Supplementary information - Examples

1 Warfarin PK/PD model 1 - Basic PK & turnover PD
   1.1 Model description
      1.1.1 Structural model
      1.1.2 Variability model
      1.1.3 Covariate model
      1.1.4 Parameter Model
      1.1.5 Observation Model
   1.2 Trial Design
   1.3 Modelling Steps
   1.4 PharmML implementation

2 Warfarin PK/PD model 2 - Inter-occasion variability on PK parameters

3 Warfarin PK/PD model 3 - Time-To-Event model for PD data

4 Warfarin PK/PD model 4 - Optimal design application
   4.1 Trial Design
   4.2 Modelling Steps - Design optimisation

5. Minimal model of glucose kinetics (re-parameterised)
   5.1 Model description
   5.2 Trial Design and Modelling Steps
   5.4 PharmML implementation

6. Mitotic oscillator model
   6.1 Model description
   6.2 Modelling steps

References
This document contains the description of the use cases introduced in the tutorial within Working with PharmML. For each example, the illustration of both the model features and of the PharmML implementation is provided. The complete PharmML codes, the correspondent implementations with at least one target tool, and the files produced executing those implementations are also provided as part of the supplementary information of this tutorial, within the “Examples” file.

1 Warfarin PK/PD model 1 - Basic PK & turnover PD

Warfarin is an anticoagulant drug administered for secondary prevention of thrombotic events. In this first example, we use data from a pharmacokinetics/pharmacodynamics (PK/PD) study in 32 patients, collected following a 1.5 mg/kg single oral dose. Data and model were derived from different papers[1-5].

1.1 Model description

This chapter describes in some detail how to define the model’s components: the structural, variability, covariate, parameter, and observation models. The subsequent warfarin related examples are largely based on the same model and the text will contain only new elements not covered here.

1.1.1 Structural model

The PK data are modelled using a one-compartment distribution model with first-order absorption (parameterized with the absorption rate constant $k_a$) and an absorption lag time (parameterized with $T_{lag}$). Warfarin concentration $CONC$ at time $t$ is given by the following algebraic equations:

$$CONC(t) = 0 \text{ for } t < (t_D + T_{lag})$$

$$CONC(t) = \frac{D}{V} \frac{k_a}{k_a - k} \left( e^{-k(t_D-T_{lag})} - e^{-k_a(t_D-T_{lag})} \right) \text{ for } t \geq (t_D + T_{lag})$$

where $t_D$ is the time when dose $D$ is given, $V$ is the compartment volume, and $k$ is the first-order elimination rate constant, which is the ratio between clearance, $CL$, and volume, $V$. When multiple doses are given at $t_{D1}$, $t_{D2}$, ..., the superposition principle applies.

The effect of warfarin is described through a turnover model reflecting the physiological processes involved in the delay between plasma concentrations and the pharmacodynamic effect, $E(t)$. PD is quantified as the prothrombin complex activity (PCA) time course: $E(t) = PCA(t)$. The turnover model is parameterised in terms of the “elimination” rate, $k_{out}$, and the “synthesis” rate, given by the product between $E_0$ and $k_{out}$, the PCA value at time zero (when steady state is assumed). Warfarin is assumed to decrease the synthesis
rate through a sigmoidal model with parameters $I_{\text{max}}$ (the maximal effect) and $IC_{50}$ (the warfarin concentration producing half of the maximal inhibitory effect). The PD model is formulated as an ordinary differential equation (ODE) and reads:

$$\frac{dE(t)}{dt} = E_0 \ k_{\text{out}} \left( 1 - \frac{I_{\text{max}} \ CONC(t)}{IC_{50} + CONC(t)} \right) - k_{\text{out}} \ E(t)$$

with the initial condition $E(t=0) = E_0$.

### 1.1.2 Variability model

This first warfarin model example assumes one subject-related variability level (inter-individual variability, IIV) for the parameters, and one observation-related variability level (residual error) for the observations. IIV and residual error are mapped to the appropriate levels declared here.

### 1.1.3 Covariate model

A covariate model accounting for the effects of body size on the model parameters is included in order to describe the allometric scaling of clearance, $CL$, and volume, $V$, with respect to the body weight, $W$. The covariate model defines a transformed continuous covariate, $\log W_{70} = \log(W/70)$, to be used in the structured parameter models for $CL$ and $V$. Also one categorical covariate, the sex ($SEX$), is taken into account and defined as $SEX = \{\text{female, male}\}$.

### 1.1.4 Parameter Model

The majority of PK parameters ($T_{\text{lag}}$, $ka$, $V$, $CL$) and PD parameters ($E_0$, $IC_{50}$, $k_{\text{out}}$) are assumed to be log-normally distributed among the individuals. Covariate effects are assumed to enter the model log-linearly. For instance, $V$ is assumed to be influenced by $W$ and $SEX$ according to the following equation:

$$\log(V) = \log(V_{\text{pop}}) + \beta_W \ \log W_{70} + \beta_{SEX} 1_{SEX=\text{F}} + \eta_V$$

(VEq)

where $V_{\text{pop}}$ is the population value of $V$, $\eta_V \sim N(0, \omega_V)$ is the inter-individual random effect with standard deviation $\omega_V$ of $V$, and $\beta_W$ and $\beta_{SEX}$ are the coefficients of the effects on $V$ of weight and sex, respectively.

One PD parameter, $I_{\text{max}}$, is constrained to have values within the interval (0,1), and is distributed according to a logit-normal distribution. Its structured model thus reads:

$$\logit(I_{\text{max}}) = \logit(pop_{I_{\text{max}}}) + \eta_{I_{\text{max}}}$$

(LogitEq)

where $pop_{I_{\text{max}}}$ is the population value of $I_{\text{max}}$, and $\eta_{I_{\text{max}}} \sim N(0, \omega_{I_{\text{max}}})$ is the inter-individual random effect with standard deviation $\omega_{I_{\text{max}}}$ on $I_{\text{max}}$. It is worth underlining here the power and advantages of this formulation.
compared to an equation-based notation which usually takes two lines to be specified. In NMTRAN code for instance, the same definition reads:

$$\begin{align*}
\text{LG\_IMAX} &= \text{POP\_LG\_IMAX} + \text{ETA\_LG\_IMAX} \\
\text{IMAX} &= \exp(\text{LG\_IMAX})/(1+\exp(\text{LG\_IMAX}))
\end{align*}$$

(LogitEq) can be parsed and its content passed to any target tool, while the latter is almost impossible to interpret without human intervention.

The random effects $\eta_V$ and $\eta_{CL}$ are assumed to be correlated: this relationship is typically formulated by introducing a correlation coefficient, $\rho_{V,CL}$, or a covariance term, $\text{cov}(\eta_V, \eta_{CL}) = \rho_{V,CL} \omega_V \omega_{CL}$, in the variance-covariance matrix, $\Omega$, for the inter-individual random effects.

1.1.5 Observation Model

In this example, the two responses ($\text{CONC}$ and $E$) are continuous variables, and we assume for them combined and additive residual error models, respectively. For the observed warfarin concentration, $\text{CONC}_{\text{obs}}$, we have

$$\text{CONC}_{\text{obs}} = \text{CONC} + (a + b \text{CONC}) \epsilon_{\text{CONC}}, \quad \epsilon_{\text{CONC}} \sim N(0,1) \quad (\text{ResidualEq1})$$

and for the observed drug effect $E_{\text{obs}}$

$$E_{\text{obs}} = E + a_2 \epsilon_E, \quad \epsilon_E \sim N(0,1) \quad (\text{ResidualEq2})$$

where $a$ and $a_2$ are the additive parameters and $b$ is the proportional parameter.

1.2 Trial Design

In this first example, the actual doses and sampling times are documented for each individual subject taking part in the clinical trial. The trial design is not encoded explicitly but is described within a dataset, “warfarin1.csv”. We therefore define the empirical design as the one sourced from the data file, consisting of $N$ arms (one per subject), each containing the actual treatment and sampling schedule for each subject taking part in the trial.

1.3 Modelling Steps

For estimation purposes, $I_{\text{max}}$ is fixed to 0.9999 for all subjects (no IIV), indicating maximum inhibition (nullifying the synthesis of PCA when concentrations go to infinity). The allometric coefficients describing the effect of $W$ on $CL$ and $V$ are fixed respectively to $\beta_{CL} = 0.75$ and $\beta_W = 1$. 


Estimation is performed in NONMEM through the FOCE algorithm with interaction (FOCEI). Population estimates are obtained together with the associated uncertainty, and the individual parameters are estimated as well.

1.4 PharmML implementation

In what follows we show how the model described above can be encoded in PharmML.

The top-level `<PharmML>` element contains a number of attributes prefixed `xmlns` (which stands for “XML namespace”) which are required by the XML Schema standard. Next, the `<Name>` element carrying the model name, which can be any string, has to be provided and can be followed by an optional `<Description>` element. The `<IndependentVariable symbId="t">` element provides the indication of the independent variable: in this case the time is chosen, whose symbol identifier is defined as “t”.

The next optional element, `<FunctionDefinition>`, is used to declare named user-defined functions which can be used later in model encoding. It proves especially useful in defining residual error models. The following snippet shows how the models called “constantErrorModel” and “combinedErrorModel” (equations ResidualEq1 and ResidualEq2) are implemented:

```xml
<ct:FunctionDefinition symbolType="real" symbId="constantErrorModel">
  <ct:FunctionArgument symbolType="real" symbId="a"/>
  <ct:Definition>
    <ct:Assign>
      <ct:SymbRef symbIdRef="a"/>
    </ct:Assign>
  </ct:Definition>
</ct:FunctionDefinition>

<ct:FunctionDefinition symbolType="real" symbId="combinedErrorModel">
  <ct:FunctionArgument symbolType="real" symbId="a"/>
  <ct:FunctionArgument symbolType="real" symbId="b"/>
  <ct:FunctionArgument symbolType="real" symbId="f"/>
  <ct:Definition>
    <ct:Assign>
      <math:Binop op="plus">
        <ct:SymbRef symbIdRef="a"/>
        <math:Binop op="times">
          <ct:SymbRef symbIdRef="b"/>
          <ct:SymbRef symbIdRef="f"/>
        </math:Binop>
      </math:Binop>
    </ct:Assign>
  </ct:Definition>
</ct:FunctionDefinition>
```

We then enter the definition of the statistical model, whose first element is `<VariabilityModel>`. In this example we actually define two variability models, one of “parameterVariability” type and one of “residualError” type, and both need one level only: the “`indiv`” and the “`residual`” level, respectively. Each model has a block identifier, here “`vm1`” and “`vm2`”:

```xml
<ModelDefinition xmlns="http://www.pharmml.org/pharmml/0.8/ModelDefinition">
  ...
</ModelDefinition>
```
Attributes `blkId` and `symbId` are used to declare block and symbol identifiers, which will be used later, wherever required, e.g. to map the random effects to the appropriate variability models and levels. 

`<CovariateModel>` is the place where the continuous (body weight, \( W \)) and categorical (sex, \( SEX \)) covariates are declared. \( W \) is used in the parameter model in a transformed form, \( \log(W/70) \), which is defined via a new identifier, `symbId="logW70"`, within the covariate model, as the following snippet shows:

```xml
<CovariateModel blkId="cm1">
  <Covariate symbId="W">
    <Continuous>
      <Transformation>
        <TransformedCovariate symbId="logW70"/>
        <ct:Assign>
          <math:Uniop op="log">
            <math:Binop op="divide">
              <ct:SymbRef symbIdRef="W"/>
              <ct:Real>70</ct:Real>
            </math:Binop>
          </math:Uniop>
        </ct:Assign>
      </Transformation>
    </Continuous>
  </Covariate>
</CovariateModel>
```

For \( SEX \), the declaration consists solely of its two categories:

```xml
<Covariate symbId="SEX">
  <Categorical>
    <Category catId="F"/>
    <Category catId="M"/>
  </Categorical>
</Covariate>
</CovariateModel>
```

Next comes the `<ParameterModel>`, here assigned the block identifier “pm1”. The first part of this element declares the set of population parameters, which are used in the definition of random effects and individual parameters. The code snippet declaring the population parameters required to describe warfarin PK reads

```xml
<ParameterModel blkId="pm1">
  <PopulationParameter symbId="pop_Tlag"/>
  <PopulationParameter symbId="omega_Tlag"/>
  <PopulationParameter symbId="pop_ka"/>
  <PopulationParameter symbId="omega_ka"/>
  <PopulationParameter symbId="pop_V"/>
  <PopulationParameter symbId="omega_V"/>
  <PopulationParameter symbId="beta_W"/>
  <PopulationParameter symbId="beta_SEX"/>
  <PopulationParameter symbId="pop_CL"/>
  <PopulationParameter symbId="omega_CL"/>
  <PopulationParameter symbId="beta_CL"/>
  <PopulationParameter symbId="rho_V_CL"/>
</ParameterModel>
```
Typically, a `<RandomVariable>` element is used to encode a random effect. Its declaration consists of two essential elements, the variability reference and the distribution definition, as the following snippet shows:

```xml
<RandomVariable symbId="eta_V">
  <ct:VariabilityReference>
    <ct:SymbRef blkIdRef="vm1" symbIdRef="indiv"/>
  </ct:VariabilityReference>
  <Distribution>
    <po:ProbOnto name="Normal1">
      <po:Parameter name="mean">
        <ct:Assign>
          <ct:Real>0</ct:Real>
        </ct:Assign>
      </po:Parameter>
      <po:Parameter name="stdev">
        <ct:Assign>
          <ct:SymbRef symbIdRef="omega_V"/>
        </ct:Assign>
      </po:Parameter>
    </po:ProbOnto>
  </Distribution>
</RandomVariable>
```

The `<VariabilityReference>` element stores the reference to the individual variability level, "indiv", of one of the variability models, i.e., the parameter related model “vm1”. The declaration of the distribution uses ProbOnto and is implemented by specifying the “code name” for the desired distribution and its parameters. “Normal1” stands for the normal distribution parameterised with mean and standard deviation. Accordingly, the parameter “mean” is assigned 0 and the standard deviation, “stdev”, is assigned a reference to the previously declared population parameter “omega_V”. The code names are not enforced by the schema and libPharmML (see the main paper for the description of libPharmML) can be used to validate the correctness of the declaration.

As described in the main text of the tutorial, the model for the individual parameters can be of different types. Here, for the log-normally distributed V the structured model is adopted and reads:

```xml
<IndividualParameter symbId="V">
  <StructuredModel>
    <Transformation type="log"/>
    <LinearCovariate>
      <PopulationValue>
        <ct:Assign>
          <ct:SymbRef symbIdRef="pop_V"/>
        </ct:Assign>
      </PopulationValue>
      <Covariate>
        <ct:SymbRef blkIdRef="cm1" symbIdRef="logW70"/>
        <FixedEffect>
          <ct:SymbRef symbIdRef="beta_W"/>
        </FixedEffect>
      </Covariate>
      <Covariate>
        <ct:SymbRef blkIdRef="cm1" symbIdRef="SEX"/>
        <FixedEffect>
          <ct:SymbRef symbIdRef="beta_SEX"/>
          <Category catId="F"/>
        </FixedEffect>
      </Covariate>
    </LinearCovariate>
  </StructuredModel>
</IndividualParameter>
```
Within the `<StructuredModel>` block, a transformation can be chosen from a build-in list of available transformations such as "identity" (the default), "BoxCox", "log", "logit" and "probit", with "log" being the option chosen for \(V\) here. If no covariates are present then specifying the `<PopulationValue>` is enough, otherwise the user can choose between `<LinearCovariate>` and `<GeneralCovariate>`. In this example the parameters can be expressed with linear covariate models. The continuous covariate, body weight, is included specifying the related fixed effect, "\(beta_W\)", and transformed covariate, "\(\log W70\)". Similarly for sex: the fixed effect, "\(beta_{SEX}\)", and the category "F" associated with it are declared. The individual parameter model is completed with the reference to its random effect, "\(eta_V\)". All remaining parameters are implemented in a similar way, except for the logit-normal distributed \(Imax\) for which the logit transformation is applied. Its implementation reads very similar to that of the log-normally distributed \(V\) shown above.

One last point worth discussing with respect to the parameter model is the correlation between the random effects, which can be implemented pairwise or using a suitable matrix. The example below shows how the former implementation is used to capture the relationship between "\(eta_V\)" and "\(eta_CL\)":

```
<Correlation>
  <ct:VariabilityReference>
    <ct:SymbRef blkIdRef="vm1" symbIdRef="indiv"/>
  </ct:VariabilityReference>
  <Pairwise>
    <RandomVariable1>
      <ct:SymbRef symbIdRef="eta_V"/>
    </RandomVariable1>
    <RandomVariable2>
      <ct:SymbRef symbIdRef="eta_CL"/>
    </RandomVariable2>
    <CorrelationCoefficient>
      <ct:Assign>
        <ct:SymbRef symbIdRef="rho_V_CL"/>
      </ct:Assign>
    </CorrelationCoefficient>
  </Pairwise>
</Correlation>
```

where "\(rho_V_CL\)" is the correlation coefficient to be estimated.

Next we define the `<StructuralModel>` including the PK and PD models. After defining the two dosing related variables, "\(D\)" (dose) and "\(tD\)" (time of dose), through the `<Variable>` element, the PK model is implemented as a piecewise algebraic formula:

```
<StructuralModel blkId="sml">
  <ct:Variable symbId="D"/>
  <ct:Variable symbId="tD"/>
```
The PD model is encoded as a single ODE for the drug effect, “E”, implemented using the `<DerivativeVariable>` element as shown below:
The initial time is 0, and is omitted here as it is the default value if not stated otherwise. The initial value for \( E, E_0 \), declared in the parameter model \( pm1 \), is unknown and has to be estimated.

The last component of the model definition is the \(<\text{ObservationModel}>\). Two continuous models are set, for the observations "CONC_obs" and "E_obs", with combined and additive errors respectively, and block identifiers "om1" and "om2". In both cases the models are well-structured and are thus described via the structured model type:

\[
<\text{ObservationModel blkId="om1"}>
<\text{ContinuousData}>
<\text{PopulationParameter symbId="add"} />
<\text{PopulationParameter symbId="prop"} />
<\text{RandomVariable symbId="epsilon_CONC"}>
<\text{VariabilityReference}>
<\text{SymbRef blkIdRef="vm2" symbIdRef="residual"} />
<\text{VariabilityReference}>
<\text{Distribution}>
<\text{ProbOnto name="StandardNormal1"} />
<\text{Distribution}>
</\text{RandomVariable}>
</\text{Standard symbId="CONC_obs"}>
<\text{Output}>
<\text{SymbRef blkIdRef="s1" symbIdRef="CONC"} />
</\text{Output}>
<\text{ErrorModel}>
<\text{Assign}>
<\text{FunctionCall}>
<\text{SymbRef symbIdRef="combinedErrorModel"} />
</\text{FunctionCall}>
</\text{Assign}>
</\text{ContinuousData}>
</\text{ObservationModel}>
As in the parameter model, the random errors "epsilon_CONC" and "epsilon_E" are implemented as standard normally distributed random variables, in this case referencing the "vm2" variability model. Note that the encoding of this distribution is notably shorter compared to other distributions due to the fact that only its code name "StandardNormal1" needs to be specified. For the residual errors, in both cases, we make use of the functions defined at the beginning of the PharmML file, "combinedErrorModel" and "constantErrorModel", by referencing suitable error parameter symbol identifiers to the function arguments. The <ResidualError> element, which refers to the random errors above, completes the observation model definitions.
The next section of the PharmML file, <TrialDesign>, consists here simply of specifying the relevant dataset, its columns, and their mapping to the corresponding model elements. The following snippet shows the dataset declaration:

```xml
<ExternalDataSet toolName="NONMEM" oid="NMoid">
  <ColumnMapping>
    <!-- Omitted here, see description below -->
  </ColumnMapping>
  <ds:DataSet>
    <ds:Definition>
      <ds:Column columnId="ID" columnType="id" valueType="string" columnNum="1"/>
      <ds:Column columnId="TIME" columnType="idv" valueType="real" columnNum="2"/>
      <ds:Column columnId="WT" columnType="covariate" valueType="real" columnNum="3"/>
      <ds:Column columnId="SEX" columnType="covariate" valueType="int" columnNum="4"/>
      <ds:Column columnId="DOSE" columnType="reg" valueType="real" columnNum="5"/>
      <ds:Column columnId="DVID" columnType="dvid" valueType="int" columnNum="6"/>
      <ds:Column columnId="DV" columnType="dv" valueType="real" columnNum="7"/>
      <ds:Column columnId="MDV" columnType="mdv" valueType="int" columnNum="8"/>
      <ds:Column columnId="TDOSE" columnType="reg" valueType="real" columnNum="9"/>
    </ds:Definition>
    <ds:ExternalFile oid="dataOid">
      <ds:path>warfarin1.csv</ds:path>
    </ds:ExternalFile>
  </ds:DataSet>
</ExternalDataSet>
```

The columns are defined using the attributes `columnId` to declare their identifiers and `columnType` to declare their use, if required, while `valueType` specifies the allowed values and `columnNum` the column number. The `<ExternalFile>` tag carries the dataset path, relative to the PharmML file and optionally its format and/or delimiter information.

The `<ColumnMapping>` elements, skipped in the snippet above, are used to map the dataset columns to the related model elements. E.g., the subject column, “ID”, (`columnType="id"`) of the dataset is mapped to the individual/subject level, “indiv”, of the parameter related variability model; the independent variable column, “TIME”, (`columnType="idv"`) to the “t” symbol; the `covariate` columns to the related covariates declared in the covariate model, etc.:

```xml
<ColumnMapping>
  <ds:ColumnRef columnIdRef="ID"/>
</ColumnMapping>
<ColumnMapping>
  <ds:ColumnRef columnIdRef="TIME"/>
</ColumnMapping>
<ColumnMapping>
  <ds:ColumnRef columnIdRef="WT"/>
</ColumnMapping>
<ColumnMapping>
  <ds:ColumnRef columnIdRef="SEX"/>
</ColumnMapping>
```
A conditional mapping of the dependent variable column, “DV”, (columnType=’dv’) is needed as well: with the <MultipleDVMapping> element, the value in the “DVID” column value maps the PK and PD observed values to the model variables “CONC_obs” and “E_obs”, respectively.

The very last section, <ModellingSteps>, holds in this case the information required for the specification of the estimation task. An optional <SoftwareSettings> element gives the opportunity to specify an external dataset file with tool specific settings (e.g., Monolix uses XML files with numeric settings). If one would like to pass additional tool/algorithm settings explicitly, it is possible within the <Operation> tag at the end of the section. When the trial design and data records are sourced from a dataset, a reference to the relevant NONMEM-type dataset can be declared within the <ExternalDataSetReference> element:

In the next step, one can specify the parameters of the models. This includes the option to assign initial values and lower/upper bounds. It is also possible to avoid estimating a parameter by fixing its value, which is done by setting the attribute fixed to “true”:

```xml
<ParametersToEstimate>
  <!-- Tlag -->
  <ParameterEstimation>
    <ct:SymbRef blkIdRef="sm1" symbIdRef="pop_Tlag"/>
```
The <Operation> tag, mentioned above, lists the detailed sub-tasks the model is used for, e.g., estimation of population and individual parameters and estimation of the Fisher Information Matrix. Also, it gives the opportunity to specify some generic information about the sub-tasks, such as the target tool or any tool-agnostic detail, like the estimation algorithm:

```xml
<Operation order="1" opType="estPop">
  <Property name="software-tool-name">
    <ct:Assign>
      <ct:String>NONMEM</ct:String>
    </ct:Assign>
  </Property>
  <Property name="software-tool-version">
    <ct:Assign>
      <ct:String>7.2.0</ct:String>
    </ct:Assign>
  </Property>
  <Algorithm definition="FOCEI" />
</Operation>
<Operation order="2" opType="estFIM"/>
<Operation order="3" opType="estIndiv"/>
</EstimationStep>
</ModellingSteps>
</PharmML>

2 Warfarin PK/PD model 2 - Inter-occasion variability on PK parameters

This example aims at explaining how multiple levels of parameter related variability are handled in PharmML. For simplicity, we use the warfarin model presented in the previous section but consider the PK part only.

The variability model captures the structure of the random variability by defining its levels and their relationships. In the mixed-effect modelling approach, each model parameter changes its value from subject to subject, therefore a subject level of variability is usually included. When occasions are envisaged, the modeller may consider an additional variability level, located below the subject level: indeed, occasions are defined as the time intervals during which the parameters of a subject remain constant, while they can exhibit some variation from interval to interval, referred to as inter-occasion variability (IOV). The
following snippet shows how such a model is typically implemented using the `<Level>` and `<ParentLevel>` elements:

```xml
<VariabilityModel blkId="vm1" type="parameterVariability">
  <Level symbId="indiv" referenceLevel="true"/>
  <Level symbId="occasion">
    <ParentLevel>
      <ct:SymbRef symbIdRef="indiv"/>
    </ParentLevel>
  </Level>
</VariabilityModel>
```

The new variability level is expressed as an additional random effect, $\eta^{(1)}$, so that the extended parameter model for $V$ (see equation VEq) now reads:

$$
log(V) = log(V_{pop}) + \beta_W \log(W70) + \beta_{SEX1SEX=F} + \eta^{(0)}_V + \eta^{(1)}_V
$$

with $\eta^{(1)}_V \sim N(0,\gamma_V)$, where $\gamma_V$ is the standard deviation of the random effect for IOV on $V$, and $\eta^{(0)}_V$ is the new symbol for what was named $\eta_V$ in the previous example. Note that, to clearly distinguish the variability levels, we use in the PharmML code the symbol “kappa_V” for $\eta^{(1)}_V$. This new random effect reads now as follows:

```xml
<RandomVariable symbId="kappa_V">
  <ct:VariabilityReference>
    <ct:SymbRef blkIdRef="vm1" symbIdRef="occasion"/>
  </ct:VariabilityReference>
  <Distribution>
    <po:ProbOnto name="Normal1">
      <po:Parameter name="mean">
        <ct:Assign>
          <ct:Real>0</ct:Real>
        </ct:Assign>
      </po:Parameter>
      <po:Parameter name="stdev">
        <ct:Assign>
          <ct:SymbRef symbIdRef="gamma_V"/>
        </ct:Assign>
      </po:Parameter>
    </po:ProbOnto>
  </Distribution>
</RandomVariable>
```

In this snippet, we map “kappa_V” to the “occasion” variability level within the `<VariabilityReference>` element. The only other change compared to the Warfarin PK/PD model 1 example is that “gamma_V” and “gamma_CL” (the standard deviation of the random effect for IOV on $CL$) need to be declared as population parameters and to be estimated.

3 Warfarin PK/PD model 3 - Time-To-Event model for PD data

In this example, we consider hemorrhaging frequency data [O'Reilly_1963, O'Reilly_1968, Holford_1986] which can be collected to evaluate side-effects of warfarin administration, and we model this data using a
time-to-event (TTE) model. TTE models are completely described by the hazard function, \( h(t) \), which in this example is assumed to consist of two components. The first component is a constant baseline hazard, \( h_{\text{base}} \), equal for all subjects, the second one linearly depends on the continuous covariate \( TRT \), i.e. the warfarin dose (in mg), via the coefficient \( \beta_{TRT} \):

\[
h(t) = h_{\text{base}} \left( 1 + \beta_{TRT} \cdot TRT \right)
\]

The complete observation model for this data type reads in PharmML

```xml
<ObservationModel blkId="om1">
  <Discrete>
    <TimeToEventData>
      <EventVariable symbId="Y"/>
      <HazardFunction symbId="h">
        <ct:Assign>
          <math:Binop op="times">
            <ct:SymbRef blkIdRef="pm1" symbIdRef="h_base"/>
            <math:Binop op="plus">
              <ct:Real>1</ct:Real>
              <math:Binop op="times">
                <ct:SymbRef blkIdRef="pm1" symbIdRef="beta_TRT"/>
                <ct:SymbRef blkIdRef="cm1" symbIdRef="TRT"/>
              </math:Binop>
            </math:Binop>
          </math:Binop>
        </ct:Assign>
      </HazardFunction>
    </TimeToEventData>
  </Discrete>
</ObservationModel>
```

The following snippets shows the `<TrialDesign>` element:

```xml
<TrialDesign xmlns="http://www.pharmml.org/pharmml/0.8/TrialDesign">
  <ExternalDataSet toolName="NONMEM" oid="NMoid">
    <ColumnMapping>
      <ds:ColumnRef columnIdRef="TIME"/>
      <ct:SymbRef symbIdRef="t"/>
    </ColumnMapping>
    <ColumnMapping>
      <ds:ColumnRef columnIdRef="TRT"/>
      <ct:SymbRef blkIdRef="cm1" symbIdRef="TRT"/>
    </ColumnMapping>
    <ColumnMapping>
      <ds:ColumnRef columnIdRef="DV"/>
      <ct:SymbRef blkIdRef="om1" symbIdRef="Y"/>
    </ColumnMapping>
    <ds:DataSet>
      <ds:Definition>
        <ds:Column columnId="ID" columnType="id" valueType="string" columnNum="1"/>
        <ds:Column columnId="TIME" columnType="idv" valueType="real" columnNum="2"/>
        <ds:Column columnId="TRT" columnType="covariate" valueType="real" columnNum="3"/>
        <ds:Column columnId="DV" columnType="dv" valueType="real" columnNum="4"/>
      </ds:Definition>
      <ds:ExternalFile oid="dataOid">
        <ds:path>warfarin3.csv</ds:path>
      </ds:ExternalFile>
    </ds:DataSet>
  </ExternalDataSet>
</TrialDesign>
```
The mapping of the observed data, column “DV”, to the <EventVariable>, “Y”, declared in the observation model is similar to the previous examples.

The <ModellingSteps> block contains the typical parameter estimation structure providing the initial estimates for the parameters, in this case “h_base_year” and “beta_TRT”. The following snippet shows the complete section:

```
<ModellingSteps xmlns="http://www.pharmml.org/pharmml/0.8/ModellingSteps">
  <EstimationStep oid="estimStep1">
    <ExternalDataSetReference>
      <ct:OidRef oidRef="NMoid"/>
    </ExternalDataSetReference>
    <ParametersToEstimate>
      <ParameterEstimation>
        <ct:SymbRef blkIdRef="pm1" symbIdRef="h_base_year"/>
        <InitialEstimate fixed="false">
          <ct:Real>0.1</ct:Real>
        </InitialEstimate>
        <LowerBound>
          <ct:Real>0</ct:Real>
        </LowerBound>
      </ParameterEstimation>
      <ParameterEstimation>
        <ct:SymbRef blkIdRef="pm1" symbIdRef="beta_TRT"/>
        <InitialEstimate fixed="false">
          <ct:Real>0.4</ct:Real>
        </InitialEstimate>
        <LowerBound>
          <ct:Real>0</ct:Real>
        </LowerBound>
      </ParameterEstimation>
    </ParametersToEstimate>
    <Operation order="1" opType="estPop">
      <ct:Description>Estimate the population parameters.</ct:Description>
      <Property name="software-tool-name">
        <ct:Assign>
          <ct:String>MONOLIX</ct:String>
        </ct:Assign>
      </Property>
    </Operation>
    <Operation order="2" opType="estFIM"/>
    <Operation order="3" opType="estIndiv"/>
  </EstimationStep>
</ModellingSteps>
```

4 Warfarin PK/PD model 4 - Optimal design application

4.1 Trial Design

In this example, we use the parameter estimates obtained in the first warfarin example to optimise the design of a future study, given constraints on possible times and doses, through the Fedorov-Wynn algorithm (D-optimal design⁶). For this purpose we need to define a starting design and the design spaces.
The starting design is based on the empirical design and uses the most frequent time-points and doses effectively used in the original dataset. Drug administrations and activities are defined in the `<Interventions>` blocks. Here we assume a 100 mg dose (<DoseAmount>) administered as a bolus at time 0 (<DosingTimes>).

```xml
<Interventions>
  <Administration oid="admin1">
    <Bolus>
      <DoseAmount>
        <ct:SymbRef blkIdRef="sm1" symbIdRef="D"/>
        <ct:Assign>
          <ct:Real>100</ct:Real>
        </ct:Assign>
      </DoseAmount>
      <DosingTimes>
        <ct:Assign>
          <ct:Real>0</ct:Real>
        </ct:Assign>
      </DosingTimes>
    </Bolus>
  </Administration>
</Interventions>
```

The initial sampling schedule is defined as a sampling window named “protA”, with a vector of 7 time points between 0.001 and 120 hours as the <ObservationTimes> block. The associated element <Continuous> states that samples are to be taken from the predictions of CONC_obs (defined in the <ObservationModel> within the <ModelDefinition>). A similar block is defined for the predictions of $E_{obs}$ (not shown).

```xml
<Observations>
  <Observation oid="protA">
    <ObservationTimes>
      <ct:Assign>
        <ct:Vector>
          <ct:VectorElements>
            <ct:Real>0.0001</ct:Real>
            <ct:Real>3</ct:Real>
            <ct:Real>24</ct:Real>
            <ct:Real>36</ct:Real>
            <ct:Real>72</ct:Real>
            <ct:Real>96</ct:Real>
            <ct:Real>120</ct:Real>
          </ct:VectorElements>
        </ct:Vector>
      </ct:Assign>
      <Continuous>
        <ct:SymbRef blkIdRef="om1" symbIdRef="CONC_obs"/>
      </Continuous>
    </ObservationTimes>
  </Observation>
</Observations>
```

We assume that in the initial design all subjects receive the same interventions and sampling schedule, therefore only one arm, “arm1”, is defined in the <Arms> element below:

```xml
<Arms>
  <Arm oid="arm1">
    <ArmSize>
      <ct:Assign>
        <ct:Real>32</ct:Real>
      </ct:Assign>
    </ArmSize>
    <InterventionSequence>
```
We assume to optimise on doses, amongst the range of doses from the empirical design, and on sampling
times, taken from a discrete set spanning 0 to 120 hours. The following snippet shows the design space for
the sampling times for warfarin concentrations:

```
<DesignSpaces>
  <DesignSpace>
    <ObservationRef oidRef="protA"/>
    <ObservationTimes>
      <ct:Assign>
        <ct:Vector>
          <ct:VectorElements>
            <ct:Real>0.0001</ct:Real>
            <ct:Real>0.5</ct:Real>
            <ct:Real>1</ct:Real>
            <ct:Real>1.5</ct:Real>
            <ct:Real>2</ct:Real>
            <ct:Real>3</ct:Real>
            <ct:Real>6</ct:Real>
            <ct:Real>9</ct:Real>
            <ct:Real>12</ct:Real>
            <ct:Real>24</ct:Real>
            <ct:Real>36</ct:Real>
            <ct:Real>48</ct:Real>
            <ct:Real>72</ct:Real>
            <ct:Real>96</ct:Real>
            <ct:Real>120</ct:Real>
          </ct:VectorElements>
        </ct:Vector>
      </ct:Assign>
    </ObservationTimes>
  </DesignSpace>
</DesignSpaces>
```

The constraints for the optimisation again reflect the original design: we assume each subject will contribute
with 7 samples per response, and the total number of subjects is fixed to 32 (as defined in `<ArmSize>` for
`arm1`):

```
<DesignSpace>
  <ObservationRef oidRef="protA"/>
  <NumberSamples>
    <ct:Assign>
      <ct:Real>7</ct:Real>
    </ct:Assign>
  </NumberSamples>
</DesignSpace>
```

4.2 Modelling Steps - Design optimisation

Design optimisation relies on prior values for the parameters. We used the parameter estimates from the
estimation performed in warfarin 1 to fill in the corresponding values for each parameter (estimation with
NONMEM). Optimal design is performed in PFIM in this example. Because PFIM does not handle
breakpoints in the model, we assume that $T_{lag}$ is fixed to its estimated value (0.858) without interindividual variability. $I_{max}$ is also fixed, to 1. Finally, the covariate model is ignored for the optimisation (only the population values are considered) as PFIM does not take into account models with continuous covariates.

Task definition and settings are defined in <ModellingSteps>. We use the Fedorov-Wynn algorithm to optimise within the set of allowed time-points.

5. Minimal model of glucose kinetics (re-parameterised)

The minimal model is a simplified model of glucose kinetics and insulin action. It was designed for the determination of insulin sensitivity form an intravenous glucose tolerance test. The model described and implemented here and the data we used have been derived from the literature (with different papers for the model\textsuperscript{8–10} and for the parameter values and the data\textsuperscript{11}). Details on the mathematical implementation can be found in the DDMoRe model repository.

5.1 Model description

The structural model consists of two ODE's, one for plasma glucose concentration $G$ and one for insulin concentration at the site of action, $Z$. The right hand side for the definition of $Z$'s derivative refers to a variable, $I$ (plasma insulin concentration), which is obtained from the piecewise linear interpolation of the measurements with respect to the independent variable, the time $t$. This is a single-subject model so no interindividual variability is included in the model. The observed variable, $G_{obs}$, is related through an additive residual error to the variable $G$.

5.2 Trial Design and Modelling Steps

In this implementation an intravenous bolus of glucose is given at time zero. The amount is fixed, 300 mg, but scaled to the body weight, $wgt$. It is also divided by the glucose volume of distribution, $V$, in order to conform to the physical nature of $G$, which is a concentration rather than an amount. Similarly, a fixed but $V$-scaled infusion (i.e., a piecewise constant additive term) is also defined on $G$. In order to match the data\textsuperscript{11}, the glucose infusion is assumed here to be constantly null, but may be different from zero when describing other experiments.

The final equation for plasma glucose concentration then reads:

$$\frac{dG(t)}{dt} = -(SG + SI Z) G + SG G_b + \frac{Inf(t)}{V} + \delta(0) \frac{bolus \, wgt}{V}$$
with $G(0) = G_b$. $SI$ and $SG$ are the equation parameters, $\delta(0)$ is the Dirac delta function centered at zero (which adds one to the value of $G$ at time zero), $bolus$ represents the bolus amount, $Inf(t)$ represents the infusion, and $G_b$ is the basal plasma glucose concentration. The latter is part of a set of covariates to be measured during the experiment, including also the basal plasma insulin, $l_b$, and the body weight $wgt$.

Before entering the details of the PharmML implementation (below), it is worth noting here that the mathematical description of infusion and bolus is not implemented within the structural model, but within the trial design definition, which deals with the model inputs. PharmML was designed in order to easily and consistently cope with scaled intravenous boluses and infusions, and more generally with any inputs to any ODE equation, which may represent compartment amounts (as usual in the pharmacometrics arena) but also other quantities, such as concentrations and electrical currents.

Besides the equation parameters, the bolus, the infusion and the three covariates, another input is required by the minimal model: the set of measurements for $I$, to be used by the structural model as explained above.

Even if the minimal model was introduced for estimation purposes, it can be also used to simulate new data. Our implementation describes both tasks: for this reason, the definition of the trial design includes both a dataset recording the measurements of “$G\_obs$”, to be used to estimate the model parameters, and the nominal observation times, to be used to simulate “$G\_obs$” with the given parameter values.

5.4 PharmML implementation

With respect to the other examples provided in this tutorial, the minimal model introduces a few different or additional PharmML features of interest. The first one is the absence of blocks depicting covariate or parameter models, or variability models with type=”parameterVariability”, as these are not needed for a single-subject model.

The second feature is the use of the <Interpolation> tag to describe the variable “$I$” as the linear interpolation of the insulin measurements:

```xml
<StructuralModel blkId="mmstruct">
  <!-- parameters definition -->
  <ct:Variable symbId="I" regressor="yes">
    <ct:Assign>
      <ct:Interpolation>
        <ct:Algorithm>linear</ct:Algorithm>
        <ct:InterpIndepVar>
          <ct:SymbRef symbIdRef="t"/>
        </ct:InterpIndepVar>
      </ct:Interpolation>
    </ct:Assign>
  </ct:Variable>
</StructuralModel>
```
The measurements are provided in the `<TrialDesign>` block through a `<LookupTable>` element, in which the columns “TIME” and “INS” of a provided dataset are mapped into “t” and “I” in the model:

```xml
<TrialDesign>
  <Observations>
    <LookupTable>
      <ColumnMapping>
        <ds:ColumnRef columnIdRef="TIME"/>
        <ct:SymbRef symbIdRef="t"/>
      </ColumnMapping>
      <Target inputTarget="variable">
        <ColumnMapping>
          <ds:ColumnRef columnIdRef="INS"/>
          <ct:SymbRef blkIdRef="mmstruct" symbIdRef="I"/>
        </ColumnMapping>
      </Target>
    </LookupTable>
  </Observations>
  <ds:DataSet>
    <ds:Definition>
      <ds:Column columnId="TIME" columnType="idv" valueType="real" columnNum="1"/>
      <ds:Column columnId="INS" columnType="dv" valueType="real" columnNum="2"/>
    </ds:Definition>
    <ds:Table>
      <ds:Row><ct:Real>0</ct:Real><ct:Real>5</ct:Real></ds:Row>
      <ds:Row><ct:Real>0.0058</ct:Real><ct:Real>4.7368</ct:Real></ds:Row>
      <ds:Row><ct:Real>160.1424</ct:Real><ct:Real>5.8</ct:Real></ds:Row>
      <ds:Row><ct:Real>180.0239</ct:Real><ct:Real>5.2171</ct:Real></ds:Row>
    </ds:Table>
  </ds:DataSet>
</TrialDesign>
```

The third feature is the description of the trial design through a single-subject explicit design. Its elements are the `<Interventions>`, the `<Observations>` and the `<Covariates>` blocks. The `<Interventions>` block describes two administrations. The first one is a `<Bolus>` scaled by two structural model parameters, “wgt” and “V”, and given at time 0:

```xml
<Administration oid="bol1">
  <Bolus>
    <DoseAmount inputTarget="derivativeVariable">
      <ct:SymbRef blkIdRef="mmstruct" symbIdRef="G"/>
    </ct:Assign>
    <math:Binop op="divide">
      <math:Binop op="times">
        <ct:Real>300</ct:Real>
        <ct:SymbRef blkIdRef="mmstruct" symbIdRef="wgt"/>
      </math:Binop>
      <ct:SymbRef blkIdRef="mmstruct" symbIdRef="V"/>
    </math:Binop>
    <ct:Assign>
      <DoseAmount/>
    </ct:Assign>
    <DosingTimes>
      <ct:Assign>
        <ct:Real>0</ct:Real>
      </ct:Assign>
    </DosingTimes>
  </Bolus>
</Administration>
```
The second administration is an <Infusion> with vectors provided for the <DoseAmount>, the <DosingTimes> and the <Rate> (note that the dose amount is also scaled, and that times and rates in this case are just zero):

```xml
<Administration oid="inf1">
  <Infusion>
    <DoseAmount>
      <ct:SymbRef blkIdRef="mmstruct" symbIdRef="G"/>
      <ct:Assign>
        <math:Binop op="divide">
          <ct:SymbRef symbIdRef="inf1Vector"/>
          <ct:SymbRef blkIdRef="mmstruct" symbIdRef="V"/>
        </math:Binop>
      </ct:Assign>
    </DoseAmount>
    <DosingTimes>
      <ct:Assign>
        <ct:Vector>
          <ct:VectorElements>
            <ct:Real>0</ct:Real>
          </ct:VectorElements>
        </ct:Vector>
      </ct:Assign>
    </DosingTimes>
    <Rate>
      <ct:Assign>
        <ct:Vector>
          <ct:VectorElements>
            <ct:Real>0</ct:Real>
          </ct:VectorElements>
        </ct:Vector>
      </ct:Assign>
    </Rate>
  </Infusion>
</Administration>
```

The <Observations> element includes the <LookupTable>, the <IndividualObservations> and the <Observation> blocks. The <LookupTable> was introduced above. The <IndividualObservations oid="mmobsdata"> block creates a new object, “mmobsdata”, to be used for estimation purposes, mapping the columns “TIME” and “GLU” of a provided dataset to “t” and “G_obs” in the model:

```xml
<IndividualObservations oid="mmobsdata">
  <ColumnMapping>
    <ds:ColumnRef columnIdRef="TIME" /><ct:SymbRef symbIdRef="t"/>
  </ColumnMapping>
  <ColumnMapping>
    <ds:ColumnRef columnIdRef="GLU" />
    <ct:SymbRef blkIdRef="mmobs" symbIdRef="G_obs"/>
  </ColumnMapping>
  <ds:DataSet>
    <ds:Definition>
      <ds:Column columnId="TIME" columnType="idv" valueType="real" columnNum="1"/>
      <ds:Column columnId="GLU" columnType="dv" valueType="real" columnNum="2"/>
    </ds:Definition>
    <ds:Table>
      <ds:Row><ct:Real>1.8634</ct:Real><ct:Real>264.7367</ct:Real></ds:Row>
      <ds:Row><ct:Real>4.2158</ct:Real><ct:Real>216.0106</ct:Real></ds:Row>
      <ds:Row><ct:Real>5.9858</ct:Real><ct:Real>196.7738</ct:Real></ds:Row>
      <!-- other rows omitted -->
    </ds:Table>
  </ds:DataSet>
</IndividualObservations>
```
The `<Observation oid="mmobssim">` block creates instead an object, “mmobssim”, to be used for simulation purposes, containing a vector of observation times and the reference to the continuous response “G_obs”:

```
<Observation oid="mmobssim">
    <ObservationTimes>
        <ct:Assign>
            <ct:Vector>
                <ct:VectorElements>
                    <ct:Real>1.8634</ct:Real>
                    <ct:Real>4.2158</ct:Real>
                    <ct:Real>5.9858</ct:Real>
                    <ct:Real>7.9548</ct:Real>
                </ct:VectorElements>
            </ct:Vector>
        </ct:Assign>
    </ObservationTimes>
    <Continuous>
        <ct:SymbRef blkIdRef="mmobs" symbIdRef="G_obs"/>
    </Continuous>
</Observation>
```

Finally, the `<Covariates>` element contains a one-row dataset providing the values for the symbols “wgt”, “Gb” and “Ib” of the structural model.

```
<Covariates>
    <IndividualCovariates>
        <ColumnMapping>
            <ds:ColumnRef columnIdRef="WGT"/>
            <ct:SymbRef blkIdRef="mmstruct" symbIdRef="wgt"/>
        </ColumnMapping>
        <ColumnMapping>
            <ds:ColumnRef columnIdRef="GB"/>
            <ct:SymbRef blkIdRef="mmstruct" symbIdRef="Gb"/>
        </ColumnMapping>
        <ColumnMapping>
            <ds:ColumnRef columnIdRef="IB"/>
            <ct:SymbRef blkIdRef="mmstruct" symbIdRef="Ib"/>
        </ColumnMapping>
        <ds:DataSet>
            <ds:Definition>
                <ds:Column columnId="GB" columnType="covariate" valueType="real" columnNum="1"/>
                <ds:Column columnId="IB" columnType="covariate" valueType="real" columnNum="2"/>
                <ds:Column columnId="WGT" columnType="covariate" valueType="real" columnNum="3"/>
            </ds:Definition>
            <ds:Table>
                <ds:Row>
                    <ct:Real>67</ct:Real><ct:Real>5</ct:Real><ct:Real>39.1</ct:Real>
                </ds:Row>
            </ds:Table>
        </ds:DataSet>
    </IndividualCovariates>
</Covariates>
```

It is worth noting that these symbols are declared in the `<StructuralModel>` block as parameters, because they do not affect any other parameters, unlike covariates in population models. Also, in this example all required datasets, e.g. the one provided within the `<LookupTable>` block, are not provided through an `<ExternalFile>` element, rather via a `<Table>` block explicitly defining the table elements row by row.

The last noteworthy feature of this example is the inclusion in the `<ModellingSteps>` element of both `<EstimationStep>` and `<SimulationStep>`. Each of them uses the `<ObservationsReference>` element to refer
to the "mmobsdata" or "mmobssim" object respectively, both declared within the <TrialDesign> block. For example:

```xml
<mstep:ModellingSteps>
  <mstep:EstimationStep oid="mmest">
    <mstep:ObservationsReference>
      <ct:OidRef oidRef="mmobsdata"/>
    </mstep:ObservationsReference>
    <mstep:ParametersToEstimate>
      <mstep:ParameterEstimation>
        <ct:SymbRef blkIdRef="mmstruct" symbIdRef="SI"/>
        <mstep:InitialEstimate fixed="false">
          <ct:Real>5.4e-4</ct:Real>
        </mstep:InitialEstimate>
      </mstep:ParameterEstimation>
      <mstep:ParameterEstimation>
        <ct:SymbRef blkIdRef="mmstruct" symbIdRef="SG"/>
        <mstep:InitialEstimate fixed="false">
          <ct:Real>.047</ct:Real>
        </mstep:InitialEstimate>
      </mstep:ParameterEstimation>
      <mstep:ParameterEstimation>
        <ct:SymbRef blkIdRef="mmstruct" symbIdRef="lambda"/>
        <mstep:InitialEstimate fixed="false">
          <ct:Real>.062</ct:Real>
        </mstep:InitialEstimate>
      </mstep:ParameterEstimation>
      <mstep:ParameterEstimation>
        <ct:SymbRef blkIdRef="mmstruct" symbIdRef="V"/>
        <mstep:InitialEstimate fixed="false">
          <ct:Real>64</ct:Real>
        </mstep:InitialEstimate>
      </mstep:ParameterEstimation>
      <mstep:ParameterEstimation>
        <ct:SymbRef blkIdRef="mnmobs" symbIdRef="alpha"/>
        <mstep:InitialEstimate fixed="false">
          <ct:Real>2</ct:Real>
        </mstep:InitialEstimate>
      </mstep:ParameterEstimation>
    </mstep:ParametersToEstimate>
  </mstep:EstimationStep>
</mstep:ModellingSteps>

The <EstimationStep> block concludes with the <Operation> element holding the target tool settings and estimation algorithm related information:

```xml
<mstep:Operation order="1" opType="estIndiv">
  <mstep:Property name="software-tool-name">
    <ct:Assign>
      <ct:String>NONMEM</ct:String>
    </ct:Assign>
  </mstep:Property>
  <mstep:Property name="software-tool-version">
    <ct:Assign>
      <ct:Real>7.2</ct:Real>
    </ct:Assign>
  </mstep:Property>
  <mstep:Property name="maxEval">
    <ct:Assign>
      <ct:Int>9999</ct:Int>
    </ct:Assign>
  </mstep:Property>
  <mstep:Property name="postHoc">
    <ct:Assign>
      <ct:String>False</ct:String>
    </ct:Assign>
  </mstep:Property>
  <mstep:Property name="NM">
    <ct:Assign>
      <ct:String>NOABORT</ct:String>
    </ct:Assign>
  </mstep:Property>
</mstep:Operation>
```
Contrary to the `<EstimationStep>` block, the `<SimulationStep>`, not described in the previous examples of this tutorial, gives the values of the model inputs not provided by the trial design via the `<VariableAssignment>` elements, e.g.:

```xml
<mstep:SimulationStep oid="mmsim">
  <mstep:ObservationsReference>
    <ct:OidRef oidRef="mmobssim"/>
  </mstep:ObservationsReference>
  <ct:VariableAssignment>
    <ct:SymbRef blkIdRef="mmstruct" symbIdRef="SI"/>
    <ct:Assign>
      <ct:Real>5.4e-4</ct:Real>
    </ct:Assign>
  </ct:VariableAssignment>
</mstep:SimulationStep>
```

Note that the inputs assigned here are the same which are estimated via the `<ParameterEstimation>` elements in the `<EstimationStep>` block.

Finally, as for the `<EstimationStep>` block, the `<Operation>` element specifies the software and the algorithm to be used to perform the task:

```xml
<mstep:Operation order="1" opType="simulatePK">
  <mstep:Property name="software-tool-name">
    <ct:Assign>
      <ct:String>NONMEM</ct:String>
    </ct:Assign>
  </mstep:Property>
  <mstep:Property name="software-tool-version">
    <ct:Assign>
      <ct:Real>7.2</ct:Real>
    </ct:Assign>
  </mstep:Property>
  <mstep:Algorithm definition="FO">
    <mstep:Property name="Seed">
      <ct:Assign>
        <ct:Int>123</ct:Int>
      </ct:Assign>
    </mstep:Property>
    <mstep:Property name="NM">
      <ct:Assign>
        <ct:String>ONLYSIM</ct:String>
      </ct:Assign>
    </mstep:Property>
  </mstep:Algorithm>
</mstep:Operation>
```

6. Mitotic oscillator model

In this final example, we show how PharmML can be used to describe models from the Systems Biology field, and in particular models encoded in SBML, the Systems Biology Markup Language. A large repository of such models is the Biomodels database, which as of October 2016 contains the implementations of
around 1500 models published in the literature. Around 600 of them have been manually curated to verify
they are syntactically correct and provide reproducible results when compared to the reference publication.
In addition, a large number of models have been included in this repository after automatically translation
from databases publishing metabolic, non-metabolic and whole genome metabolism pathways.

6.1 Model description

BIOMD0000000003 model was downloaded from the Biomodels database and automatically converted
from SBML to PharmML. The conversion processes involves reading an SBML model in jSBML\textsuperscript{12},
translating the SBML declarations to form an equation system and encoding those equations in
libPharmML for conversion into PharmML.

This model represents a minimal cascade model for the mitotic oscillator, and describes the cellular
mechanism regulating the onset of mitosis through the regulation of cdc2 kinase\textsuperscript{13}, first studied in early
amphibian embryos. In this model, the accumulation of a protein, called cyclin for its involvement in the cell
cycle, causes the activation of cdc2 kinase and the formation a complex, the maturation or M-phase
promoting factor. This complex triggers mitosis and cyclin degradation, which inactivates cdc2 kinase,
resetting the cell for a new division cycle.

Figure 1 shows a schematic of a minimal model describing this behaviour, and including a dual cycle
triggering the oscillation pattern. It involves 3 entities: cyclin, C, cyclin protease, X, and cdc2 kinase, M.
Cyclin C is assumed to be synthesized at a constant rate $v_i$, and its elimination is governed by the sum of a
first-order process with rate constant $k_d$ and a saturable process depending on both cyclin concentration
and cyclin protease X. The first cycle represents the activation and deactivation cycle of cdc2 kinase M,
triggered by cyclin, and the second cycle represents the activation and deactivation of cyclin protease X.
These two cycles involve conversion enzymes with concentrations assumed to be in excess and constant
over the duration of the observation period. The entire system can be described through the following set of
3 differential equations:

$$\frac{dC}{dt} = v_i - v_d X \frac{C}{K_d + C} - k_d C$$

$$\frac{dM}{dt} = \frac{C}{K_C + C} V_M1 \frac{1 - M}{K_1 + (1 - M)} - V_2 \frac{M}{K_2 + M}$$

$$\frac{dX}{dt} = M V_M3 \frac{1 - X}{K_3 + (1 - X)} - V_4 \frac{X}{K_4 + X}$$
Without loss of generality the equations for $M$ and $X$ have been scaled so that these variables represent the fraction of active cdc2 kinase and the fraction of active cyclin protease, respectively. $v_i$ represents the constant rate of cyclin synthesis, $k_d$ the rate constant for the non-specific degradation of cyclin, $v_d$ the maximum rate of cyclin degradation, $K_d$ and $K_c$ the Michaelis-Menten constants for cyclin degradation and activation, respectively. The normalised parameters $V_{M1}$, $V_{M3}$, $V_2$, $V_4$ and $K_j$ ($j = 1$ to $4$) characterise the kinetics of the enzymes involved in the two cycles of post-translational modifications. For consistency we also define $V_1 = V_{M1} C/(K_c+C)$ and $V_3 = M V_{M3}$ as time-varying parameters.

![Diagram of the minimal model for the mitotic oscillator](image)

**Figure 1:** minimal model for the mitotic oscillator

The PharmML implementation for the model definition just summarized is not detailed here as it simply includes differential equations.

### 6.2 Modelling steps

The model is used in a predictive manner to explore the behaviour of the system over time. The values of the parameters may be modified to assess the conditions giving rise to an oscillatory evolution; in this tutorial the values reported in the related publication are used$^{13}$: $v_i = 0.025$, $k_d = 0.01$, $v_d = 0.25$, $K_d = 0.02$, $K_C = 0.5$, $K_1 = K_2 = K_3 = K_4 = 0.005$, $V_{M1} = 3$, $V_2 = 1.5$, $V_{M3} = 1$, $V_4 = 0.5$. The initial values for the system are set to $C(0) = 0.01$, and $X(0) = M(0) = 0.01$.

There is no trial design in the original publications, and the trial design for the PharmML file was generated automatically via the converter. The section `<TrialDesign>` is very simple and consists in defining the observation times for a simulation task. Since the observation times are equal for the three state variables of interest ($C$, $M$, $T$), they are merged into one `<Observation>` tag, as the following snippet shows, and the states of the system are generated for a sequence of time values from 0 to 100 min:
As this last example on the PharmML implementation of a mitotic oscillator model shows, System Biology models do not impose any new requirements on PharmML compared to standard PK and PD models. Nor is size an issue, as PharmML can store any number of algebraic or differential equations. The same applies for Physiology-Based Pharmacokinetics (PBPK) models, nowadays largely used by the Pharmacometricians.

References

1. O’Reilly, R. A. & Aggeler, P. M. Studies on coumarin anticoagulant drugs. Initiation of warfarin therapy without a loading dose. *Circulation* **38**, 169–177 (1968).

2. O’Reilly, R. A., Aggeler, P. M. & Leong, L. S. Studies on the coumarin anticoagulant drugs: the pharmacodynamics of warfarin in man. *J. Clin. Invest.* **42**, 1542–1551 (1963).

3. Holford, N. H. Clinical pharmacokinetics and pharmacodynamics of warfarin. Understanding the dose-effect relationship. *Clin. Pharmacokinet.* **11**, 483–504 (1986).
4. Nagashima, R., O’Reilly, R. A. & Levy, G. Kinetics of pharmacologic effects in man: the anticoagulant action of warfarin. *Clin. Pharmacol. Ther.* **10**, 22–35 (1969).

5. Abbrecht, P. H., O'Leary, T. J. & Behrendt, D. M. Evaluation of a computer-assisted method for individualized anticoagulation: retrospective and prospective studies with a pharmacodynamic model. *Clin. Pharmacol. Ther.* **32**, 129–136 (1982).

6. Retout, S., Comets, E., Samson, A. & Mentré, F. Design in nonlinear mixed effects models: optimization using the Fedorov-Wynn algorithm and power of the Wald test for binary covariates. *Stat. Med.* **26**, 5162–5179 (2007).

7. Retout, S., Mentré, F. & Bruno, R. Fisher information matrix for non-linear mixed-effects models: evaluation and application for optimal design of enoxaparin population pharmacokinetics. *Stat. Med.* **21**, 2623–2639 (2002).

8. Bergman, R. N., Ider, Y. Z., Bowden, C. R. & Cobelli, C. Quantitative estimation of insulin sensitivity. *Am. J. Physiol.-Endocrinol. Metab.* **236**, E667–E677 (1979).

9. Bergman, R. N., Phillips, L. S. & Cobelli, C. Physiologic evaluation of factors controlling glucose tolerance in man: measurement of insulin sensitivity and beta-cell glucose sensitivity from the response to intravenous glucose. *J. Clin. Invest.* **68**, 1456–1467 (1981).

10. Mari, A. Assessment of insulin sensitivity with minimal model: role of model assumptions. *Am. J. Physiol.-Endocrinol. Metab.* **272**, E925–E934 (1997).

11. Finegood, D. T., Pacini, G. & Bergman, R. N. The insulin sensitivity index. Correlation in dogs between values determined from the intravenous glucose tolerance test and the euglycemic glucose clamp. *Diabetes* **33**, 362–368 (1984).

12. Rodriguez, N. *et al.* JSBML 1.0: providing a smorgasbord of options to encode systems biology models. *Bioinforma. Oxf. Engl.* **31**, 3383–3386 (2015).

13. Goldbeter, A. A minimal cascade model for the mitotic oscillator involving cyclin and cdc2 kinase. *Proc. Natl. Acad. Sci. U. S. Am. Proc. Natl. Acad. Sci. U. S. Am.* **88**, **88**, 9107, 9107–9111 (1991).