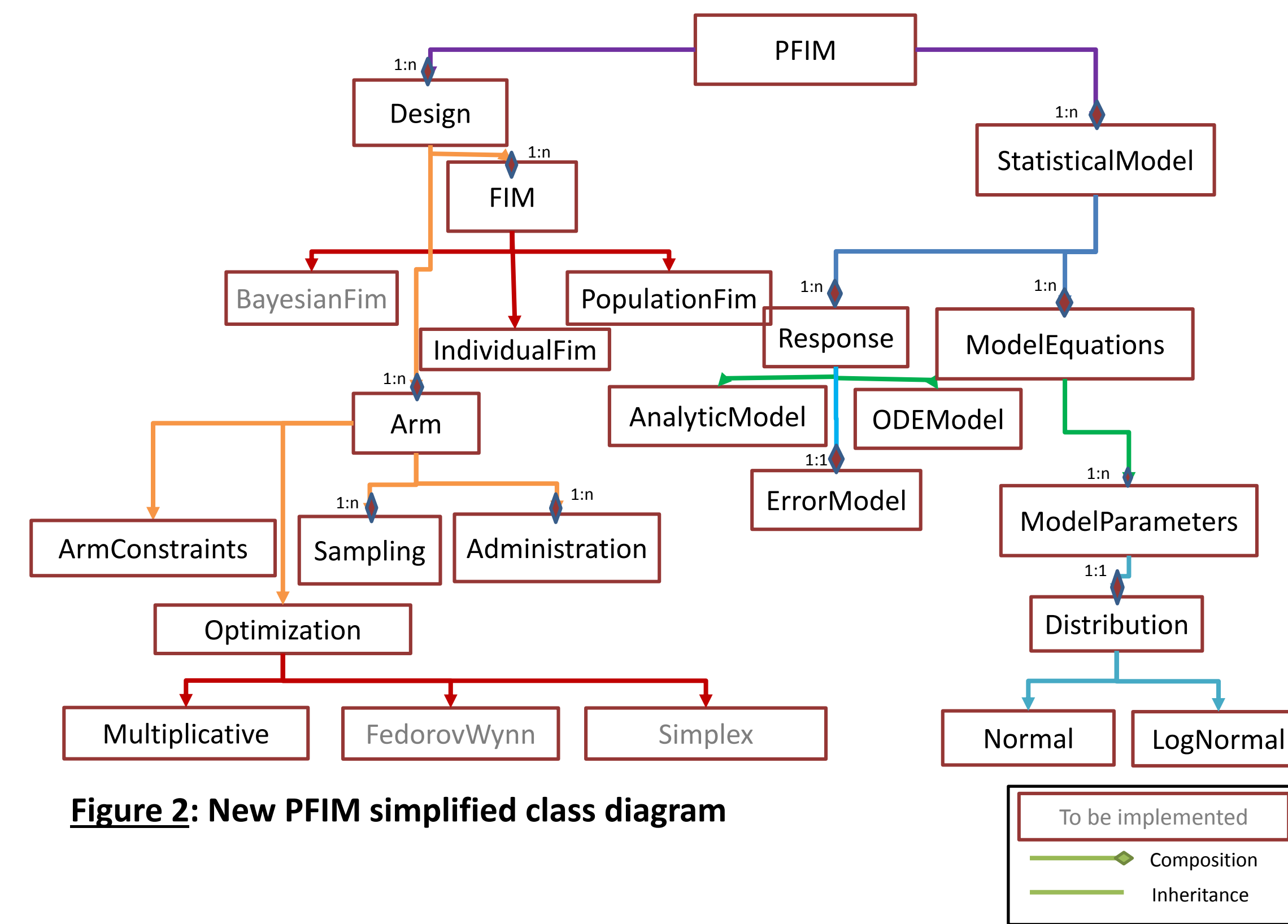


Figure 2: New PFIM simplified class diagram

Example of design evaluation for a PKPD model with the new PFIM

```
## Create a project
myProject <- PFIM(name = "Test PFIM")

## Create the ODE model
myModel <- ODEStatisticalModel()

## Create a list of model equations
myModelEquations <- ModelODEEquations( list( "Resp1" = expression( q1 ),
                                              "Resp2" = expression( q2 ) ),
                                          list( "deriv.q1" = expression( 100*ka/V * exp(-ka*t) - c1/V * q1 ),
                                              "deriv.q2" = expression( Rin * (1-(1^(q1)))/(q1<c50) -kout*q2 ) )

## Assign the equations to the model
myModel <- defineModelEquations( myModel, myModelEquations )

## Define the variables of the ode model
vQ1 <- ModelVariable( "q1" )
vQ2 <- ModelVariable( "q2" )
myModel <- defineVariable( myModel, vQ1 )
myModel <- defineVariable( myModel, vQ2 )

## Set fixed effects (mu), standard deviation of random effects (omega) and distribution of each parameter
pka <- ModelParameter( "ka", mu = 1.6,
                      omega = sqrt( 0.7 ),
                      distribution = LogNormalDistribution() )

pv <- ModelParameter( "v", mu = 8,
                    omega = sqrt( 0.02 ),
                    distribution = LogNormalDistribution() )

pc1 <- ModelParameter( "c1", mu = 0.13,
                    omega = sqrt( 0.06 ),
                    distribution = LogNormalDistribution() )

prin <- ModelParameter( "Rin", mu = 5.4,
                    omega = sqrt( 0.2 ),
                    distribution = LogNormalDistribution() )

pkout <- ModelParameter( "kout", mu = 0.06,
                    omega = sqrt( 0.02 ),
                    distribution = LogNormalDistribution() )

pc50 <- ModelParameter( "c50", mu = 1.2,
                    omega = sqrt( 0.01 ),
                    distribution = LogNormalDistribution() )

## Assign the parameters to the statistical model
myModel <- defineParameter( myModel, pka )
myModel <- defineParameter( myModel, pv )
myModel <- defineParameter( myModel, pc1 )
myModel <- defineParameter( myModel, prin )
myModel <- defineParameter( myModel, pkout )
myModel <- defineParameter( myModel, pc50 )

## error model (standard deviations)
myModel <- addResponse( myModel, Response( "Resp1", Combined( sigma_inter = 0.6, sigma_slope = 0.07 ) ) )
myModel <- addResponse( myModel, Response( "Resp2", constant( sigma_inter = 4 ) ) )

## Assign the model to the project
myProject <- defineStatisticalModel( myProject, myModel )

## Create a design
myDesign <- Design()

## For each arm create and add the sampling times for each response
arm1 <- arm( name="traz test", arm_size = 32, cond.int=list( "q1"=0, "q2"= expression(Rin/kout) ) )
arm1 <- addSampling( arm1, samplingtimes( outcome = "Resp1", sample.time = c(0.5, 1, 2, 6, 9, 12, 24, 36, 48, 72, 96, 120), initial.time = 0 ) )
arm1 <- addSampling( arm1, samplingtimes( outcome = "Resp2", sample.time = c(0, 24, 36, 48, 72, 96, 120, 144), initial.time = 0 ) )

## Add the arm to the design
myDesign <- addArm( myDesign, arm1 )

## Add the design to the project
myProject <- addDesign( myProject, myDesign )

## Evaluate the PopulationFIM
myEvaluationPop <- evaluatePopulationFIM( myProject )

#show the results of the project
show(myEvaluationPop)
showFIM(myEvaluationPop)
showDesigns(myEvaluationPop)
```

Figure 3: Example of design evaluation in new PFIM

- Example description
 - PKPD Model written as ODE (Parameters are given in Table 1)
 - Monocompartmental PK
 - Turnover/full Imax PKPD
 - Design
 - N=32
 - Oral administration, dose = 100 mg
 - 12 PK samples from 0.5 to 120 hr, 8 PD samples from 0 to 144 hr
- The complete R script is given in Figure 3

Table 1: Model parameters

Parameter	μ	ω	distribution	σ
ka (h ⁻¹)	1.6	0.84	lognormal	-
V (L)	8	0.14	lognormal	-
Cl (L.h ⁻¹)	0.13	0.24	lognormal	-
Rin (h ⁻¹)	5.4	0.45	lognormal	-
kout (h ⁻¹)	0.06	0.14	lognormal	-
C50 (mg.L ⁻¹)	1.2	0.1	lognormal	-
$\sigma_{inter,PK}$	-	-	-	0.6
$\sigma_{slope,PK}$	-	-	-	0.07
$\sigma_{inter,PK}$	-	-	-	4

μ : fixed effect
 ω : standard deviation of random effect
 σ : standard deviation of the residual error

Expected standard errors			
**** fixed effects			
	value	se	rse(%)
μ_{ka}	1.60	0.245399629	15.337477
μ_V	8.00	0.219676219	2.745953
μ_{Cl}	0.13	0.005970197	4.592460
μ_{Rin}	5.40	0.448031367	8.296877
μ_{kout}	0.06	0.002019822	3.366369
μ_{C50}	1.20	0.044647680	3.720640
**** variance of random effects			
	value	se	rse(%)
ω^*_{ka}	0.70	0.187321551	26.76022
ω^*_V	0.02	0.005953426	29.76713
ω^*_{Cl}	0.06	0.016098758	26.83126
ω^*_{Rin}	0.20	0.053003777	26.50189
ω^*_{kout}	0.02	0.008496468	42.48234
ω^*_{C50}	0.01	0.010472338	104.72338
**** error model			
	value	se	rse(%)
σ_{inter_Resp1}	0.60	0.08617462	14.36244
σ_{slope_Resp1}	0.07	0.01256800	17.95428
σ_{inter_Resp2}	4.00	0.22607882	5.65197

***** EXPECTED STANDARD ERRORS *****			
----- Fixed Effects Parameters -----			
	Beta	StdError	RSE
ka	1.60	0.245399645	15.337478 %
v	8.00	0.219676207	2.745953 %
cl	0.13	0.005970198	4.592460 %
Rin	5.40	0.448031346	8.296877 %
kout	0.06	0.002019822	3.366369 %
C50	1.20	0.044647686	3.720641 %
----- Variance of Inter-Subject Random Effects -----			
	omega2	StdError	RSE
ka	0.70	0.187321581	26.76023 %
v	0.02	0.005953426	29.76713 %
cl	0.06	0.016098760	26.83127 %
Rin	0.20	0.053003775	26.50189 %
kout	0.02	0.008496461	42.48231 %
C50	0.01	0.010472334	104.72334 %
----- Standard deviation of residual error -----			
	Sigma	StdError	RSE
sig.interA	0.60	0.08617452	14.362420 %
sig.slopeA	0.07	0.01256800	17.954291 %
sig.interB	4.00	0.22607882	5.651971 %

Figure 4: Expected standard errors with new PFIM (left) and PFIM 4.0. (right)

- The expected standard errors of parameters are consistent with those obtained with PFIM 4.0 (Figure 4)

Perspectives

- To implement features which are in PFIM 4.0.: Fedorov-Wynn and Simplex algorithms for design optimisation, Bayesian FIM to give shrinkage predictions [5], discrete covariates and Wald test power predictions [6]
- To implement new features such as alternative methods to evaluate the FIM (e.g. MC/AGQ [7]) for discrete response models and robust design optimisation accounting for model and/or parameter uncertainty [8,9]
- To submit PFIM package to CRAN
- To increase the interoperability of PFIM with estimation parameter softwares, as aimed by the DDMoRe project.