User guide for npde v1.2

November 2007

Emmanuelle Comets, Karl Brendel [1]***, France Mentré***

INSERM, U738, Paris, France

Université Paris 7, UFR de Médecine site Bichat, Paris, France.

npde website: www.npde.biostat.fr

[1]*During this work, Karl Brendel was supported by a grant from the Institut de recherches internationales Servier, Courbevoie, France*

Contents

1 Introduction

This is the User's guide for the add-on package npde (version 1.2, November 2007) for the R language. npde computes normalised prediction distribution errors, a metric to evaluate nonlinear mixed-effect models such as those used in population pharmacokinetic or pharmacodynamic studies. Prediction distribution errors were developed under the name prediction discrepancies by Mentré and Escolano [1]. Brendel et al [2] proposed an improved version taking into account repeated observations within each subject, which were called prediction distribution errors.

The program is an add-on package (or library) running under the R software [3]. Please install and launch R to use npde.

In section 2, we describe briefly the method referenced in [2]. Details concerning the context and the methods can be found in this publication. In section 3, we describe how to use the program to compute, plot and test prediction distribution errors. Finally, in section 4, we run an example (included in the package).

1.1 Citing npde

If you use this program in a scientific publication, we would like you to cite the following reference:

> Brendel K, Comets E, Laffont CM, Laveille C, Mentré F. Metrics for external model evaluation with an application to the population pharmacokinetics of gliclazide. *Pharmaceutical Research* 2006, 23:2036–2049.

A BibTeX entry for LaTeX users is:

```
@Article{,
author ={Karl Brendel and Emmanuelle Comets and C{\'e}line Laffont and
Christian Laveille and France Mentr{\'e}},
title ={Metrics for external model evaluation with an application to the
population pharmacokinetics of gliclazide},
volume ={23},
pages ={2036--49},
journal ={Pharmaceutical Research},
year =2006 }
```

1.2 License

npde is a software distributed under the terms of the GNU GENERAL PUBLIC LICENSE Version 2, June 1991. The terms of this license are in a file called COPYING which you should have received with this software.

If you have not received a copy of this file, you can obtain one via the world wide web at http://www.gnu.org/copyleft/gpl.html, or by writing to:

The Free Software Foundation, Inc.,
51 Franklin Street, Fifth Floor, Boston, MA 02110-1301, USA.

2 Prediction distribution errors

2.1 Models and notations

Let B denote a learning dataset and V a validation dataset. B is used to build a population pharmacokinetic model called M^B.

Let i denote the i^{th} individual (i = 1,..., N) and j the j^{th} measurement in an individual (j = 1,..., n_i, where n_i is the number of observations for subject i). Let Y_i be the n_i-vector of observations observed in individual i. Let the function f denote the nonlinear structural model. f can represent for instance the pharmacokinetic model. The statistical model for the observation y_{ij} in patient i at time t_{ij}, is given by:

$$y_{ij} = f(t_{ij}, \theta_i) + \varepsilon_{ij} \tag{1}$$

where θ_i is the p-vector of the individual parameters and ε_{ij} is the residual error, which is assumed to be normal, with zero mean. The variance of ε_{ij} may depend on the predicted concentrations $f(t_{ij}, \theta_i)$ through a (known) variance model. Let σ denote the vector of unknown parameters of this variance model.

In pharmacokinetic (PK)/pharmacodynamic (PD) studies for instance, it is usually assumed that the variance of the error follows a combined error model:

$$\text{var}(\varepsilon_{ij}) = (\sigma_{\text{inter}} + \sigma_{\text{slope}}\ f(t_{ij}, \theta_i))^2 \tag{2}$$

where σ_{inter} and σ_{slope} are two parameters characterising the variance. In this case, $\sigma = (\sigma_{\text{inter}}, \sigma_{\text{slope}})'$. This combined variance model covers the case of an homoscedastic variance error model, where $\sigma_{\text{slope}} = 0$, and the case of a constant coefficient of variation error model when $\sigma_{\text{inter}} = 0$. Another parameterisation often found is:

$$\text{var}(\varepsilon_{ij}) = \sigma^2_{\text{inter}} + \sigma^2_{\text{slope}}\ f(t_{ij}, \theta_i)^2 \tag{3}$$

Another usual assumption in population PK/PD analyses is that the distribution of the individual parameters θ_i follows a normal distribution, or a log-normal distribution, as in:

$$\theta_i = h(\mu, X_i)\, e^{\eta_i} \tag{4}$$

where μ is the population vector of the parameters, X_i a vector of covariates, h is a function giving the expected value of the parameters depending on the covariates, and η_i represents the vector of random effects in individual i. η_i usually follows a normal distribution $\mathcal{N}(0,\Omega)$, where Ω is the variance-covariance matrix of the random effects, but other parametric or non-parametric assumptions can be used for the distribution of the random effects, as in the first paper using this method, in the context of non-parametric estimation [4]. Although npde were developed in the context of pharmacokinetic and pharmacodynamic analyses, it is a general approach that can be used to evaluate mixed effect models.

We denote P the vector of population parameters (also called hyperparameters) estimated using the data in the learning dataset B: $\mathrm{P} = (\mu, \mathrm{vec}(\Omega), \sigma)'$, where $\mathrm{vec}(\Omega)$ is the vector of unknown values in Ω. Model M^B is defined by its structure and by the hyperparameters $\hat{\mathrm{P}}^B$ estimated from the learning dataset B.

2.2 Method

Evaluation methods compare the predictions obtained by M^B, using the design of V, to the observations in V. V can be the learning dataset B (internal validation) or a different dataset (external validation). The null hypothesis (H_0) is that data in the validation dataset V can be described by model M^B. Prediction distribution errors are a metric designed to test this assumption. In this section, we describe how they are computed.

Let F_{ij} denote the cumulative distribution function (cdf) of the predictive distribution of Y_{ij} under model M^B. We define the prediction discrepancy pd_{ij} as the value of F_{ij} at observation y_{ij}, $F_{ij}(y_{ij})$. F_{ij} can be computed using Monte-Carlo simulations.

Using the design of the validation dataset V, we simulate under model M^B K datasets $\mathrm{V}^{sim(k)}$ (k=1,...,K). Let $\mathrm{Y}_i^{sim(k)}$ denote the vector of simulated observations for the i^{th} subject in the k^{th} simulation.

The prediction discrepancy pd_{ij} for observation y_{ij} is computed as the percentile of y_{ij} in the empirical distribution of the $y_{ij}^{sim(k)}$:

$$\mathrm{pd}_{ij} = F_{ij}(y_{ij}) \approx \frac{1}{K}\sum_{k=1}^{K} \delta_{ijk} \tag{5}$$

where $\delta_{ijk} = 1$ if $y_{ij}^{sim(k)} < y_{ij}$ and 0 otherwise.

Under H_0, if K is large enough, prediction discrepancies (pd) follow $\mathcal{U}(0,1)$ by construction. However, when multiple observations are available for one subject as is usually the case in population analyses, the pd are correlated within each subject [1]. To correct for this correlation, we compute the empirical mean $E(Y_i)$ and variance $\text{var}(Y_i)$ over the K simulations. Decorrelation is performed simultaneously for simulated data:

$$Y_i^{sim(k)*} = \text{var}(Y_i)^{-1/2}(Y_i^{sim(k)} - E(Y_i))$$

and for observed data:

$$Y_i^* = \text{var}(Y_i)^{-1/2}(Y_i - E(Y_i))$$

Decorrelated pd are then obtained using the same formula as 5 but with the decorrelated data, and we call the resulting variables prediction distribution errors (pde):

$$\text{pde}_{ij} = F_{ij}^*(y_{ij}^*) \approx \frac{1}{K}\sum_{k=1}^{K} \delta_{ijk}^* \tag{6}$$

where $\delta_{ijk}^* = 1$ if $Y_{ij}^{sim(k)*} < y_{ij}^*$ and 0 otherwise.

It can happen that some observations lie either below or above all the simulated data corresponding to that observation. In this case, we define the corresponding pde_{ij}:

$$\text{pde}_{ij} = \begin{cases} 1/K & \text{if } y_{ij}^* < y_{ij}^{sim(k)*} \quad \forall k \\ 1 - 1/K & \text{if } y_{ij}^* > y_{ij}^{sim(k)*} \quad \forall k \end{cases} \tag{7}$$

Under H_0, if K is large enough, the distribution of the prediction distribution errors should follow a uniform distribution over the interval [0,1] by construction of the cdf. Normalized prediction distribution errors (npde) can then be obtained using the inverse function of the normal cumulative density function implemented in most software:

$$\text{npde}_{ij} = \Phi^{-1}(\text{pde}_{ij}) \tag{8}$$

By construction, if H_0 is true, npde follow the $\mathcal{N}(0,1)$ distribution without any approximation and are uncorrelated within an individual.

2.3 Tests and graphs

Under the null hypothesis that model M^B describes adequately the data in the validation dataset, the npde follow the $\mathcal{N}(0,1)$ distribution. We report the first three central moments of the distribution of the npde: mean, variance, skewness, as well as the kurtosis, where we define kurtosis as the fourth moment minus 3 so that the kurtosis for $\mathcal{N}(0,1)$ is 0 (sometimes called excess kurtosis). The expected

value of these four variables for the expected $\mathcal{N}(0,1)$ are respectively 0, 1, 0 and 0. We also give the standard errors for the mean (SE=$\sigma/\sqrt{N}$) and variance (SE=$\sigma\sqrt{2/(N-1)}$).

We use 3 tests to test the assumption that the npde follow the $\mathcal{N}(0,1)$ distribution: (i) a Wilcoxon signed rank test, to test whether the mean is significantly different from 0; (ii) a Fisher test for variance, to test whether the variance is significantly different from 1; (iii) a Shapiro-Wilks test, to test whether the distribution is significantly different from a normal distribution. The package also reports a global test, which consists in considering the 3 tests above with a Bonferroni correction. The p-value for this global test is then reported as the minimum of the 3 p-values multiplied by 3, the number of simultaneous tests (or 1 if this value is larger than 1) [5]. A graphical code is used in the library to highlight significant results, similar to the code used by other statistical functions in R such as lm (see example). The normality test is very powerful, especially with large amount of observations. When the test remains significant even after model refinement, QQ-plots should be used to assess model adequacy in addition to the 3 statistical tests. This is especially useful in large datasets where the sheer amount of data will lead to reject even reasonable models.

Graphs can be used to visualise the shape of the distribution of the npde. Classical plots include quantile-quantile plots (QQ-plots) of the distribution of the npde against the theoretical distribution, as well as histograms of the npde. We also find that scatterplots of the npde versus the independent variable, the predicted dependent variables, or covariates, can help pinpoint model deficiencies. Some of these graphs are plotted by default (see section 3.5.2). The package computes for each observation the predicted value as the empirical mean over the k simulations of the simulated predicted distribution (denoted $E_k(y_{ij}^{sim(k)})$), which is reported under the name ypred along with the npde and/or pd.

3 Using the package

3.1 Download and installation

npde can be obtained at the following URL: http://www.npde.biostat.fr/. The website also contains information concerning the updates to npde, the most recent version of this User Guide, and references to some publications describing and using npde.

The program is distributed as an add-on package (library) for R. The current version is 1.2. Linux/Unix users should download the package npde_1.2.tar.gz while Windows users should download the package npde_1.2.zip. To install the package in Unix/Linux systems, type:

```
R CMD INSTALL npde_1.2.tar.gz
```

Alternatively, the command install.packages() can be used to install the npde package from within R. Assuming the package has been downloaded in the directory subdir, installation from disk can be performed in R with the command:

```
install.packages(pkgs="subdir/npde_1.2.tar.gz",repos=NULL)
```

Superuser privileges may be required for a system-wide installation.

Under Windows, use the GUI menu and select the package npde_1.2.zip. The inline installation can also be used:

```
install.packages(pkgs="subdir/npde_1.2.zip",repos=NULL)
```

Please consult the *R Installation and Administration* manual (section 6) provided with R (or available from the CRAN) for further details concerning the installation of packages.

Uninstalling: prior to installation, previous versions of npde should preferably be uninstalled. Under Windows, please remove the directory npde from the library directory (path RHOME\library). Under Linux, use the remove option (usually requires superuser privileges), assuming the variable $RHOME contains the path to R:

```
sudo R CMD REMOVE -l $RHOME/library/ npde
```

3.2 Loading the library

Loading the library is done as usual in R by typing:

```
library("npde")
```

in the R command window. Starting from version 1.1 downloaded after July 27th, 2007, a message will be printed to state the version and date of the library.

3.3 Preparation of the input

The library needs two files:

- the file containing the dataset to be evaluated (hereafter named 'observed data')
- the file containing the simulations (hereafter named 'simulated data')

The library does not perform the simulations. Either R, NONMEM [6], MONOLIX [7] or any program of your choice can be used for that purpose.

3.3.1 Observed data

The observed data file must contain at least the following 3 columns:

1. id : patient identification
2. xobs: independent variable (time, X, ...)
3. yobs: dependent variable (DV, concentrations, effects...)

An additional column may be present in the dataset to indicate missing data (MDV). In this case, this column should contain values of 1 to indicate missing data and 0 to indicate observed data (as in NONMEM or MONOLIX. Alternatively, missing data can be coded using a dot ('.') or the character string NA directly in the column containing yobs. The computation of the prediction distribution errors will remove missing observations from the observed dataset reported in the output (see section 3.5).

Other columns may be present but will not be used by the library. The actual order of the columns is unimportant, since the user may specify which column contain the requested information, but the default order is 1. id, 2. xobs, 3. yobs and no MDV column. A file header may be present, and column separators should be one of: blank space(s), tabulation mark, comma (,) or semi-colon (;). Finally, the digit mark should be a dot as in English (eg a number would read 4.5) and not a comma as in French (4,5).

3.3.2 Simulated data

The user must provide a file containing the K simulated datasets stacked one after the other. Within each simulated dataset, the order of the observations must be the same as within the observed dataset. The dimensions of the two datasets must be compatible: if n_{obs} is the number of lines in the observed dataset, the file containing the simulated datasets must have $K \times n_{obs}$ lines.

The simulated data file must contain at least 3 columns, in the following order:

1. id : patient identification
2. xsim: independent variable (time, X, ...)
3. ysim: dependent variable (DV, concentrations, effects...)

Additional columns may be present but will not be used by the library.

The length of the id (resp xobs) column must be equal to the length of the id (resp xobs) column of the observed dataset repeated K times.

An example of how to set up simulations for an example dataset can be found in section 4, and examples of a simulated and observed dataset are available in the subdirectory doc/inst of the library.

3.3.3 BQL data

Important note: BQL (below the quantification limit LOQ) or otherwise censored data are currently not handled by npde.

If an estimation method taking censored data into account has been used for the estimation, these data should be removed from the dataset or set to missing, using for example an MDV item, pending future extensions of npde. On the other hand, if BQL data were kept as is during the estimation process, or set to LOQ/2, they should remain in the dataset. npde will likely detect model misspecification related to these data, and we would suggest for the evaluation to remove time-points where a significant proportion of the data is BQL.

During the simulations, negative or BQL data may be simulated due to the error model. At present, these data should be kept as is to avoid truncating the predictive distribution, but further work is needed to propose a better method of dealing with censored data both in the simulated and observed datasets.

3.3.4 Number of simulations

Based on the results of the simulation study, we recommend to use at least K=1000 but the actual number may depend on the dataset involved, and should be increased when the dataset includes a large number of subjects. This will be investigated in more details in future work on npde. A warning will be issued when K is smaller than 1000 (see example in section 4).

3.4 Execution

3.4.1 Interactive execution

The interactive mode is called by the function npde():

```
myres<-npde()
```

The user will be prompted to enter successively:

- the name of the file or dataframe containing the observed data
- the columns in which id, xobs, dependent variable yobs, and possibly column with missing data MDV can be found (the default order is 1, 2, 3 and no MDV column)
- the name of the file or dataframe containing the simulated data

- whether results should be saved to disk; if so, the user must also enter
  - the format of the graph (one of: Postscript, JPEG, PNG or PDF)
  - the name of the files: an extension .npde will be added to this name for the file in which numerical results are to be saved (see section 3.5), and an extension depending on the format of the graph will be added to this name for the file in which graphs are to be saved (respectively .eps, .jpeg, .png, .pdf for the formats above). For instance, if the user enters myoutput and requests that the graphs be saved in PDF format, the results file will be named myoutput.npde and the graph files will be myoutput.pdf.
- whether npde should be computed
- whether pd should be computed
- whether a message should be printed as the computation of npde begins in a new subject
- whether the function should return values (see section 3.5.1)

Alternatively, one or both filenames for the observed and simulated data can be replaced by a dataframe if the data has already been loaded in R (see example in the online documentation provided with the package).

3.4.2 Non-interactive execution

In the non-interactive mode, the required information is fed to the function autonpde() instead of answering a series of questions. The minimum input should include the name of the observed data file (for example, theopp.tab) and the name of the simulated data file (for example, simtheopp.tab), as in:

```
autonpde("theopp.tab","simtheopp.tab")
```

A number of options can also be set as arguments, and are given in table I.

Here is an example of the call to autonpde() with a number of arguments (see example in section 4.4 for an illustration):

```
autonpde<-function(namobs="theopp.tab",namsim="simtheopp.tab",iid=1,ix=2,iy=3,
    imdv=0,namsav="output.eps",boolsave=T,type.graph="eps",output=F,verbose=T)
```

The arguments have the following meaning:

namobs: name of the observed data file (or name of the R dataframe)

Option	Effect	Default value
iid	number of the column where id is located in the observed data file	1
ix	number of the column where X is located in the observed data file	2
iy	number of the column where Y is located in the observed data file	3
imdv	number of the column where MDV is located in the observed data file	0 (none)
namsav	name of the files where results will be saved (without extension)	output
boolsave	whether results should be saved to disk	T
type.graph	graph format	eps (postscript)
output	whether the function returns the results	T
verbose	whether a message should be printed as the computation of npde begins in a new subject	F
calc.npde	whether normalised prediction distribution errors should be computed	T
calc.pd	whether prediction discrepancies should be computed	F

Table I:*Options available for the autonpde function.*

namsim: name of the simulated data file (or name of the R dataframe)

iid: number of the column where id is located in the observed data file; defaults to 1

ix: number of the column where xobs is located in the observed data file; defaults to 2

iy: number of the column where yobs is located in the observed data file; defaults to 3

imdv: number of the column where MDV is located in the observed data file; defaults to 0 (no MDV item)

namsav: name of the files to which results are to be saved (defaults to output, which will produce a file called output.eps (if the default format of postscript is kept, see type.graph below for the other extensions possible) for the graphs and a file called output.npde for the numerical results

boolsave: whether graphs should be saved to a file; defaults to T

type.graph: graph format (one of: Postscript, JPEG, PNG or PDF, corresponding to extensions .eps, .jpeg, .png and .pdf respectively for the graph file); defaults to postscript

output: whether the function returns the results; defaults to T

verbose: whether a message should be printed as the computation of npde begins in a new subject; defaults to F

calc.npde: whether normalised prediction distribution errors should be computed; defaults to T

calc.pd: whether prediction discrepancies should be computed; defaults to F

3.5 Results

Both execution modes will produce the same results. Three types of results are produced by default, but options can be used so that only some of them are created:

1. an R object containing several elements, including the npde and/or pd (see section 3.5.1). With the option output=F the object is not returned.
2. a graph file containing diagnostic plots of the npde (see section 3.5.2). The graph also appears in the graphic window of the current R session. With the option boolsave=F the graph is shown but not saved to a file.
3. a text file with the same name as the graph file and extension .npde containing the following data, organised in columns: id, xobs, ypred, npde, pd With the option boolsave=F, the results are not saved.

3.5.1 Value

By default, the function returns an object which can be redirected to an object, as below:

```
myres<-npde()
```

Elements of myres can then be accessed as usual with R objects, e.g. normalised prediction distribution errors can be obtained as:

```
myres$npde
```

This behaviour can be changed using the option output=F.

The object returned by the function contains 7 elements:

1. a data frame obsdat containing the observed data, with the following elements: id, xobs, yobs
2. ydobs: the decorrelated observed data y^*_{ij}
3. ydsim: the decorrelated simulated data $y^{sim(k)*}_{ij}$
4. ypred: the empirical mean of the simulated predicted distribution for each observation ($E_k(y^{sim(k)}_{ij})$)
5. xerr: an integer valued 0 if no error occurred during the computation or a positive number (1 or 2) depending on the error encountered, if an error occurred
6. npde: the normalised prediction distribution errors (if calc.npde=T, a vector of NA otherwise)
7. pd: the prediction discrepancies (if calc.pd=T, a vector of NA otherwise)

Note that obsdat has been stripped of the lines corresponding to missing or absent observations (including dose events in NONMEM) and so may not match the observed file exactly.

3.5.2 Graphs

Four graphs are produced:

1. a quantile-quantile plot: plot of the npde versus the corresponding quantiles of a normal distribution
   - the line $y = x$ is also drawn
2. a histogram of the npde
   - the shape of the normal distribution $\mathcal{N}(0,1)$ is also shown
3. a plot of the npde versus the independent variable X
4. a plot of the npde versus ypred
   - for these last two graphs, we plot the lines corresponding to $y = 0$ and to the critical values 5% and 95% (delimiting the 90% confidence interval in which we expect to find the bulk of the npde).

These graphs are designed as diagnostics for the npde; a function providing similar graphs for pd is plotpd.

3.6 Error during execution

Sometimes the function is unable to compute the decorrelated prediction distribution errors for one or more subjects. The following error messages can appear:

```
The computation of the npde has failed for subject xx because the Cholesky
decomposition of the covariance matrix of the simulated data could not be
obtained.
```

or

```
The computation of the npde has failed for subject xx because the covariance
matrix of the simulated data could not be inverted.
```

followed by:

```
This usually means that the covariance matrix is not positive-definite. This can
be caused by simulations widely different from observations (in other words, a
poor model). We suggest to plot a prediction interval from the simulated data
to check whether the simulations are reasonable, and to consider prediction
discrepancies.
Prediction discrepancies will now be computed.
```

In our experience, this usually happens when the model is so ill-conditioned that the matrices involved in the computation of the prediction distribution errors are singular, and mostly happens when the model predicts the data very poorly. A prediction interval (or Visual Predictive Check) can be plotted to check this.

When npde cannot be computed, the program computes automatically pd even if the calc.pd=F option was used. The following graphs are plotted using pd instead of npde

1. a quantile-quantile plot: plot of the pd versus the corresponding quantiles of a uniform distribution
   - the line $y = x$ is also drawn
2. a histogram of the pd with the uniform density $\mathcal{U}(0,1)$ overlain
3. a plot of the pd versus the independent variable X
4. a plot of the pd versus ypred

- for these last two graphs, we plot the lines corresponding to $y = 0$ and to the 5% and 95% critical values (delimiting the 90% confidence interval in which we expect to find the bulk of the pd).

3.7 Other functions

The function testnpde() can be used to perform the tests on the normalised prediction distribution errors. Assuming the vector npde contains the npde, the function is called by typing:

```
testnpde(npde)
```

It prints four tests and returns a vector with the p-values of:

- a Wilcoxon signed rank test, to test whether the mean is significantly different from 0
- a Fisher test for variance, to test whether the variance is significantly different from 1
- a Shapiro-Wilks test, to test whether the distribution is significantly different from a normal distribution
- the adjusted p-value corresponding to the minimum of the 3 previous p-values multiplied by the number of tests (3), or 1 if this value is larger than 1.

The function plotnpde() can be used to plot the graphs. It requires as input both the normalised prediction distribution errors and the observed X. When npde() or autonpde() are called with the option output=T (default), the resulting object contains both these vectors:

```
myres<-npde(output=T)
xobs<-myres$obsdat$xobs
npde<-myres$npde
ypred<-myres$ypred
plotnpde(xobs,npde,ypred)
```

The function plotpd can be used to plot similar graphs for pd instead of npde, since these graphs are not plotted by default:

```
pd<-myres$pd
plotpd(xobs,pd,ypred)
```

The npde library contains 14 functions. Figure 1 presents the functions hierarchy starting with function npde. A similar graph is obtained with function autonpde without the call to function pdemenu.

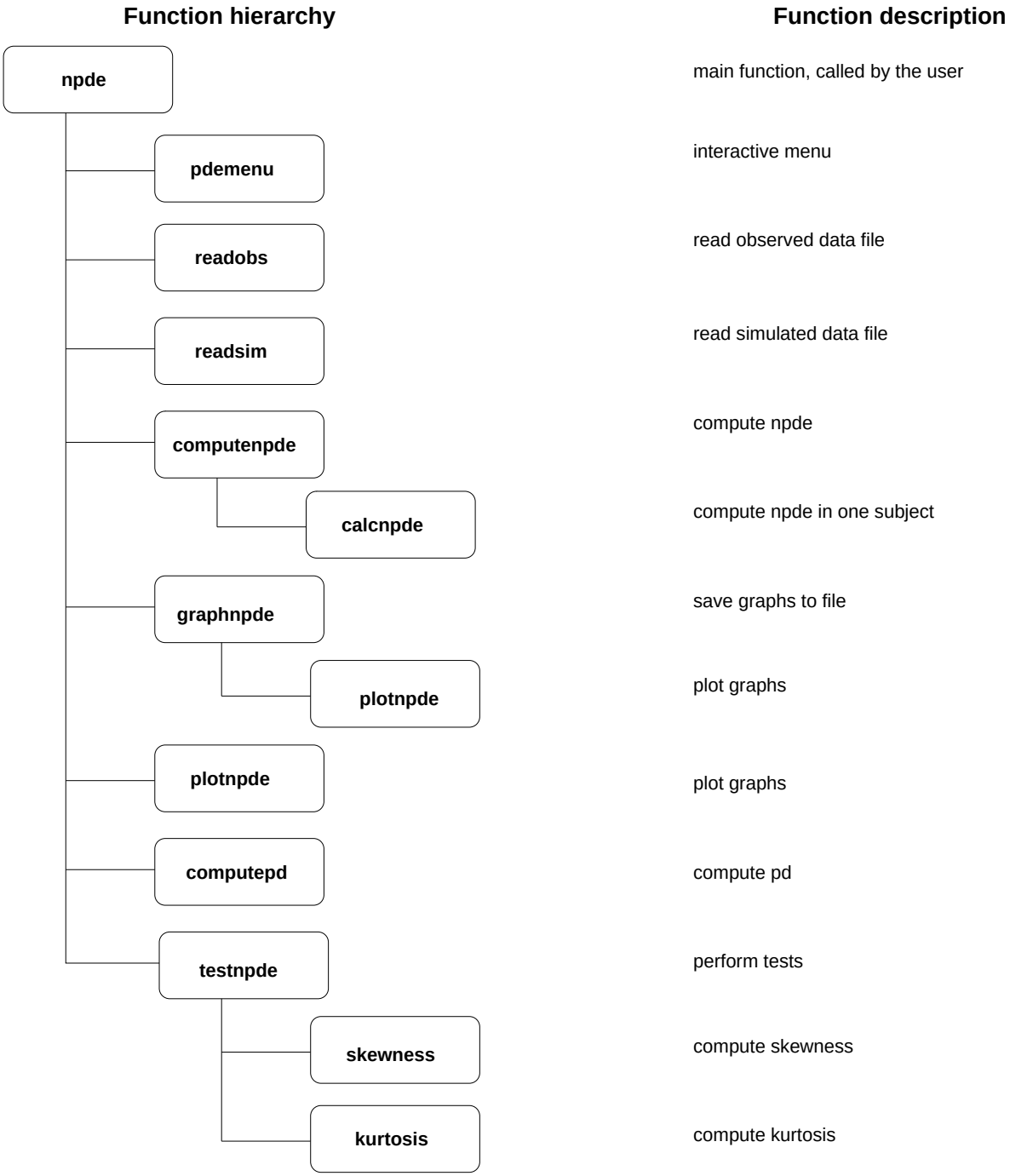

Figure 1 : *Function hierarchy for the npde package, and brief description of each function. The functional hierarchy is given for a user call to npde. With autonpde, the hierarchy is the same save for the initial call to pdemenu.*

4 An example

The subdirectory doc contains a full working example of application of npde. We used the theopp.tab dataset provided with NONMEM as an example which most users are already familiar with. This dataset is also available under the name Theoph in the dataset package in R, under a slightly different format. This dataset was provided by a study by Dr. Robert Upton of the kinetics of the anti-asthmatic drug theophylline [6].

The subdirectory doc contains the following files:

theopp.tab	the observed data
simtheopp.tab	the simulated data (with K=100)[1]
fittheop.ctr	the control file used for the estimation
simultheop.ctr	the control file used for the simulations
runtheo.res	the result file from the estimation
theophylline.eps	the graphs
theophylline.npde	the file containing the results
npde_userguide.pdf	the present user guide
vtrue.dat[2]	a file with data simulated under H_0
vfalse.dat[2]	a file with data simulated assuming a bioavailability divided by 2

[1] *We used K=100 to provide a very quick computation of the* npde *and to avoid including a large file in the package, however we recommend using at least K=1000 for the simulations.*

[2] *These datasets were simulated as examples of external validation datasets in a poster [8]*

4.1 Data

Theophylline concentrations were measured in 12 patients over a period of 24 hr after a single oral dose of the drug. Each patient received a different dose. The data file has the following structure:

Column number	1	2	3	4	5
Column name	ID	Dose	Time	Conc	Wt
Item meaning	Patient id	Dose	Time	Concentrations	Weight

Doses are given in mg, times in hours and concentrations are reported in mg.L^{-1}.

The data for the first two patients is shown in the appendix (see page 26)

4.2 Model

The data was analysed with a one-compartment model with first-order absorption and elimination, parameterised in absorption rate constant k_a (units hr^{-1}) volume of distribution V (units L) and elimination rate constant k (units hr^{-1}). Concentrations at time 0 were removed from the dataset. The model did not include covariates. Interindividual variability was modelled using an exponential model for the three pharmacokinetic parameters. A covariance between the parameters k and V was assumed, yielding the following variance-covariance matrix:

$$\Omega = \begin{pmatrix} \omega^2_{k_a} & 0 & 0 \\ 0 & \omega^2_V & \text{cov}(\eta_k, \eta_V) \\ 0 & \text{cov}(\eta_k, \eta_V) & \omega^2_k \end{pmatrix} \tag{9}$$

The residual error model was a combined additive and proportional error model as in equation 2.

These data were analysed with the software NONMEM version 5.1. The ADVAN2 routine was used. The estimation method was the FOCE algorithm with the INTERACTION option. The control file is given in the Appendix (see page 27) and the relevant results in the ouput file runtheo.res are shown on page 28.

The following parameter estimates were obtained:

Population mean		Interindividual variability	
k_a (hr^{-1})	1.51	ω_{k_a} (-)	0.67
V (L)	0.46	ω_V (-)	0.12
k ($L.hr^{-1}$)	0.087	ω_k (-)	0.13
σ_{inter} ($mg.L^{-1}$)	0.088	$\text{cor}(\eta_k, \eta_V)$ (-)	0.99
σ_{slope} (-)	0.26		

4.3 Simulations

The simulations were also performed using NONMEM version 5.1. The control file used for the simulations is given in the Appendix (see page 29). The beginning is identical to the control file used for the analysis (page 28); the initial values in the $THETA, $OMEGA, $SIGMA blocks have been changed to the values estimated with the model, the $ERROR block includes a line to output the simulated data, and the $TABLE block has been changed to output the simulated data in a file.

The number of simulations can be changed with the SUBPROBLEMS options in the $SIMULATION block. Here, we use 100 simulations to compute the npde quickly as an illustration, but larger numbers are more appropriate (we recommend at least 1000 simulations). Simulations were saved in the file simtheopp.tab.

4.4 **Computing** npde

The interactive version of the program was run below. In a first step, the user was prompted to enter all details necessary for the computations (text **in purple** show values entered by the user while text in black is printed by the program):

myres<-npde()

Name of the file containing the observed data: **theopp.tab**

I'm assuming file theopp.tab has the following structure:

ID X Y ...

To keep, press ENTER, to change, type any letter: **n**

Column with ID information ? **1**

Column with X (eg time) information ? **3**

Column with Y (eg DV) information ? **4**

Column signaling missing data (eg MDV, press ENTER if none) ?

Name of the file containing the simulated data:

simtheopp.tab

Do you want results and graphs to be saved to files (y/Y) [default=yes] ? **y**

Different formats of graphs are possible:

1. Postscript (extension eps)
2. JPEG (extension jpeg)
3. PNG (extension png)
4. Acrobat PDF (extension pdf)

Which format would you like for the graph (1-4) ? **1**

Name of the file (extension will be added, default=output): **theophylline**

Do you want to compute npde (y/Y) [default=yes] ? **y**

Do you want to compute pd (y/Y) [default=no] ? **y**

Do you want a message printed as the computation of npde begins in a new subject (y/Y) [default=no] ? **y**

Do you want the function to return an object (y/Y) [default=yes] ? **y**

In the second step, the program computed the normalised prediction distribution errors, plotted the corresponding graphs and performed the statistical tests for npde, then computed the prediction discrepancies (for which no tests are reported). A warning is issued here because the number of simulations is considered too small.

```
Warning: the number of simulations is 100 which may be too small.
We advise performing at least 1000 simulations to compute npde.
Computing npde
Computing the npde for subject  1
Computing the npde for subject  2
Computing the npde for subject  3
Computing the npde for subject  4
Computing the npde for subject  5
Computing the npde for subject  6
Computing the npde for subject  7
Computing the npde for subject  8
Computing the npde for subject  9
Computing the npde for subject  10
Computing the npde for subject  11
Computing the npde for subject  12
Saving graphs in file keepnpde/inst/doc/theophylline.eps
-------------------------------------------
Distribution of npde:
           mean= 0.05641   (SE= 0.092 )
       variance= 1.024   (SE= 0.13 )
       skewness= 0.4065
       kurtosis= 0.0888
-------------------------------------------

Statistical tests
  Wilcoxon signed rank test  : 0.883
  Fisher variance test       : 0.823
  SW test of normality       : 0.00509 **
Global adjusted p-value      : 0.0153 *
---
Signif. codes: '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1
-------------------------------------------
Computing pd
Saving results in file keepnpde/inst/doc/theophylline.npde
```

Alternatively, the first step can be run non-interactively, with the following command:

```
myres<-autonpde("theopp.tab","simtheopp.tab",1,3,4,0,"theophylline",verbose=T,
    calc.pd=T)
```

The results of the statistical tests show that the normality assumption for the normalised prediction distribution errors is rejected according to the Shapiro-Wilks test for normality, as can be seen in the plots in the next section (figure 2). The adjusted p-value for the 3 tests taken simultaneously using a Bonferroni correction therefore rejects the assumption that the model describes the data adequately.

4.5 Graphs

The graphs in figure 2 are plotted in a window, and saved to a file (unless boolsave=F). The

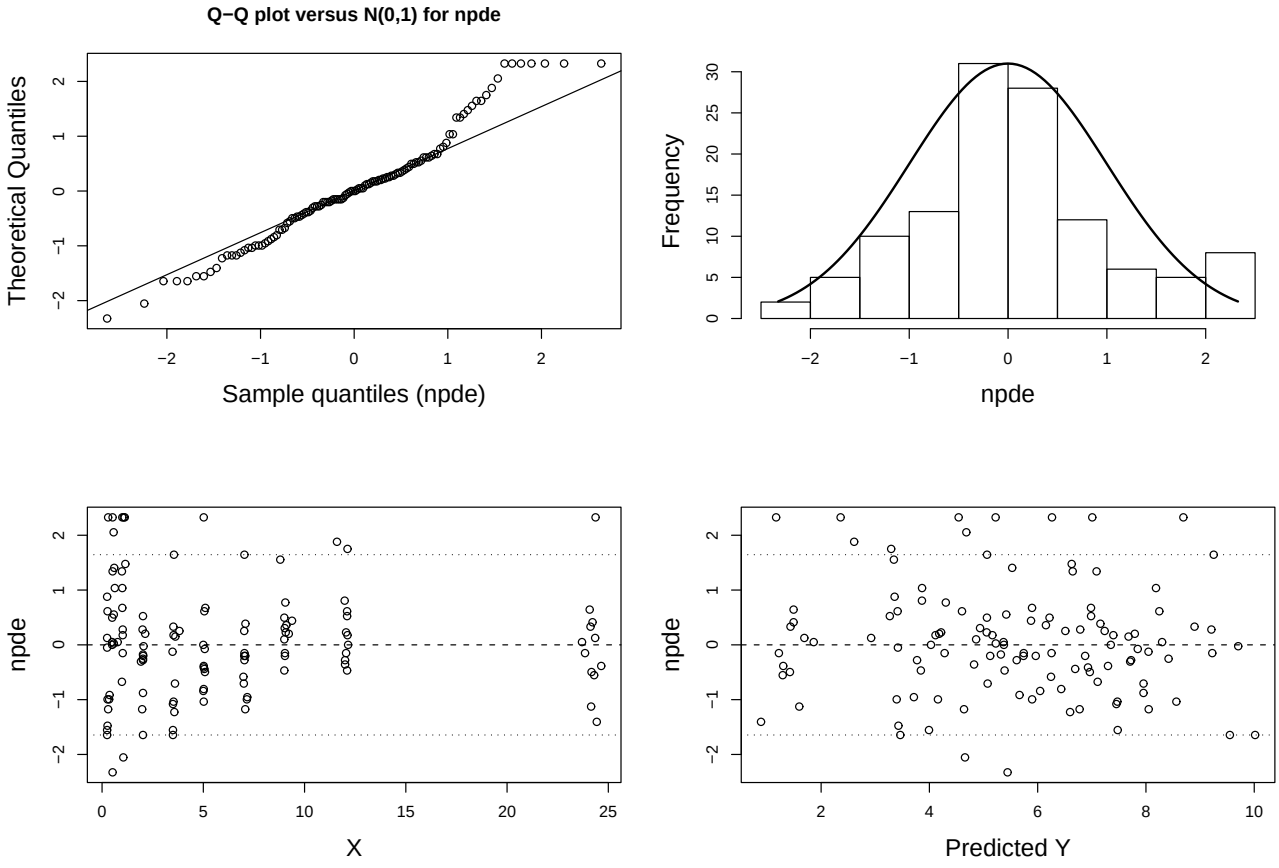

Figure 2 : *Graphs plotted by the* npde() *or* autonpde() *functions.*

quantile-quantile plot and the histogram show a group of values corresponding to npde = 2.33, corresponding to predicted distribution errors set at 0.99. This indicates observations larger than all the

100 corresponding simulated values. This often happens when K is small as is the case in this example (K=100). However, even increasing the number of simulations to 1000 or 2000 does not in this example yield a non-significant test, meaning the model does not describe the data adequately (results not shown).

4.6 Tests

The tests in section 4.4 can be regenerated easily without running the computation all over again, using the function testnpde(). In the example above, the npde were saved to a file named theophylline.npde. The following code reads the results from this file and computes the same tests as above:

```
dat<-read.table("theophylline.npde",header=T)
y<-testnpde(dat$npde)
```

which yields the same results as previously:

```
------------------------------------------
Distribution of npde:
           mean= 0.05641   (SE= 0.092 )
       variance= 1.024   (SE= 0.13 )
       skewness= 0.4065
       kurtosis= 0.0888
------------------------------------------

Statistical tests
  Wilcoxon signed rank test  : 0.87
  Fisher variance test       : 0.823
  SW test of normality       : 0.00509 **
Global adjusted p-value      : 0.0153 *
---
Signif. codes: '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1
------------------------------------------
```

Here, the four p-values are redirected to the R object y.

References

[1] F. Mentré and S. Escolano. Prediction discrepancies for the evaluation of nonlinear mixed-effects models. *Journal of Pharmacokinetics and Pharmacodynamics* **33**:345–67 (2006).

[2] K. Brendel, E. Comets, C. Laffont, C. Laveille, and F. Mentré. Metrics for external model evaluation with an application to the population pharmacokinetics of gliclazide. *Pharmaceutical Research* **23**:2036–49 (2006).

[3] R Development Core Team. *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria (2006). ISBN 3-900051-07-0.
URL http://www.R-project.org

[4] F. Mesnil, F. Mentré, C. Dubruc, J. Thénot, and A. Mallet. Population pharmacokinetics analysis of mizolastine and validation from sparse data on patients using the nonparametric maximum likelihood method. *Journal of Pharmacokinetics and Biopharmaceutics* **26**:133–161 (1998).

[5] S. Wright. Adjusted P-Values for Simultaneous Inference. *Biometrics* **48**:1005–13 (1992).

[6] L. Sheiner and S. Beal. *NONMEM Version 5.1*. University of California, NONMEM Project Group, San Francisco (1998).

[7] M. Lavielle. *MONOLIX (MOdèles NOn LInéaires à effets miXtes)*. MONOLIX group, Orsay, France (2005).
URL http://www.monolix.org

[8] E. Comets, K. Brendel, and F. Mentré. Normalised prediction distribution errors in R: the npde library. *16th meeting of the Population Approach Group in Europe, Copenhagen, Denmark* page Abstr 1120 (2007).

Appendix

The data

Below, the data for the first two subjects in the data file is shown:

```
1      4.02      0.          .       79.6
1       .        0.25      2.84       .
1       .        0.57      6.57       .
1       .        1.12     10.5        .
1       .        2.02      9.66       .
1       .        3.82      8.58       .
1       .        5.1       8.36       .
1       .        7.03      7.47       .
1       .        9.05      6.89       .
1       .       12.12      5.94       .
1       .       24.37      3.28       .
2      4.4       0.          .       72.4
2       .         .27      1.72       .
2       .         .52      7.91       .
2       .        1.        8.31       .
2       .        1.92      8.33       .
2       .        3.5       6.85       .
2       .        5.02      6.08       .
2       .        7.03      5.4        .
2       .        9.        4.55       .
2       .       12.        3.01       .
2       .       24.3        .90       .
```

Control file used for the analysis

```
$PROB  THEOPHYLLINE   POPULATION DATA
$INPUT      ID DOSE=AMT TIME DV WT
$DATA       theopp.tab
$SUBROUTINES  ADVAN2 TRANS2

$PK
;THETA(1)=MEAN ABSORPTION RATE CONSTANT (1/HR)
;THETA(2)=MEAN ELIMINATION RATE CONSTANT (1/HR)
;THETA(3)=SLOPE OF CLEARANCE VS WEIGHT RELATIONSHIP (LITERS/HR/KG)
   CALLFL=1
   KA=THETA(1)*EXP(ETA(1))
   V=THETA(2)*EXP(ETA(2))
   K=THETA(3)*EXP(ETA(3))
   CL=K*V
   S2=V

$ERROR
SLOP=THETA(4)
SINT=THETA(5)
IPRED=F
W=SLOP*F+SINT
Y=F+W*EPS(1)
IRES=IPRED-DV
IWRES=IRES/W

$THETA  (.1,3,5) (0,0.5,) (.004,.1,2) (0,0.2,) (0,0.1,)
$OMEGA 0.2
$OMEGA BLOCK(2) 0.2 0.05 0.2
$SIGMA 1 FIX

$EST NOABORT METHOD=COND INTERACTION MAXEVAL=2000  PRINT=5
$COV
```

Results obtained with NONMEM

```
 ********************************************************************************************
 **********                                                                        *********
 **********                MINIMUM VALUE OF OBJECTIVE FUNCTION                     *********
 **********                                                                        *********
 ********************************************************************************************
 ***************************          86.664     ************************************
 ********************************************************************************************
 **********                                                                        *********
 **********                      FINAL PARAMETER ESTIMATE                         **********
 **********                                                                        *********
 ********************************************************************************************

 THETA - VECTOR OF FIXED EFFECTS PARAMETERS   *********

            TH 1      TH 2      TH 3      TH 4      TH 5
         1.51E+00  4.60E-01  8.73E-02  8.81E-02  2.58E-01

 OMEGA - COV MATRIX FOR RANDOM EFFECTS - ETAS  ********

            ETA1      ETA2      ETA3
 ETA1
+        4.43E-01
 ETA2
+        0.00E+00  1.45E-02
 ETA3
+        0.00E+00  1.62E-02  1.82E-02

 SIGMA - COV MATRIX FOR RANDOM EFFECTS - EPSILONS  ****

            EPS1
 EPS1
+        1.00E+00
```

Control file used for the simulations

```
$PROB  THEOPHYLLINE   POPULATION DATA
$INPUT      ID DOSE=AMT TIME DV WT
$DATA       theopp.tab
$SUBROUTINES  ADVAN2 TRANS2

$PK
;THETA(1)=MEAN ABSORPTION RATE CONSTANT (1/HR)
;THETA(2)=VOLUME OF DISTRIBUTION (LITERS)
;THETA(3)=MEAN ELIMINATION RATE CONSTANT (1/HR)
   CALLFL=1
   KA=THETA(1)*EXP(ETA(1))
   V=THETA(2)*EXP(ETA(2))
   K=THETA(3)*EXP(ETA(3))
   CL=K*V
   S2=V
$ERROR
SLOP=THETA(4)
SINT=THETA(5)
IPRED=F
W=SLOP*F+SINT
Y=F+W*EPS(1)
FSIM=Y

$THETA 1.51 0.46 0.0873 0.0881 0.258
$OMEGA 0.443
$OMEGA BLOCK(2) 0.0145 0.0162 0.0182
$SIGMA 1 FIX

$SIMULATION (82015831)  ONLYSIM SUBPROBLEMS=100
$TABLE ID TIME FSIM IPRED NOPRINT NOHEADER NOAPPEND FILE=simtheopp.tab
```

Simulated data

Below, the first few lines of the simulated data file created by the previous control file are shown:

```
1.0000E+00  0.0000E+00 -9.0212E-02  0.0000E+00
1.0000E+00  2.5000E-01  2.2892E+00  2.5691E+00
1.0000E+00  5.7000E-01  4.2279E+00  4.6287E+00
1.0000E+00  1.1200E+00  5.4979E+00  6.3074E+00
1.0000E+00  2.0200E+00  7.9173E+00  6.8719E+00
1.0000E+00  3.8200E+00  5.3943E+00  6.1422E+00
1.0000E+00  5.1000E+00  4.3926E+00  5.4793E+00
1.0000E+00  7.0300E+00  5.0335E+00  4.5902E+00
1.0000E+00  9.0500E+00  3.3301E+00  3.8114E+00
1.0000E+00  1.2120E+01  3.3686E+00  2.8730E+00
1.0000E+00  2.4370E+01  6.0324E-01  9.3011E-01
2.0000E+00  0.0000E+00  2.0597E-01  0.0000E+00
2.0000E+00  2.7000E-01  2.3492E+00  3.1259E+00
2.0000E+00  5.2000E-01  4.4722E+00  5.1687E+00
2.0000E+00  1.0000E+00  5.7317E+00  7.5603E+00
2.0000E+00  1.9200E+00  8.6685E+00  9.1770E+00
2.0000E+00  3.5000E+00  9.6393E+00  8.9901E+00
2.0000E+00  5.0200E+00  7.9179E+00  8.1473E+00
2.0000E+00  7.0300E+00  6.7075E+00  7.0363E+00
2.0000E+00  9.0000E+00  5.4641E+00  6.0802E+00
2.0000E+00  1.2000E+01  5.0094E+00  4.8659E+00
2.0000E+00  2.4300E+01  2.0761E+00  1.9518E+00
```

This dataset contains the following columns: patient ID (idsim), time (xsim), simulated data (ysim), individual predictions. The program only uses the first 3 columns.

Saved results

In the example, the results are saved to a file called theophylline.npde. The first lines of this file, corresponding to the first 2 subjects are shown below:

```
id xobs  yobs  ypred           npde             pd
1  0.25  2.84  2.9238643    0.125661346855074  0.55
1  0.57  6.57  4.6822991    2.05374891063182   0.85
1  1.12  10.5  6.264357     2.32634787404084   0.99
1  2.02  9.66  6.986255     0.524400512708041  0.98
1  3.82  8.58  6.511039     0.253347103135800  0.93
1  5.1   8.36  5.895675     0.674489750196082  0.96
1  7.03  7.47  5.064736     1.64485362695147   0.97
1  9.05  6.89  4.302909     0.772193214188685  0.99
1  12.12 5.94  3.29402      1.75068607125217   0.99
1  24.37 3.28  1.16874348   2.32634787404084   0.99
2  0.27  1.72  3.39568076  -0.994457883209753  0.16
2  0.52  7.91  5.222963     2.32634787404084   0.9
2  1     8.31  6.984615     0.674489750196082  0.71
2  1.92  8.33  7.707843    -0.305480788099397  0.64
2  3.5   6.85  7.47791     -1.55477359459685   0.33
2  5.02  6.08  6.43454     -0.80642124701824   0.43
2  7.03  5.4   5.612031    -0.279319034447454  0.48
2  9     4.55  4.862751     0.100433720511470  0.43
2  12    3.01  3.771684    -0.279319034447454  0.26
2  24.3  0.9   1.2906205   -0.553384719555673  0.31
```