Computer Methods and Programs in Biomedicine (2004) **76**, 31—40

ELSEVIER

Computer Methods and
Programs in Biomedicine

www.intl.elsevierhealth.com/journals/cmpb

Non-linear mixed-effects pharmacokinetic/pharmacodynamic modelling in NLME using differential equations

Christoffer W. Tornøe[a,*], Henrik Agersø[a], E. Niclas Jonsson[b], Henrik Madsen[c], Henrik A. Nielsen[c]

[a] *Experimental Medicine, Ferring Pharmaceuticals A/S, Kay Fiskers Plads 11, DK-2300 Copenhagen S, Denmark*
[b] *Department of Pharmaceutical Biosciences, Division of Pharmacokinetics and Drug Therapy, Uppsala University, Box 580, S-751 23 Uppsala, Sweden*
[c] *Informatics and Mathematical Modelling, Technical University of Denmark, DK-2800 Lyngby, Denmark*

Received 5 July 2003; received in revised form 7 January 2004; accepted 7 January 2004

KEYWORDS
Non-linear mixed-effects models;
NLME;
nlmeODE;
Population analysis;
Pharmacokinetic/pharmacodynamic modelling

Summary The standard software for non-linear mixed-effect analysis of pharmacokinetic/phar-macodynamic (PK/PD) data is NONMEM while the non-linear mixed-effects package NLME is an alternative as long as the models are fairly simple. We present the nlmeODE package which combines the ordinary differential equation (ODE) solver package odesolve and the non-linear mixed effects package NLME thereby enabling the analysis of complicated systems of ODEs by non-linear mixed-effects modelling. The pharmacokinetics of the anti-asthmatic drug theophylline is used to illustrate the applicability of the nlmeODE package for population PK/PD analysis using the available data analysis tools in R for model inspection and validation. The nlmeODE package is numerically stable and provides accurate parameter estimates which are consistent with NONMEM estimates.

1. Introduction

Mixed-effects modelling is the most used method for analysis of population pharmacokinetic/pharmacodynamic (PK/PD) data. The standard software for non-linear mixed-effect analysis of PK/PD data is NONMEM [1] while the non-linear mixed-effects package NLME [2] previously has not been able to handle differential equations thereby limiting its use in PK/PD modelling. In this paper, we present the nlmeODE package, a mixed-effects modelling package in R [3] for population PK/PD analysis using ordinary differential equations (ODEs). The odesolve package [4] which can handle stiff and non-stiff systems of first-order ODE's is used in combination with the NLME package for parameter estimation in non-linear mixed-effects models. In order to investigate the possibility of increasing the numerical stability and the rate of convergence of the NLME algorithm, a gradient attribute is included by simultaneous solution to the associated sensitivity equations to be used in the Gauss-Newton non-linear least-squares optimization algorithm.

*Corresponding author.
E-mail address: christoffer.tornoe@ferring.com (C.W. Tornøe).

0169-2607/$ — see front matter © 2004 Elsevier Ireland Ltd. All rights reserved.
doi:10.1016/j.cmpb.2004.01.001

2. Computational methods and theory

The nlmeODE package is written in the S language as implemented in the statistical modelling environment R [3]. The nlmeODE package combines the odesolve package [4] with the NLME package [2] thereby enabling the use of ODE's in non-linear mixed-effects models. In the following, the computational methods for the implementation of ODE's is presented along with the theory behind non-linear mixed-effects modelling.

2.1. Ordinary differential equation solver

The odesolve package [4] provides an interface to the Fortran ODE solver lsoda (livermore solver for ordinary differential equations, with automatic algorithm selection) [5] which can be used to solve initial value problems for systems of first-order ODE's of the form

$$\frac{\mathrm{d}\boldsymbol{x}}{\mathrm{d}t} = \boldsymbol{g}(\boldsymbol{x}, t, \boldsymbol{p}), \qquad \boldsymbol{x}(t_0) = \boldsymbol{x}_0, \qquad t \geq t_0 \tag{1}$$

where $\boldsymbol{x}$ is an N-dimensional dependent-variable vector, $\boldsymbol{x}_0$ the initial conditions, $\boldsymbol{g}$ the structural model, t is the independent variable while $\boldsymbol{p}$ is an M-dimensional vector of model parameters.

The algorithm used in lsoda includes automatic selection between methods for stiff and non-stiff systems of ODE's. The definition of a stiff system is a system with two or more eigenvalues where the real parts of the eigenvalues are very different in magnitude. Adam's method is used for solving non-stiff systems of ODE's while the backwards differentiation formula (BDF) is used for stiff systems [5]. These methods are not able to handle systems with discontinuities arising from multiple doses or infusions. These problems are overcome by using the critical time argument in lsoda (`tcrit`) and by calling lsoda separately for each discontinuity so that the solver does not attempt to integrate over the discontinuity.

In order to investigate the possibility of increasing the numerical stability and the rate of convergence of the Gauss-Newton non-linear least-squares optimization algorithm in NLME, the gradient matrix $\partial\boldsymbol{x}/\partial\boldsymbol{p}$ is calculated by simultaneous solution of the sensitivity equations associated with (1). The sensitivity equations are automatically derived by differentiating (1) with respect to $\boldsymbol{p}$, i.e.

$$\frac{\mathrm{d}\boldsymbol{S}}{\mathrm{d}t} = \boldsymbol{J} \cdot \boldsymbol{S} + \frac{\partial \boldsymbol{g}}{\partial \boldsymbol{p}} \tag{2}$$

where $\boldsymbol{S}$ is the $N \times M$ sensitivity coefficient matrix $S_{ij} \equiv \partial x_i/\partial p_j$, $\boldsymbol{J}$ the $N \times N$ Jacobian matrix $J_{ij} \equiv \partial g_i/\partial x_j$, while $\partial\boldsymbol{g}/\partial\boldsymbol{p}$ is an $N \times M$ matrix of partial derivatives $\partial g_i/\partial p_j$ [6].

2.2. Non-linear mixed-effects modelling

Non-linear mixed-effects models for repeated measures can be thought of as a hierarchical model involving both fixed-effects associated with the population parameters and random-effects accounting for unexplained inter- and intra-individual variability [7,8].

At the first-stage model, the intra-individual (residual) variability describing the difference between the individual predicted values and the observations is modelled as

$$y_{ij} = f(\boldsymbol{\phi}_i, \boldsymbol{x}_{ij}) + \epsilon_{ij}, \qquad i = 1, \ldots, N, \qquad j = 1, \ldots, n_i \tag{3}$$

where y_{ij} is the jth response for the ith individual, $f(\cdot)$ a non-linear function of an individual-specific parameter vector $\boldsymbol{\phi}_i$ and predictor vector $\boldsymbol{x}_{ij}$, N the number of individuals, and n_i the number of measurements for individual i. The residual error terms ϵ_{ij} are assumed independently and identically distributed normal random variables with mean zero and variance σ^2.

At the second-stage of the hierarchy, the model relates the parameters of the different individuals, i.e.

$$\boldsymbol{\phi}_i = \boldsymbol{A}_{ij}\boldsymbol{\beta} + \boldsymbol{B}_{ij}\boldsymbol{b}_i \tag{4}$$

where $\boldsymbol{A}_{ij}$ and $\boldsymbol{B}_{ij}$ are design matrices for the fixed-effects vector $\boldsymbol{\beta}$ and random-effects vector $\boldsymbol{b}_i$, respectively. The inter-individual variability (IIV) is modelled by the random-effects vector $\boldsymbol{b}_i$ which consists of k zero-mean variables assumed to be independent and identically distributed (traditionally thought to be the multivariate normal distribution) with variance-covariance matrix $\boldsymbol{\Psi}$. The residual error terms ϵ_{ij} and $\boldsymbol{b}_i$ are assumed independent for all i and j.

The parameters in the mixed-effects model described by (3) and (4) are estimated either by maximum likelihood (ML) or by restricted maximum likelihood (REML) based on the marginal density of $\boldsymbol{y}$

$$p(\boldsymbol{y}|\boldsymbol{\beta}, \sigma^2, \boldsymbol{\Psi}) = \int p(\boldsymbol{y}|\boldsymbol{\beta}, \sigma^2, \boldsymbol{b})p(\boldsymbol{b}|\boldsymbol{\Psi})\,\mathrm{d}\boldsymbol{b} \tag{5}$$

where the conditional density of $\boldsymbol{y}$ given the random-effects $\boldsymbol{b}$ is denoted by $p(\boldsymbol{y}|\boldsymbol{\beta}, \sigma^2, \boldsymbol{b})$ while $p(\boldsymbol{b}|\boldsymbol{\Psi})$ is the marginal distribution of $\boldsymbol{b}$ [2,9]. In general, the REML approximation gives larger values for the estimates of the variance components than the ML method [9].

Since the structural model $f(\cdot)$ can be non-linear in the random-effects, the integral in (5) does generally not have a closed-form expression [2]. The method proposed by Lindstrom and Bates [10] for approximating the likelihood function in (5) is implemented in NLME. The method alternates between two steps, a penalized non-linear least-squares (PNLS) step and a linear mixed-effects (LME) step. In the PNLS step, the conditional modes of the random-effects $\boldsymbol{b}$ and the conditional estimates of the fixed effects $\boldsymbol{\beta}$ based on the current estimate of $\boldsymbol{\Psi}$ are obtained by minimizing the PNLS objective function, i.e.

$$O_{\text{PNLS}} = \sum_{i=1}^{N} \frac{||\boldsymbol{y}_i - \boldsymbol{f}_i(\boldsymbol{\beta}, \boldsymbol{b}_i)||^2}{\sigma^2} + \boldsymbol{b}_i^{\mathrm{T}} \boldsymbol{\Psi}^{-1} \boldsymbol{b}_i \tag{6}$$

where $[\boldsymbol{f}_i(\boldsymbol{\beta}, \boldsymbol{b}_i)]_j = f(\boldsymbol{\phi}_i, \boldsymbol{x}_{ij})$ with $i = 1, \ldots, N$ and $j = 1, \ldots, n_i$ [9].

In order to update the estimate of $\boldsymbol{\Psi}$, the model function $\boldsymbol{f}(\cdot)$ is linearized in the LME step using a first-order Taylor expansion around the current estimates of $\boldsymbol{\beta}$ and the conditional modes of the random effects $\boldsymbol{b}$. The approximative log-likelihood function for the estimation of $\boldsymbol{\Psi}$ in the LME step can thereby be written as [9]

$$\begin{aligned}
&\log L_{\text{LME}}(\boldsymbol{\beta}, \sigma^2, \boldsymbol{\Psi}) \\
&\quad = -\frac{M}{2} \log 2\pi\sigma^2 \\
&\qquad - \frac{1}{2} \sum_{i=1}^{N} \left\{ \log \left| \left(\sigma^2 \boldsymbol{I} + \frac{\partial \boldsymbol{f}_i}{\partial \boldsymbol{\beta}^{\mathrm{T}}} \boldsymbol{\Psi} \frac{\partial \boldsymbol{f}_i}{\partial \boldsymbol{\beta}^{\mathrm{T}}}^{\mathrm{T}} \right) \right| \right. \\
&\qquad + \left[\boldsymbol{y}_i - \boldsymbol{f}_i(\boldsymbol{\beta}, \boldsymbol{b}_i) + \frac{\partial \boldsymbol{f}_i}{\partial \boldsymbol{b}_i^{\mathrm{T}}} \hat{\boldsymbol{b}}_i \right]^{\mathrm{T}} \\
&\qquad \times \left(\sigma^2 \boldsymbol{I} + \frac{\partial \boldsymbol{f}_i}{\partial \boldsymbol{\beta}^{\mathrm{T}}} \boldsymbol{\Psi} \frac{\partial \boldsymbol{f}_i}{\partial \boldsymbol{\beta}^{\mathrm{T}}}^{\mathrm{T}} \right)^{-1} \\
&\qquad \left. \times \left[\boldsymbol{y}_i - \boldsymbol{f}_i(\boldsymbol{\beta}, \boldsymbol{b}_i) + \frac{\partial \boldsymbol{f}_i}{\partial \boldsymbol{b}_i^{\mathrm{T}}} \hat{\boldsymbol{b}}_i \right] \right\}
\end{aligned} \tag{7}$$

where $M = \sum_{i=1}^{N} n_i$ is the total number of observations for the N individuals and $\boldsymbol{I}$ the identity matrix. The log-likelihood function in (7) is identical to that of a linear mixed-effects model [11].

3. Program description

The current version of nlmeODE is 0.2-3 and is compatible with R version 1.8.1, odesolve version 0.5-8 and NLME version 3.1-45. The basic idea of nlmeODE is to use the same notation as NONMEM [1], the *de facto* standard software for non-linear mixed-effects analysis of PK/PD data. The commands and R objects are printed in monospaced typewriter font `like this`.

The data object is constructed using the `groupedData` function by attaching a formula as an attribute of the data along with labels and units for default plots, i.e.

```
> data <- groupedData (formula, data,
labels, units)
```

The formula should as a minimum specify which columns in the data object is the dependent variable (e.g. `Conc`), independent variable (e.g. `Time`), and grouping factor (e.g. `Subject`). Optional columns in the data object are the dose (`Dose`), dosing compartment (`Cmt`), rate of infusion (`Rate`), covariate measurements, etc.

The models which the data should be analyzed with are entered into a list, i.e.

```
> model <- list(DiffEq, ObsEq, Parms,
States,Init)
```

where the elements in the model object are explained below.

`DiffEq` A list of formulas containing the ODE's for the system in the same order as the compartment numbers, i.e. formula *i* is the ODE for compartment *i*.

`ObsEq` A list of formulas specifying which states are observed along with possible scaling parameters.

`Parms` A vector with the names of the parameters used in `DiffEq`, `ObsEq`, and `Init`.

`States` A vector with the names of the states in `DiffEq`.

`Init` A list with the same length as `States` specifying the initial states of the system.

The model function `ODEmodel` which can be used to fit non-linear mixed-effects models in NLME is created by calling `nlmeODE` with the model and data object as arguments, i.e.

```
> ODEmodel <- nlmeODE(model,data,...)
```

The reason for creating `ODEmodel` using `nlmeODE` is to make different internally created objects (e.g. dosing information) available in the NLME call using the lexical scoping rules in *R* (see [12] for further details). Furthermore, the arguments in `ODEmodel` are required to be of the same length as the dependent variable in NLME which makes this work around necessary in order to be able to specify the model object with which the data should be analyzed. The optional arguments for nlmeODE are

`LogParms` If TRUE, the parameters are reparameterized in terms of the logarithm of the parameters (Default is TRUE).

`JAC` If TRUE, the Jacobian of the system of ODE’s is computed and passed to the ODE solver. In some circumstances, supplying the Jacobian can speed up the computations if the system is stiff (Default is TRUE).

`SEQ` A logical value, that when TRUE, adds a gradient attribute to the returned value calculated by simultaneous solution of the sensitivity equations associated with the system of ODE’s (Default is FALSE).

`rtol` Relative error tolerance for lsoda (Default is 0.01).

`atol` Absolute error tolerance for lsoda (Default is 0.01).

`tcrit` Time beyond which the integration should not proceed (Default is NULL).

`hmin` Minimum value of the integration stepsize (Default is 0).

`hmax` Maximum value of the integration stepsize (Default is infinity).

A typical NLME call looks like

```
> model.fit <- nlme(Conc ~
  ODEmodel(Parms,Time,Subject),
data, fixed, random, start, ...)
```

where `Conc` is the dependent variable name, `Parms` are the model parameters in the same order as in the model object, `Time` is the independent variable name, and `Subject` is the grouping factor name where all of the variable names are to be evaluated in the `groupedData` object `data`. The `fixed` argument includes the fixed-effects parameters to be estimated, `random` specifies which of the parameters in `fixed` should have an associated random effect, `start` are the starting estimates of the fixed-effects parameters, and ‘‘...’’ are optional arguments for NLME.

4. Sample of typical program run

The purpose of this section is to show a typical program run using the nlmeODE package and to show the available data analysis tools in *R* for mixed-effects model building, inspection, and validation. The R code for the graphs in this paper is shown in Appendix A.

The pharmacokinetic (PK) data of the anti-asthmatic drug theophylline is used for showing a typical nlmeODE run since these data have been extensively reported and analyzed in [1,2,13]. The drug was administered orally to twelve subjects and serum concentrations were measured at eleven time points per subject.

4.1. Model building

The theophylline data which is distributed with NLME and NONMEM is converted to a `groupedData` object by calling

```
> Theoph <- groupedData(conc ~
  Time|Subject,
data = read. table (‘‘Theoph. dat’’,
  header = TRUE),
labels = list (x = ‘‘Time since drug
  administration’’,
y = ‘‘Theophylline serum
  concentration’’),
units = list(x = ‘‘(hr)’’,
  y = ‘‘(mg/L)’’) )
```

The pharmacokinetics of theophylline is modelled using a one-compartment open model with first-order absorption and elimination. The model object `OneComp` used for analyzing the theophylline data is created by

```
> OneComp <- list(
DiffEq = list(dy1dt = ~ -ka*y1,
  dy2dt = ~ ka*y1-ke*y2),
ObsEq = list(SC ~ 0, Cp ~ y2/CL*ke),
Parms = c(‘‘ka’’,’’ke’’,’’CL’’),
States = c(‘‘y1’’,’’y2’’),
Init = list(0,0))
```

There are two differential equations in the `OneComp` model object, `dy1dt` and `dy2dt`, representing the absorption and serum compartments, respectively. Since it is only the theophylline serum concentration which is measured, the observation equation is zero for compartment 1 while that of compartment 2 is equal to `y2` divided by the serum volume `V = CL/ke`. The parameters in `OneComp` are the first-order absorption rate constant `ka`, the first-order elimination rate constant `ke`, and the clearance `CL`. The states in `DiffEq` are `y1` and `y2` and the initial states are set to zero in the `Init` argument since there is no theophylline in the two compartments before drug administration.

The model function to be used with NLME is created by calling `nlmeODE` with the model object

OneComp and the groupedData object Theoph as arguments, i.e.

```
> TheophModel <- nlmeODE(OneComp,
  Theoph,
LogParms = TRUE, JAC = TRUE, SEQ =
  FALSE, rtol = 0.01, atol = 0.01)
```

where the parameters are reparameterized in terms of the logarithm of the parameters in order to ensure non-negative parameter estimates while keeping the optimization problem unconstrained. The Jacobian matrix JAC is automatically computed using the deriv function and passed to the ODE solver due to the stiffness of the system while the sensitivity equations SEQ are not included since they do not increase the numerical stability or the rate of convergence for the particular problem. The relative and absolute error tolerance for lsoda are set to 0.01.

The NLME call is executed by

```
> Theoph1.nlme <- nlme(
conc ~ TheophModel(ka,ke,CL,Time,
  Subject),
data = Theoph, fixed = ka+ke+CL~1,
  random = pdDiag(ka+ke+CL~1),
start = c(ka = log(1.65),
  ke = log(0.08),CL = log(0.05)),
control = list(returnObject = T,
  msVerbose = T), verbose = T)
```

Each parameter in the model has an associated fixed effect. The argument fixed is a formula defining the structures of the fixed effects in the model. The right-hand side of fixed is 1 which indicates that a single parameter is associated with each fixed effect. If a fixed-effects parameter should depend linearly on a covariate, the covariate should be included on the right-hand side of the fixed formula for the particular parameter. The fixed-effects parameters may or may not have an associated random effect. If the random formula is omitted, all fixed effects are by default assumed to have an associated random effect with a general positive-definite variance-covariance matrix $\boldsymbol{\Psi}$ with Log-Cholesky parametrization (pdLogChol). In this case, the random effects for ka, ke, and CL are assumed independent of each other by specifying a positive-definite diagonal variance covariance matrix using the pdDiag function. The NLME package provides several other classes of positive-definite variance-covariance matrices for the random effects (see [2] for further details on how to construct your own pdMat functions). The start argument provides a list of starting estimates for NLME for the fixed effects only. The control option returnObject is set to TRUE to get the fitted object returned even if the maximum number of iterations is reached without convergence of the algorithm. The msVerbose and verbose arguments enable the user to trace the convergence of the iterative NLME algorithm. The returned object is of class NLME for which several methods are available for examining the fitted model. Detailed information about the fit is displayed using the summary method (see below).

The standard deviation for the random effect associated with ke is very small compared to the fixed effect and is therefore excluded from the model. The estimated covariances between the fixed-effects parameter estimates in the model are investigated using the vcov function (see below). The covariances between the parameters are very small indicating that the parameters are all identifiable.

```
> summary(Theoph1.nlme)
Random effects: list(ka ~ 1, ke ~ 1, CL ~ 1)
                                ka            ke          CL     Residual
StdDev                       0.667      1.84E-05       0.166        0.712
Fixed effects: ka + ke + CL ~ 1
                             Value    Std. Error     t-value      P-value
ka                           0.490         0.200        2.45         0.02
ke                           -2.46        0.0471       -52.3         0.00
CL                           -3.23        0.0588       -55.0         0.00
Correlation
                                ka            ke
ke                          -0.096
CL                          -0.029         0.518
```

```
> vcov(Theoph1.nlme)
          ka          ke          CL
ka    0.0391     -0.0009     -0.0003
ke   -0.0009      0.0022      0.0014
CL   -0.0003      0.0014      0.0034
```

The Theoph1.nlme model is revised to only include random effects associated with ka and CL by using the update function, i.e.

```
> Theoph2.nlme <- update(Theoph1.
nlme, random = pdDiag(ka+CL~1))
```

Most PK/PD data follow a proportional (heteroscedastic) residual error model but when the observations are near the lower limit of quantification of the assay, it is often useful to model the residual error using a combined additive (homoscedastic) and proportional error model [1]. Extended residual variance models can be implemented using the varFunc class objects in NLME. The constant plus power variance model varConstPower is defined as

$$\mathrm{Var}(\epsilon_{ij}) = \sigma^2(\delta_1 + |f(\boldsymbol{\phi}_i, \boldsymbol{x}_{ij})|^{\delta_2})^2 \tag{8}$$

which has the additive and proportional error models as special cases (i.e. $\delta_2 = 0$ for an additive and $\delta_1 = 0 \wedge \delta_2 = 1$ for a proportional error model) and smoothly interpolates between them. More complicated residual variance models for e.g. multilevel models with different types of measurements are available in the varFunc class functions in NLME and it is furthermore possible to specify user-defined variance functions.

The standardized residuals of Theoph2.nlme seems to indicate increasing residual error variability with increasing theophylline serum concentrations. The Theoph2.nlme model is updated using the varConstPower function by

```
> Theoph3.nlme <- update(Theoph2.nlme,
weights = varConstPower(const = 0.7,
  power = 0.3))
```

The variance parameter estimates of Theoph3.nlme using the intervals function indicate that the power variance estimate power is insignificant on a 5% level.

```
> intervals(Theoph3.nlme)
```

```
           lower    estimate    upper
const      0.219       0.593    1.606
power     -0.290       0.188    0.665
```

The significance of the variance parameters const and power is tested by comparing the likelihood ratio statistics of Theoph2.nlme and Theoph3.nlme which confirms the joint significance of power and const. The assumption of homoscedasticity (power = 0) can therefore be rejected.

```
> anova(Theoph2.nlme,Theoph3.nlme)
```

```
             df logLik L.Ratio p-value
Theoph2.nlme  6   -191
Theoph3.nlme  8   -183    15.6   4E-04
```

It is furthermore possible to include covariates in the model in order to explain some of the inter-individual variability. The estimated individual parameters from the fitted model are extracted using the coef function with the augFrame option.

```
> Theoph3.coef <- coef(Theoph3.nlme,
augFrame = T)
```

and the scatter plot of the weight covariate Wt versus the estimated individual parameters is shown in Fig. 1.

The clearance CL of theophylline seems to decrease with increasing weight Wt in Fig. 1 while the opposite is true for the relationship between the absorption rate constant ka and Wt. Since the dose is weight-adjusted, the weight is included as a covariate on the scaling parameter CL rather than on ka by

```
> Theoph4.nlme <- update(Theoph3.nlme,
  fixed = list(ka+ke~1,CL~Wt),
start = c(ka = 0.5, ke = -2.5, CL = -3.2,
  Wt = -0.005))
```

where CL is the intercept and Wt is the slope parameter of the CL and Wt relationship.

The significance of the fixed-effects slope parameter CL.Wt is tested by

```
> anova(Theoph4.nlme,Terms=
  c(‘‘CL.Wt’’))
F-test for: CL.Wt
```

```
  numDF   denDF   F-value    P-value
  1       117     1.907204   0.1699
```

The F-test shows that the added term CL.Wt is insignificant and the weight covariate is therefore left out of the final model. The iterative procedure of building a PK model for the theophylline data is hereby terminated with Theoph3.nlme being the final model.

4.2. Model validation

The inspection of the final Theoph3.nlme model is based on goodness-of-fit (GOF) graphical analysis of residuals and predictions, distribution of the residual errors and random-effects parameters. When examining a fitted model, the two main assumptions which should be tested are that (1) the normalized residuals are independent and identically normally distributed with mean zero and variance σ^2, and (2) the random effects should be normally distributed with mean zero and covariance matrix $\boldsymbol{\Psi}$ and independent of the residual errors [14].

The augmented predictions of the fitted model can be obtained using the augPred method which produces a data frame containing the population

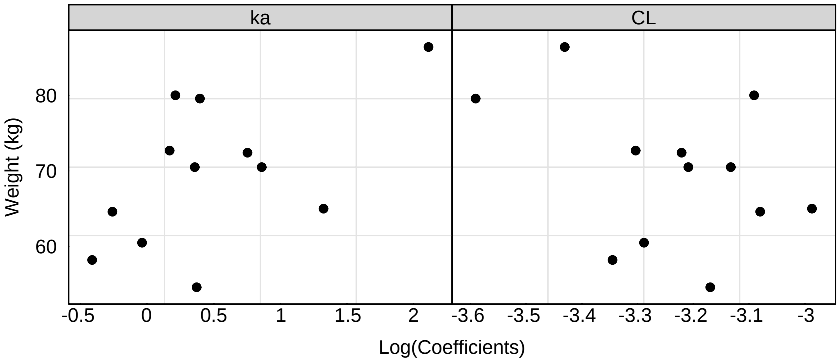

Fig. 1 Scatter plot of the weight covariate Wt vs. the estimated individual parameters (left) ka and (right) CL.

predictions (obtained by setting the random effects to zero), the individual predictions (using the estimated random effects) along with the observed data (see Fig. 2). The fitted model seems to capture the PK of theophylline nicely since the individual predictions follows the observations.

The basic GOF graph of the observed data versus the population and individual predictions in Fig. 3 indicates good agreement between the observed serum theophylline concentrations and the model predictions since the dots are nicely scattered around the line of identity.

The residual errors are tested next. The plot of normalized residuals versus fitted values for Theoph3.nlme is shown in Fig. 4 (left) where the observations with absolute normalized residuals greater than the 0.975 quantile of the standard normal distribution are identified with the subject number. The ''S'' shaped curve in the quantile-quantile ($Q-Q$) plot (Fig. 4 middle) im-

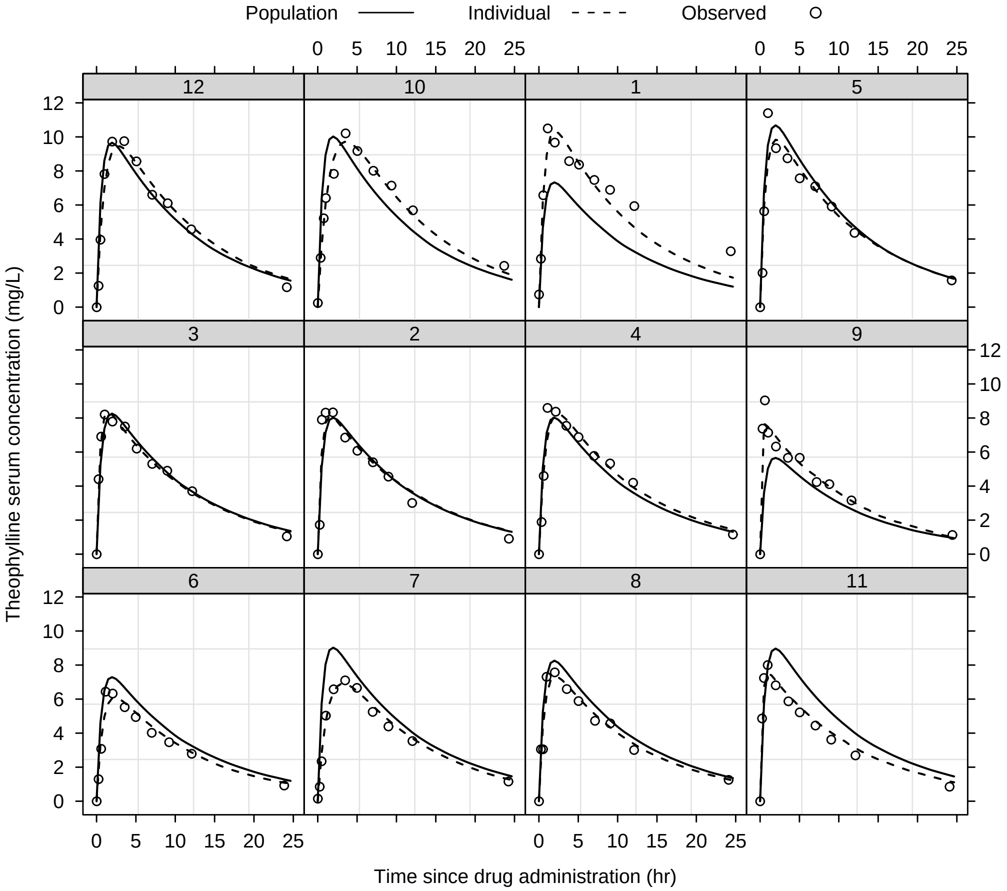

Fig. 2 Observed and predicted theophylline serum concentrations plotted vs. time. Population predictions (solid line), individual predictions (dashed line), and observed theophylline concentrations (circles).

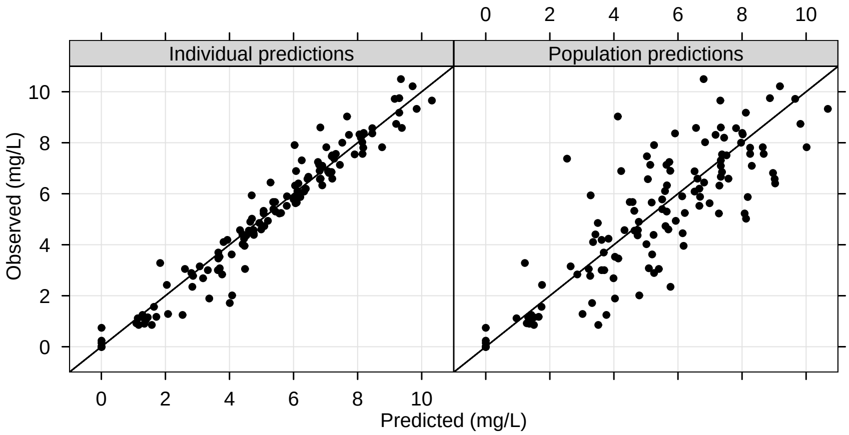

Fig. 3 Basic goodness-of-fit graph for `Theoph3.nlme`. (Left) Plot of observed vs. individual predicted and (right) observed vs. population predicted serum theophylline concentrations. The solid lines are the lines of identity.

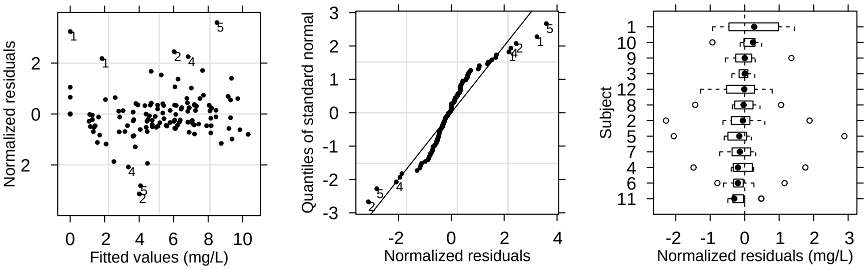

Fig. 4 Residual error plots for `Theoph3.nlme`. (Left) Normalized residuals vs. fitted values. (Middle) Q—Q plot of normalized residuals. (Right) Box plot of subject vs. normalized residuals.

plies that the normalized residual error distribution has longer tails than the Gaussian. The box plot in Fig. 4 (right) shows the quartiles of the normalized residual errors for each subject where the dot inside the box represents the median while the extreme residual errors are highlighted as circles. The residual error plots in Fig. 4 show that the normalized residuals are distributed symmetrically around zero with an approximative constant variance. The normalized residuals are not perfectly normal distributed but no serious violations of the residual error assumptions seem to be present.

The autocorrelation function (ACF) is a useful tool for investigating the correlation structure of the residual errors. The empirical ACF of the normalized residuals of `Theoph3.nlme` can be assessed using the `ACF` function (see Fig. 5). The reason for only calculating the ACF for the first 8 lags is because only eleven theophylline concentration measures are available from each subject in the `Theoph` data. When investigating the ACF in Fig. 5, no significant autocorrelation correlation seems too be present indicating that the normalized residuals can be considered as white noise. In case of serially correlated residual errors, the `corStruct` class objects for residual error correlation structures can be called with the `corr` argument. Implemented in NLME are 10 standard classes of correlation structures including autocorrelation-moving average structures (e.g. `corAR1`, `corCAR1`, `corARMA`) along with some spatial correlation structures.

Finally, the assumptions on the random effects are assessed by plotting the quantiles of the standard normal distribution versus the random-effects parameters in Fig. 6. The plot does not indicate any serious violations of the assumption of normality for the random effects.

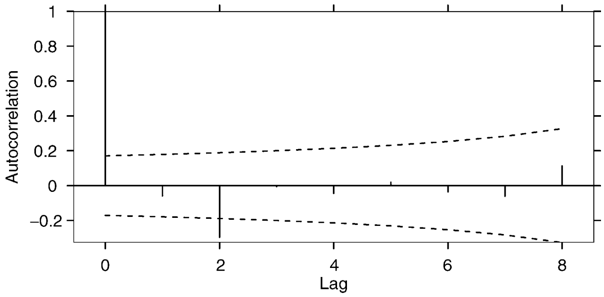

Fig. 5 Empirical autocorrelation function of the normalized residuals of `Theoph3.nlme` for lag 0—8 along with the 95% confidence interval (dotted line) under the hypothesis that the normalized residuals are white noise.

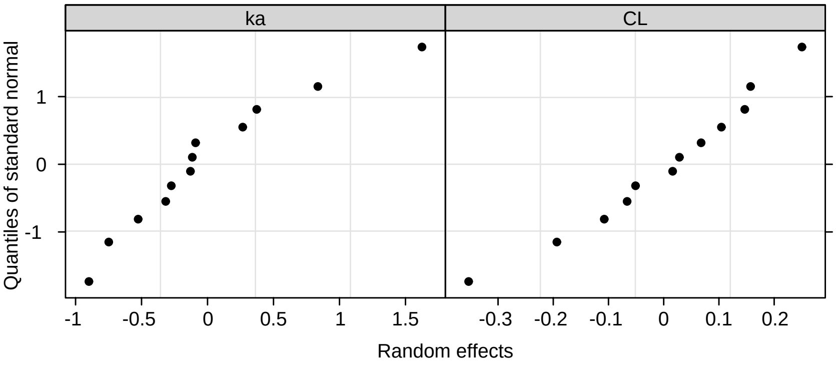

Fig. 6 *Q*—*Q* plot of the random effects for `Theoph3.nlme`.

5. Discussion

The nlmeODE package provides accurate parameter estimates which are consistent with the results obtained using NONMEM and the `SSfol` function in NLME (not shown) [1,2,13]. It is numerical stable with and without simultaneous solution of the associated sensitivity equations which only seems to reduce the number of iterations but not the estimation time.

The implementation of covariate relationships as well as correlation and variance structures in NLME is straightforward to implement by updating previously fitted models. Model validation of a fitted non-linear mixed-effects model is made feasible with the available data analysis tools in *R* as shown in Section 4.2.

The odesolve package is only available in *R* and not S-PLUS, but the plan is to port it along with the nlmeODE package to S-PLUS and make it compatible with Xpose [15], a model building aid for population PK/PD modelling.

In summary, we have demonstrated the use of the nlmeODE package together with NLME and odesolve for non-linear mixed-effects PK/PD modelling. The implementation of ODE’s in the non-linear mixed-effects library NLME makes it a promising tool for population PK/PD analysis of complicated systems of ODE’s with and without a closed-form solution. The nlmeODE package enables the analysis of multiple dose and infusion PK studies as well as simultaneous or sequential analysis of PD data by e.g. turnover models. The nlmeODE package is free of charge and can be obtained on the internet addresses http://www.cran.r-project.org and http://nlmeODE.sourceforge.net.

Acknowledgements

This work was sponsored by Ferring Pharmaceuticals A/S, Denmark, the Swedish foundation for strategic research, Sweden, and Center for Information Technology, Denmark. The help and support of R Development Core Team member Peter Dalgaard and the odesolve package author R. Woodrow Setzer Jr., with the development of the nlmeODE package is greatly appreciated.

Appendix A. R code

The code for creating the graphs in this paper is shown below.

Figure	Code
1	`> plot(Theoph3.coef, form = ~Wt)`
2	`> plot(augPred(Theoph3.nlme, level = 0:1))`
3	`> plot(Theoph3.nlme, conc~fitted(.,0), abline = c(0,1))`
3	`> plot(Theoph3.nlme, conc~fitted(.,1), abline = c(0,1))`
4	`> plot(Theoph3.nlme, resid(.,1,type = ‘‘n’’)~fitted(.,1), id = 0.05, abline = 0)`
4	`> qqnorm(Theoph3.nlme, ~resid(.,1,type = ‘‘n’’), id = 0.05)`
4	`> plot(Theoph3.nlme, Subject~resid(.,1,type = ‘‘n’’), abline = 0)`
5	`> plot(ACF(Theoph3.nlme, maxLag = 8), alpha = 0.05)`
6	`> qqnorm(Theoph3.nlme, ~ranef(.))`

References

[1] S.L. Beal, L.B. Sheiner (Eds.). NONMEM User's Guides. NONMEM Project Group, University of California, San Francisco, 1994.

[2] J.C. Pinheiro, D.M. Bates, Mixed-Effects Models in S and S-PLUS. Springer-Verlag, New York, 2000.

[3] R. Ihaka, R. Gentleman, A language for data analysis and graphics, J. Comput. Graph. Stat. 5 (3) (1996) 299—314.

[4] R.W. Setzer, The odesolve Package. Solvers for Ordinary Differential Equations, http://www.cran.r-project.org.

[5] L.R. Petzold, Automatic selection of methods for solving stiff and and nonstiff systems of ordinary differential equations, Siam J. Sci. Stat. Comput. 4 (1983) 136—148.

[6] J.R. Leis, M.A. Kramer, The simultaneous solution and sensitivity analysis of systems described by ordinary differential equations, ACM Trans. Math. Software 14 (1) (1988) 45—60.

[7] A. Racine-Poon, J. Wakefield, Statistical methods for population pharmacokinetic modelling, Stat. Methods Med. Res. 7 (1) (1998) 63—84.

[8] L. Sheiner, J. Wakefield, Population modelling in drug development, Stat. Methods Med. Res. 8 (3) (1999) 183—193.

[9] J.C. Pinheiro, D.M. Bates, Approximations to the loglikelihood function in the nonlinear mixed effects model, J. Comput. Graph. Stat. 4 (1) (1995) 12—35.

[10] M.J. Lindstrom, D.M. Bates, Nonlinear mixed effects models for repeated measures data, Biometrics 46 (3) (1990) 673—687.

[11] M.J. Lindstrom, D.M. Bates, Newton—Raphson and EM algorithms for linear mixed-effects models for repeated-measures data, J. Am. Stat. Assoc. 83 (404) (1988) 1014—1022.

[12] J. Fox, G. Monette, An R and S-Plus Companion to Applied Regression, Sage, Thousands Oaks, CA, 2002.

[13] M. Davidian, D.M. Giltinan, Nonlinear Models for Repeated Measurement Data, Chapman & Hall, London, 1995.

[14] M.O. Karlsson, E.N. Jonsson, C.G. Wiltse, J.R. Wade, Assumption testing in population pharmacokinetic models: illustrated with an analysis of moxonidine data from congestive heart failure patients, J. Pharmacokinet. Biopharm. 26 (2) (1998) 207—246.

[15] E.N. Jonsson, M.O. Karlsson, Xpose—an S-PLUS based population pharmacokinetic/pharmacodynamic model building aid for NONMEM, Comput. Methods Programs Biomed. 58 (1) (1999) 51—64.