

1 Sensitivity_SobolMethod Tutorial

2 Introduction

In this tutorial we will be looking at the Sobol method. It is a quantitative sensitivity analysis method, which assigns a percentage of output variance to each input or combination of inputs. This tutorial will build on the concepts introduced in the `MorrisMethod` tutorial. It may be useful to have a look at the `MorrisMethod` tutorial first, if you have not already done so. The `MorrisMethod` tutorial introduces sensitivity analysis inputs and outputs in the context of APSIM. It also introduces the nitrogen leaching simulation used in this tutorial.

The Sobol method is a robust, reproducible sensitivity analysis method. It is based on the idea that a mathematical function of m inputs can be decomposed into 2^m functions, each corresponding to one combination of inputs. It belongs to a class of sensitivity analysis methods named variance-based methods. A good overview of these may be found in [Pianosi et al., 2016](#).

Just like the Morris method, the Sobol method can be used to identify inputs that have a large contribution to output variability, or a negligible contribution to output variability. The Sobol method has two major advantages however:

- * The Sobol method can *quantify* the contribution of each input, as a percentage of total output variance.
- * The Sobol method can also quantify the contribution of interactions between inputs.

The additional information comes at a price: the Sobol method is very computationally intensive. This is an issue for larger models, with many inputs or long run times. The Sobol method is most often used as a follow-up to less computationally intensive methods, for example the Morris method. A typical sensitivity analysis may involve the following steps:

1. Use the Morris method to analyse *all* inputs of interest, and identify non-influential inputs.
2. Fix non-influential inputs to a constant value.
3. Use the Sobol method on the remaining inputs, assigning a percentage of output variance to each input or combination of inputs.

Note we do not need to use the above approach in this example, as we are only analysing a handful of inputs.

This tutorial will show you how to set up a Sobol analysis in APSIM. It will then look at the analysis results, and compare them to the results of the Morris method. The limitations of the current Sobol method implementation will also be highlighted.

3 Sobol Method Setup

4 Base Simulation

The structure of the Sobol method is very similar to the structure of the Morris method. The Sobol method is run from a `Sobol` node, which can be found in `Models`, and should be added at the same level as the `DataStore`. Like the `Morris` node, the `Sobol` node requires a base simulation. We are using the same simulation (`UrinePatchSimulation`), but with a few modifications. You may wish to expand the base simulation, to remind yourself of its structure. The current implementation of the Sobol method cannot analyse multiple values of the same output. Hence the simulation runs for 3 years only, and reports a single value of nitrogen leached (`LeachN`).

The Sobol method shares some additional reporting requirements with the Morris method:

- * The report node must be named 'Report'.
- * `[Clock].Today.Year` must be included as a reporting variable.

The finished base simulation is copied into the Sobol node. The original simulation may again be deleted.

5 Sobol Node

Select the Sobol node, `SobolMethodExample`. It has a very similar structure to the Morris node. Do not let this fool you: The underlying analysis is in fact very different! The input information is again entered in the bottom panel of the user interface. If you are unsure of the process, revisit the Morris method tutorial.

The `LowerBound` and `UpperBound` are now used a little differently. In this implementation, each input range is used to define a uniform distribution:

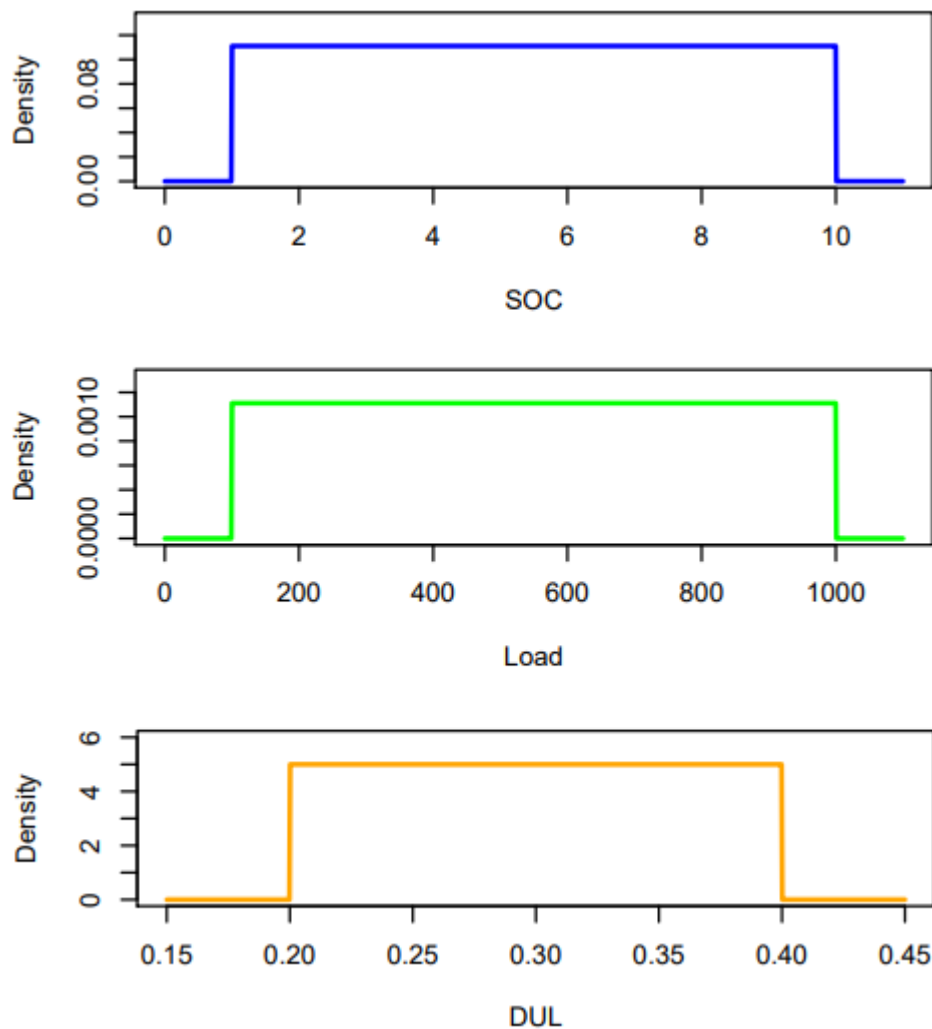


Figure 1: Input Uniform Distributions

Each uniform distribution is used to produce a sample of corresponding input values. Like with the Morris method, the range is one of the most important pieces of information fed into the analysis. See the Morris method tutorial, or Section 4 of [Pianosi et al., 2016](#), if you need more information.

The size of the sample drawn from each uniform distribution corresponds to the `Number of paths` (top panel of the user interface). Note that the meaning of path here is slightly different than in the Morris node. Later on, you will again see how to assess the adequacy of the `Number of paths` value. Here the value is small, to reduce analysis run time. In practice, several hundred paths may be required for a useful result.

6 The Sobol Method Algorithm

Now run APSIM from the `Simulations` node, if you have not done so already. It may again take a couple of minutes to finish. The algorithm behind the Sobol method is complex, and largely beyond the scope of this tutorial. The method typically produces two sensitivity measures for each input:

* A first-order index. The i th first-order index is the proportion of output variance due to varying input i alone (i.e. not considering any interaction effects).

* A total-order index. The i th total-order index is the proportion of output variance due to varying input i , taking into account all interaction effects involving input i .

Currently, only the latter is returned to the user in APSIM. Some additional information regarding the Sobol method is provided below, for the interested reader.

The columns from the two matrices are 'mixed and matched' to produce the sensitivity indices. Note that in theory any

order index may be produced (i.e. an index for any *group* of inputs). In practice, this has a prohibitive associated computational cost.

The (very) mathematically inclined reader may find a detailed account of Saltelli's version of the Sobol method in [Saltelli, 2002](#).

7 Analysis Results

The Sobol analysis produces two data sources: `Report` and `Statistics`. `Report` is not very interesting in this example, so we will focus on `Statistics`. Select the graph node, `TotalEffectIndices`. The provided graph (Figure 1) contains the total effect indices for each input. The orange dots indicate a 95% confidence interval. The confidence interval is quite large in this case, and the estimates not particularly useful. The interval may be reduced by increasing the sample size (`Number of paths`). If you have time, you may wish to try increasing the `Number of paths` to 100. The analysis will take longer to run, but the results will be more useful.

Compare this to the Morris analysis plot (`MorrisMethod` tutorial, Figure 3). Hint: use the filter feature in the `Series` node to select the year 1974. This way you can directly compare the two sets of results. Note that the results of both the Morris and Sobol method will differ somewhat between runs. The Sobol method identifies `Load` as the most influential input, which aligns with the general findings in the Morris method tutorial. The inputs `SOC` and `DUL` also account for some variability. This again aligns with the Morris method, where `SOC` and `DUL` were shown to be important in some years.

8 TotalEffectIndices

TotalEffectIndices

9 Conclusion

The urine patch simulation was used to demonstrate the use of another sensitivity analysis method. The Sobol method provided quantitative sensitivity estimates for each input (i.e. total-effect indices). The total-effect indices provided more detailed sensitivity information, but largely aligned with the findings of the Morris method. `Load` was found to be the most influential input, however `SOC` and `DUL` were also of some importance. Temporal variability was not analysed in this example, but additional Sobol analyses could be carried out to do so.

10 References

Pianosi, Francesca, Beven, Keith, Freer, Jim, Hall, Jim W., Rougier, Jonathan, Stephenson, David B., Wagener, Thorsten, 2016. Sensitivity analysis of environmental models: A systematic review with practical workflow. *Environmental Modelling and Software* 79, 214-232.

Saltelli, Andrea, 2002. Making best use of model evaluations to compute sensitivity indices. *Computer Physics Communications* 145 (2), 280-297.