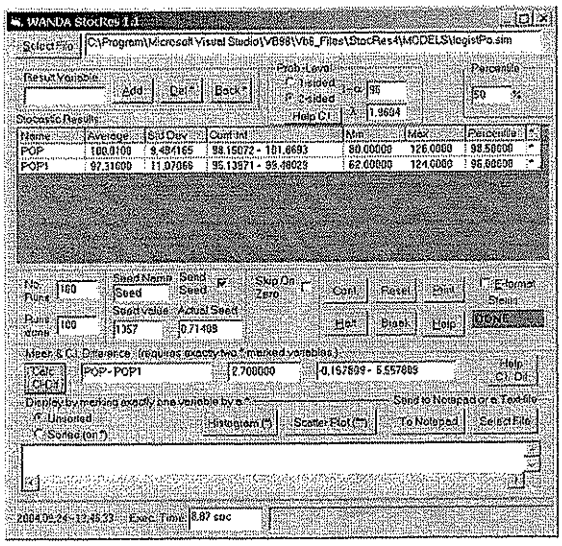
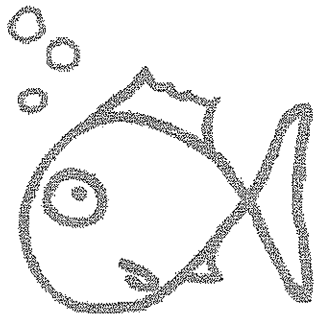
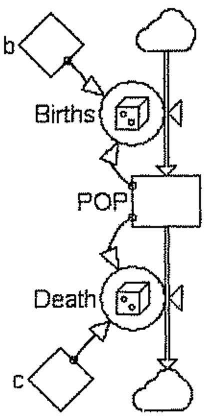
SLU

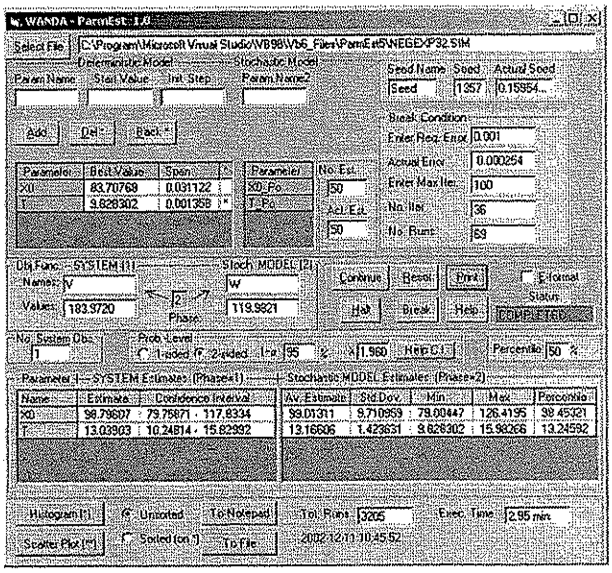
Tools for Statistical Handling of Poisson Simulation: Documentation of StocRes and ParmEst

Leif Gustafsson









SLU
Institutionen för biometri och teknik

SLU
Department of Biometry and Engineering

Rapport – biometri 2004:01

Uppsala 2004
ISSN 1652-3253

ABSTRACT

In many studies of dynamic systems, the stochastic aspects are as important as the dynamic. It is then important to consider uncertainty in the results. Furthermore, dynamics and stochastics interact because the stochastics excite the dynamics and the dynamics change the conditions for the stochastics. Poisson Simulation is an extension of Continuous System Simulation to model and simulate dynamic and stochastic processes. The power of this idea has been described and verified.

However, introducing randomness into a CSS model makes the results stochastic. Therefore, a number of simulation runs followed by a statistical analysis have to be performed. Two cases of the statistical analysis must be distinguished:

I. Estimation of output quantities, which is straight-forward, and

II. Estimation of parameters, which requires a more complex procedure based on bootstrap methods.

This documentation presents methods to solve these problems with the help of specially designed tools for repeated simulation and subsequent statistical analysis. Part I presents the tool **StocRes** for estimation of output quantities and Part II the tool **ParmEst** for parameter estimations.

ACKNOWLEDGEMENTS

StocRes and ParmEst are part of the Wanda project to handle Poisson Simulation supported by the Swedish Council for Planning and Co-ordination of Research (FRN) and The Swedish Research Council for Environment, Agricultural Sciences and Spatial Planning (FORMAS).

Contents **Page**

Part I

Documentation of StocRes - A tool for Statistical Handling of Poisson Simulation

1. INTRODUCTION

Poisson simulation is an extension of Continuous System Simulation to the stochastic domain where the stochastics of the flows are modelled, see Gustafsson (2000, 2003, 2004 and submitted papers). This means that a study must include a number of runs to analyse the results by statistical means. The dynamics of the model are taken care of by the CSS language used, but a device to collect the results from the runs and to analyse them statistically has to be added.

These results can be of two kinds: A) Statistical analysis of *model output*; and B) Statistical analysis of *parameter values* for a fitted model. In this part of the documentation A) is treated. The device to perform this is called StocRes (Stochastic Results). It is a programme written in Visual Basic 6.0 (Perry, 1998) that connects a Poisson Simulation model written in Powersim Constructor (Powersim Corporation, 1996).

2. PHILOSOPHY BEHIND STOCRES AND THE POISSON SIMULATION MODEL FILE

Before constructing a device to handle the statistics from a Poisson simulation model it is important to have a clear philosophy of what to handle in this device and what to handle in the model file.

This philosophy is based on several requirements and preferences.

- StocRes should be a superior programme that connects to the Powersim model, specifies the study, controls the runs, retrieves the results, calculates statistical estimates and presents the statistical results. More precisely, StocRes should be able to
  - Specify and connect to a certain model file
  - Specify the variables to be analysed
  - If wanted, make the study reproducible by controlling the seeds for the simulations
  - Order a number of N runs of the model
  - Receive the results of specified variables
  - Make ordinary statistical analyses of the results from the N simulations
  - Present the results in an appropriate way
- The communication between StocRes and the model should be restricted to before and after full simulation runs. (No intervention should be made during the simulation runs)
- What can be done in the model should be done there. The mechanisms in StocRes should be as few and simple as possible. For example various mechanisms to calculate objective

functions, accumulate results, count events etc. during the run belong to the model. Initialisation of random seeds should also mainly be done in the model (as will be demonstrated later, it is sufficient to provide a single seed for each run from StocRes).

- The user's interface should be as small and comprehensive as possible without being difficult to understand and use.
- The technical interface between StocRes and the Powersim model file should be called from few procedures in StocRes in order to facilitate the portability to other simulation languages. In this implementation StocRes communicates with Powersim via the Powerim API PsCnct, see Appendix 1. This Powersim connection control has the events PreBeginSim, which enables the transfer of values to specified model quantities before a simulation run, and EndSim, which transfers results back to the calling programme. The StocRes communication is therefore restricted to the two small procedures: PsCnct_PreBeginSim and PsCnct_EndSim. (There are in fact a few exceptions to this, e.g. a few lines of code for testing whether a specified variable name exists in the model and for sending the PsCnct.Play statement that starts the simulation and that also calls the PsCnct interface.)

3. THE USER'S INTERFACE OF STOCRES AND ITS FEATURES

The user's interface is shown in Figure 1.

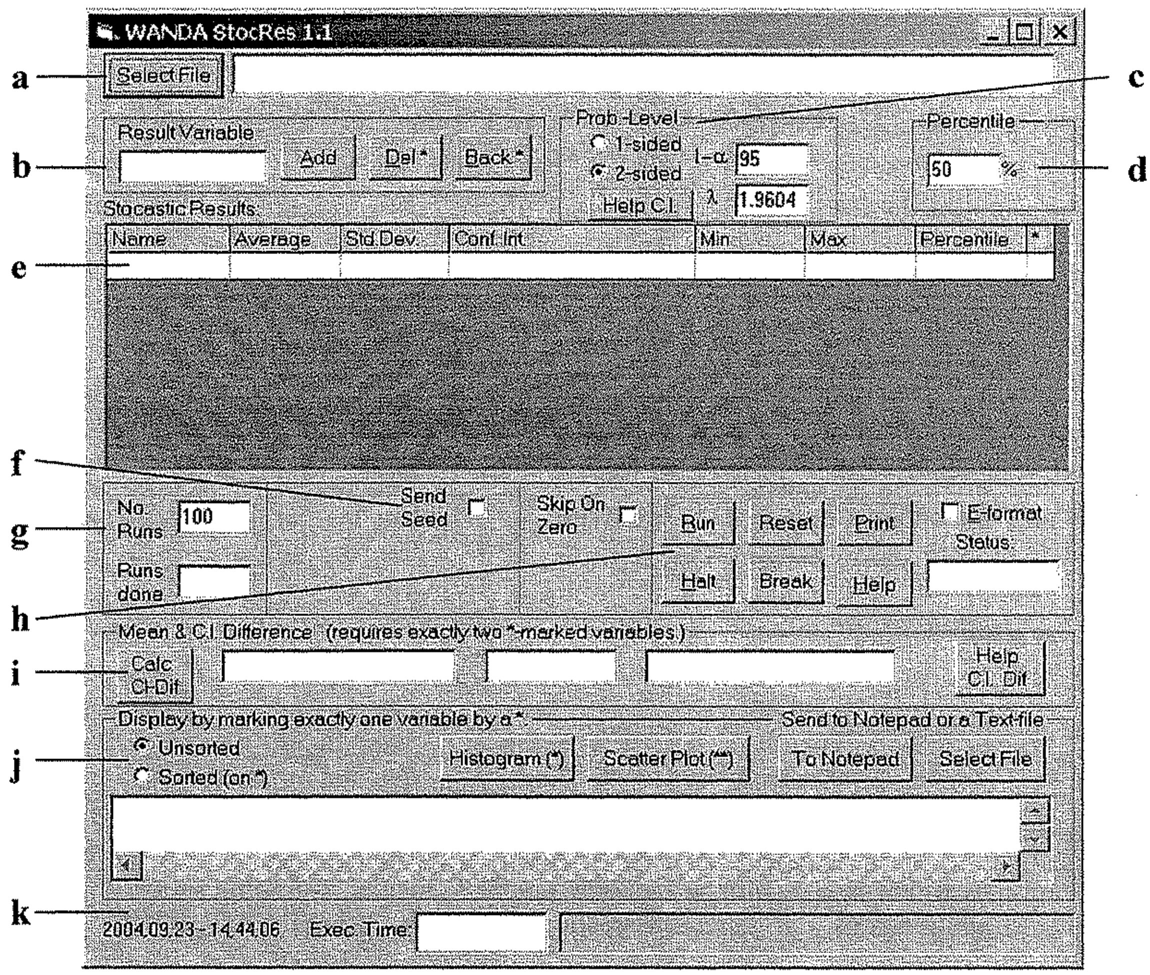


Figure 1. The StocRes user's interface.

As seen from the figure, the user's interface is composed of the following parts:

a) Model-file selection. Press <Select File> to open the file selection facility. Move around in the file structure and open the appropriate model file. This will display the path and name of the file in the textbox to the right of the button and connect StocRes to the file via the PsCnct API.

b) Each variable to study is specified in the 'Result Variable' text box and the <Add> button is pressed. This will check whether the specified variable exists in the model and, if so, add it to the 'Stochastic Results' grid. The <Del> and <Back> buttons also enable removal or change of order of the already specified variables.

c) The 'Prob.-Level' frame allows you to specify 1- or 2-sided intervals for confidence interval estimates and in the '$1-\alpha$' textbox you can specify the confidence level as a percentage (Default 2-sided interval and 95%.) The corresponding λ-values are automatically calculated by an inverse normal distribution. (After the simulation you can also change these options.) A <Help C.I.> button gives the theoretical background (see Appendix 2b).

d) In the 'Percentile' frame you can specify any percentile between 0 and 100 %. The default is 50%, which means the median value. (After the simulation, you can also change the percentile.)

e) In the 'Stochastic Results:' grid, only the specified variables can be seen before the simulation runs. After the runs the Average, Standard Deviation, specified Confidence Interval of the Average estimate, Min value, Max value, and specified Percentile value for each specified variable are presented. (Confidence interval requires at least 20 runs.) Also note that the confidence interval for the average estimate can of course be decreased by increasing the number of runs. The theory behind the calculations is described in Appendix 2b.

   The very last column of the grid can be empty or hold a star '*'. This is automatically inserted for the last added item and cleared by the Run button. You can also mark a number of grid rows, which results in a star on these rows or by pressing <Ctrl> on the keyboard while clicking a row, mark or unmark that row. The star is needed for several purposes. For example the <Del> and the <Back> buttons operate on marked rows only. See also below under point j).

f) In the 'Send Seed' frame only the empty check box is visible if unchecked. If 'Send Seed' is checked <u>one</u> seed is sent to the model for each run. By this device you can make the study reproducible and use variance reduction with common random numbers. (If the random number generators in the simulation model support antithetic sampling, this form of variance reduction can also be used. Unfortunately, the Powersim generators do not support antithetic sampling.)

   When 'Send Seed' is checked (not shown in Figure 1, but in Figure 3 on page 9) the three textboxes: 'Seed Name', 'Seed' and 'Actual Seed' appear. In the 'Seed Name' textbox you specify the name (default 'Seed') which has to correspond to a quantity in the model to be accepted. An integer value (default 1357) is specified in the 'Seed' textbox. The actual seed, which is a number between zero and one, that for each run is sent to the model can be observed (only) in the last of these textboxes. It is then the task of the operator to use the transferred seed to initiate all the random number generators used in the model. This can easily be done by: Seed1=(Seed+s1) MOD 1, .. SeedN=(Seed+sN) MOD 1. s1 ..sN can e.g. be different numbers between zero and one. This guarantees that

Seed1 .. SeedN are independent individual seeds of the same generality as if s1 .. sN had been specified entirely in the model.

g) Below the grid is the 'Run' frame. Here you specify the number of runs to be ordered in the 'No.Runs' textbox. In the 'Runs done.' textbox you can follow the actual run sequence.

h) The buttons within this frame are: <Run>, <Halt>, <Reset>, <Break>, and <Print> and <Help>.

- <Run> saves the specifications done and starts the simulation run sequence. By increasing 'No.Runs' you can press this button again (which is now labelled 'Cont.'.)
- <Halt> halts the run sequence (after a full simulation run).
- <Reset> resets the specifications to the state at last run.
- <Break> ends the session.
- <Print> prints the form as is.
- <Help> gives instructions about how to run StocRes. (See Appendix 2a.)

In this frame you also find the check box 'E-format'. The default is unchecked, which gives maximal numerical information within 9 positions. If 'E-format' is checked the results are presented as: X.XX E±XX.

The Status after the simulation (ERROR, HALTED, DONE) is presented to the right of this frame.

i) The 'Average & C.I. Difference' frame provides a way to calculate the difference and confidence intervals for the difference between two variables. This compares the results for one variable over the N simulation runs to the results of another. (An alternative way is to study paired runs and calculate the difference in the model.) To use this device, exactly two variables have to be *-marked. Then press the <Calc. CI-Dif> button, which fills in the names of the variables, average of the difference and the estimated confidence interval for this estimate in the three textboxes. A <Help C.I. Dif> button is also available (see Appendix 2c for a description of the calculations).

j) To see the crude data results from the simulation runs (not the statistical estimates of it), and perhaps to analyse it with other tools, there are some devices in the bottom frame. By marking exactly one variable in the grid, the result of each run is automatically displayed in the display window. With the radio buttons 'Unsorted' and 'Sorted' you decide whether the data should be displayed in the order of the simulation runs (Unsorted) or in ascending order (Sorted).

With the <To Notepad> and <Select File> buttons you can send all data to Notepad or to a specified file. Data from this session such as the model file name, date and time etc are then sent to Notepad or the specified file followed by the data results of all variables in separate columns. If the radio button 'Sorted' is selected you have to mark exactly one variable in the grid, which should be presented in ascending order. All variables from each run are, of course, presented together on the same row in Notepad or the selected file.

There are also two further buttons: <Histogram(*)> and <Scatter Plot (**)> The stars indicate how many parameters have to be selected to display a Histogram of one parameter or a Scatter plot of two parameters.

k) Date and Time, Execution Time for the simulation runs and a field for Free Text.

4. A POWERSIM MODEL EXAMPLE

A simple example of a logistic model (Braun, 1993) in two copies is presented below. The structure of this model is dX/dt=F1-F2; X(0)=Poisson(25, Seed2); F1=Poisson(b*X*dt, Seed1)/dt; F2= Poisson(c*X*X*dt, Seed3)/dt. In the first copy POP=X, Birth=F1 and Death=F2. For the second copy the naming is similar but with an '1' added to the names. (Death stands for death because of competition, which is why it is described by a quadratic term.)

These two copies are identical except for the three random number generators of each copy having different seeds – all seeds calculated from the seed called 'Seed' and sent from StocRes. The six generators are: Seed1=(Seed+0.1) Mod 1, Seed2=(Seed+0.2) Mod 1, Seed3= (Seed+0.3) Mod1 and Seed11=(Seed+0.11) Mod 1, Seed21=(Seed+0.21) Mod 1, Seed31= (Seed+0.31) Mod1.

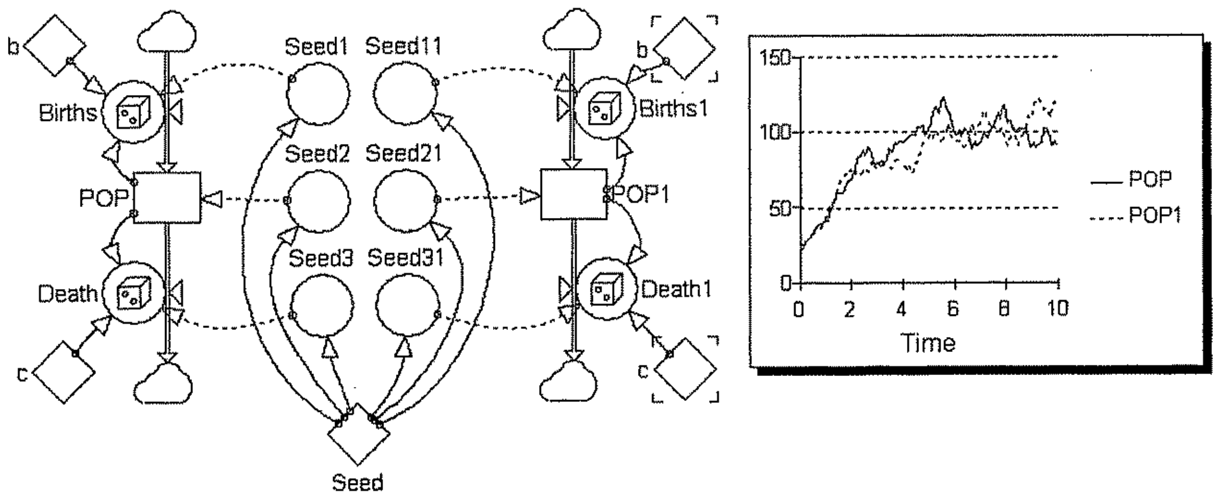


Figure 2. A Powersim model requiring six random number generator seeds. To the right the outcome of one replication of the two stochastic models.

In Figure 3, the results of 100 simulation runs are presented. Calculation of the difference between POP and POP1 and the confidence interval for this difference is also calculated.

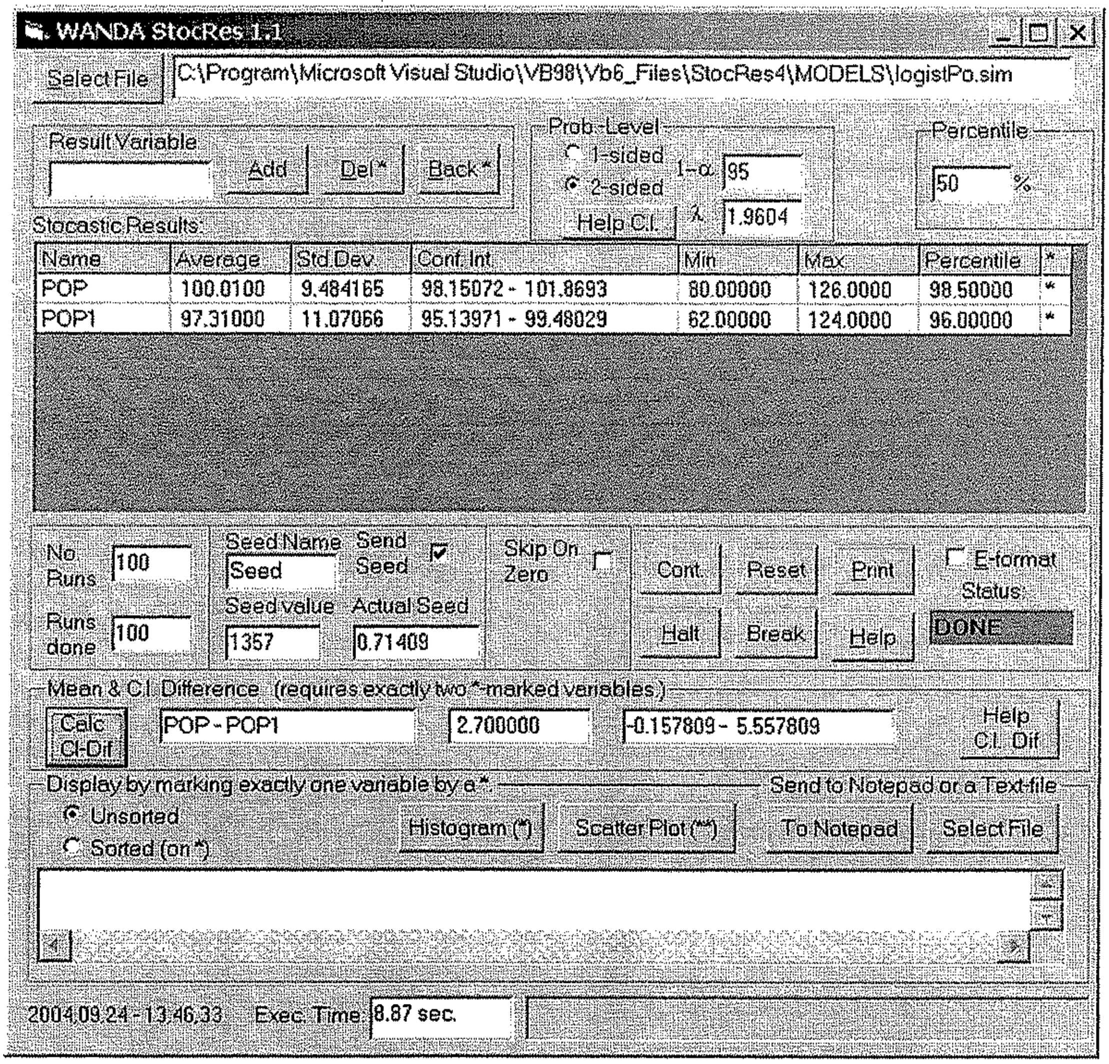


Figure 3. The results after 100 simulation runs.

As seen, the estimate of the Average of POP is 100.01 and of POP1 is 97.31. The difference is 2.70. Since the 2-sided confidence interval on 95% confidence level for this difference (-0.15 – 5.56) includes the value zero, no difference could be inferred (which would not be expected since the two copies of the model are identical except for seed values).

The Standard Deviation was 9.48 for POP and 11.07 for POP1. Estimates of Confidence intervals, Min, Max and the 50% Percentile (median) for POP and POP1 are also shown in Figure 3.

The Execution Time with the step-size equal to 0.1 was 8.8 seconds on a PIII/800 MHz computer.

This example also displays many other statistical results. For example, the Median (50% percentile) is somewhat lower (for both POP and POP1) than the Average for this non-linear model.

5. TESTS AND ERROR CHECKING

A large proportion of the about 1500 lines of code are devoted to error checking. For example, a variable cannot be added before StocRes is connected to a selected model file. Then Add checks that a variable with the specified name exists in the model. A check is made that the simulation run works. On many occasions, there have to be tests that certain quantities are specified. If a seed is sent, a corresponding quantity with that name must exist in the model. Furthermore, this Seed name may not coincide with variable names specified in the grid. When an operation that requires one or two *-marked rows in the grid is to be performed, a check of this condition is first made, etc.

It is a large task to test all aspects of such a complicated programme. StocRes should therefore be regarded as a beta-release in order to identify possible errors and to improve the programme. The following test types in particular have been performed.

- Tests of performance and technical reliability.
- Tests of Calculations of Average, Standard Deviation, Confidence Intervals, Min, Max and Percentiles.
- Tests of Average and Confidence Interval for a difference
- Tests of Displaying data and of sending data to Notepad or a specified file.
- Test of the Send Seed mechanism.

All known errors have been corrected.

6. REFERENCES

Braun M. 1993. *Differential Equations and Their Applications*. Springer-Verlag, New York. ISBN:0-387-97894-1.

Gustafsson L. 2000. Poisson Simulation – A Method for Generating Stochastic Variations in Continuous System Simulation. *Simulation* 74:5, pp. 264-274.

Gustafsson L. 2003. Poisson Simulation as an Extension of CSS for the Modeling of Queuing Systems. *Simulation* 79:9, pp. 528-541.

Gustafsson L. 2004. Studying dynamic and stochastic systems using Poisson Simulation. In Liljenström H. and Svedin U. Eds. *Micro – Meso – Macro: Addressing Complex Systems Couplings.* World Scientific Publishing Company: Singapore.

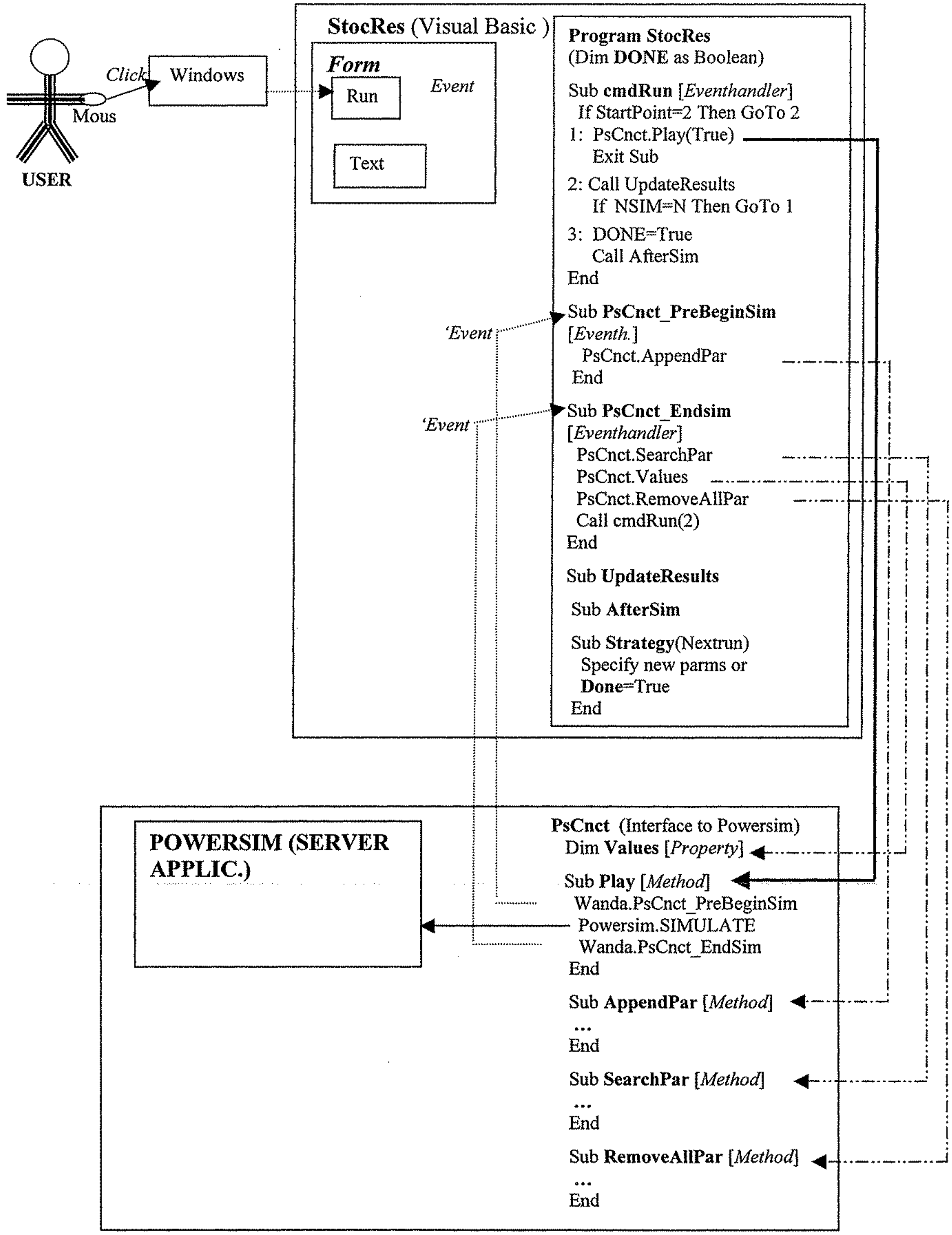
Gustafsson L. Methods and Tools for Statistical Analysis of Poisson Simulation. Submitted.

Gustafsson L. Poisson Simulation in Theory and Practice. Submitted.

Perry G. 1998. *Teach Yourself Visual Basic 6 in 21 Days*. Sams Publishing: Indianapolis, Indiana. ISBN: 0-672-31310-3.

Powersim Corporation. 1996. *Powersim Reference Manual, Powersim Press*. Powersim Corporation, 1175 Herndon Parkway, suite 600, Herndon, VA 22170, USA. (Note that the time-step is denoted TIMESTEP in Powersim.)

APPENDIX 1. Interfacing StocRes to Powersim via the API PsCnct



APPENDIX 2. Help Texts for StocRes

The help texts are:

a) Help StocRes
b) Help Statistics
c) Help Statistic comparison

They appear when help buttons in StocRes are pressed.

a) 

HELP STOCRES

StocRes is the module for collection of results from N simulations and for statistical analysis of these data.

Quantities to be studied (e.g. States, Flows, Differences, Ratios, Quantities cumulated within a simulation) have to be modelled in Powersim and their resulting values are after each run transferred to StocRes.
StocRes will calculate: Average, Standard Deviation, Confidence Interval, Min, Max, and Percentiles (Percentiles).
The Level for the Conf. Int. can be specified as well as 1 or 2-sided intervals.

(Default is 95% and 2-sided interval.) For more information: Press 'Help C.I.'

The Percentile can also be specified. (Default =50%=Median.)

To use the StocRes module you have to:

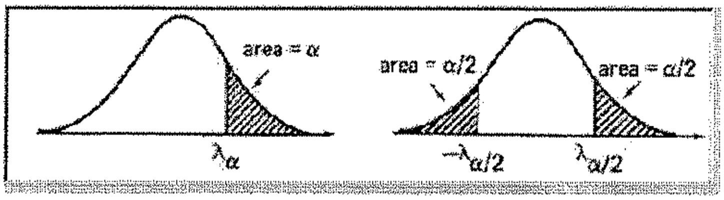
1. Select the file of the Powersim Model.
2. Specify and Add the names of the quantities to be studied.
[3. Specify the Level for the Confidence Interval & 1 or 2-sided.]
[4. Specify the Percentile.]
5. Specify the 'No. Runs'.
6. Press the Run button.

(NOTE: Only write in Yellow fields.)

b) Help C.I.

HELP STATISTICS

```
After N simulations statistics can be calculated:

Mean = M = SUM(X1..XN)/N
StdDev = s = Sqrt{SUM[(Xi-M)(Xi-M)]/(N-1)} 'Std. dev. of a sample.

Choose 1-sided or 2-sided conf intervals
Confidence limits for the mean are set by 1-alpha (e.g. 90%, 95%, etc.)
The corresponding lambda (= L1 or L2) is then calculated (see Figure).

Confidence limits for the mean are calculated using the Normal
approximation. (This requires at least N=20 simulations.)

The C.I. interval is then:
1-sided
Interval = X < M + L1·s/Sqrt(N)    or    X > M - L1·s/Sqrt(N)

2-sided
Interval = (M - L2·s/Sqrt(N); M + L2·s/Sqrt(N))

(Note: You can reduce the C.I. by increasing the number
of simulations.)
```



c) Help C.I. Dif

HELP STATISTICS COMPARISON

To compare the results of two situations (or two models) make N1 simulations with the first one and N2 with the second one. (In this implementation the two models have to be located in the same Powersim file so the number of simulations: N1=N2=N.)

Statistics can then be calculated for each case. A Confidence Interval for the difference between these trials can also be made.

```
Mean = M(i) = SUM(X1(i)..XN(i))/N  i=1,2.
StdDev = s(i) = Sqrt{SUM[(Xi(i)-M(i))(Xi(i)-M(i))]/(N(i)-1)}
(Std. Dev. of a sample).
```

For COMPARISON
Choose 1-sided or 2-sided conf intervals at the bottom [COMPARE (M1-M2)]. Confidence limits for the Mean=M(1)-M(2) are set by 1-alpha (e.g. 90%, 95%, etc.) The corresponding lambda (= L1 or L2) is then calculated.

Confidence limits for the Mean difference (M(1)-M(2) are calculated using the Normal approximation. (This requires at least N(1)=20 and N(2)=20 simulations.)

The C.I. interval is then:

```
1-sided
Interval = X < M1-M2 + L1·Sqrt[(s1·s1+s2·s2)/N]   or
                X > M1-M2 - L1·Sqrt[(s1·s1+s2·s2)/N]

2-sided
Interval = {M1-M2 - L2·Sqrt[(s1·s1+s2·s2)/N];
                M1-M2 + L2·Sqrt[(s1·s1+s2·s2)/N]}
```

(Note: You can reduce the C.I. by increasing the number of simulations.)
(Note: To mark two non-consecutive variables, press Ctrl-key when clicking.)

Part II

Documentation of ParmEst - A tool for Statistical Handling of Poisson Simulation

1. INTRODUCTION

Poisson simulation is an extension of Continuous System Simulation to the stochastic domain where the stochastics of the flows are modelled, see Gustafsson (2000, 2003, 2004 and submitted papers). Introducing stochastics means that a study must include a number of runs to analyse the results by statistical means. The dynamics of the model are taken care of by the CSS language used, but a device to collect the results from the runs and to analyse them statistically has to be added.

These results can be of two kinds: A) Statistical analysis of *model output*; and B) Statistical analysis of *parameter values* (model input) for a fitted model. A) is treated in Part I of this documentation, while in this part B) is treated. The device to perform parameter estimation is called ParmEst (Parameter Estimation). It is a programme written in Visual Basic 6.0 (Perry, 1998) that connects a Poisson Simulation model written in Powersim Constructor (Powersim Corporation, 1996).

Estimation of model parameter values is more complicated and computer intensive than estimation of model outcomes. To make it easy we describe the problem with the help of a simple example.

Example. Assume that a unit of a compound contains on average $X0_{Syst}$ radioactive atoms of a certain kind that decay with a time constant T_{Syst}. Each time we take such a unit and study it we get slightly different results because the number of radioactive atoms is only approximately $X0_{Syst}$ and the number of decays during a study time will only give a time constant of approximately the value T_{Syst}. If we have *one sample* we can only study the process once. Then we can build a model with the parameters X0 and T and fit these parameters so that the model behaviour is as similar as possible to that of the system. These values are of course not identical to $X0_{Syst}$ and T_{Syst}, but they are the best estimate we can get from a single experiment (sequence of observed data). To judge the credibility of the estimates X0 and T it would be valuable to know how much X0 and T would vary if we could repeat the experiment a number of times and thus get estimates of standard deviation or confidence intervals. When it is not possible to repeat the experiment on the system, we still have the possibility to study how much X0 and T vary in our model if we can make it stochastic in a realistic way. Provided that this stochastic model is a good description (dynamically and stochastically) of the real system, the estimates of the model variations around X0 and T are about the same as the real variations would have been around $X0_{Syst}$ and T_{Syst}.

Thus, we get the estimates, X0 and T, from fitting the deterministic model to the system, and the variations of the estimates from fitting the deterministic model to the stochastic one. This technique is called resampling (Press et al., 1989) and these ideas are in statistics called Bootstrap methods (Davison, 1997).

However, to perform this we need 1) **Data** from observations of the real system behaviour,

2) A **deterministic model** to fit X0 and T against the real data, 3) A **stochastic model** that display the variations, and 4) A device that measures the difference between system and deterministic model behaviours and a similar device that measures the difference between stochastic and deterministic model behaviours.

The procedure is performed in two stages or phases. In Phase 1 the values of the parameters are estimated by fitting the deterministic model to system data. In Phase 2 the standard deviations of the parameters are obtained by running the stochastic model a number of times and for each such run fitting the deterministic model to the results of the stochastic model. This results in a number of estimations of the parameters.

Finally, statistical methods can be used to produce e.g. standard deviation or confidence intervals based on the variations of the parameter estimates. ■

For more information, see Press et al. (1989) and Davison (1997).

2. PHILOSOPHY BEHIND PARMEST AND THE POISSON SIMULATION MODEL FILE

Before constructing a device to handle the statistics from a Poisson simulation model it is important to have a clear philosophy of what to handle in this device and what to handle in the model file.

This philosophy is based on several requirements and preferences.

- ParmEst should be a superior programme that connects to the Powersim model, specifies the study, controls the runs, retrieves the results, calculates statistical estimates and presents the statistical results in appropriate ways. Therefore ParmEst should be able to
    - Specify and connect to a certain model file
    - Specify the variables to be analysed
    - Control the seeds for the simulations
    - Order one parameter estimation for the deterministic model against system data
    - Transfer the estimated values to the stochastic model
    - Order a number of parameter estimations of the deterministic model against results from stochastic model runs
    - Receive the results of specified parameters
    - Make ordinary statistical analyses of the results from the N simulations
    - Present the results in an appropriate way
- The user's interface should be as small and comprehensive as possible without being difficult to understand and use.
- The communication between ParmEst and the model should be restricted to before and after full simulation runs. (No intervention should be made during the simulation runs.)
- What can be done in the model file should be done there. The mechanisms in ParmEst should be as few and simple as possible.

The **model file** should contain: 1) The system data from the single observation, 2) The deterministic model, and 3) A corresponding stochastic model that displays the variations. It should also contain 4) An objective function that measures how well the deterministic model fits the system data, and another objective function that measures how well the deterministic model fits the stochastic model.

In addition, the model file may also include e.g. mechanisms to accumulate results, count events etc. during the run. Initialisation random seeds should also mainly be done in the model. (It is sufficient to provide one single seed for each run from ParmEst.)

- The technical interface between ParmEst and the Powersim model file should be called from only few procedures in ParmEst to facilitate the portability to other simulation languages. In this implementation ParmEst communicates with Powersim via the Powerim API PsCnct, see Appendix 1. This Powersim connection control has the events PreBeginSim, which enables the transfer of values to specified model quantities before a simulation run, and EndSim, which transfers results back to the calling programme. The ParmEst communication is therefore restricted to the two small procedures: PsCnct_PreBeginSim and PsCnct_EndSim. (There are in fact a few exceptions to this. For example a few lines of code for testing if a specified variable name exists in the model and for sending the PsCnct.Play statement that starts the simulation and that also calls the PsCnct interface.)

3. THE USER'S INTERFACE OF PARMEST AND ITS FEATURES

The user's interface is shown in Figure 1.

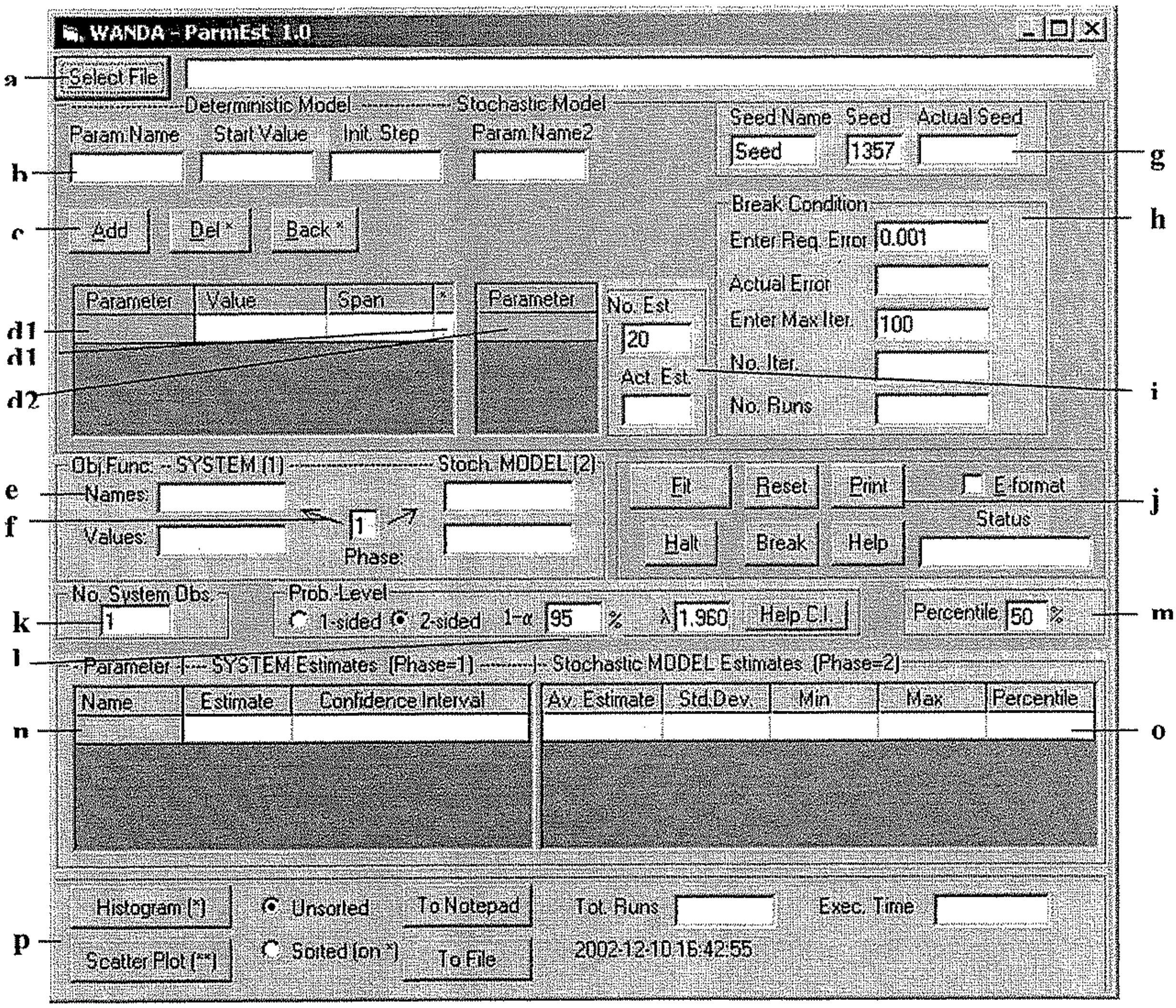


Figure 1. The ParmEst user's interface.

As seen from Figure 1, the user's interface is composed of the parts described below:

To understand this user's interface, it must be kept in mind that the underlying model file is composed of several parts: 1) The data describing the real system behaviour, 2) The deterministic model, and 3) The stochastic model. However, 4) An objective function describing the cumulative squared difference between the deterministic model and system data is also needed, as is another objective function describing the corresponding difference between the deterministic and the stochastic model.

a) Model-file selection. Press <Select File> to open the file selection facility. Move around in the file structure and select and open the appropriate model file that holds the system data, the deterministic model and the stochastic model together with the two objective functions mentioned above. This will display the path and name of the file in the textbox to the right of the button and connect ParmEst to that file via the PsCnct API.

b) Each variable to study in the deterministic model has to be specified by Param.Name, Start Value and Init.Step together with the corresponding Param.Name2 of the stochastic model. Start Value tells ParmEst where to start the parameter search and Init.Step the size of steps to use when changing the initial parameter value. (With <Alt><Tab> you can toggle between the ParmEst user's interface and the model file to see the names of the parameters and objective functions.)

c) The <Add> button When pressed will check whether the specified variable exists in the model and, if so, add it to the left 'Parameter grid'. The <Del> and <Back> buttons also enable removal or change of order of the already specified variables.

d) In the first 'Parameter grid' (**d1**) before the fitting the specified variables, their initial values and step sizes of the deterministic model are shown. To the right a second 'Parameter grid' (**d2**) containing only the names of the corresponding parameters of the stochastic model is shown. During Phase 1 the deterministic model is fitted to system data. The estimated parameters are then transferred to the stochastic model. There follows Phase 2 where the stochastic model shows a (pseudo-)stochastic behaviour (depending on the seed used). In this phase the deterministic model is fitted to the stochastic one using the second objective function. The result is a new set of parameter estimates. This is repeated N times with a new seed for each fit, so in the end we have N sets of parameter estimates. After these fittings the N sets of parameter estimates are used to calculate the results.

   The very last column of the first Parameter grid (**d1***) can be empty or hold a star '*'. This is automatically inserted for the last added item and cleared by the Fit button. You may also mark a number of grid rows, which results in a star on these rows or by pressing <Ctrl> on the keyboard while clicking a row, mark or unmark that row. The star is needed for several purposes. For example the <Del> and the <Back> buttons operate on marked rows only. See also under point p).

e) Below the Parameter grid are fields for name and value of the two Objective functions used in Phase 1 and 2. The Names given here must be the ones in the model file. The resulting values of the objective functions are delivered from the model during the fitting process.

f) The phase is shown in this text field. Phase 1 means fitting the deterministic model to system and Phase 2 means fitting it to the stochastic model.

g) To the right there is a frame for 'Seed Name', 'Seed Value' and 'Actual Seed'. Specify a Seed Name, which must exist in the model, and a Seed Value for it. In Phase 2 <u>one</u> Actual Seed is sent to the model for each fit. To the next fit a new Seed Value is calculated by the Visual Basic built-in random number generator. By this device you can make the study reproducible or perform variance reduction using common random numbers. (Unfortunately, the Powersim generators do not support antithetic sampling.) The actual seed, which is a number between zero and one, that for each run is sent to the model can be observed (only) in the last of these textboxes. It is the task of the operator to initiate all the random number generators used in the model from the transferred seed. This can easily be done by: Seed1=(Seed+s1) MOD 1, ... SeedN=(Seed+sN) MOD 1. s1 ...sN can e.g. be different numbers between zero and one. This guarantees that Seed1 ... SeedN are independent individual seeds of the same generality as if s1 ... sN had been specified in the model file.

h) To the right you also find the Break Condition frame with a number of fields for specifying the required accuracy of the fitting process (Enter Req.Error), and in the field below you can follow the development of the actual error during the fitting process. 'Enter Max Iter' will stop the fitting process if it requires more than the specified number of iterations in the simplex optimiser used, each iteration takes several simulations. (A simplex optimiser of the actual type is described in Press et al. (1989)).

i) To the right of the Parameter grids there is a text field No.Est. where you specify the Number of Parameter Estimates you want. In the Act.Est. you can follow the process showing the number of parameter estimation rounds performed so far.

j) Below the Break Condition frame is a frame with the buttons: <Fit>, <Halt>, <Reset>, <Break>, <Print> and <Help>.

- <Fit> saves the specifications done and starts the simulation run sequence. By increasing 'No.Est.' you can press this button again (which is now labelled 'Continue'.)
- <Halt> halts the run sequence after a full fitting round.
- <Reset> resets the specifications to the state before <Fit> was last pressed.
- <Break> ends the session by closing ParmEst and the connected model file.
- <Print> prints the form as is.
- <Help> gives instructions about how to run ParmEst. (See Appendix 2a and 2a1.)

In this frame you also find the checkbox 'E-format'. The default is unchecked, which gives maximal numerical information within 9 positions. If 'E-format' is checked the results are presented as: X.XX E±XX.

The status after the parameter estimation session (ERROR, HALTED, DONE) is presented to the right in this frame.

k) To the left is No.System Obs. which means the number of observations you have from the studied system. Usually you have only one. This value affects the calculation of the confidence intervals around the parameter estimates.

l) The 'Prob.-Level' frame allows you to specify a 1- or 2-sided confidence interval and in the '$1-\alpha$' textbox you can specify the Confidence Level as a percentage (Default 2-sided interval and 95%.) The corresponding λ-value is automatically calculated by an inverse normal distribution. (After the simulation you can also change these options.) A <Help C.I.> button gives the theoretical background. How the statistical calculations are performed is displayed by pressing the Help C.I. button (see Appendix 2b).

m) In the 'Percentile' frame you can specify any percentile between 0 and 100 %. The default is 50% which means the median value. (Afterwards you can also change the percentile.)

n) In the Phase 1-Result grid, the Parameter estimates from Phase 1 and the Confidence Intervals are calculated. (The Confidence Interval is calculated around the parameter estimates from Phase 1, and based on the Number of System Observations, the Standard Deviation (from Phase 2) and from λ. Confidence interval requires at least 20 runs.) The theory behind the calculations is described in Appendix 2b.

o) In the Phase 2-Result grid the stochastic results: Av.Estimate, Std.Dev., Min, Max, and Percentile of the N fits are presented. The theory behind the calculations is described in Appendix 2b.

p) To see the N individual parameter estimates (not the statistical results of it), and perhaps to analyse them with other tools, there are some devices in the bottom frame. With the radio buttons 'Unsorted' and 'Sorted' you decide whether the data should be displayed in the order of the simulation runs (Unsorted) or in ascending order (Sorted) by the *-marked parameter. At the bottom of the form you also find the two buttons: <Histogram(*)> and <Scatter Plot (**)>. The stars tells how many parameters have to be selected to display a Histogram or a Scatter plot.

With the <To Notepad> and <Select File> buttons you can send all data to Notepad or to a specified file. Data from the session such as the model file name, date and time etc. are then sent to Notepad or the specified file followed by the data results of all variables in separate columns. If the radio button 'Sorted' is selected you have to mark *exactly one* variable in the grid, which should be presented in ascending order. All variables from the same fitting are presented together on the same row in Notepad or the selected file.

The Total Number of Simulations and the Total Execution Time for the process are also displayed.

4. A POWERSIM MODEL FILE EXAMPLE

A simple example of a negative exponential process, presented in the Introduction above, is given in Figure 2.

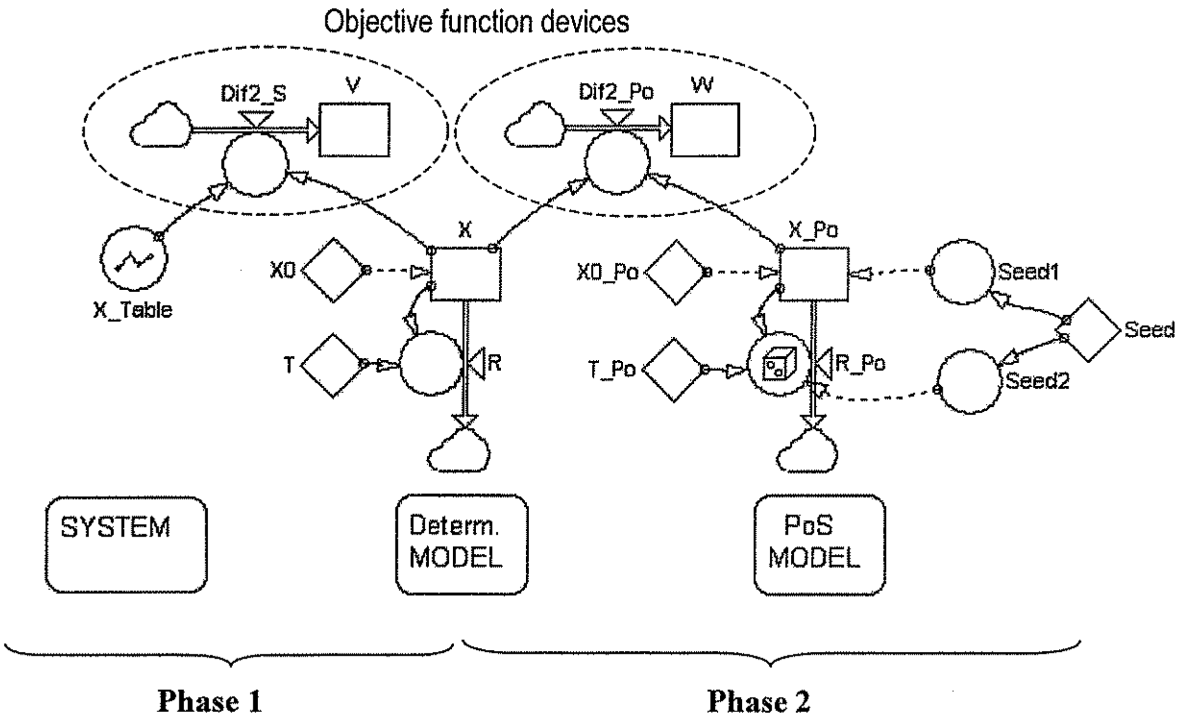


Figure 2. The Powersim model file.

The system data are taken from a table look-up function describing the number of radioactive atoms over time. During Phase 1 the parameters X0 (starting value of X) and T (decay time constant) are estimated by minimising the difference between the behaviours of the system and the deterministic model in a least square sense. Technically, for each time-step this difference is squared and cumulated into the objective function V, whose value is returned to ParmEst after the simulation.

To the right of Figure 2 the stochastic model is shown. At the end of Phase 1 the estimated values of the parameters X0 and T are transferred to the corresponding parameters X0_Po and T_Po of the stochastic model. During Phase 2 the stochastic model will display different stochastic realisations depending on the delivered seed ('Seed'). For each of N given seed values the deterministic model is fitted to the stochastic one by minimising W. This gives N new (resampled) estimates of the parameters X0 and T.

Finally, these N estimates are statistically treated and presented in terms of Average, Standard Deviation, Min, Max and Percentile. Finally, the Confidence Intervals around the system (Phase 1) estimates are calculated. This calculation also involves the Standard Deviation estimates from Phase 2.

The structure of the deterministic model is:

dX/dt = –R

R = X/T.

The state X is initialised by X(0)=X0.

The structure of the stochastic model is:

dX_Po/dt = –R_Po

R_Po = Poisson(dt*X_Po/T_Po, Seed2)/dt.

where X0_Po and T_Po get the values of X0 and T estimated in Phase 1. The state X_Po is initialised by X_Po(0)=Poisson(X0_Po, Seed1).

The two seeds used are calculated from the single seed called 'Seed' provided by ParmEst. The two seeds used in the model were calculated as: Seed1=(Seed+0.1) Mod 1 and Seed2=(Seed+0.2) Mod 1.

In Figure 3, the results of 50 fits are presented.

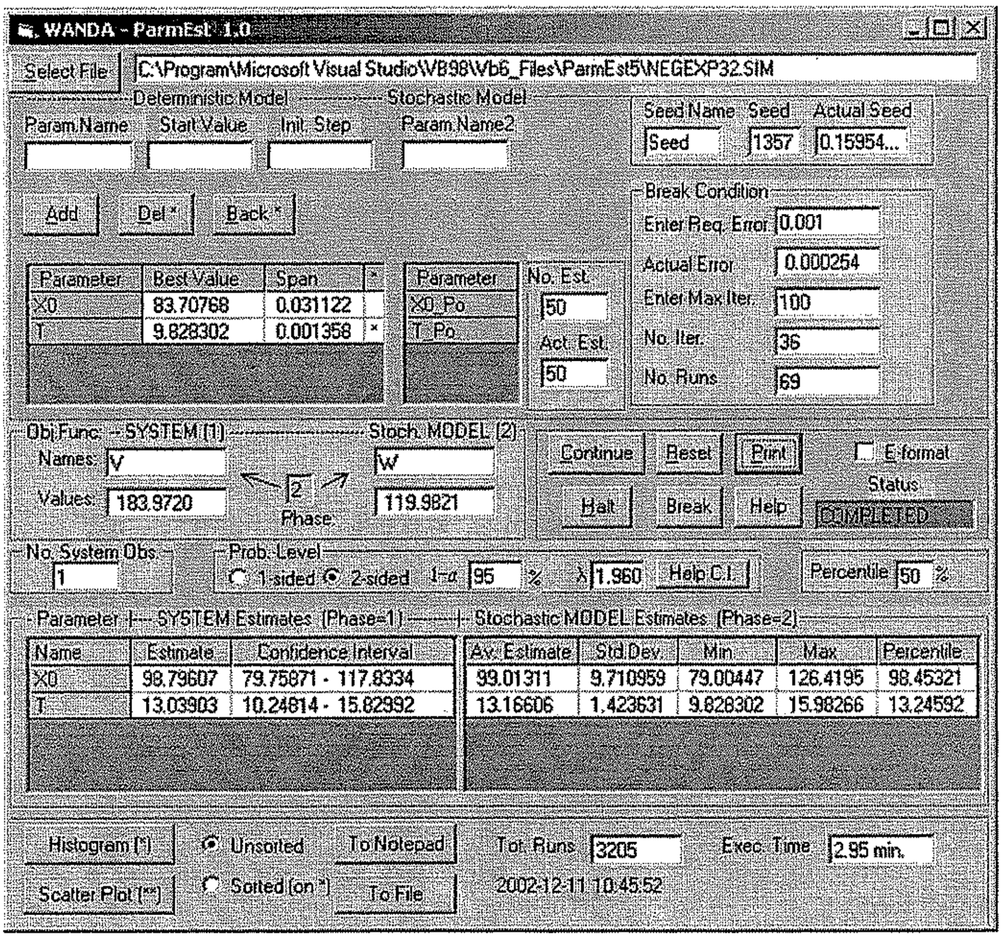


Figure 3. The results after 50 resamplings.

As seen, the parameter estimates against the real system (Phase 1) were X0 = 98.8 and T = 13.0.

For the 50 resamplings with the stochastic model (Phase 2) we got the Average estimates X0 = 99.0 and T = 13.2.

Note: The estimate of the parameters in Phase=1 are transferred to the stochastic model. But since the results from a stochastic model vary, the Average estimates of the parameters in Phase=2 may differ somewhat from the Phase=1 estimates. This difference is called *bias*.

In this case the bias was 0.2 for X0 and 0.2 for T.

The Standard Deviation for X0 was 9.7 and for T it was 1.4. In the figure, Min, Max and the median (50% percentile) are also shown.

As a 2-sided confidence interval on the 95% confidence level (around the Phase 1 estimates) we got 79.8 – 117.8 for X0 and of 10.2 – 15.8 for T. The intervals will, of course, decrease for a larger number of estimation rounds.

At the bottom of the form, you can see that it required 3205 runs to achieve these results, and that it took less than 3 minutes of execution time (on a PIII/800 MHz computer with the model time-step=0.5 from time 0 to 50).

5. TESTS AND ERROR CHECKING IN PARMEST

A large part of the about 2500 lines of code are devoted to error checking. For example, a variable cannot be added before ParmEst is connected to a selected model file. Then Add checks that a variable with the specified name exists in the model. A check is also made that the simulation runs work. On many occasions, there have to be tests that certain quantities are specified. If a seed is sent, a corresponding quantity with that name must exist in the model file. Furthermore, this Seed name may not coincide with variable names specified in the grid. When an operation that requires one or two *-marked rows in the Parameter grid is to be performed, a check of this condition is first made.

It is a large task to test all aspects of such a complicated programme. ParmEst should therefore be regarded as a beta-release in order to find possible errors and to improve the programme. The following test types have been performed.

- Tests of performance and technical reliability.
- Test of the fitting during the two phases.
- Tests of Calculations of Average, Standard Deviation, Min, Max, Percentiles and Confidence Intervals.
- Tests of Displaying data and of sending data to Notepad or a specified file.
- Test of the Send Seed mechanism.

All known errors have been corrected.

6. REFERENCES

Davison A.C. 1997. *Bootstrap Methods and Their Application.* Cambridge University Press: Cambridge, UK. ISBN:0-521-57391-2.

Gustafsson L. 2000. Poisson Simulation – A Method for Generating Stochastic Variations in Continuous System Simulation. *Simulation* 74:5, pp. 264-274.

Gustafsson L. Poisson 2003. Simulation as an Extension of CSS for the Modeling of Queuing Systems. *Simulation* 79:9, pp. 528-541.

Gustafsson L. 2004. Studying dynamic and stochastic systems using Poisson Simulation. In: Liljenström H. and Svedin U. (ed) *Micro – Meso – Macro: Addressing Complex Systems Couplings.* World Scientific Publishing Company: Singapore.

Gustafsson L. Methods and Tools for Statistical Analysis of Poisson Simulation. Submitted.

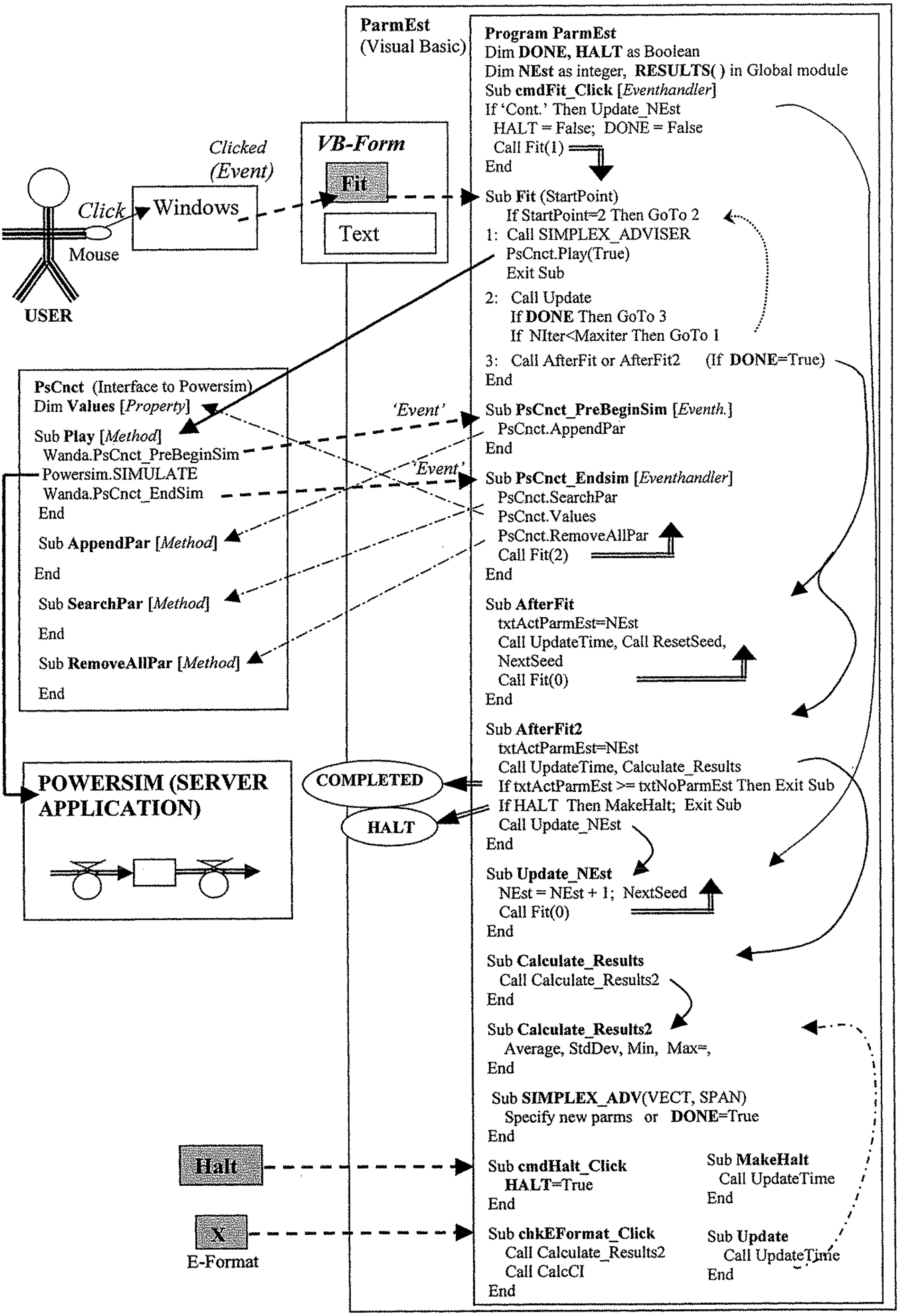
Gustafsson L. Poisson Simulation in Theory and Practice. Submitted.

Perry G. 1998. *Teach Yourself Visual Basic 6 in 21 Days.* Sams Publishing: Indianapolis, Indiana. ISBN: 0-672-31310-3.

Powersim Corporation. 1996. *Powersim Reference Manual, Powersim Press.* Powersim Corporation, 1175 Herndon Parkway, suite 600, Herndon, VA 22170, USA. (Note that the time-step is denoted TIMESTEP in Powersim.)

Press W.H., Flannery B.P., Teukolsky S.A. and Vetterling W.T. 1989. *Numerical Recipes in Pascal – The Art of Scientific Computing.* Cambridge University Press: Cambridge, UK. (The book is also available in FORTRAN or C.) ISBN:0-521-37516-9.

APPENDIX 1. Interfacing ParmEst to Powersim via the API PsCnct



APPENDIX 2. Help Texts for ParmEst

The help texts are:
a) Help ParmEst
a1) - Help Model File (Appears on Help Model in Help ParmEst.)
b) Help Statistics
They appear when help buttons in ParmEst are pressed.

a) Help

```
HELP PARMEST

WANDA ParmEst first fits a DETERMINISTIC MODEL to the SYSTEM
and then to a STOCHASTIC MODEL. The Model file should be
written in Powersim according to a certain layout.          Help Model
More about the Model File if you press this button.

To use ParmEst you have to  (Note: use Decimal point NOT comma):

   1. Select the file of the Powersim Model.
   2. Specify each Parameter and Add it:
      a. Type Name the DETERMINISTIC MODEL Parameter.
      b. Type Starting Value of the parameter.
      c. Type initial Step Length of the parameter.
      d. Type Name the corresponding STOCHASTIC MODEL Parameter.
      e. Press the Add button.
   3. Enter and Select the Name of the TWO Objective Functions.
   4. Enter Requested Error (e.g. 0.001 or 1E-5).
  [5. Enter Maximum Number of Iterations (Default=100).]
   6. Press the Fit button.
  [7. When ended, the fitting procedure may continue after change of
      Max Iter or Req. Error.]

  [You can also Halt and/or Reset the procedure.]
```

Specific help is obtained by placing the cursor over a Button, Text field etc.

a1) Help Model

HELP MODEL FILE

The MODEL FILE must consist of the following parts (See example above):

A) A description of the SYSTEM OUTPUT. Usually given as a table function.

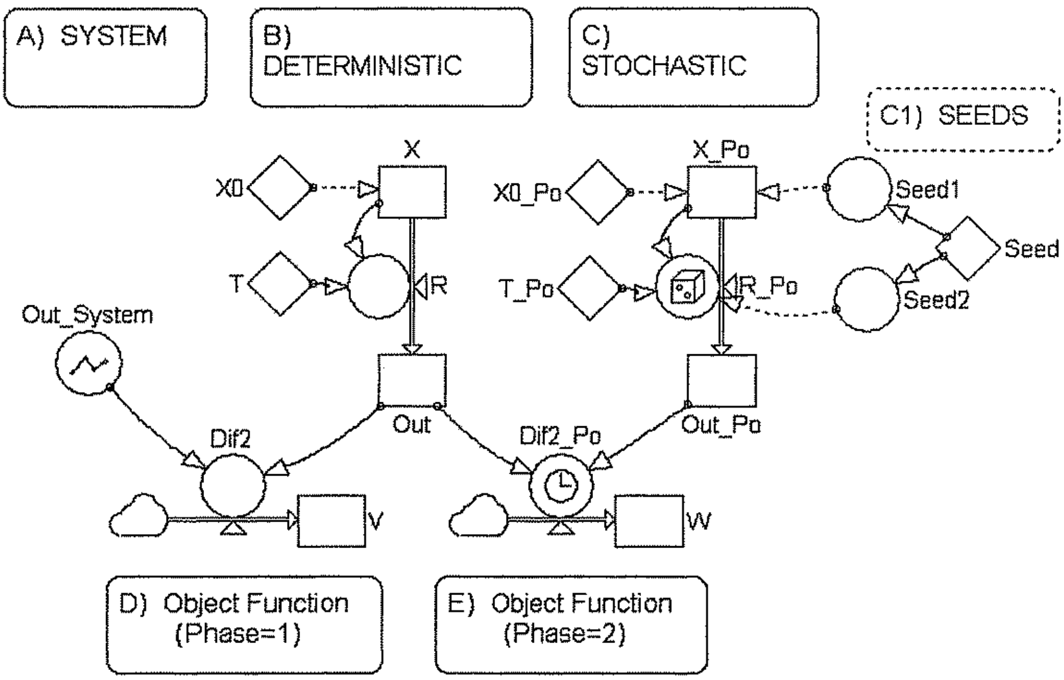
B) A DETERMINISTIC MODEL with parameters (Here X0 and T) to be fitted first to the SYSTEM (Phase 1) and then to the STOCHASTIC MODEL (Phase 2).

C) A STOCHASTIC MODEL with the same structure and corresponding parameters (here X0_Po and T_Po) as for the DETERMINISTIC MODEL.

C1) One Seed 'Seed' provided by ParmEst to control the stochastic model. 'Seed' is then split into several seeds (here 'Seed1' and 'Seed2') to deliver one seed to each Random Number Generator of the STOCHASTIC MODEL. The seeds are held constant during the simulations in a fitting round, but are changed for the next round.

D) An OBJECTIVE FUNCTION (here V) to compare the fitting of the DETERMINISTIC MODEL to the SYSTEM. ParmEst will minimise V during Phase 1.

E) Another OBJECTIVE FUNCTION (here W) to compare the fitting of the DETERMINISTIC MODEL to the STOCHASTIC MODEL. ParmEst will minimise W during Phase 2.



b) 

HELP STATISTICS

After N simulations statistics can be calculated:

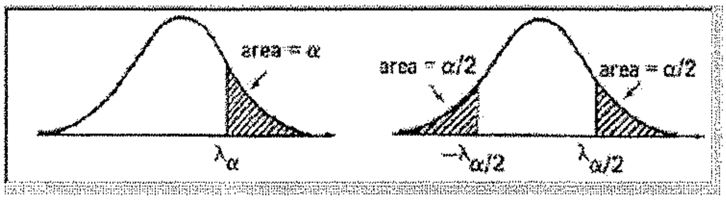
Mean = M = SUM(X1..XN)/N
StdDev = s = Sqrt{SUM[(Xi-M)(Xi-M)]/(N-1)} 'Std. dev. of a sample.

Choose 1-sided or 2-sided conf intervals
Confidence limits for the mean are set by 1-alpha (e.g. 90%, 95%, etc.)
The corresponding lambda (= L1 or L2) is then calculated (see Figure).

Confidence limits around the SYSTEM mean estimate are calculated using
the Normal approximation. (This requires at least N=20 simulations.)

The C.I. interval is then:
1-sided
Interval = X < M + L1·s/Sqrt(N)   or   X > M - L1·s/Sqrt(#Syst.Obs.)

2-sided
Interval = (M - L2·s/Sqrt(N); M + L2·s/Sqrt(#Syst.Obs.))



Institutionerna för biometri och informatik respektive lantbruksteknik gick samman 2003-07-01 och blev Institutionen för biometri och teknik.

Tidigare utgåvor från Institutionen för biometri och teknik

Licentiatavhandlingar

002 Sundberg, Cecilia. Food waste composting – effects of heat, acids and size.

Rapport – miljö, teknik och lantbruk

04:01 Bernesson, S. Life cycle assessment of rapeseed oil, rape methyl ester and ethanol as fuels – a comparison between large- and smallscale production.

Examensarbeten

04:01 Ericsson, Niclas. Uthållig sanitet i Peru – En förstudie i staden Picota.
04:02 Ekvall, Cecilia. LCA av dricksvattendesinfektion – en jämförelse av klor och UV-ljus.
04:03 Wertsberg, Karin. Behandling av lakvatten med kemiska oxidationsmedel för att delvis bryta ned oönskade organiska föreningar – En studie utförd vid Hovgårdens avfallsanläggning i Uppsala.
04:04 Degaart, S. Humanurin till åkermark och grönytor – avsättning och organisation i Göteborgsområdet.

Tidigare utgåvor från Institutionen för biometri och informatik

Institutionsrapporter

2003
80 Edlund, T. Pluripolar Completeness of Graphs and Pseudocontinuation. Licentiatavhandling.
79 Nilsson, K. Macrolide antibiotics – mode of action and resistance mechanisms. Licentiatavhandling.
78 Sahlin, U. Analysis of forest field data with a spatial approach. Examensarbete.
77 Seeger, P. Nested t by 2 Row-Column-Designs suitable for bridge competitions.

2002
76 Wörman, A. Low-Velocity Flows in Constructed Wetlands: Physico-Mathematical Model and Computer Codes in Matlab-Environment.
75 Huber, K.T., Moulton, V. & Steel, M. Four characters suffice to convexly define a phylogenetic tree.
74 Ekbohm, G. Induktion, biometri, vetenskap.
73 Huber, K.T:, Moulton, V. & Semple, C. Replacing cliques by stars in quasi-median graphs.
72 Huber, K.T. Recovering trees from well-separated multi-state characters.
71 Holland, B.R., Huber, K.T., Dress, A. & Moulton, V. δ-plots: A tool for analyzing phylogenetic distance data.
70 Huber, K.T., Koolen, J.H. & Moulton, V. The Tight Span of an Antipodal Metric Space: Part II – Geometrical Properties.
69 Huber, K.T., Langton, M., Penny, D., Moulton, V. & Hendy, Michael. Spectronet: A package for computing spectra and median networks.

68 Åsenblad, N. Multivariate Linear Normal Models for the Analysis of Cross-Over Designs. Filosofie Licentiatavhandling i biometri med inriktning mot matematisk statistik.

Tidigare utgåvor från Institutionen för lantbruksteknik

Institutionsrapporter

248 Lundh, J-E., Huisman, M. En jämförande studie av några maskinella och motormanuella röjningsmetoder utmed järnväg – uppföljning av skottutveckling efter röjning samt utvärdering av selektiv röjning.
249 Ljungberg, D Gebresenbet, G Eriksson, H SAMTRA - samordning av godstransporter: Undersökning av möjligheter och hinder för samordnad varudistribution i centrala Uppsala.
250 Larsolle, A., Wretblad, P. & Westberg, C. A comparison of biological effect and spray liquid distribution and deposition for different spray application techniques in different crops.
251 Tidåker, P. Life Cycle Assessment of Grain Production Using Source-Separated Human Urine and Mineral Fertiliser.
252 Perez Porras, J., Gebresenbet, G. Biogas development in developing countries.
253 Wikner, I. Environmental conditions in typical cattle transport vehicles in Scandinavia.
254 Sundberg, C. Food waste composting – effects of heat, acids and size.
255 Nilsson, D. Harvesting and handling of flax for the production of short fibres under Swedish conditions. A literature review.

Institutionsmeddelanden

02:01 Fredriksson, H. Storskalig sommarskörd av vass - energiåtgång, kostnader och flöden av växtnäring för system med skörd och efterföljande behandling.
02:02 Björklund, A. Latrin och matavfall i kretslopp i Stockholms skärgård.
02:03 Jannes, S. Hantering av slaggvatten på högdalenverket – ett helhetsgrepp på hanteringen av förorenade vattenströmmar till och från slaggvattensystemet.
02:04 Flodman, M. Emissioner av metan, lustgas och ammoniak vid lagring av avvattnat rötslam.
02:05 Andersson, A. & Jensen, A. Flöden och sammansättning på BDT-vatten, urin, fekalier och fast organiskt avfall i Gebers.
02:06 Hammar, M. Organiskt avfall för biogas produktion i Götene, Lidköping, Skara och Vara kommuner
02:07 Nilsson, D. Småskalig uppvärmning med biobränslen. Kurskompendium.
03:01 Sjöberg, C. Lokalt omhändertagande av restprodukter från enskilda avlopp i Oxundaåns avrinningsområde.
03:02 Nilsson, D. Production and use of flax and hemp fibres. A report from study tours to some European countries.
03:03 Rogstrand, G. Beneficial Management for Composting of Poultry Litter and Yard-Trimmings- Environmental Impacts, Compost Product Quality and Food Safety.
03:04 Lundborg, M. Inverkan av hastighet och vägförhållande på bränsleförbrukning vid körning med traktor.
03:05 Ahlgren, S. Environmental impact of chemical and mechanical weed control in agriculture. A comparing study.
03:06 Kihlström, M. Possibilities for intermodal grain transports in the Mälardalen region – environmental and economical aspects.