Modelling the Degradation of Sewer Pipelines in CareS:
The `GompitZ` Tool User's Guide – version 2.08

Yves Le Gat - Cemagref/REBX – October 2011

1. Introduction

The `GompitZ` tool is a software developed at the Hydraulics and Civil Engineering (REBX) Research Unit of the *Cemagref* Centre of Bordeaux (France), in the framework of the WP2.2 of the FP5 Programme CareS. This computer programme is written in C language (ANSI C90) ; the development is carried out on a Linux SlackWare 12.0 platform, using the GNU development tool GCC, under GNU public license (GPL) ; Code::Blocks is used to compile for Windows NT platform (Mingw32 is an alternative to produce Win32 object code under Linux).

`GompitZ` consists in 2 executable modules, `Gompcal` and `Gompred`, that are respectively designed for:

- By Stratum Model Calibration,
- Simulation of Rehabilitation Scenarios and Prediction of Future Pipeline Condition.

The underlying statistical model of GompitZ is a Non Homogeneous Markov Chain, formalised by a matrix of time varying state transition probabilities. These transition probabilities are functions of parameters which account for:

- the states concerned by the transition ("Alpha" parameters),
- explanatory factors proper to the pipeline and its environment assumed to influence the initial (at the laying out of the pipeline) condition ("Beta0" parameters),
- explanatory factors assumed to influence the deterioration speed ("Beta1" parameters),
- the standard deviation of the individual random effects ("Individual Frailty Factors" = IFF) distribution ("Sigma" parameter).

The model parameters are estimated by carrying out a by stratum (notion to be explained later in this document) Multi-state Mixed Gompit Analysis, i.e. an equivalent of Probit Analysis that uses the Gompertz distribution instead of the Gaussian one, and accounts for over-dispersion through the IFF distribution (as in usual mixed models). Multi-state Mixed Gompit Analysis technically consists in calibrating survival functions, i.e. functions giving the probability of a pipeline to be in a given condition or better, this probability being conditional on the values of the pipeline explanatory factors and its IFF. The parameter estimates are obtained by maximising the marginal likelihood of the sample of inspected pipes, i.e. the likelihood conditional on the IFF values integrated over the IFF possible values ; the IFF distribution is taken to be Gaussian with mean 0 and standard deviation Sigma. The model is hence designed to fully account for longitudinal data, i.e. repeated inspections of the same pipe ; the sample of inspections may then contain single or multiple inspections per pipe.

This document presents, in section 2, the input data structure for the `Gompcal` module, the `Gompcal` output files in section 3, the `Gompred` input files in section 4, the `Gompred` output files in section 5, the contents of the `GompitZ` distribution package, the syntax of compilation of the C codes and execution of the object codes in section 6.

2. Input Data for `Gompcal` module

2.1 Input Data - Pipe Condition and Description

The `GompitZ` tool expects an input file to be provided by the Rehab Manager that merges by pipe identifier the following two types of information:

- Pipeline Description Data, consisting in a pipeline identifier, and a list of fields giving the characteristics of the pipeline (material, diameter, installation year, length, depth, type of effluent, etc.), and of its environment (traffic above, type of soil occupation above, type of embedding soil, etc.),

- Inspection Record Data, consisting in a pipeline identifier, the inspection year, the condition class code, and a special variable called "weight" the meaning of which is explained later in this document.

The pipeline characteristics are variables that are called covariates when they are used as explanatory factors in a statistical model. It is possible to parameterize a model without any covariate. However an installation year must be given for inspected pipes in order to run the `Gompcal` module. Additionally, to run `Gompred`, the length of all pipes for which predictions are wanted must be given.

2.2 Strata

In statistical modelling, a stratum may be defined as a group of individuals that homogeneously respond to the factors driving the process they are subjected to. The input data must thus provide information about the strata which pipelines belong to. The notion of stratum is central for the model calibration process, as a separate parameter set is estimated for each stratum. `GompitZ` can handle strata with different numbers of covariates, as well as different numbers of observed states.

In matter of pipeline deterioration, strata may e.g. be defined by grouping materials that deteriorate according to the same physical and chemical mechanisms: cast iron, "cementitious" materials, plastic materials, brick, masonry, clay, etc..

2.3 Numerical Fields Format in Input Data File

All numerical fields are assumed to use the dot as decimal delimiter. The scientific notation is allowed. As mentioned later, the fields containing a year (installation or inspection) or a covariate status are integral, and must thus exclusively be composed of numerical characters (`0`, `1`, `2`, `3`, `4`, `5`, `6`, `7`, `8`, `9`).

2.4 Alphanumerical Fields in Input Data File

Alphanumerical fields are allowed to be composed of any ASCII printable character, except the one used as field delimiter (see below). They must be at most 49 character wide.

2.5 File Delimiter

`GompitZ` assumes all files it has to deal with have all their record fields delimited by the same character. The characters the most widely used as delimiter in database systems are the coma "`,`" or the semicolon "`;`". Any other ASCII printable character can however be used provided it is not used inside numerical (e.g. "+", "-" or decimal separator are prohibited) or alphanumerical fields. The horizontal tab character is prohibited as field delimiter. As mentioned later in section 6, the field delimiter must be specified by the user as an argument of the command that launches a `GompitZ` executable.

The semicolon is used in the examples given in this document.

2.6 Input Data File Headlines

The current `GompitZ` version assumes the input data file contains (`3+nbstra`) headlines, where `nbstra` stands for the number of strata.

The first row must list the theoretical condition labels, ordered from the best to the worst one. These conditions are theoretical in the sense they are defined by a standard ; some of them may however be unobserved in some strata of the available pipelines sample.

The second row must list the strata labels (any order is convenient).

The third row must list the labels of all the covariates susceptible to be used in any stratum (not necessarily in alphabetical order). One of these labels must be "`LENGTH`", and must correspond to the length of the pipelines, even if this variable is not used as a covariate. The pipeline length is indeed indispensable for simulating rehabilitation. Any categorical covariate must have its label prefixed with an underscore "_" (e.g. `_TYPE`). Any covariate the label of which is not "_" prefixed is considered as being numerical.

Next rows: one row per stratum, with the stratum label as its first field, and then a list of integers equal to `0`, `1`, `2` or `3`, that stand for the covariate status, and must then be listed in the same order as the covariates. The meaning of the status values is as follows:

- `0` = covariate not used in that stratum
- `1` = covariate used in that stratum as a factor influencing the initial deterioration state
- `2` = covariate used in that stratum as a factor influencing the deterioration speed
- `3` = covariate used in that stratum as a factor influencing both the initial deterioration state and the deterioration speed

The headlines of the sample file `obs.txt` provided with the current `GompitZ` distribution is shown below:

```
C4;C3;C2;C1
Cast Iron;Brick
DIAM;_TYPE;LENGTH;ROAD
Cast Iron;3;2;0;1
Brick;3;3;0;2
```

2.7 Input Data Files Rows

Each record must begin with the 6 following fields in that order:

- Stratum alphanumerical label (value must be one of those mentioned in the 2nd row of the headlines),
- Pipeline Identifier alphanumerical label,
- Installation Year (integer),
- Inspection Year (integer – void field if not inspected),
- Condition Class alphanumerical label (value must be one of those mentioned in the 1st row of the headlines - void field if not inspected),
- Weight of the pipeline for the calibration.

The meaning of the weight variable deserve some explanations. This variable is used by the model calibration process to modulate the relative importance of the observations. The weight should be equal to `1.0` for all inspected pipes if they are all assumed to be equally representative, or if their inspection are equally reliable. Weighted model calibration is a rigorous way of handling a sample of inspections that was not fully randomly collected, i.e. that suffers a bias of selection.
The weight can also be used for handling pseudo-continuous condition scales, i.e. cases where the outcome of inspection is not an integer (e.g. "1", "2", "3" or "4"), but can also be a decimal value (e.g. "2.4" or "3.6"); an inspection scored "2.4" must be reported in the input data file as **2 rows**:

- one row with condition equal to "2" and weight 0.6,
- and another row with condition equal to "3" and weight 0.4.

The decimal condition is therefore defined as 2 X 0.6 + 3 X 0.4 = 2.4.

The additional fields contain the covariate values ; these values must appear in the same order as the covariate labels they correspond to in the 3rd headline row. As mentioned earlier, at least the pipeline length must be given. Any numerical covariate (i.e. not "_" prefixed) value in data rows must be a valid numerical value ; otherwise and provided this value is required for calibration or prediction computations, the corresponding data row is discarded. In the `obs.txt` example, variables `DIAM`, `_TYPE`, `LENGTH` and `ROAD` are respectively discrete numerical

(values in {100, 200, ..., 900, 1000}), categorical alphanumerical (values in {Combined, Rain, Waste}), continuous numerical (values in [20.0, 80.0]) and discrete numerical (value = 0 or 1, according to absence / presence of the attribute "located under a road way"). The last covariate ROAD is an indicator variable and could equivalently be handled as a numerical or categorical covariate (should be labelled _ROAD in this last case). The data set is purely fictitious as it is only designed for program testing purpose.
Here are the first 5 data rows of obs.txt:

```
Cast Iron;Pipe0001;1870;2003;C1;1.0;600.0;Combined;32.0;0.0
Cast Iron;Pipe0002;1959;1991;C4;1.0;300.0;Combined;72.0;0.0
Cast Iron;Pipe0003;1888;1995;C1;1.0;200.0;Rain;79.0;1.0
Cast Iron;Pipe0004;1938;1991;C3;1.0;600.0;Combined;56.0;1.0
Brick;Pipe0005;1963;;;1.0;800.0;Waste;51.0;1.0
```

3. Gompcal output files

3.1 The Theoretical Model and the Parameter Estimation

As specified in the reference WP2.2 document entitled "Modelling the Degradation of Sewer Pipelines: The GompitZ Program", the GompitZ statistical tool is based on a Mixed Generalised Linear Regression Model, derived from the classical Probit Regression Model using the Gompertz probability distribution (alternative to the Gaussian distribution). The resulting model can be defined as a set of equations that enables the condition probabilities to be calculated for any age of the pipeline, conditionally on the covariates values and on the IFF, and possibly on the state observed at a previous inspection (namely the most recent one if more than one are reported). The model parameters are:

- as many "Alpha" parameters as there are many condition classes minus 1,
- "Beta0" and "Beta1" parameters - one per covariate with status = 1 or 2, two per covariate with status = 3 (and obviously zero per covariate with status = 0) ; there is always at least one Beta1 value estimated for the time variable,
- a "Sigma" parameter as the standard deviation of the 0 mean Gaussian distribution of the IFF.

The method used to calibrate the model is the maximisation of the log of the marginal likelihood of the observations (i.e. the natural logarithm of the joined probability of the condition of the inspected pipelines integrated over the IFF). The maximisation is worked out using the Levenberg-Marquardt algorithm, the gradient vector and Hessian matrix being computed by finite differences approximations ; the initial parameter vector is estimated by the Kaplan-Meyer method ; the integral of the conditional log-likelihood is approximated with a Gauss-Legendre quadrature.
The role of the covariates consists in modulating the survival functions in each condition, whereas the stratum label is used to define categories of pipelines, for each of which a separate calibration (i.e. parameters estimation) is carried out. If for instance the user chooses to consider the sole variables material and diameter, and to use their crossed levels to define strata, the resulting model has no covariates and the model predictions will be at the group level. A pure at group level analysis may also be obtained by considering no stratification and the sole variables material and diameter as covariates. If a stratification e.g. by material is chosen, and diameter, length and traffic are selected as covariates, the effect of the covariates are allowed to vary across materials ; the predictions are in this case at the pipeline level, in the sense that it is unlikely that 2 different pipelines could have the same values for the stratification variable and the covariates.
The theoretical reason for having added the random effect IFF in the version 2.00 of GompitZ lies in the frequent presence in practical datasets of observed conditions that sensitively depart from the ones that have the highest theoretical probabilities conditionally on the covariate values ; it is e.g. not rare to observe rather old pipes in a surprisingly good state. The IFF can then be seen as a correction on the deterioration speed that account for such an over-dispersed observation.

3.2 The `calibr.txt` output file

`Gompcal` documents the calibration results by printing in `calibr.txt` for each stratum, the number of pipelines described, the number of inspections reported, in total and split by observed condition, the final covariance matrix, and a table of the parameter estimates, along with their estimation standard errors and a Wald Chi-Square test for their significance:

- Concerning the Alpha's, they are tested against the value of the better adjacent condition class (against 0 for the Alpha of the best condition class), providing so a test of the relevance of the condition class definition (2 adjacent condition classes having alpha values not significantly different are likely to occur nearly at the same age).
- All Beta values are tested against 0, because the 0 value means no effect on the survival functions.

There is currently no automated covariates selection process implemented ; it is hence advisable to try first a complete set of available covariates, and deselect step by step those which appear to be not significant. It is particularly irrelevant to keep in the covariate set a not significant factor either influencing the initial condition and whose magnitude of estimate exceeds the first Alpha estimate, or influencing the deterioration speed and whose magnitude of estimate exceeds the "T" estimate ; `Gompred` may then produce questionable predictions.

If in a given stratum, either all observed states are identical, or no valid inspection data is present in the input data set, or all reported inspections concern pipe of the same age, or too few inspections are reported for a given condition class, no calibration is carried out for this stratum, and `calibr.txt` only contains a warning message.

```
GompitZ v2.07 - Calibration results


*** Stratum Clay ***
10 (total weight = 10.00) pipelines described in this stratum

0 (total weight = 0.00) inspections reported:
Warning: Stratum void of inspection data - Calibration aborted

*** Stratum Cast Iron ***
4065 (total weight = 4065.00) pipelines described in this stratum

3647 (total weight = 3647.00) inspections reported:
720 (total weight = 720.00) of which in condition C4
916 (total weight = 916.00) of which in condition C3
1225 (total weight = 1225.00) of which in condition C2
786 (total weight = 786.00) of which in condition C1

Convergence achieved in 47 iterations
Log-Likelihood = -3166.450737

Final Covariance Matrix =
{ +2.364e-02 , +2.472e-02 , +2.720e-02 , -3.223e-05 , -3.251e-03 , -6.417e-03 , +8.571e-06 , +1.700e-04 }
{ +2.472e-02 , +3.090e-02 , +3.535e-02 , -3.078e-05 , -3.704e-03 , -7.967e-03 , +8.839e-06 , +1.035e-04 }
{ +2.720e-02 , +3.535e-02 , +5.135e-02 , -2.904e-05 , -4.327e-03 , -9.965e-03 , +9.063e-06 , -3.522e-04 }
{ -3.223e-05 , -3.078e-05 , -2.904e-05 , +6.769e-08 , -6.616e-07 , +7.956e-06 , -1.635e-08 , -4.555e-08 }
{ -3.251e-03 , -3.704e-03 , -4.327e-03 , -6.616e-07 , +6.144e-03 , +1.551e-04 , +2.346e-07 , +2.821e-05 }
{ -6.417e-03 , -7.967e-03 , -9.965e-03 , +7.956e-06 , +1.551e-04 , +3.028e-03 , -3.474e-06 , -1.700e-05 }
{ +8.571e-06 , +8.839e-06 , +9.063e-06 , -1.635e-08 , +2.346e-07 , -3.474e-06 , +6.064e-09 , +2.529e-08 }
{ +1.700e-04 , +1.035e-04 , -3.522e-04 , -4.555e-08 , +2.821e-05 , -1.700e-05 , +2.529e-08 , +1.640e-04 }

Parameter estimates and Wald Chi2 tests
-----------------------------------------------------------------------------------------------
Label                                              Estimate     Std. Error  DF  Chi2        Pr>Chi2
-----------------------------------------------------------------------------------------------
Alpha(C4) (vs 0)                                   -1.1547e+00  1.5374e-01  1   5.6413e+01  0.000000
Alpha(C3) (vs Alpha(C4))                           -3.2318e+00  1.7578e-01  1   8.4605e+02  0.000000
Alpha(C2) (vs Alpha(C3))                           -6.3562e+00  2.2660e-01  1   8.4573e+02  0.000000
DIAM (vs 0)                                        -1.7599e-03  2.6017e-04  1   4.5758e+01  0.000000
ROAD (vs 0)                                        +3.0056e-01  7.8386e-02  1   1.4702e+01  0.000126
T (vs 0)                                           -2.9890e+00  5.5030e-02  1   2.9503e+03  0.000000
T*DIAM (vs 0)                                      +4.5103e-04  7.7874e-05  1   3.3544e+01  0.000000
Sigma (vs 0)                                       +3.5279e-01  1.2806e-02  1   7.5892e+02  0.000000
-----------------------------------------------------------------------------------------------
```

```
*** Stratum Asbestos ***
10 (total weight = 10.00) pipelines described in this stratum

0 (total weight = 0.00) inspections reported:
Warning: Stratum void of inspection data - Calibration aborted

*** Stratum Brick ***
3177 (total weight = 3177.00) pipelines described in this stratum

2839 (total weight = 2839.00) inspections reported:
1342 (total weight = 1342.00) of which in condition C4
723 (total weight = 723.00) of which in condition C2
774 (total weight = 774.00) of which in condition C1

Convergence achieved in 53 iterations
Log-Likelihood = -1897.810240

Final Covariance Matrix =
{ +7.864e-02 , +8.090e-02 , -7.359e-05 , -4.699e-02 , -3.915e-02 , -2.237e-02 , +2.008e-05 , +1.199e-02 ,
+9.859e-03 , +9.107e-04 , +3.863e-04 }
{ +8.090e-02 , +8.995e-02 , -7.026e-05 , -4.778e-02 , -3.830e-02 , -2.403e-02 , +2.014e-05 , +1.208e-02 ,
+9.845e-03 , +8.067e-04 , -1.608e-06 }
{ -7.359e-05 , -7.026e-05 , +1.660e-07 , +1.117e-05 , -1.239e-06 , +1.799e-05 , -3.832e-08 , -3.298e-06 ,
+1.704e-06 , -2.529e-07 , -2.292e-07 }
{ -4.699e-02 , -4.778e-02 , +1.117e-05 , +6.429e-02 , +4.105e-02 , +1.239e-02 , -2.840e-06 , -1.788e-02 ,
-1.056e-02 , -1.829e-04 , +1.665e-05 }
{ -3.915e-02 , -3.830e-02 , -1.239e-06 , +4.105e-02 , +1.106e-01 , +8.994e-03 , +2.436e-06 , -1.068e-02 ,
-2.487e-02 , -2.226e-04 , -7.221e-05 }
{ -2.237e-02 , -2.403e-02 , +1.799e-05 , +1.239e-02 , +8.994e-03 , +8.821e-03 , -7.277e-06 , -3.691e-03 ,
-3.184e-03 , -9.608e-04 , -1.950e-04 }
{ +2.008e-05 , +2.014e-05 , -3.832e-08 , -2.840e-06 , +2.436e-06 , -7.277e-06 , +1.268e-08 , +5.905e-07 ,
-8.281e-07 , +3.505e-07 , +1.716e-07 }
{ +1.199e-02 , +1.208e-02 , -3.298e-06 , -1.788e-02 , -1.068e-02 , -3.691e-03 , +5.905e-07 , +6.410e-03 ,
+3.432e-03 , -3.383e-05 , -6.492e-05 }
{ +9.859e-03 , +9.845e-03 , +1.704e-06 , -1.056e-02 , -2.487e-02 , -3.184e-03 , -8.281e-07 , +3.432e-03 ,
+7.159e-03 , +2.447e-04 , +9.319e-05 }
{ +9.107e-04 , +8.067e-04 , -2.529e-07 , -1.829e-04 , -2.226e-04 , -9.608e-04 , +3.505e-07 , -3.383e-05 ,
+2.447e-04 , +9.967e-04 , +9.305e-05 }
{ +3.863e-04 , -1.608e-06 , -2.292e-07 , +1.665e-05 , -7.221e-05 , -1.950e-04 , +1.716e-07 , -6.492e-05 ,
+9.319e-05 , +9.305e-05 , +3.043e-04 }

Parameter estimates and Wald Chi2 tests
-------------------------------------------------------------------------------------------------
Label                                          Estimate     Std. Error  DF  Chi2        Pr>Chi2
-------------------------------------------------------------------------------------------------
Alpha(C4) (vs 0)                               -2.0593e+00  2.8042e-01  1   5.3928e+01  0.000000
Alpha(C2) (vs Alpha(C4))                       -4.0967e+00  2.9991e-01  1   6.1245e+02  0.000000
DIAM (vs 0)                                    -2.7097e-03  4.0744e-04  1   4.4228e+01  0.000000
_TYPE[Combined] (vs Waste)                     +4.1116e-01  2.5355e-01  1   2.6296e+00  0.104889
_TYPE[Rain] (vs Waste)                         -6.8701e-01  3.3252e-01  1   4.2687e+00  0.038821
T (vs 0)                                       -3.5658e+00  9.3921e-02  1   1.4414e+03  0.000000
T*DIAM (vs 0)                                  +7.3472e-04  1.1262e-04  1   4.2559e+01  0.000000
T*_TYPE[Combined] (vs Waste)                   -2.8725e-01  8.0060e-02  1   1.2873e+01  0.000333
T*_TYPE[Rain] (vs Waste)                       +4.2900e-01  8.4610e-02  1   2.5708e+01  0.000000
T*ROAD (vs 0)                                  +4.2627e-01  3.1571e-02  1   1.8230e+02  0.000000
Sigma (vs 0)                                   +3.2737e-01  1.7444e-02  1   3.5221e+02  0.000000
-------------------------------------------------------------------------------------------------


*** Stratum Concrete ***
10 (total weight = 10.00) pipelines described in this stratum

0 (total weight = 0.00) inspections reported:
Warning: Stratum void of inspection data - Calibration aborted
```

3.3 The `param.txt` output file

The parameter estimates are also sent to a text file `param.txt`, to be used by the prediction module `Gompred` in conjunction with the input data file.

```
C4;C3;C2;C1
Cast Iron;Brick
DIAM;_TYPE;LENGTH;ROAD
```

```
_TYPE;Combined;Rain;Waste
Cast Iron;C4;C3;C2;C1
Cast Iron;3;2;2;0;0;1
Cast
Iron;-1.218158e+000;-3.304839e+000;-6.452599e+000;-1.667648e-003;2.896731e-001;-2.936680e+000;3.977357e-004;-
1.887854e-001;2.185972e-001;3.070101e-001
Brick;C4;C2;C1
Brick;3;3;3;0;0;2
Brick;-2.059462e+000;-4.096813e+000;-2.709714e-003;4.114559e-001;-6.867287e-001;-3.565743e+000;7.347455e-004;
-2.873364e-001;4.289248e-001;4.262730e-001;3.273713e-001
```

4. The Prediction Module `Gompred`

4.1 Prediction computation

For pipelines not inspected, i.e. with a missing inspection year, prediction computations use the direct method of subtracting the values of the marginal survival functions of the adjacent condition classes. Whereas for inspected pipes, Non Homogeneous Markov Chain calculations are carried out, starting at the age reached by the pipeline at its most recent inspection, the starting state probability vector being the one observed. Additionally the deterioration speed of each inspected pipe is corrected by an IFF computed by Bayesian estimation. In the case of repeated inspections of the same pipe, as many IFF values are computed as there are many inspections; each IFF value relates to a given inspection and the previous ones ; the variation over time of the IFF values reflects then the changes in the level of information gained after each inspection.
These IFF values are outputted in the `iff.txt` texte file. As shown below, the first row recalls the version number of `GompitZ`, and the purpose of the output. The next rows relate each to an inspection, and print the pipe identifier, the installation year, the pipe length, the inspection year, the condition observed, and the IFF value.

```
GompitZ v2.07 - Individual Frailty Factor
Pipe0001;1983;67.000000;1986;C4;-0.003662
Pipe0001;1983;67.000000;1996;C4;-0.032104
Pipe0002;1955;51.000000;1986;C4;-0.016968
Pipe0002;1955;51.000000;1996;C4;-0.046265
```

4.2 Simulation of rehabilitation

The rehabilitation of a pipeline is simply simulated by resetting its installation year at the rehabilitation year, and its condition probability vector at the “as good as new” values. The selection for rehabilitation is driven by the computation of a deterioration index, calculated as the mean condition rank (the ranks are taken to be those of the theoretical condition label list). The rehabilitation simulation is designed to start after the year of last inspection.

4.2 Strategy Specification Files `ST`*`n`*`.txt`

As mentioned in 3.2, `Gompred` retrieves the model parameter values per stratum from the `Param.txt` file. `Gompred` also needs rehabilitation strategy specifications which are given in the `ST`*`n`*`.txt` files. The structure of these files has been defined by the TUD team responsible for the WP6.3.

`Gompred` reread the entire input data file, and performs for each pipeline prediction computations and rehabilitation simulations in the range of future years that is specified in one of the `ST`*`n`*`.txt` file, according to the option number `n` that is specified as 3rd argument in the launching command. S

The meaning of the option number `n` is as follows:
- `n = 0` stands for “do-nothing” strategy ;
- `n = 1` stands for a length driven strategy, that consists in rehabilitating each year the most deteriorated pipelines up to a total length annually specified in `ST1.txt ;`

- `n = 2` stands for a budget driven strategy, that consists in rehabilitating each year the most deteriorated pipelines up to a budget annually specified in `ST2.txt ;`
- `n = 3` stands for an optimised strategy specified in `ST3.txt`, that consists in rehabilitating each year an optimised number of pipelines among the most deteriorated so as to bring the network under a given condition probability vector at a given future year, and then maintain it in that condition.

Since version 2.05, GompitZ enables for the "do-nothing" strategy the computation of predictions for pipes installed after the starting year of simulation; if however the pipe is installed after the stopping year of simulation, no prediction is carried out. This feature is not yet available for the other strategies.

The `ST`*`n`*`.txt` files also specify the unit rehabilitation cost (in monetary unit per length unit, e.g. €/m) in each condition, which allows to estimate the rehabilitation cost for each pipeline.

As exemplified here below, the file `ST0.txt` contains 3 rows respectively specifying:

- the starting year of simulation,
- the stopping year of simulation,
- the list of condition labels.

```
2005
2007
C4;C3;C2;C1
```

The file `ST1.txt` contains 4 rows respectively specifying:

- the starting year of simulation,
- the stopping year of simulation,
- the list of condition labels,
- the list of unit rehabilitation costs per condition,

plus one row per simulation year specifying the annual total length to be rehabilitated.

```
2005
2007
C4;C3;C2;C1
200;300;400;500
2005;100
2006;110
2007;120
```

The file `ST2.txt` contains 4 rows respectively specifying:

- the starting year of simulation,
- the stopping year of simulation,
- the list of condition labels,
- the list of unit rehabilitation costs per condition,

plus one row per simulation year specifying the annual total budget of rehabilitation operations.

```
2005
2007
C4;C3;C2;C1
200;300;400;500
2005;100000
2006;110000
2007;120000
```

The file `ST3.txt` contains 6 rows respectively specifying:

- the starting year of simulation,
- the stopping year of simulation,
- the list of condition labels,
- the list of unit rehabilitation costs per condition,
- the list of maximum tolerated proportion of the network length in each condition,
- the year at which the proportions of the network length in each condition must have been brought just below their maximum tolerated value.

```
2005
2015
C4;C3;C2;C1
200;300;400;500
1.0;0.7;0.2;0.05
2010
```

In the above example, `Gompred` simulates, from year 2005 to year 2010, the annual rehabilitation of as many pipes among the most deteriorated so as at year 2010 the proportion of the network length in condition C4 not exceeds 1.0 (100%), 0.7 in condition C3, 0.2 in condition C2, and 0.05 in condition C1. `Gompred` simulates then, from year 2011 to year 2015, the annual rehabilitation of as many pipes among the most deteriorated so as to maintain these proportions.

5. The outputs of `Gompred`

`Gompred` outputs its results into 2 files:

- `predict`*n*`.txt`
- `rehab`*n*`.txt`

There is of course no `rehab0.txt` file.

5.1 `Predict`*n*`.txt`

`Predict`*n*`.txt` contains one row per pipeline and per prediction year. Each record is composed of the following fields:

- stratum number, this number being the appearance rank in the list of stratum labels given the second row of the observation file,
- pipeline identifier,
- year of prediction (that ranges in this example from 2005 to 2007),
- condition probabilities (for state `C4` to state `C1` in this example – void field if the state is not observed in obs.txt),
- deterioration index,
- pipeline length.

```
0;Pipe1246;2005;0.000000;0.000000;0.000000;1.000000;3.000000;72.000000
0;Pipe1246;2006;0.915137;0.066170;0.016130;0.002563;0.106118;72.000000
0;Pipe1246;2007;0.911041;0.069330;0.016937;0.002692;0.111280;72.000000
0;Pipe2598;2005;0.000000;0.000000;0.000000;1.000000;3.000000;67.000000
0;Pipe2598;2006;0.000000;0.000000;0.000000;1.000000;3.000000;67.000000
0;Pipe2598;2007;0.915137;0.066170;0.016130;0.002563;0.106118;67.000000
```

5.2 `Rehab`*n*`.txt`

`Rehab`*n*`.txt` contains one row per rehabilitated pipeline. Each record is composed of the following fields:

- year of rehabilitation
- pipeline identifier
- pipeline length
- rehabilitation cost

```
2005;Pipe2072;34.0;7411.47
2005;Pipe2544;43.0;9373.33
2005;Pipe2954;62.0;13001.18
2006;Pipe1246;72.0;15164.05
2006;Pipe2572;79.0;16246.30
2007;Pipe2598;67.0;14110.99
2007;Pipe1967;60.0;12636.71
```

6. Compilation and Execution of the Programs

6.1 Contents of the `GompitZ` package

The `GompitZ` package is distributed in the `GompitZ2.08.zip` archive file, that contains the following files:

- `/Readme.txt`
- `/doc/GompitZ2_08UserGuideOct2011.pdf`
- `/example/calibr.txt`
- `/example/iff.txt`
- `/example/obs.txt`
- `/example/param.txt`
- `/example/predict`*n*`.txt (`*n*`=0,1,2,3)`
- `/example/rehab`*n*`.txt (`*n*`=1,2,3)`
- `/example/ST`*n*`.txt (`*n*`=0,1,2,3)`
- `/source/gompcal.c`
- `/source/gompred.c`
- `/bin/gompcal`
- `/bin/gompred`
- `/win32/gompred.exe`
- `/win32/gompcal.exe`

Only the binary files in directories /bin or `/win32` (depending on the host platform) are really necessary. All files in `/example` directory are example ones, i.e. generated by running the executable files. The `/doc` directory contains the present document. The C source codes are provided in the `/source` directory.

6.2 Compilation

For each tool, the C code has been entirely gathered into a single ".c" code file for practicality reasons. Using the GNU tool **gcc** (natively available on a Linux station, but necessitates Code::Blocks under Windows NT), it is particularly simple to compile:

```
#> gcc -lm -Wall -g gompcal.c -o gompcal
#> gcc -lm -Wall -g gompred.c -o gompred
```

This produces executable programmes called `gompcal` and `gompred` under Linux, and `gompcal.exe` and `gompred.exe` under Windows NT.

6.3 Execution

Assuming an input data file `obs.txt` is available and compliant with the format described in sections 2.6 and 2.7, the program can be launched by typing under Linux:

```
#> ./gompcal obs.txt ";"
#> ./gompred obs.txt ";" 1
```

or under Windows NT:

```
#> gompcal obs.txt ";"
#> gompred obs.txt ";" 1
```

assuming here that the user wants to perform strategy 1 computations with `Gompred`.

The file `obs.txt` is here assumed to be present in the same library as the executable. It is obviously possible to use complete or relative paths in the above syntax examples, either for the executables or the input file. The output files `param.txt`, `calibr.txt`, `predictn.txt` and `rehabn.txt` are created in the current directory. It is to be noticed that `gompred` absolutely needs as input file a param.txt previously generated by `gompcal`.