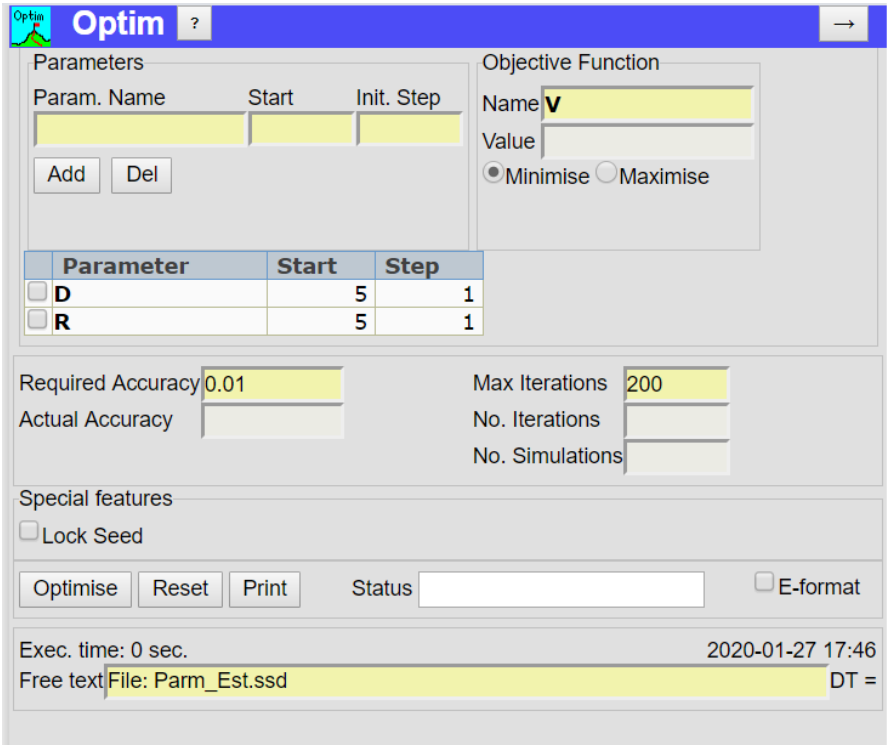
Model fitting, Optimization and Sensitivity analysis

Leif Gustafsson © LAB-2_Model fitting and Optimization.docx 2020-04-17

Contents:



"A good tool improves the way you work.

A great tool improves the way you think."

Name:	**Date:**
Course:	**Approved:**

1. Introduction to Optimization and Model fitting

First, read this short description about the key concepts of *optimization* and *model fitting*.

- **Optimization** means to search for the max or min of something – in practice to *maximize* what we want or *minimize* what we dislike.
- **Objective function**, $V(p_1, p_2, \ldots, p_n)$, is a function that describes how your objective (goal) V depends on the set of parameters $p_1, p_2, \ldots, p_n$. The optimization is the systematic search for the set (combination) of values for these parameters that maximizes or minimizes V. For example, what combination of protein (p_1), fat (p_2), and carbohydrates (p_n) should I eat to maximize my performance (V).
- **Model fitting** is the procedure to make the model behave as similarly as possible to that of the system under study (with regard to the purpose of the model study). Model fitting has two main aspects:
  1) Finding the best *model structure*, and
  2) Finding the *set of values for the parameters* $p_1, p_2, \ldots, p_n$, that minimizes the difference between the Model's and System's behaviours (for a given model structure). This is called **parameter estimation**.
- **Parameter estimation** is thus a minimization task, i.e. it is a special case of optimization.
- **An optimizer** is a programme that maximizes or minimizes an objective function $V(p_1, p_2, \ldots, p_n)$ by systematically searching for the optimal set of values for the parameters $p_1, p_2, \ldots, p_n$. In StochSD the optimizer is called **Optim**.

The interplay between a *simulation model* and an *optimizer* is shown in Figure 1.

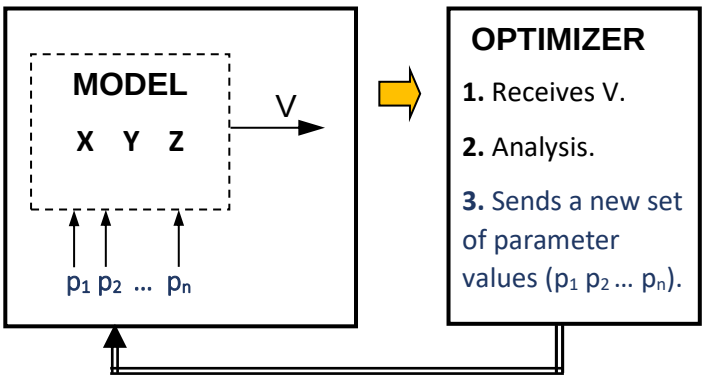


Figure 1. *Optimization* (minimization or maximization) of a model means that an *objective function* V is calculated from the quantities of interest in the model. The objective function V can be any element (X, Y or Z) or any combination of them; for example V=Revenue-Cost-Tax.
V may refer to the state at the End of the replication or be summed up over the time-steps.
The optimizer then analysis if the optimization process is going in the right or wrong direction and sends a new sets of parameter values to be tested by the model.

2. The System under study and a Model of it

We will now study a non-infectious disease that has a long treatable pre-stage before it becomes serious. The *incidence* (i.e. rate) of new pre-cases over age is known. However, an unknown proportion of the Pre-Stage cases will heal (*regress*) spontaneously and the average sojourn (stay) time in the Pre-stage is also unknown.

By screening the entire population, 75% of these Pre-Stage cases can be found (75% *sensitivity* of the screening test) and eliminated. The structure of the disease is shown in Figure 2.

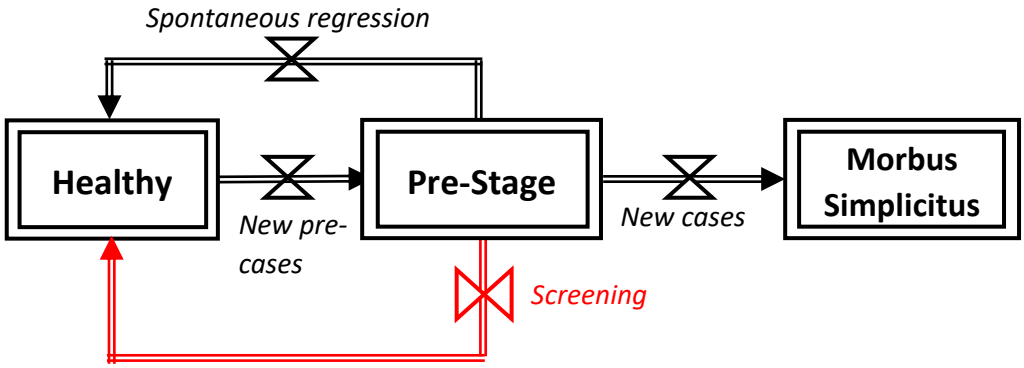


Figure 2. A conceptual model of Morbus Simplicities. Stages in double frames. The Screening & Treatment mechanism is also shown.

In these exercises we want the *focus to be on model fitting*, *optimization* and *sensitivity analysis*, why we want the model to be as simple as possible. Therefore, we will study the unique disease that we name *Morbus Simplicitus*, because it is the only disease where the stages can be described by *single* compartments! (This means that the sojourn time distribution is exponential which is biologically very rare.) See Figure 3.

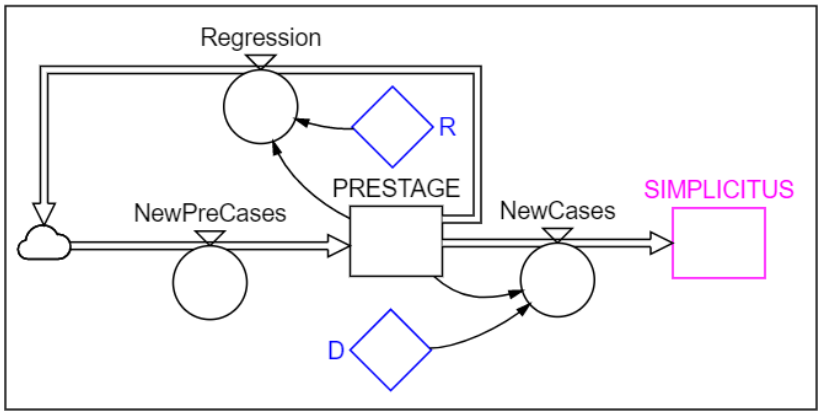


Figure 3. A StochSD model of the disease process (The screening will be implemented later on). The flow of new pre-cases to the pre-stage is known, but the parameters D and R must be estimated. SIMPLICITUS contains the number of individuals with the serious disease that we want to minimize.

Explanations of the very simplified model shown in Figure 3

The purpose of the StochSD model is that it should be *as simple as possible* to keep the focus on how to technically perform *model fitting*, *optimization* and *sensitivity analysis*. This purpose admits the simplifications:

- The Healthy stage is of no interest when we know the flow of NewPreCases, so there is no need to model it.
- We describe the Pre-Stage by *a single compartment* named **PRESTAGE**.
- The stage denoted Morbus Simplicitus is just a counter of the cases getting the serious disease why it can be represented by a single compartment (and a Number Box).
- D and R are parameters that tell the flow rates (fraction of PRESTAGE per time unit) that progress to SIIMPLICITUS or regress (to Healthy).
- Screening will be added later on.

Exercise 1

Start StochSD and define the Time Unit as '**Year**'.

Build the StochSD model shown in Figure 3.

Hint: When drawing a flow, you hold down the left mouse button. To bend the flow, also click the right mouse button when drawing the flow (and continue in a perpendicular direction).

To test that the model will execute, include simple *test values* for the three unknowns: NewPreCases, D and R.

Set NewPreCases to a step that starts at age 20, has its peak at 100. This can be accomplished by a step function: Step(Start time, Height). For example: Step(15, 100).

Set R and D to e.g. 5 years each.

Set simulation Length to 90 years and Time Step to e.g. 0.5 years.

Test that the model works and look at the flow NewPreCases in a Time Plot, and the stocks PRESTAGE (Left axis) and SIMPLICITUS (Right axis) in another.

When the model works, save it as: **Disease_Model.ssd** ■

COMMENT for the interested:
The progressing fraction: p = NewCases/(NewCases+Regression) = (X/D)/(X/D+X/R) = 1/D/(1/D+1/R) = R/(D+R), where X stands for PRESTAGE.

Further, *the sojourn time* T in the PRESTAGE with two outflows is obtained from the differential equation: dX/dt = X/D+X/R = X·(1/D+1/R) = X/T. Thus, 1/T= 1/D+1/R = (D+R)/(D·R), which gives T= D·R/(D+R).

When using *several compartments* in order to model of the pre-stage more realistically, *re-parameterising* to p & T instead of D & R makes the modelling easier, and the global concepts (fraction p and sojourn time T) is easier to interpret than the local time constants R and D (that only relates to the progressing and regressing flows, respectively). However, here we will use the simplest model. □

3. Model fitting

Model fitting is the procedure to make a model behave as similarly as possible as the system under study (with regard to the purpose of the model study).

Model fitting has two main aspects:

1) Finding the best model structure, and

2) Parameter estimation: To Find the set of parameter values, p_1, p_2, …, p_n, that minimizes the difference between the Model's and System's behaviours (for a given model structure).

Finding the best model structure is only mentioned here. For this system under study, you could test different structures for the pre-stage, divide the population into males and females, etc. For each structure you also have to perform a parameter estimation and find out which of the candidate models behaves most similar to that of the system under study.

Parameter estimation is a special case of *optimization* where we *minimize the difference* in behaviour between System and Model by trimming the unknown parameters of the model. See the left side of Figure 4.

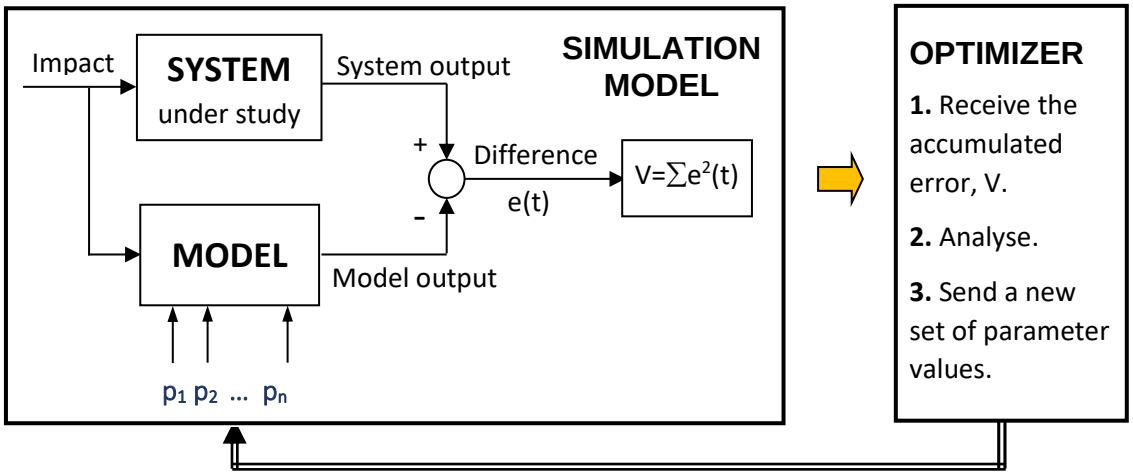


Figure 4. *Parameter estimation* of a model means that the difference, e(t), between the outputs of the system and the model is accumulated into V over the simulation. The optimizer then knows if the optimization process is going in the right or wrong direction and sends a new set of parameter values to be tested by the model. (Minimizing $V=\sum e^2(t)$ means to fit the model's trajectory to that of the system in a *least square* meaning.)

Technically, the comparison of System and Model behaviours must refer to the same situation. If the system is affected by inputs from the environment, this must also be the case for the model. The output for the System under study must also be known, so it is comparable with the Model output.

A **Total model** for parameter estimation has three parts:

A) The SYSTEM description, usually in form of tabulated values over time for input and output.

B) The MODEL with its unknown parameters to be fitted.

C) The OBJECTIVE FUNCTION (V) that measures the difference in the relevant outputs of SYSTEM and MODEL. In a least square minimization, the difference e(t) = SystemOutput(t) – ModelOutput(t) is calculated at each time-step, e(t) is then squared, and $e^2(t)$ is cumulated into the objective function V.

The **Total model** is connected to an **Optimizer**, that orders a simulation run with an initial set of parameter values. Then the **Total model** performs the specified run and delivers the value of V to the **Optimizer**. The **Optimizer** then changes the set of parameter values in accordance with a search algorithm and orders a new run. When the adjustment of the parameters only produces a very small improvement in V (defined by you) the minimization process is terminated, and the *optimal set of parameter values*, and the *value of V* is presented.

Now we will apply this knowledge to the model of the disease process. Therefore, construct the Total model shown in Figure 5.

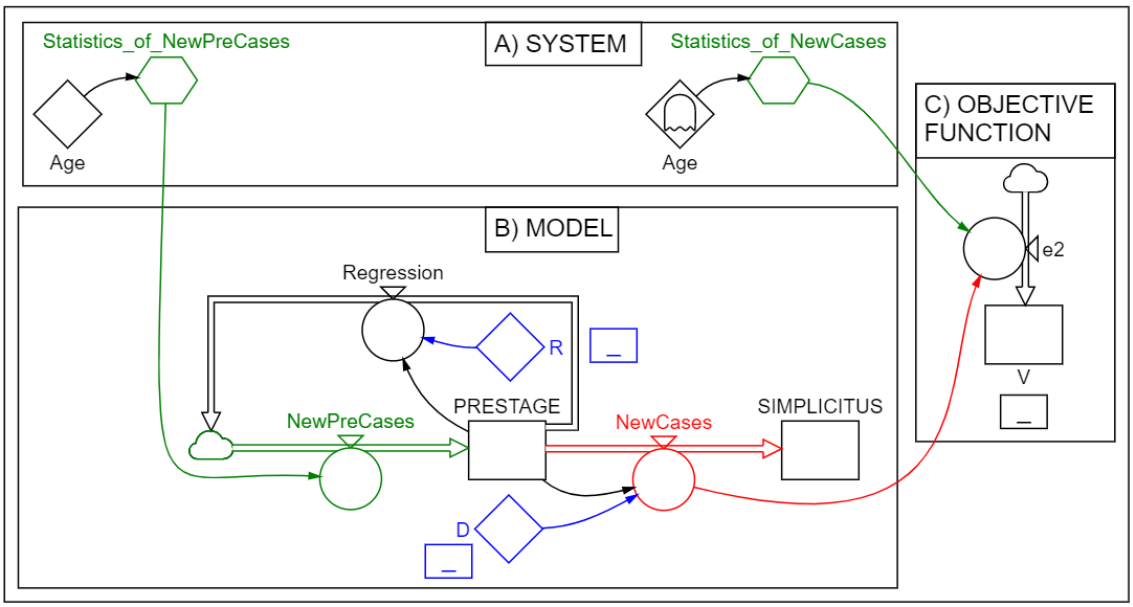


Figure 5. The TOTAL model for parameter estimation of D and R in the Morbus Simplicitus model.

Exercise 2

Open the **Disease_Model.ssd** and save it under the name: **Param_Estimation.ssd** so that most of the Disease model in Figure 3 can be reused. (Start Time: 0, Length: 90 years and Time Step: 0.5 years should be preserved.) Then, complete the model in Figure 5 in accordance with the instructions, below.

A) The System under study

In this case we describe the input of the studied system by statistics of *new pre-cases* over 0 to 90 years of age in a Converter, and output of *new cases* over the same age span in another Converter. The value pairs x,y; are here: Age,NumberOfCases; .

Statistics_of_NewPreCases = 0,0; 5,0; 10,0; 15,1.6; 20,21.2; 25,55.6; 30,136.8; 35,166; 40,143.6; 45,94.4; 50,80; 55,66.8; 60,41.6; 65,48; 70,45.6; 75,36; 80,41.6; 85,29.2; 90,27.2

Statistics_of_NewCases = 0,0: 5,0; 10,0; 15,0; 20,0.8; 25,8.4; 30,25.6; 35,45.6; 40,52; 45,56.4; 50,45.2; 55,40.8; 60,35.6; 65,32.8; 70,23.6; 75,20.8; 80,22; 85,18.8; 90,12.4;

The Converters require incoming x-values, i.e. Age. The easiest way provide this is to start the simulation at time zero so that Age becomes equal to Time. Then link a Parameter named Age, defined by the time function T(), to the Converters, as shown in Figure 5A.

Task: Include the two tables (Converters) driven by Age in the model, *and test that it works*. (By plotting the converters, you can often detect unreasonable table values.)

B) The disease Model

The disease model in Figure 5 is the same as in Figure 3. In this section we focus on *model fitting* and assume that we have an appropriate model structure. Then it remains to estimate the unknown values of the parameters D and R so that the Model will behave as similarly as as possible as the System just described by its input and output.

Therefore, we use the System input: **Statistics_of_NewPreCases** as input to the flow **NewPreCases** in the Model, and the System output: **Statistics_of_NewCases** will be used for comparisson with the output flow **NewCases** in the Model. See Figure 5B.

Task: Link **Statistics_of_NewPreCases** to **NewPreCases** (replacing the test data). *Test that it works!*

C) The Objective Function

Now, we want to fit the output of the Model to that of the System over time in a least squaremeaning by adjusting the parameters D and R. Therefore, build a simple devise that sums up the squared differences (**Statistics_of_NewCases – NewCases)^2** which we denote **e2**. The squared differences of the flow **e2(t)** are then summed up in an empty Stock named e.g. **V**. See Figure 5C.

Task: Include the Objective Function device in the model. *Test that the model works!*

The optimizer

An **optimizer** is a programme that maximizes or minimizes an objective function V(p_1, p_2, …, p_n) by systematically searching for the optimal set of values for the parameter p_1, p_2, …, p_n. In StochSD the optimizer is called **Optim**. See Figure 6. (In the Help menu, you will find the manual describing the search method used and how to use Optim.)

Task: Open **OPTIM** from the Tools menu. The model on the screen and the opened tool are now connected.

Then specify **D** in the 'Param. Name' field, give a value of it to test (e.g. 5) for the first run (in the 'Start' field), and an increment of this value (in the 'Init. Step' field) to be use for the following run. Then press the **Add** button.

Do the same for **R**.

Give the name for the Objective Function – in the model we named it **V**.

Specify the **Required Accuracy** to e.g. 0.01. (When the improvements become smaller than this, the Optim will terminate the search for better values of **R** and **D**.)

Check that Minimize (and not Maximize) of the objective function is selected.

(Max iterations has a default value of 200 runs. There is no reason to change it.)

(You may also add specific information in free text for documentation, for example the name of the model file.)

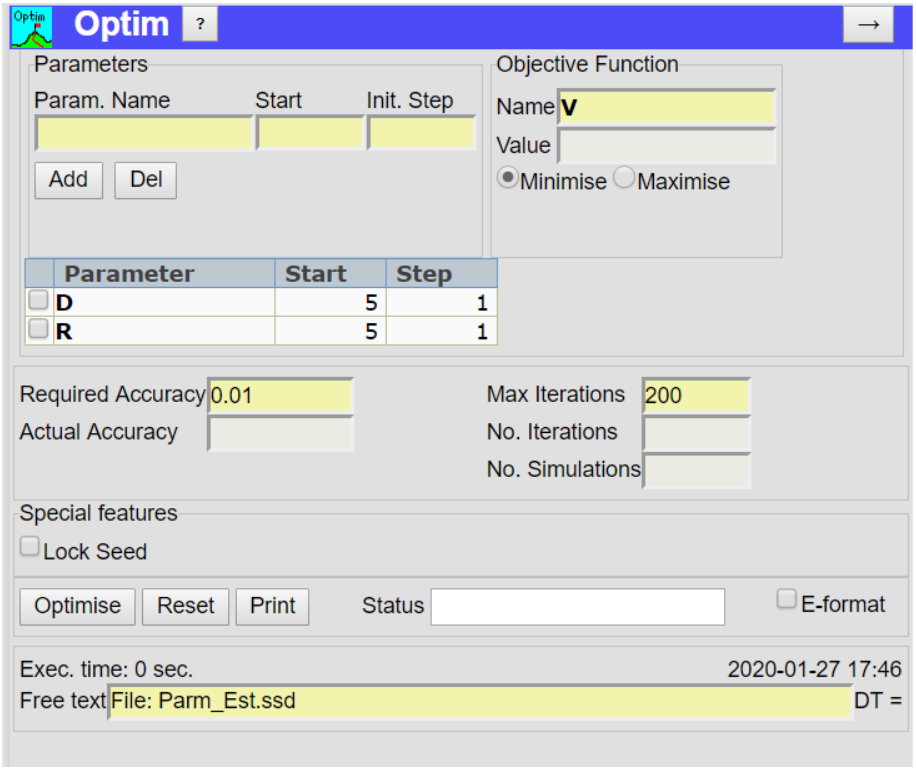


Figure 6. The Optim form ready to start by clicking the **Optimize** button.

Finally, click the **Optimize** button to start the search process of finding the values of **D** and **R** that minimizes **V**. (To speed up the process, graphic outputs in the model are automatically disconnected during the search.)

What was the estimates of the parameters?

Answer: **D =** ……. **R =** ……..

What was the cumulated least squares of the difference?

Answer: **V =** …..……. (**Note:** The value of this minimal V is of interest when you compare it with that of another model structure in order to find the best structure.)

How many individuals will get SIMPLICITUS according to the Model?

Answer: ……

How many individuals got SIMPLICITUS according to the System under study?

Answer: ……. (You can easily obtain this by cumulating the **Statistics_of_NewCases** into an empty Stock.)

Save this model. (You have already named it '**Param_Estimation.ssd**'.) ■

Now we have a model ready for use. In the next Section, we will include and optimize the use of screening.

4. Optimization

With the parameters **D** and **R** estimated and the **Statistics_of_NewPreCases** linked to the flow **NewPreCases**, we now have a complete model of how the disease develops over age for the studied *birth cohort* (people born the same year).

Exercise 3

Start by making a copy of the file **Param_Estimation.ssd** saved as **Opt_Screen.ssd**. Then, start to modify the model to what is shown in Figure 7.

Move the **Age**, **Link** and **Statistics_of_NewPreCases** to the MODEL (part B of Figure 5). Then remove the SYSTEM and the OBJECTIVE FUNCTION parts. The rest of Figure 5A and the whole of Figure 5C can now be removed.

Now, we will include three rounds of screening for the studied birth cohort. We will assume that a screening of the population has the ***Sensitivity*** to find (and eliminate) 75% of the pre-stages in order to reduce future cases of Morbus Simplicitus.

The task is to distribute these three screening rounds over age (i.e. time) in an optimal way so that the number of **SIMPLICITUS** is minimized. (Still, we want the model to be as simple as possible why we e.g. neglect deaths by other reasons, which would make the model more realistic but also more complex, require a survival table, and take a longer time to build and test.)

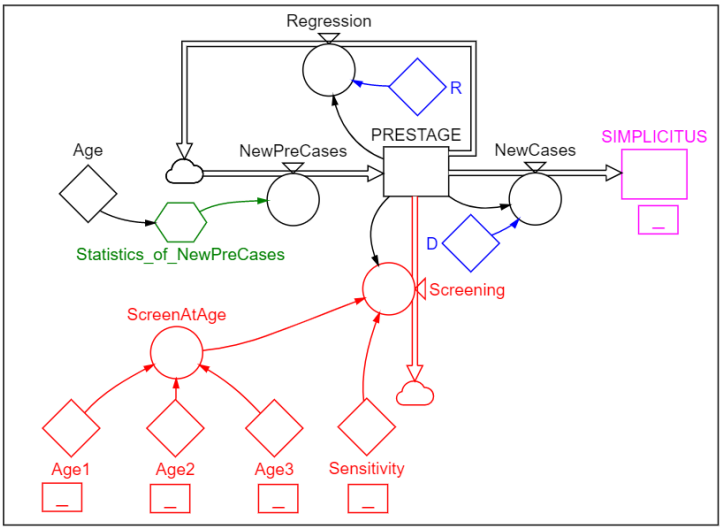


Figure 7. Minimization of SIMPLICITUS by finding the optimal ages for Age1, Age2 & Age3.

Construct the screening devise, beginning with of three age parameters telling when screening should be performed, and one sensitivity parameter as shown in Figure 7.

Then connect **Age1**, **Age2** and **Age3** to the auxiliary **ScreenAtAge**, and define **ScreenAtAge** so that it is active only at the three time-steps surrounding Age1, Age2 and Age3:

IfThenElse(T() >= [Age1]-DT()/2 AND T() < [Age1]+DT()/2
OR T() >= [Age2]-DT()/2 AND T() < [Age2]+DT()/2
OR T() >= [Age3]-DT()/2 AND T() < [Age3]+DT()/2, 1/DT(), 0)

(**Explanation.** The overall structure of this statement is:

IfThenElse(Test Condition, Value if True, Value if False).

The test condition is here weather Age_i (i=1, 2, 3) falls within the current time-step (i.e. between Time-DT/2 and Time+DT/2). If the Value is True then the statement returns 1/DT, and if it is False it returns the value 0. The If-true-case is 1/DT because the screening is performed during a single time-step, and the shorter the DT the larger the action must be during this time-step. (The symbol >= means ≥.)

Also note that it is more secure to include the elements ([Age1], [Age2], [Age3]) by clicking these linked primitives and to get the functions: IfThenElse(…), T(), and DT() from the function lists than to write them.)

Also define the outflow **Screening** as: [PRESTAGE]*[ScreenAtAge]*[Sensitivity].

Give start values to Age1, Age2 and Age3, for example 25, 50 and 75, respectively, and set Sensitivity to 0.75.

Run the model and see what happens in a Time Plot. When the model works well – save it!

Now make a *Compare Simulations Plot*, open the plot and check **PRESTAGE** and **Keep Results**. Then set **Sensitivity** to 0 to eliminate the effects of screening and run the model. Also read the number of cases in **SIMPLICITUS**.

Answer: The number getting SIMPLICITUS *without* screening was: …… cases.

Now, set **Sensitivity** = 0.75 and run again.

Answer: The number getting SIMPLICITUS *with* screening was: …… cases.

Answer: The reduction of SIMPLICITUS because of screening was: …… cases.

Sketch the results without and with screening in the *Compare Simulations Plot* in Figure 8.

Don't forget to check the 'Keep Results' box so that you can compare with older simulations.

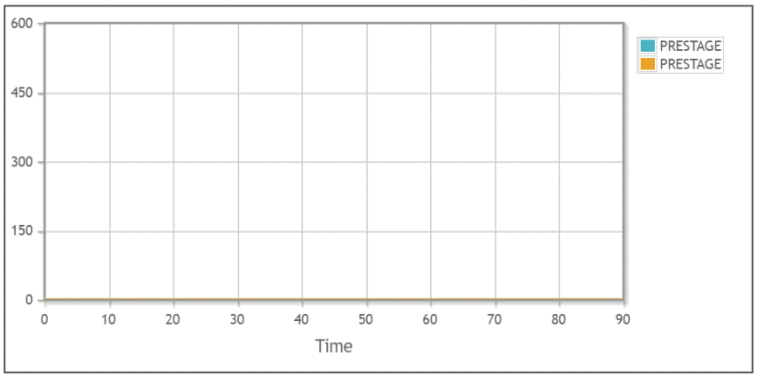


Figure 8. Simulations without and with screening.

Task: Explain what you see by inserting the numbers from the three answers above, for the areas under or between the curves in Figure 8.

Age1, Age2 and Age3 are the set of parameters to be optimized in order to minimize the content in SIMPLICITUS. So, open **Optim** again. If it contains R, D and V from a previous session, then click the **Reset** button so that you can delete R, D and V. (Check a quantity and click **Del.**)

Now, with Sensitivity = 0.75. Then insert e.g.: Age1, 25, 1 and click **Add**. Age2, 50, 1 and **Add**. Age3, 75, 1 and **Add**. Since the purpose is to *minimize* SIMPLICITUS, this will be the Objective function to enter.

Required Accuracy can be 0.01.

Check that **Minimise** is selected.

Then click the **Optimize** button.

What is the set of optimal screening ages? How many cases of SIMPLICITUS will then happen?

Answer: Age1 = ……, Age2 = …..., and Age3 = ….. ⇒ SIMPLICITUS = …… cases.

What is the *reduction* by optimal screening according to the model? **Answer:** ....… cases. ■

Comment: The parameter set that maximizes an objective function V is the same set that minimizes -V. Therefore, maximization and minimization are performed by the same algorithm (with a change of sign). For example, the Screening flow could be accumulated in a Stock of Removed_PreCases, and *maximization* of the content in this Stock could be performed. □

Comment: Maximization can be regarded as the search for the top of a hill in a parameter space with a vertical axis for the objective function (and minimization as the search for the bottom of a cavity). See Figure 9.

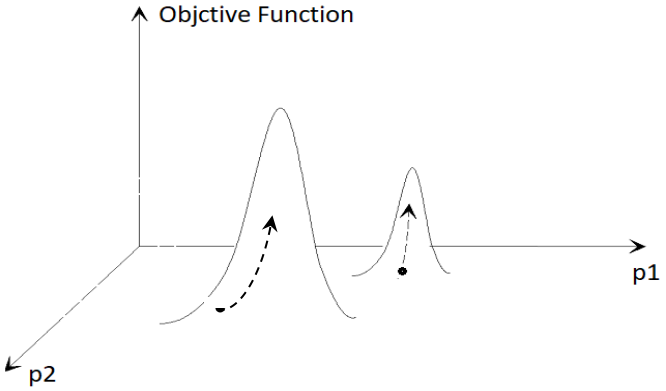


Figure 9. The search method used in an optimizer strives upwards (or downwards) from its starting position. So, there is a risk of finding a *local* instead of *global* optimum depending on from where the search is started.

For more complex models with several optima, you have the risk of finding a *Local optimum* instead of the *Global* one. Knowledge about the model or starting from different points can usually handle this problem. □

5. Sensitivity analysis

5.1 Introduction

Sensitivity analysis is an important and versatile technique to investigate how the output of a model depends on its inputs. Such inputs can be: Parameters, Timing of an action, Initial values, Impacts from the environment, Structural changes of the model, etc.

Sensitivity analysis is used for a wide range of purposes during a modelling project such as:

Model building
- Model simplification – finding and removing model inputs that have no or little effect on the output, or identifying and removing redundant parts of the model structure.
- Sensitivity analysis can examine where improved precision of input is necessary.

Validation of the model
- Testing the robustness of the results of a model.
- Searching for errors in the model (by detecting unexpected relationships between model inputs and outputs).
- To compare important causal connections observed in the system under study with the corresponding input-output relation in the model.

Model analysis and evaluation
- Increase understanding of the relationships between input and output.
- To evaluate the confidence in a model requires **1)** an *uncertainty analysis* to quantify the uncertainty in model results, and **2)** a *sensitivity analysis* to evaluate how much each input contributes to the output uncertainty.

A typical procedure for sensitivity analysis
A common and simple approaches is to change **one-factor-at-a-time** to see how this affects the output. This procedure includes the following steps:

1. Quantify the uncertainty in each input.
2. Identify the model output (often an objective function) to be analysed.
3. Make a *baseline* simulation.
4. Run the model once for each input to be changed from the baseline settings.
5. Use the resulting model outputs to calculate the sensitivity measure of interest.

Comment: There are more advanced versions e.g. to handle:
- *Correlated inputs***:** Most common sensitivity analysis methods assume independence between model inputs, but sometimes inputs can be strongly correlated.
- *Model interactions:* Interactions occur when *simultaneous* perturbations of inputs causes a change in the output that is larger than the sum of changing each of the inputs.

There are also methods for *stochastic models.*

Sensitivity measures

A measure often used is the ratio between the change of an output, ΔV, (often the objective function) and the change of an affecting input, Δp.

The sensitivity can be expressed in absolute terms as: $Absolute\ sensitivity = \frac{\Delta V}{\Delta p}$,

or in relative terms (e.g. in percent) as: $Relative\ sensitivity = \frac{\Delta V/V}{\Delta p/p}$; where Δ denotes a *small* change. (E.g. telling how many *percent* V will change when p is changed by one *percent*?)

5.2 Sensitivity analysis of the model

Now we want to study what effect the change in some factors in our model will make by using the **Sensi** tool. Factors we *could* examine are:

1. The **Sensitivity** of screening.
2. The progressing time constant **D** and the regressing time constant **R**.
3. The size of the inflow **NewPreStages**.
4. Timing of actions: **Age1**, **Age2**, **Age3**.
5. The **Initial values** (Not applicable in this model. But if you could be born with the disease or its pre-stage as the case for e.g. HIV, then we could test how it affects the output.)
6. **Qualitative changes** (e.g. we include more screening rounds).
7. **Structural changes** (e.g. a pre-stage is represented by several compartments).

Exercise 4

Task: *Specify the **baseline settings*** and *the **outcome of interest*** for the case with the optimal timing of the three screening rounds (see Section 4, above), and include them in the Table.

The baseline settings

D_0 =

R_0 =

k_0*NewPreCases, where k_0=1

Initial values of $PRESTAGE_0$ = 0

Initial values of $SIMPLICITUS_0$ = 0

Number of compartments describing the pre-$stage_0$ = 1

Optimal screening:

Number of $screening\ rounds_0$ = 3

$Age1_0$ =

$Age2_0$ =

$Age3_0$ =

$Sensitivity_0$ = 0.75

⇓

Number of $SIMPLICITUS_0$ over life =

Task: Now, open the **Sensi** tool. Enter **D**, **R**, **Age1** and **Sensitivity** together with their *baseline values* and *increments* (Δp_i: see the left column in the table below) and add them to the Result grid below. Also add the output **SIMPLICITUS**, and then press the **Run** button.

Sensi will now make a baseline run, and then run the model for each perturbed value ($p_i+\Delta p_i$) with restoring to the baseline settings after each simulation. Finally, the sensitivities are calculated and presented. (With the 'radio buttons', you switch between Absolute and Relative sensitivity.)

Changed settings	$\frac{\mathbf{\Delta V}}{\mathbf{\Delta} p}$	$\frac{\mathbf{\Delta V/V}}{\mathbf{\Delta p}/p}$
$D_0 \rightarrow D_0 + 1$ year		
$R_0 \rightarrow R_0 + 1$ year		
$Age1_0 \rightarrow Age1_0 + 1$ year		
$Sensitivity_0 \rightarrow Sensitivity_0 + 0.01$		

Task: Which of these changes had the strongest effect on the output in *absolute* and in *relative* terms?

Answer: .....

…… ■