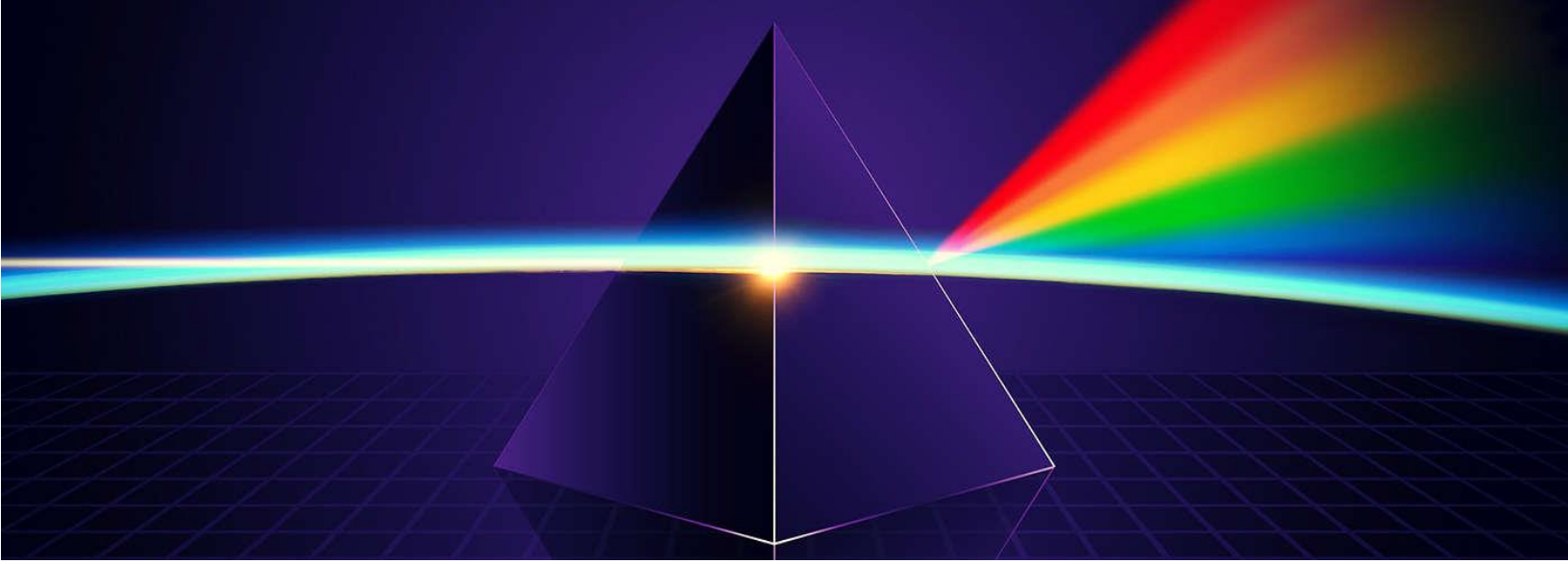


VARS-TOOL

A toolbox for

Sensitivity and Uncertainty Analysis

USER'S MANUAL

Lead Developer: Saman Razavi

November, 2018

Table of Contents

1. What is VARS-TOOL? 4
2. What is New in This Version? 5
3. Features of VARS-TOOL at a Glance 6
4. How to use this Toolbox? 7
5. VARS 8
   5.1. VARS Run File 8
   5.2. VARS Input Files 9
   5.3. VARS Output Files 11
   5.4. Simulation Model File 13
   5.5. Beyond MATLAB, Off-line Mode, and Parallel Simulations 14
   5.6. How to Interpret VARS Results? 14
6. Sobol 18
   6.1. Sobol Run File 18
   6.2. Sobol Output Files 19
7. Morris 20
   7.1. Morris Run File 20
   7.2. Morris Output Files 21
8. RSA 22
   8.1. RSA Run File 22
   8.2. RSA Output Files 23
9. Model Emulation: Handling Model Crashes 25
   9.1. Model Emulation Run File 25
   9.2. Model Emulation Output File 26
   9.3. Example with HBV-SASK Model 26
10. Example 1: VARS Off-line Mode with a Single-Output Model 30
11. Example 2: Models with time series output via the Generalized Global Sensitivity Matrix (GGSM) approach 35

Appendix A: The HBV-SASK Hydrologic Model 39
   A1. Model Description 39
   A2. How To Run HBV-SASK? 41

A3. Case Studies 41
A4. Model Performance 43
Appendix B: Example of external model runs 44
B1. Running models offline with OSTRICH 44
1) Running VARS to sample parameter sets 44
2) Running Model (MESH) offline using OSTRICH 45
2.1. Compiling MPI Ostrich and Model (MESH) in Linux 46
2.2. Arranging structure of files/directories for linking OSTRICH and the model 47
2.3. Preparing OSTRICH files 47
2.4. Submit the job to execute OSTRICH 49
3) Running VARS to generate sensitivity indices 50
B2. Model description and case study 51

1. What is VARS-TOOL?

VARS-TOOL is a next-generation, multi-method sensitivity and uncertainty analysis software toolbox, applicable to the full range of computer simulation models, including Dynamical Earth Systems Models (DESMs). Developed primarily around the powerful "Variogram Analysis of Response Surfaces" (VARS) framework, VARS-TOOL provides a comprehensive suite of algorithms and methods for global sensitivity analysis (GSA), including ones based on the derivative-based (such as the method of Morris), distribution-based (particularly the variance-based method of Sobol'), and variogram-based (such as VARS) approaches.

Equally important is the fact that VARS-TOOL includes a set of highly efficient sampling techniques that help to reduce costs while maximizing robustness and rapid convergence to stable sensitivity estimates; these include a new technique called Progressive Latin Hypercube Sampling (PLHS) that preserves the distributional properties of the sample as additional sample points are progressively collected.

Special features of VARS-TOOL include (1) tools for analysis of dynamical systems models based on the new Generalized Global Sensitivity Matrix (GGSM) approach, (2) factor grouping for dealing with high-dimensional problems, (3) visualization tools for monitoring stability and convergence, (4) model emulation for handling model failures, and (5) an interface that allows working with any model in any programming language and operating system.

As a test bed for training and research, VARS-TOOL also includes a set of mathematical test functions and the HBV-SASK rainfall-runoff (dynamical systems) model. VARS-TOOL is under continuous development and new capabilities and features are forthcoming.

This version of VARS-TOOL has been written in MATLAB environment. However, it has the capability to be used for any computer simulation models (for example, executable files, or directly in any programming languages), as explained later in this manual.

A free copy of VARS-TOOL can be downloaded for research purposes from www.VARS-tool.com. For commercial purposes, please contact, please contact Dr. Saman Razavi at saman.razavi@usask.ca.

2. What is New in This Version?

The Version 1 (V1) of VARS-TOOL was released in February 2016, which mainly included the STAR-VARS algorithm. The following changes have been made in Version 2 (V2):

- Some bugs have been fixed.
- In V1, factor rankings were such that #1 corresponded to the least influential factor (lowest sensitivity). In V2, this numbering is reversed such as #1 corresponds to the most influential factor (most sensitive).
- An algorithm to directly compute Morris derivative-based sensitivity indices has been added to the toolbox.
- An algorithm to directly compute Sobol' variance-based sensitivity indices has been added to the toolbox.
- A new, unique feature has been added that accounts for the time-dependent properties of dynamical systems models in global sensitivity analysis.
- An algorithm to compute sensitivity indices based on Regional Sensitivity analysis (Monte-Carlo Filtering) has been added to the toolbox.
- The toolbox has been enabled with a range of new sampling strategies, including Progressive Latin Hypercube Sampling (PLHS).
- A new "factor grouping" strategy has been implemented to handle high-dimensional problems. Through this strategy, GSA problems having hundreds of factors become tractable.
- A new "model emulation" strategy has been added to handle model failures (crashes) in the course of GSA.
- A tool for visualization of results and monitoring the GSA performance has been added to the toolbox.
- A test bed including a range of test functions have been added to the toolbox for learning and research purposes.
- A hydrologic model, called HBV-SASK, which is an example of dynamical systems models have been added to the toolbox.

3. Features of VARS-TOOL at a Glance

Table below provides a list of GSA algorithms, sampling strategies, special features, synthetic test functions, and dynamical systems models case studies.

Table 3.1. A summary of tools, capabilities, and test beds available in VARS-TOOL

Sensitivity Analysis Algorithms	Sampling Strategies	Other Features	Test Functions	Real-World Case Study
Derivative-based (Morris) Algorithm that computes "elementary effects" based indices of *Morris (1991)*, *Campolongo et al. (2007)*, and *Sobol and Kucherenko (2009)* (main_Morris.m) Variance-based (Sobol) Algorithm that computes "main effect" and "total-order effect" indices *(Saltelli et al., 2008)* (main_Sobol.m) Variogram-based algorithm that computes IVARS indices (including total-variogram effect), total-order effects, and elementary effects indices, all together, through STAR-VARS *(Razavi and Gupta, 2016b)* (main_VARS.m) The Generalized Global Sensitivity Matrix (GGSM) algorithm coupled with STAR-VARS that computes "time-varying" and "total-period time aggregate" sensitivity indices *(Gupta and Razavi, in review; Razavi and Gupta, in review)* (main_VARS.m) Monte-Carlo Filtering Algorithm (also called Regional Sensitivity Analysis) *(Spear et al., 1994)* (main_RSA.m)	Random Sampling (RND) Latin Hypercube Sampling* *(McKay et al., 1979)* (LHS) Symmetric Latin Hypercube Sampling *(Ye et al., 2000)* (SLHS) Halton Sequence* *(Halton, 1960)* (Halton) Sobol Sequence* *(Sobol', 1967)* (Sobol) Progressive Latin Hypercube Sampling *(Sheikholeslami and Razavi, 2017)* (PLHS) STAR Sampling that links with any of the above (*Razavi and Gupta, 2016b)* (STAR)	Bootstrapping to estimate confidence Intervals on sensitivity indices and reliabilities in factor rankings *(Razavi and Gupta, 2016b)* Factor grouping to identify and group factors of comparable effects (*Sheikholeslami et al., in review)* Model emulation to handle failures of simulation models *(Sheikholeslami et al., in prep.)* Online visualization to monitor stability and convergence of the GSA algorithms Interface to link VARS-TOOL with any model in any programing language and operating system	These functions provide synthetic examples for systems with scalar-model outputs Multi-scale wavy function Dimensions=6 *(Razavi and Gupta, 2016a)* Ishigami Dimensions: 3 *(Ishigami and Homma, 1990)* g-function (SobolG) Dimensions:D *(Saltelli et al., 2008)* Sobol-Levitan (soblev99) Dimensions:D *(Sobol and Levitan, 1999)* Welch function Dimensions:20 *(Welch et al., 1992)* Levy function Dimensions:2 *(Laguna and Martí, 2005)* Eggholder Dimensions:2 *(Jamil and Yang, 2013)* Langermann function Dimensions=D *(Jamil and Yang, 2013)*	The HBV-SASK hydrologic model set up for two case studies, the Banff and Oldman Rivers. These case studies provide real-world examples of dynamical systems with time-dependent (vector) outputs The MESH (Modélisation Environmentale communautaire – Surface & Hydrology) land surface-hydrology model set up for Canadian watersheds. MESH is based on Canadian Land Surface Scheme (CLASS). This is a computationally-intensive, physically-based Earth system model connected to VARS-TOOL via OSTRICH Toolkit

* These sampling strategies are built-in MATLAB functions that are available to VARS-TOOL.

4. How to use this Toolbox?

Download VARS-Tool-v2 from http://vars-tool.com/ and unzip the file. The folder VARS-Tool-v2 can be placed anywhere in the hard drive. No installation is required. VARS-Tool should work on almost any version of MATLAB. VARS-TOOL has been written with the basic (built-in) functions of MATLAB, and for the convenience of users, MATLAB functions that are available only in particular MATLAB toolboxes have been avoided.

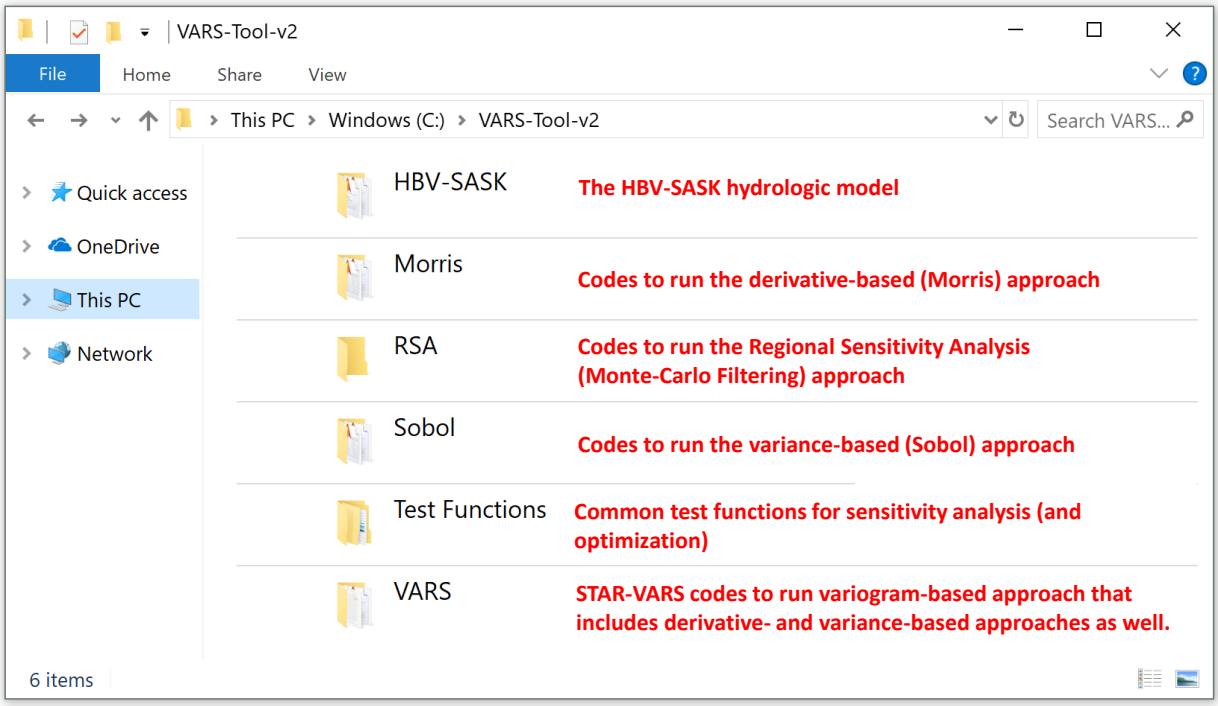


Figure 4.1. VARS-TOOL main folder

Files in VARS-Tool-v2 are arranged in the folders shown in the figure above. In the following sections, descriptions of the files and how to run the different algorithms with examples are presented.

5. VARS

The algorithm in this folder runs STAR-VARS and generates derivative-, and variance-, and variogram-based indices of global sensitivity, all together in one run. The run file, input files, and output files of VARS are illustrated in the following.

5.1. VARS Run File

main_VARS.m

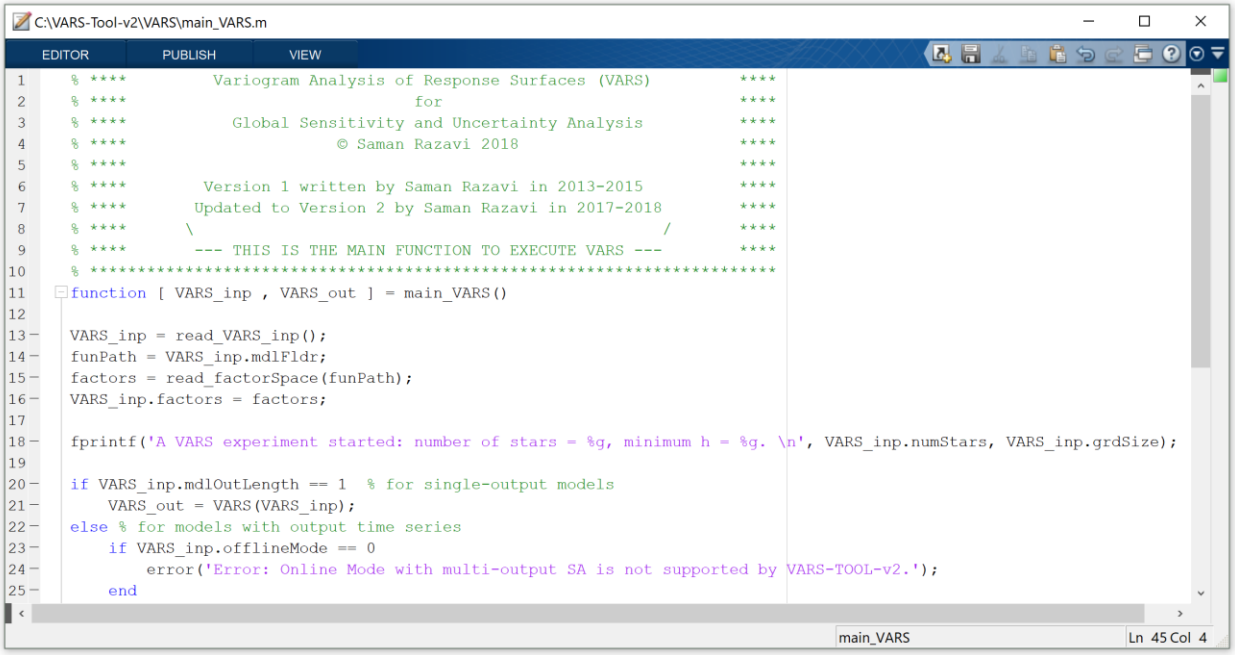


Figure 5.1. main_VARS.m

VARS can be run by **main_VARS.m** located in VARS directory. This is the only MATLAB .m file that the user needs to deal with. The following figure shows example messages that will appear on Command Window upon running this file.

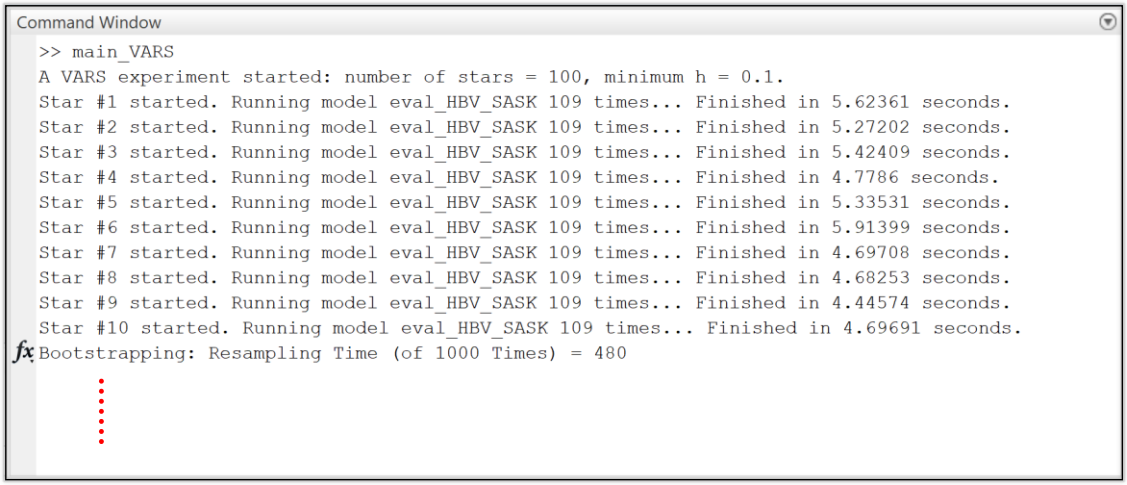


Figure 5.2. A screenshot of command window when running main_VARS.m

5.2. VARS Input Files

VARS_inp.txt

This is the main input file, where the model file, factor ranges, and algorithm parameters are specified. This file must be located in the VARS main directory. The line numbers and orders must not be changed. The percent sign (%) at each line denotes that the following information are comments and will not be used by VARS.

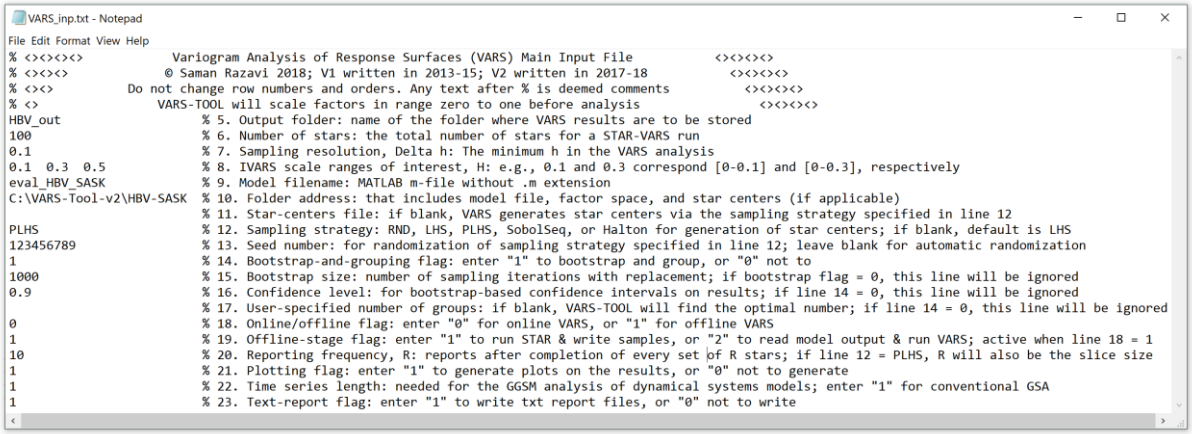


Figure 5.3. VARS_inp.txt

The function of each line is as follows:

Lines 1-4: Comment lines. Any explanation by the user can be included here.

Line 5: Name of the output folder where the results of VARS runs will be stored. This folder will be created in the VARS main directory. If a folder with this name already exists, the existing files in it will be rewritten.

Line 6: Number of stars, *m*. It can take any positive integer number.

Line 7: Resolution of STAR sampling, Δh. It can take any real number between 0 and 0.5. A good choice is to set it to 0.1. Note that VARS scales all factor ranges to [0-1] before processing, therefore, for example, Δh=0.1 refers to 10% of the factor ranges.

Line 8: Scale ranges for IVARS, *H*. It can take any positive real number between 0 and 0.5. For example, to get $IVARS_{30}$, set *H*=0.3 and this refers to a scale range of zero to 30% of the factor ranges. Note that multiple numbers can be put here separated by spaces. The recommended *H* values to be considered are 0.1, 0.3, and 0.5. If only having one sensitivity metric is of interest (a comprehensive metric), $IVARS_{50}$, also referred to as "total-variogram effect" will be a good choice. Note that if any of the specified *H* values is not an integer multiple of Δh, the respective IVARS will be calculated by interpolation.

Line 9: Model filename. The file name is only required with the "on-line" mode for running the computer simulation model. In the "off-line" model this line is discarded.

Line 10: Folder address of inputs. This folder must include factorSpace.txt, which is a necessary input file to VARS and is explained separately below. This folder may also include the model file as specified in line 9 and the star centers file as specified in line 11.

Line 11: Star centers file with its extension (e.g., randSample.txt). This file should be located in the folder specified in Line 10. In this file, the number of rows is *m*, as specified in Line 6, and the

number of columns is *D* (i.e., the number of factors). If no name is provided on this line, VARS will generate star centers via the sampling strategy specified in Line 12.

Line 12: The sampling strategy to generate star centers. Options include RND (i.e., random sampling), LHS (i.e., Latin hypercube sampling), PLHS (i.e., progressive Latin hypercube sampling), SobolSeq (i.e., Sobol' sequence), Halton (i.e., Halton sequence). If the user leaves this line blank, VARS-TOOL uses LHS.

Line 13: Seed number for randomization of the sampling strategy specified in line 12. This seed number is used only for the generation of star centers, and bootstrapping, if activated, will be automatically randomized. If no number is provided on this line, VARS will automatically randomize the random number generator.

Line 14: Bootstrap-and-grouping flag. Enter "1" to run the bootstrap and grouping procedures; and "0" to turn them off. The bootstrap procedure generates confidence levels on sensitivity metrics and reliabilities on factor rankings, and the grouping procedure clusters the factors into groups of similar influence.

Line 15: Bootstrap size. Number of iterations for sampling with replacement, and can take any positive integer number. This number is only needed if bootstrap flag in line 14 is "1". A minimum reasonable bootstrap size is about 100, but the larger the better.

Line 16: Confidence level. The confidence level at which the bootstrap-based confidence intervals (lower and upper bounds) on sensitivity metrics are to be reported. It can take any real number between 0 and 1, with 0.9 as a good choice. This number is only needed if bootstrap flag in line 14 is "1".

Line 17: User-specified number of factor groups. If the user leaves this line blank, the grouping procedure will find and report the optimal number of groups. If line 14 = 0, this line will be ignored.

Line 18: On-line/off-line (internal/external) mode flag. Enter "0" to run the computer simulation model on-line inside VARS, or "1" to run the model in the off-line model. See Section 3.4.

Line 19: Offline stage flag. This flag is active only when line 18 is set to 1. Enter "1" to run STAR and write samples, and "2" to read model output and run VARS. This will be further explained later in Section 3.4.

Line 20: Reporting frequency, *R*. It can take any integer number between 1 and *m* (i.e., the number of stars). After every set of *R* stars, VARS calculates and reports the results.

Line 21: Plotting flag. Enter "1" to plot the results while VARS is running or "0" to turn off the plotting capability. If on, the plots will be updated according to the reporting frequency specified in Line 20.

Line 22: Length of time-ordered model output. This line enables the Generalized Global Sensitivity Matrix (GGSM) approach for dynamical systems models. If this number (length) is greater than one, the GGSM approach will be activated. For example, if the model output is a 365-day time series (365 entries), the length should be 365. Enter "1" for conventional GSA for models with a single response.

Line 23: Text-report flag. Enter "1" to write text-report files at the frequency specified in Line 20 or "0" not to write those files. When the length specified in Line 22 is large, the user may want to turn off this reporting option to save the runtime. If this is off, the user can still access the results in the MATLAB environment.

factorSpace.txt

This file specifies the number of factors to be considered for sensitivity analysis and their lower and upper bounds. The first line is the header (not to be read by VARS), and each of the following lines represents a factor. The first column is for the factor numbers, and the second and third columns are for the lower and upper bounds on the factors, respectively. The lower and upper bounds can take any real values and are dependent on the computer simulation model under investigation. VARS-TOOL will automatically rescale these factor ranges between zero and one, prior to analyses. The fourth column includes the names of the factors (strings), which, if provided, will be used in reporting. The percent sign (%) indicates the beginning of the notes/comments that users may want to leave there; these notes will not be read by VARS-TOOL.

This file should be located in the folder specified at line 11 of **VARS_inp.txt**.

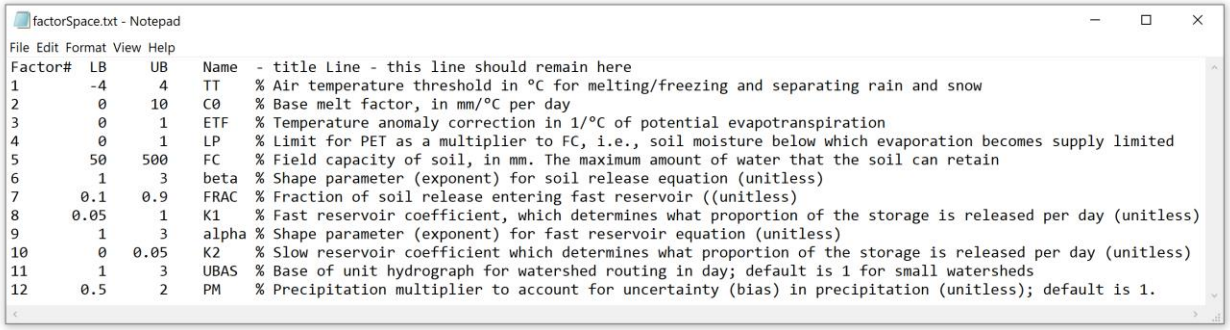


Figure 5.4. factorSpace.txt: example from the HBV-SASK case study

STAR_in.smp

This file is an input file that is only needed for the "off-line mode" (see Section 5.5). Each row in this file is the model response for the respective factor set given in **STAR_out.smp** (see Section 5.3). Note that the order of model response values in this file should be exactly the same as the order of points (factor sets) in **STAR_out.smp**. See VARS Example 1 in Section 8 for full details.

5.3. VARS Output Files

VARS_out_XX.txt

After running **main_VARS.m**, this is the main output file, where XX above is the number of stars used in the analyses (see also the description of reporting frequency in line 20 of **VARS_inp.txt**). The dashed area in the screenshot below was cut to save space here.

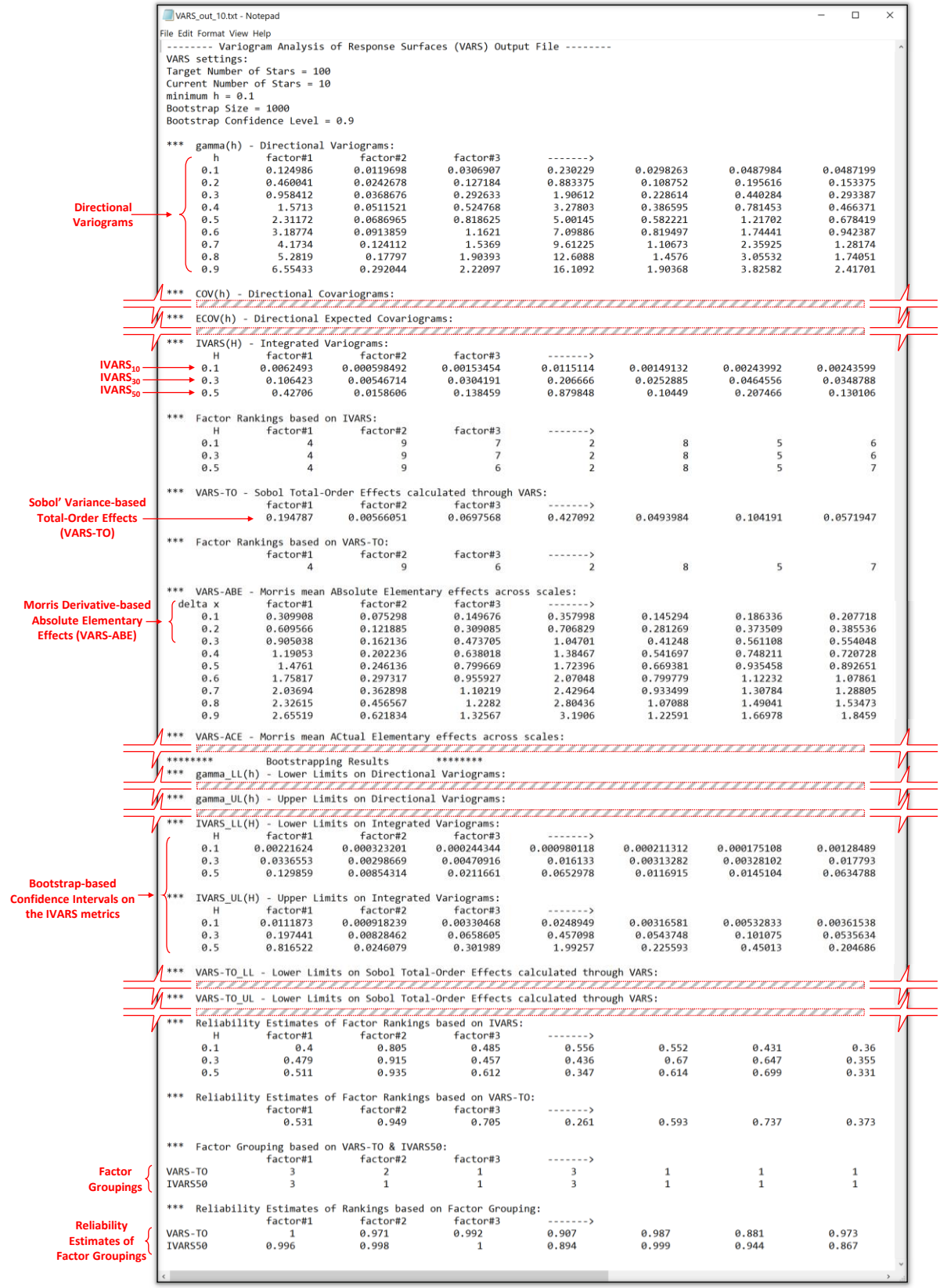


Figure 5.5. VARS_out_XX.txt: example results from the HBV-SASK case study

STAR_out.smp

This file contains the sampled points generated via STAR. VARS-TOOL generates this file, but if the on-line mode is selected (see line 18 of **VARS_inp.txt**), this file does not need to be dealt with.

The sampled points are arranged in this file as follows. Each row represents a point in the D-dimensional problem space (a set of values for all factors). Each column represents a factor (there are D columns). See VARS Example 1 in Section 8 for full details.

STAR_out.mat

This file includes the VARS run information that will only be required by the algorithm in off-line mode (see line 18 of **VARS_inp.txt**).

5.4. Simulation Model File

In the "on-line" mode, the model file specified in line 9 of **VARS_inp.txt** can be an .m file written in MATLAB that receives a vector of factor values (e.g., model parameter values) from VARS and returns the associated model response to VARS. The actual simulation model may be implemented in MATLAB and run by this file, or may be an external command, e.g., an executable file that is called in this file via DOS command. The model file of the example given below is a test function included in VARS-TOOL.

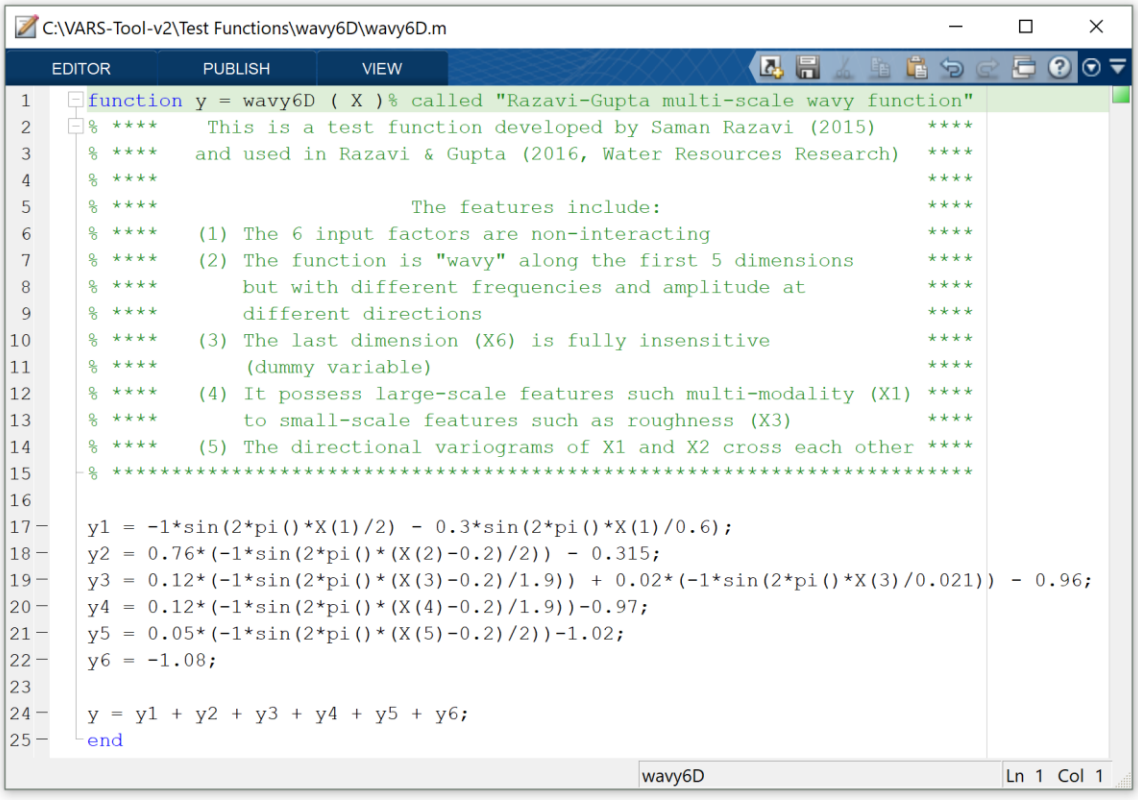


Figure 5.6. Example simulation file: Razavi-Gupta Multi-Scale Wavy Function

5.5. Beyond MATLAB, Off-line Mode, and Parallel Simulations

Two modes for VARS execution are available:

- **On-line (internal):** This mode can be used when the computer simulation model is setup to be called and run through the MATLAB environment. With this mode, a MATLAB .m file is needed that receives a sample set of factors as the input, runs the computer simulation model, and returns the model response as the output. When this mode is selected, sampled points in the factor space will be evaluated (in a serial manner) by the computer simulation model as they are generated.

- **Off-line (external):** This mode is to be used when the computer simulation model needs to be run externally (outside the MATLAB environment), or in a parallel manner for computational efficiency. Three steps are involved: (1) MATLAB generates all of the sampled points and stores them in a text file. (2) The computer simulation model is run externally for all of the sampled points in a Monte-Carlo-type setting and stores the respective model response values in a text file. These runs can be parallelized if multiple processors are available. (3) MATLAB reads in the model runs results and generates sensitivity metrics.

The off-line (or external) mode of the VARS toolbox has been designed for use with any model, running on any operating system, and coded in any programming language. This mode can also be used to parallelize computer simulation models when multiple processors are available.

The files shown in the previous sub-sections with the HBV-SASK case study are to run **main_VARS.m** in the on-line mode. In Sections 8 and 11, two examples are given that run this file in the off-line mode. Another example is given in Appendix B, where a complex physically-based model, MESH, is run externally using OSTRICH toolkit.

5.6. How to Interpret VARS Results?

VARS stores all the sensitivity metrics in **VARS_out_XX.txt**, where XX represents the number of stars used in the analysis. These results include the directional variograms, IVARS metrics (Integrated Variogram Across a Range of Scales) for the scale ranges specified in Line 8 of **VARS_inp.txt**, VARS-based estimates of variance-based Total-Order Effects (Sobol's approach), and VARS-based estimates of different types of derivative-based Elementary Effects (Morris' approach) for a range of step sizes (scales). Table below presents a summary of these metrics.

Table 5.1. Summary of VARS products for Global Sensitivity Analysis

No.	VARS Product	Description
1	$\gamma(h)$	directional variogram
2	$IVARS_{10}$	Integrated Variogram Across a Range of Scales: scale range = 0-10%
3	$IVARS_{30}$	Integrated Variogram Across a Range of Scales: scale range = 0-30%
4	$IVARS_{50}$	Integrated Variogram Across a Range of Scales: scale range = 0-50%
5	VARS-TO	variance-based Total-Order effect (Sobol)
6	VARS-ACE	mean ACtual Elementary effect across scales
7	VARS-ABE	mean ABsolute Elementary effect across scales

Directional variogram represents the variance of change in model response as a function of perturbation scale in a particular direction (distance in the associated direction) in the factor space. In the *n*-dimensional factor space where $\boldsymbol{x} = \{x_1, \dots, x_i, \dots, x_n\}$ represents a location in the space, $y = f(x_1, \dots, x_n)$ represents the model response, and h_i is an increment (size of change) in the i^{th} direction (i=1, …, n), the directional variogram is formulated as

$$\gamma(h_i) = \frac{1}{2} V\big(y(x_1, \dots, x_i + h_i, \dots, x_n) - y(x_1, \dots, x_i, \dots, x_n)\big)$$

where $V(.)$ is variance function. As such, directional variogram represents the rate of variability (i.e., sensitivity) across a range of scales in the factor space - directional variogram is a measure of scale-dependent sensitivity.

IVARS, or Integrated Variogram Across a Range of Scales, integrates the directional variogram over a scale range from zero to H_i in the i^{th} direction

$$\Gamma(H_i) = \int_0^{H_i} \gamma(h_i)\, dh_i$$

and therefore, provides a summary metric for global sensitivity for any given scale range. IVARS_{XX} refers to integrated variogram with an H_i value of XX% (0.XX) of the factor range. The use of IVARS_{10}, IVARS_{30}, and IVARS_{50} are recommended (computed for 0.1, 0.3, and 0.5 of the factor range, respectively). The user is encouraged to investigate the full spectrum of sensitivity information at different scales, however, if having a *single* global sensitivity metric is desired, the user may choose IVARS_{50} as the most comprehensive metric for global sensitivity.

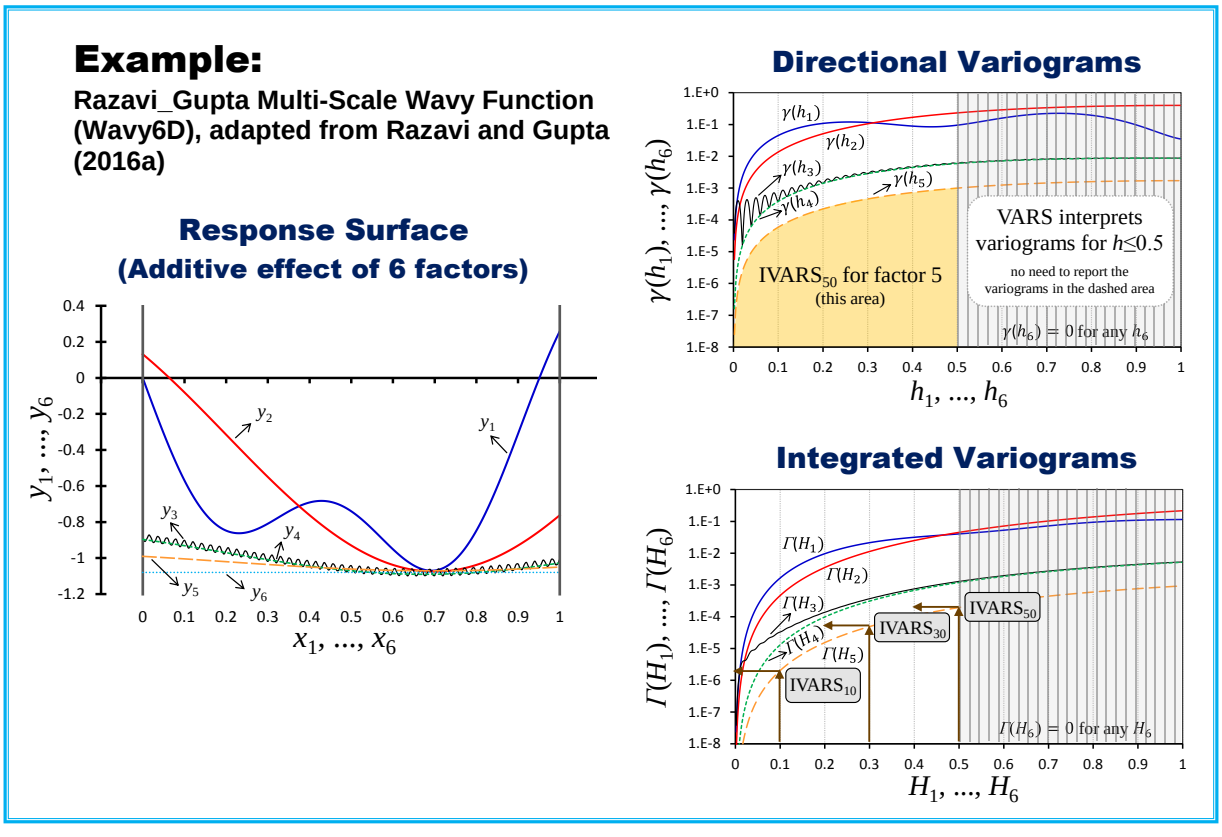


Figure 5.7. Example directional variograms and integrated variograms for Razavi-Gupta Multi-Scale Wavy Function

VARS-TO is the VARS-based estimate of Total-Order effect. Total-order effect is based on a variance-based approach (Sobol's approach) to global sensitivity analysis, where the total variance of a response surface is decomposed into the variance contributions of its different factors and their interactions.

VARS-ACE, VARS-ABE, VARS-SQE are VARS-based estimates of different types of Elementary effects. Elementary effect is based on a derivative-based approach (Morris' approach) to global sensitivity analysis, where sensitivity is seen as some average of numerical derivatives with a specific step size across the factor space. The difference between the three types is related to the treatment of the derivative values (local sensitivities) before averaging (i.e., averaging actual, absolute, or squared values). The unique feature of these VARS-based metrics is that, as opposed to their original versions, they do not depend on a specific step size and provide the derivative-based sensitivity information across the full range of scales.

Sensitivity is a relative concept, and the significance of sensitivity to a factor based on a particular sensitivity metric should only be interpreted relative to those to other factors.

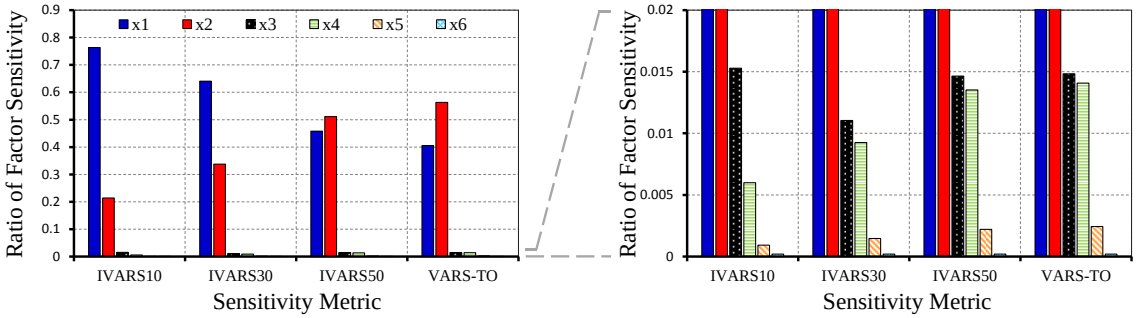
Via its own bootstrap procedure, VARS also generates:

→ **Confidence intervals on sensitivity metrics** – the confidence interval refers to a range wherein the true value of a sensitivity metric lies with a given confidence level (e.g., 90%).

→ **Reliability estimates of factor rankings** – reliability refers to the probability that the estimate of a factor ranking inferred by a sensitivity metric equal the true ranking based on that metric.

Example:

Razavi_Gupta Multi-Scale Wavy Function (Wavy6D), adapted from Razavi and Gupta (2016a)

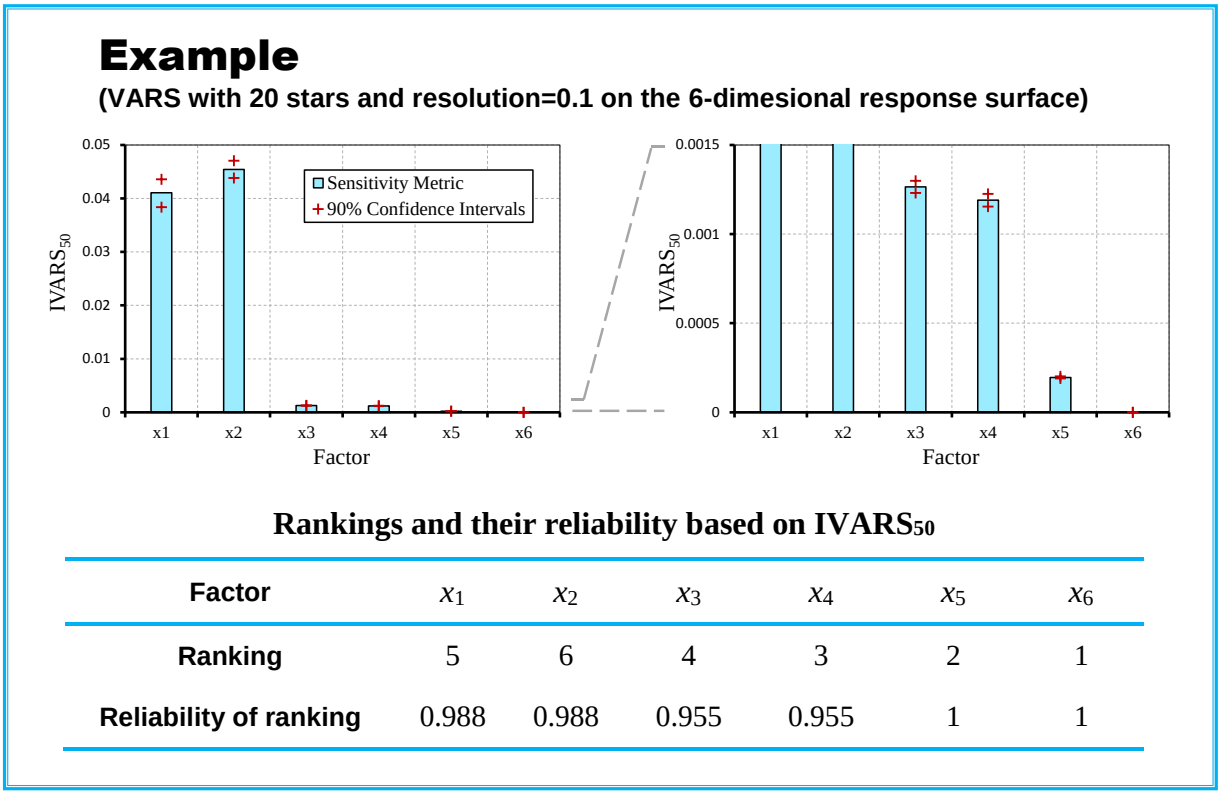


Note: the ratio of factor sensitivity is the value of each metric divided by the summed values of that metric over all of the factors.

Figure 5.8. Example VARS-based sensitivity indices for Razavi-Gupta Multi-Scale Wavy Function

If required, the "true" value of a sensitivity metric and its inferred factor ranking might be obtainable analytically for very simple models. For practical models, however, it typically needs an excessively large number of model runs (e.g., VARS with a large number of stars and a fine resolution) to converge

closely to the true values. The bootstrap feature of this toolbox provides the user with a measure of confidence (uncertainty) and robustness that can be ascribed to the different VARS-based sensitivity metrics. **For more details of VARS, readers are advised to refer to *Razavi and Gupta* (2016a&b).**



Factor	x_1	x_2	x_3	x_4	x_5	x_6
Ranking	5	6	4	3	2	1
Reliability of ranking	0.988	0.988	0.955	0.955	1	1

Figure 5.9. Example VARS-based confidence intervals and reliabilities of the GSA for Razavi-Gupta Multi-Scale Wavy Function

6. Sobol

6.1. Sobol Run File

main_Sobol.m

This is the main file that runs Sobol' variance-based algorithm. The users set control parameters and specifications within this file, as shown in the figure below.

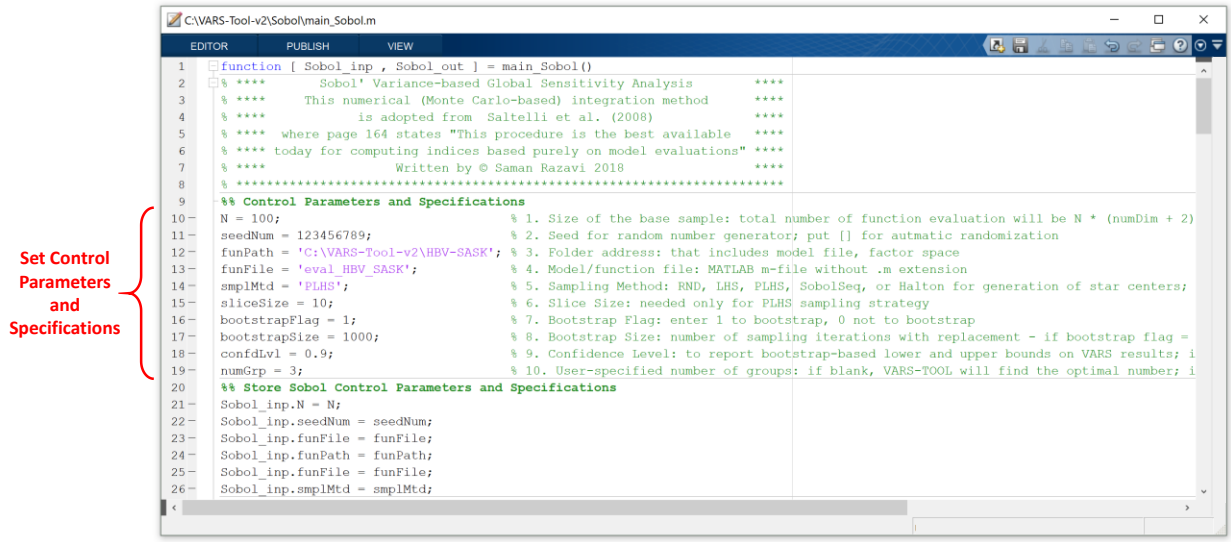


Figure 6.1. main_Sobol.m

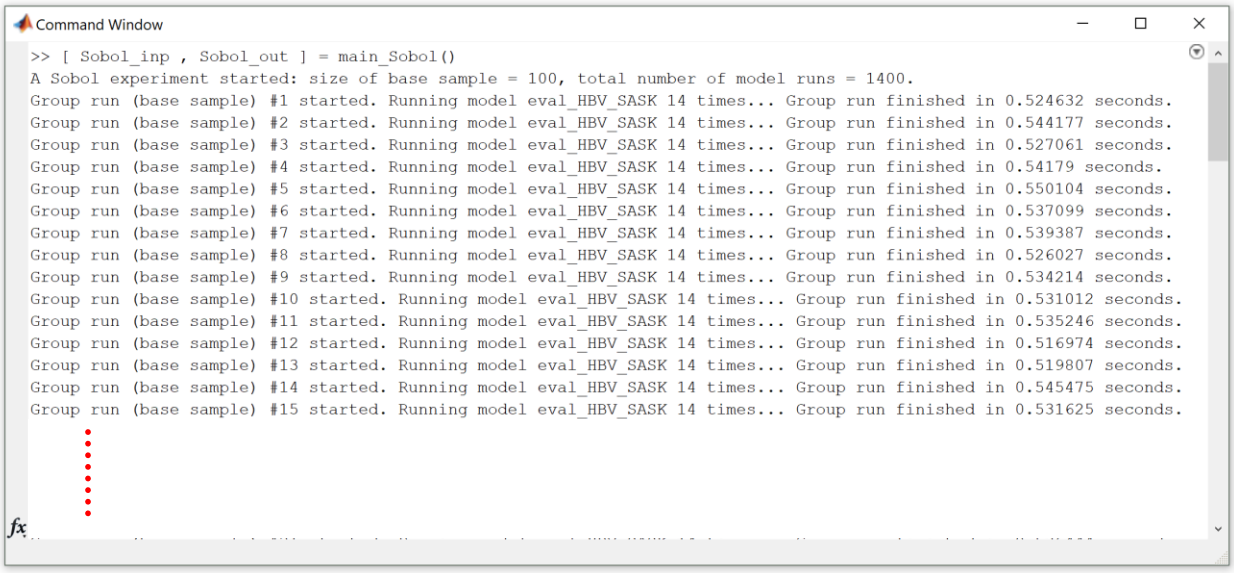


Figure 6.2. A screenshot of command window when running main_Sobol.m

6.2. Sobol Output Files

After running **main_Sobol.m**, it returns two structure arrays: **Sobol_inp** and **Sobol_out**. The former contains the control parameters and specifications, while the latter includes the results of Sobol' run. See the figures below. These two arrays are stored in **Results_Sobol.mat** in the Sobol folder.

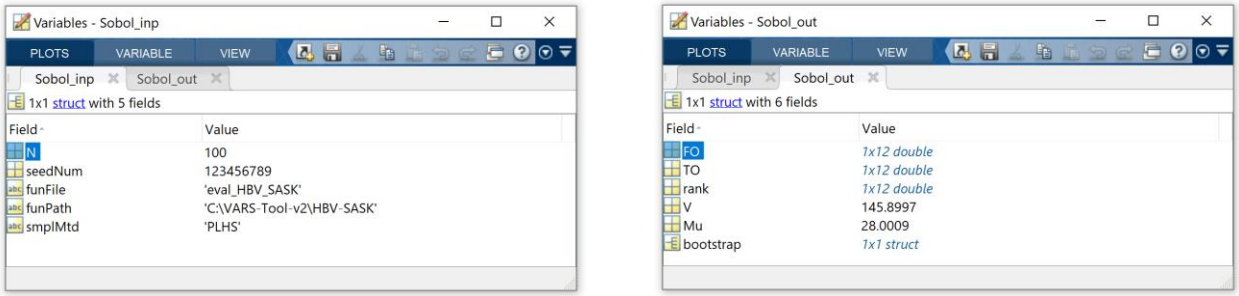


Figure 6.3. Sobol_inp and Sobol_out

7. Morris

7.1. Morris Run File

main_Morris.m

This is the main file that runs Morris' derivative-based algorithm. It returns the GSA indices proposed by *Morris* (1991), *Campolongo et al*. (2007), and *Sobol and Kucherenko* (2009). They include mean (actual) elementary effects, standard deviation of elementary effects, mean absolute elementary effects, and mean squared elementary effects. The users set control parameters and specifications within this file, as shown in the figure below.

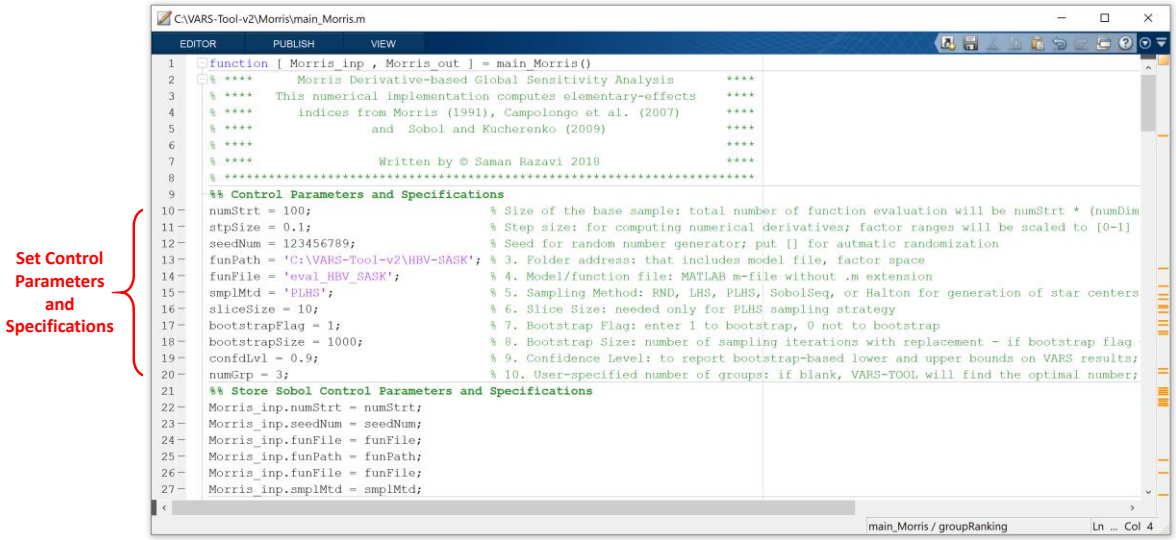


Figure 7.1. main_Morris.m

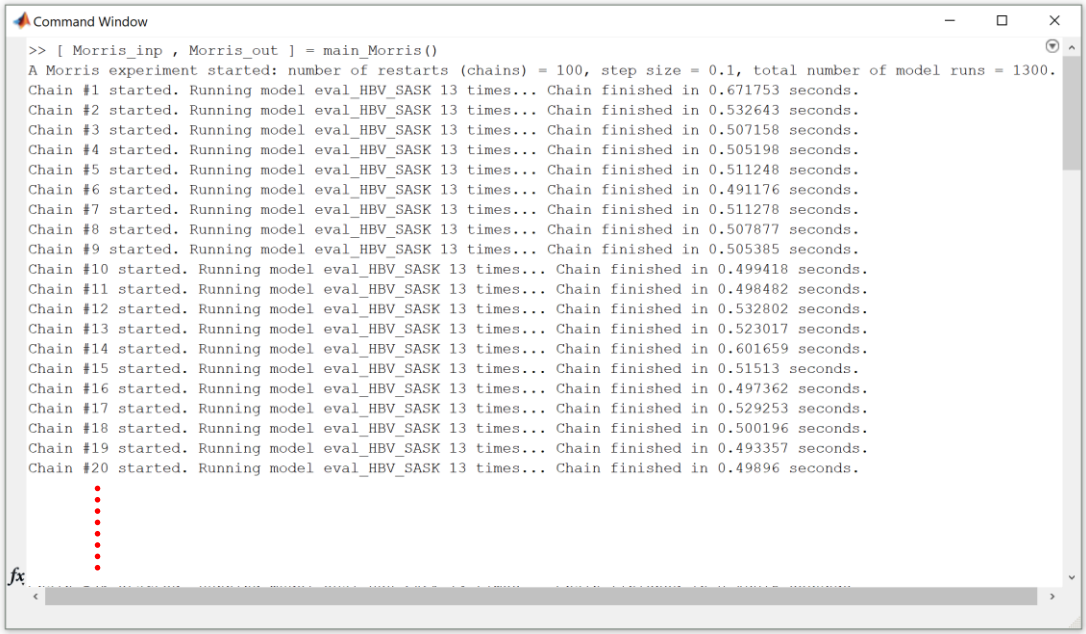


Figure 7.2. Screenshot of command window when running main_Morris.m

7.2. Morris Output Files

After running **main_Morris.m**, it returns two structure arrays: **Morris_inp** and **Morris_out**. The former contains the control parameters and specifications, while the latter includes the results of Morris run. See the figures below. These two arrays are stored in **Results_Morris.mat** in the Morris folder.

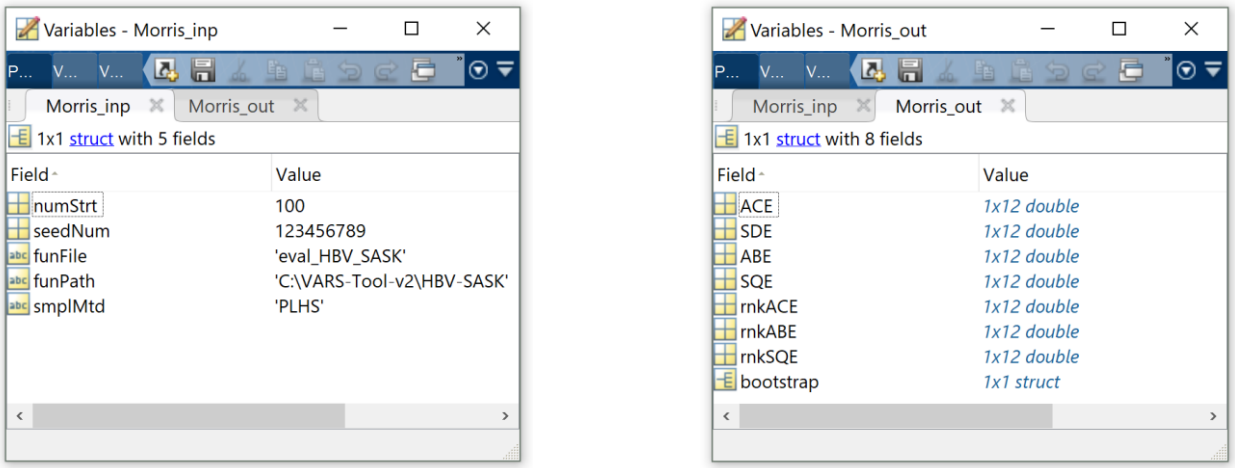


Figure 7.3. Morris_inp and Morris_out

8. RSA

8.1. RSA Run File

main_RSA.m

This is the main file that runs Regional Sensitivity Analysis (RSA) algorithm introduced by *Hornberger and Spear (1981)*. It calculates the GSA indices based on the classical Kolmogorov-Smirnov (K-S) distance measure (*Kolmogorov* 1933). The users set control parameters and specifications within this file, as shown in the figure below.

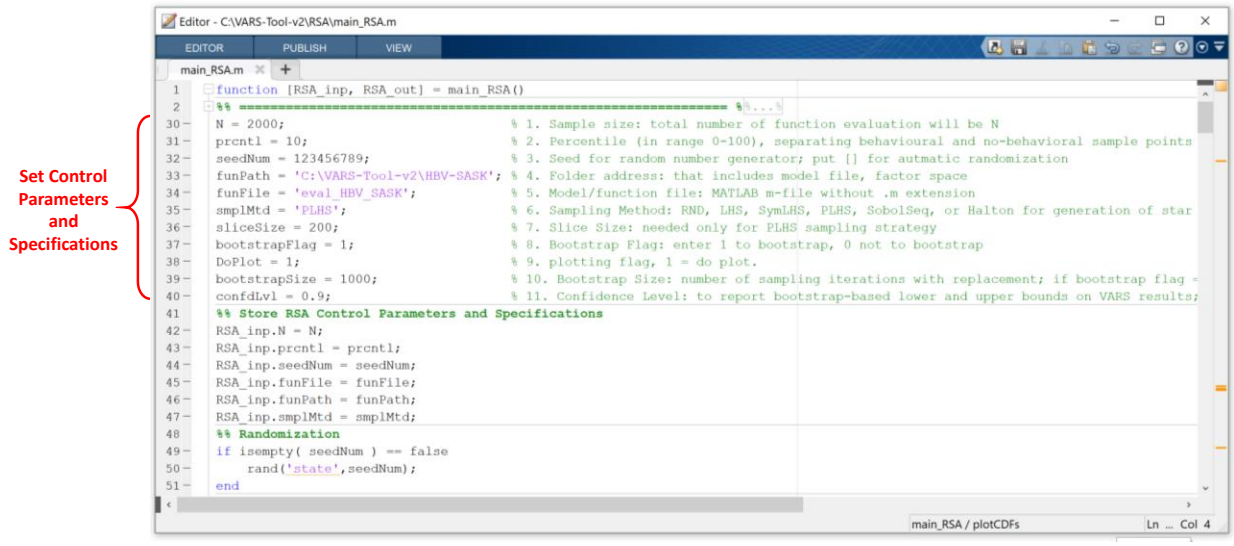


Figure 8.1. main_RSA.m

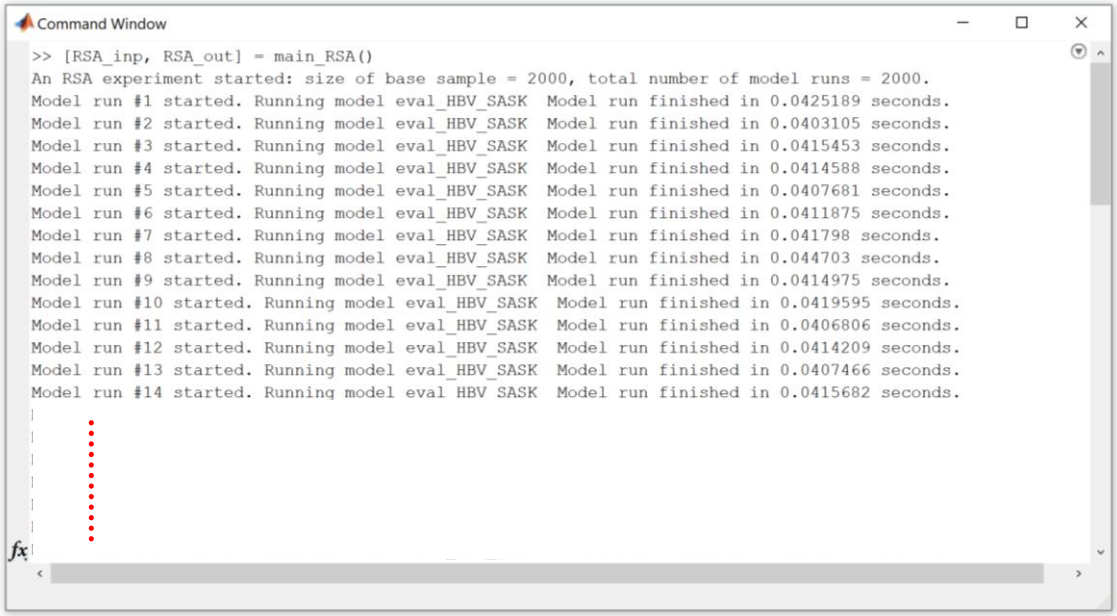


Figure 8.2. Screenshot of command window when running main_RSA.m

8.2. RSA Output Files

After running **main_RSA.m**, it returns two structure arrays: **RSA_inp** and **RSA_out**. The former contains the control parameters and specifications, while the latter includes the results of RSA run. See the figures below. These two arrays are stored in **Results_RSA.mat** in the RSA folder.

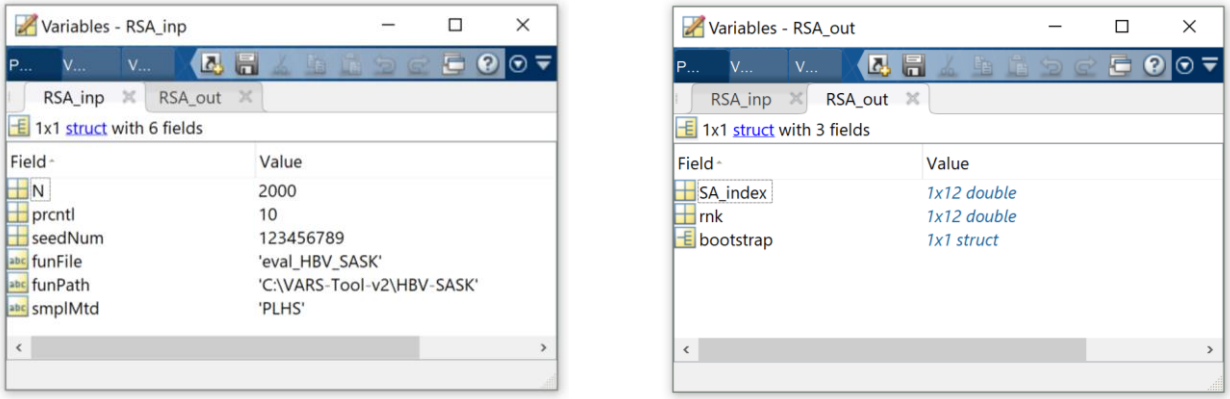


Figure 8.3. RSA_inp and RSA_out

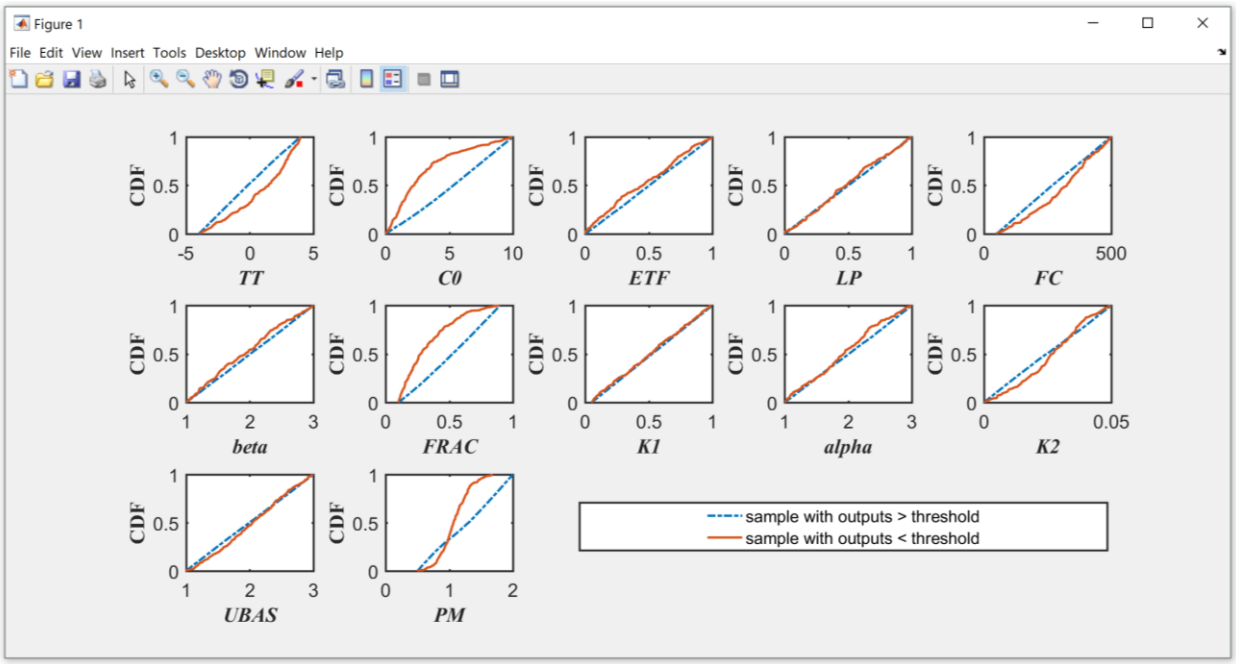


Figure 8.4. A sample plot generated by main_RSA.m for the HBV-SASK case study on the Oldman Basin. Root mean squared error was used as model response.

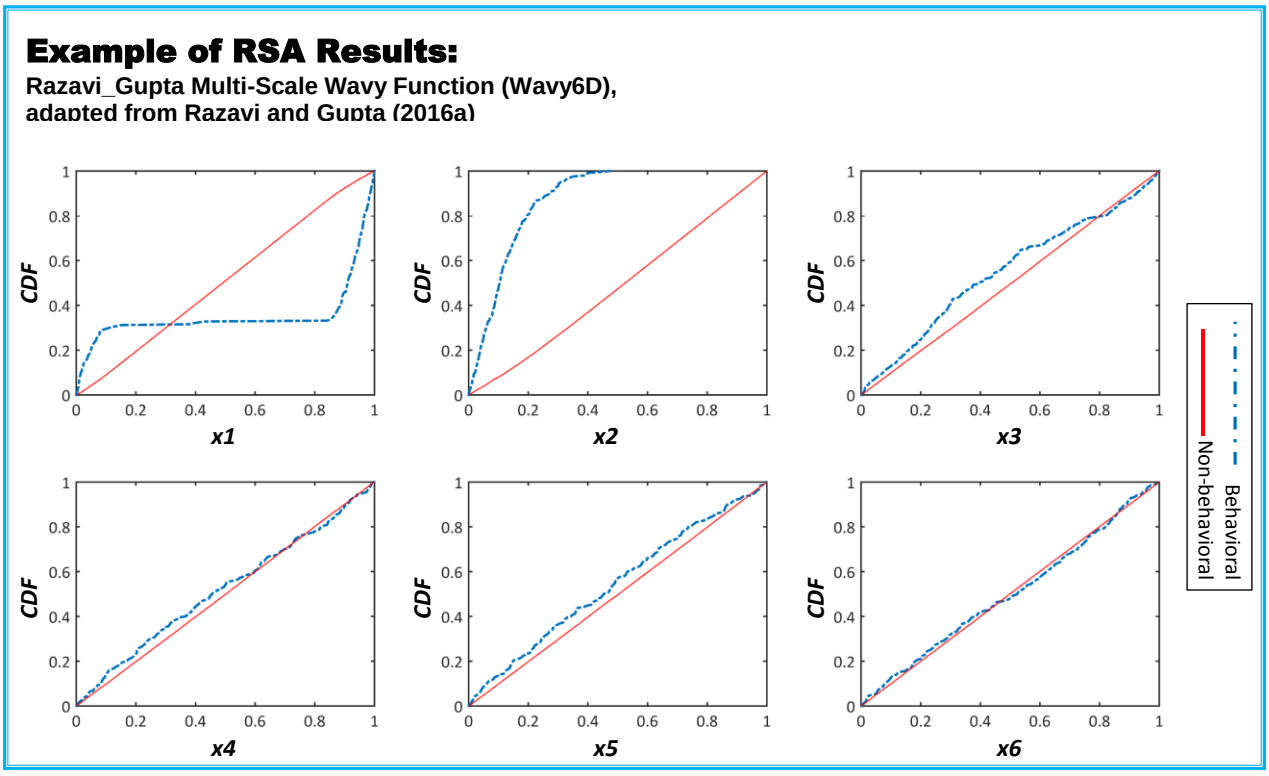


Figure 8.5. Example of RSA results for Razavi-Gupta Multi-Scale Wavy Function

9. Model Emulation: Handling Model Crashes

"Parameter-induced simulation crashes" is a commonly faced problem in numerical models, which happens mostly due to a violation of numerical stability conditions. Model crashes can impede accomplishment of sampling-based analyses such as global sensitivity analysis, as they require running numerical models for many configurations of parameter values, some of which are prone to model crashes. The common practice in handling this issue is discarding the failed designs and re-running sensitivity analysis on reduced parameter ranges. However, identifying regions of parameter space responsible for crashes is extremely difficult and requires analyzing the behavior of models across the parameter space.

To address this challenge, VARS-TOOL has been enabled by alternative strategies to handle model crashes. Basically, VARS-TOOL treats computer crashes as missing data and considers the model responses as an incomplete data matrix. Three efficient strategies, including median substitution, nearest neighbor, and response surface modelling were implemented in VARS-TOOL

9.1. Model Emulation Run File

main_crash_handling.m

After running a model, the user must replace the missing responses/outputs (simulation failures/crashes) before performing sensitivity analysis. This can be easily done using the **main_crash_handling.m** code.

Set Control Parameters and Specifications

```
1  function [imputed_output] = main_crash_handling ( )
2  %% Description: %...%
21
22 %% Control Parameters and Specifications
23 funPath = 'C:\VARS-Tool-v2\Model Emulation'; % 1. Folder address: that includes model output, design points
24 outputFile = 'model_output.smp';            % 2. Model responses file (.smp format)
25 sample_setFile = 'sample.smp';              % 3. Design points/sample set (.smp format)
26 strategy = 'RBF';                          % 4. Crash handling strategies: (default = 'NN')
27 %                                               'median' = Median substitution
28 %                                               'NN' = Single Nearest neighbor substitution
29 %                                               'RBF' = Response surface modeling-based substitution (here we used Radial Basi
30 %% Read inputs
31 currDir = pwd;
32 cd(funPath);
33 model_output = importdata( outputFile );
34 design_points = importdata( sample_setFile);
35 cd (currDir);
36
37 %% Main code
38 c_indx = find(isnan(model_output)); % identify the crashed runs
39 if isempty(c_indx)
40     error('There is no failed simulation in model outouts')
41 end
42 switch strategy
43     case 'median'
44         % Strategy No.1
```

Figure 9.1. main_crash_handling.m

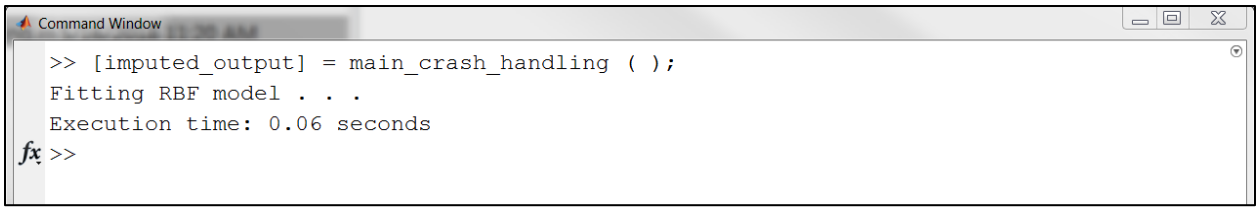


Figure 9.2. Screenshots of command window when running main_crash_handling.m

9.2. Model Emulation Output File

After running **main_crash_handling.m**, it returns an output vector called: **imputed_output** which contains the model responses where the missing are filled in using the pre-specified strategy. See the figure below. This array is stored in **filled_output.mat** in the Model Emulation folder.

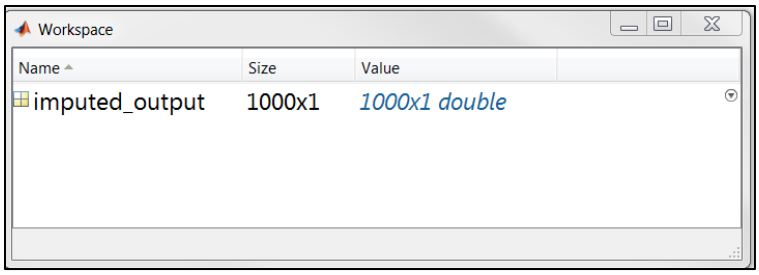


Figure 9.3. imputed_output

9.3. Example with HBV-SASK Model

This example shows how to use the Model Emulation package to handle computer crashes on the HBV-SASK hydrologic model. Note that, in this example, we assume that 10% of the model outputs are missing completely at random (i.e. 100 out of 1000 runs).

Step 1. Run the HBV-SASK model for all the parameters.

Step 2. Replace the failed runs by 'NaN' in the model output file.

Step 3. Write the model output and parameter set (parameters should be scaled to [0,1] for better performance) in model_output.smp and samples.smp, respectively.

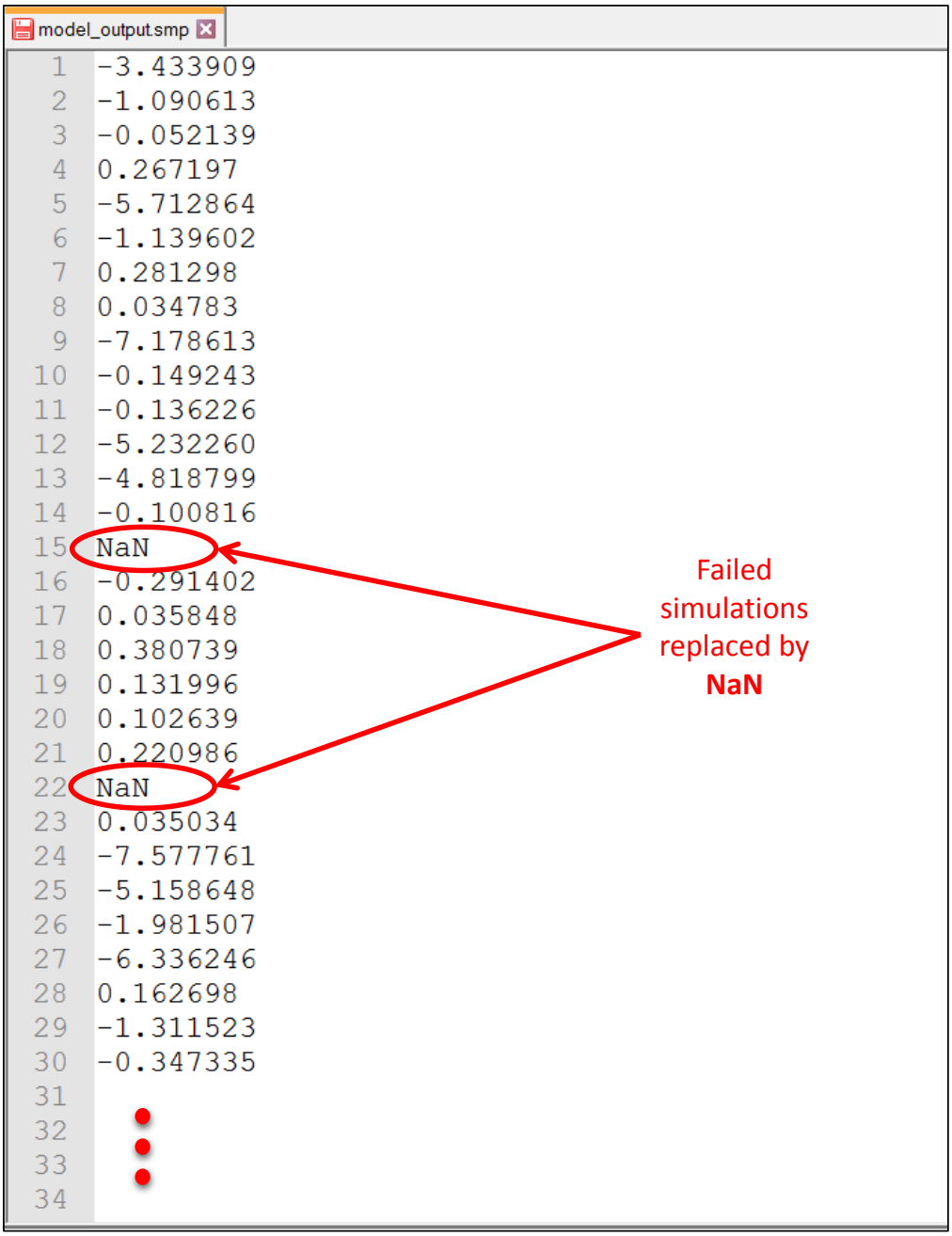


Figure 9.4. model_output.smp.

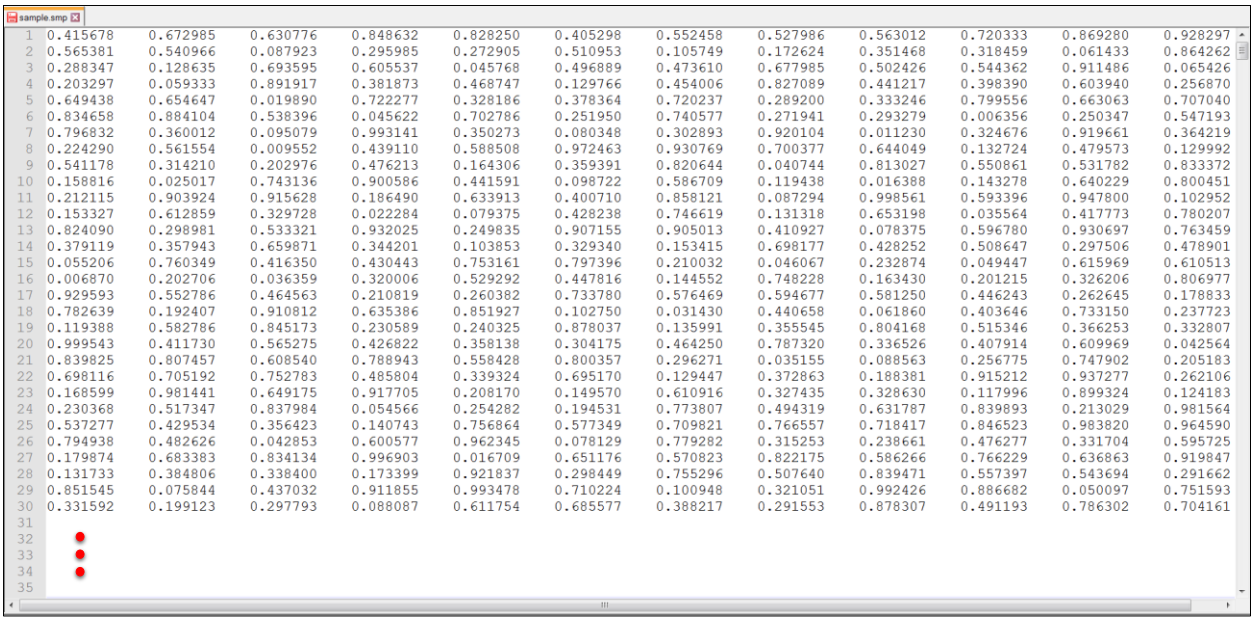


Figure 9.5. samples.smp.

Step 4. Specify the crash handling strategy in **main_crash_handling.m**. There are three strategies in VARS Tool, including 'median', 'NN', and 'RBF'. The default strategy is 'RBF' which uses radial basis functions as an interpolant to fill in missing values in model output.

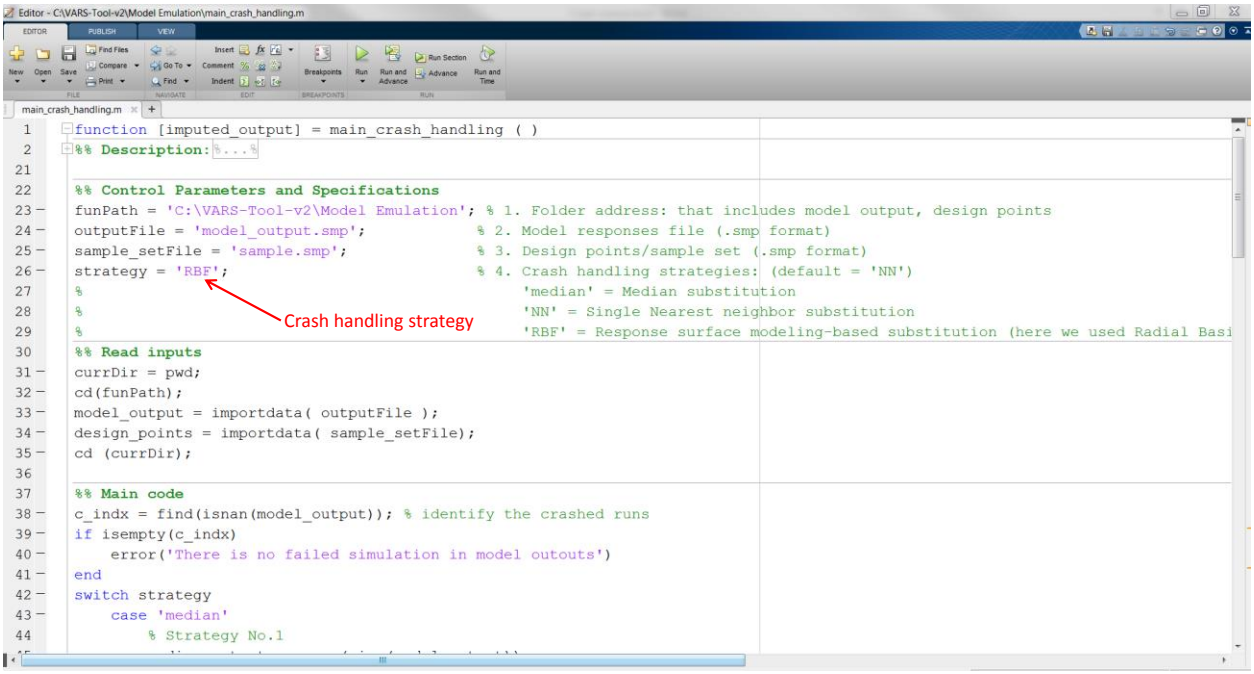


Figure 9.6. Selecting crash handling strategy.

Step 5. Run the **main_crash_handling.m** code. It saves the results in both **.mat** and **.smp** formats. After replacing the missing values in model output, user can apply any sensitivity analysis algorithm to determine the influence of HBV-SASK parameters on the model output.

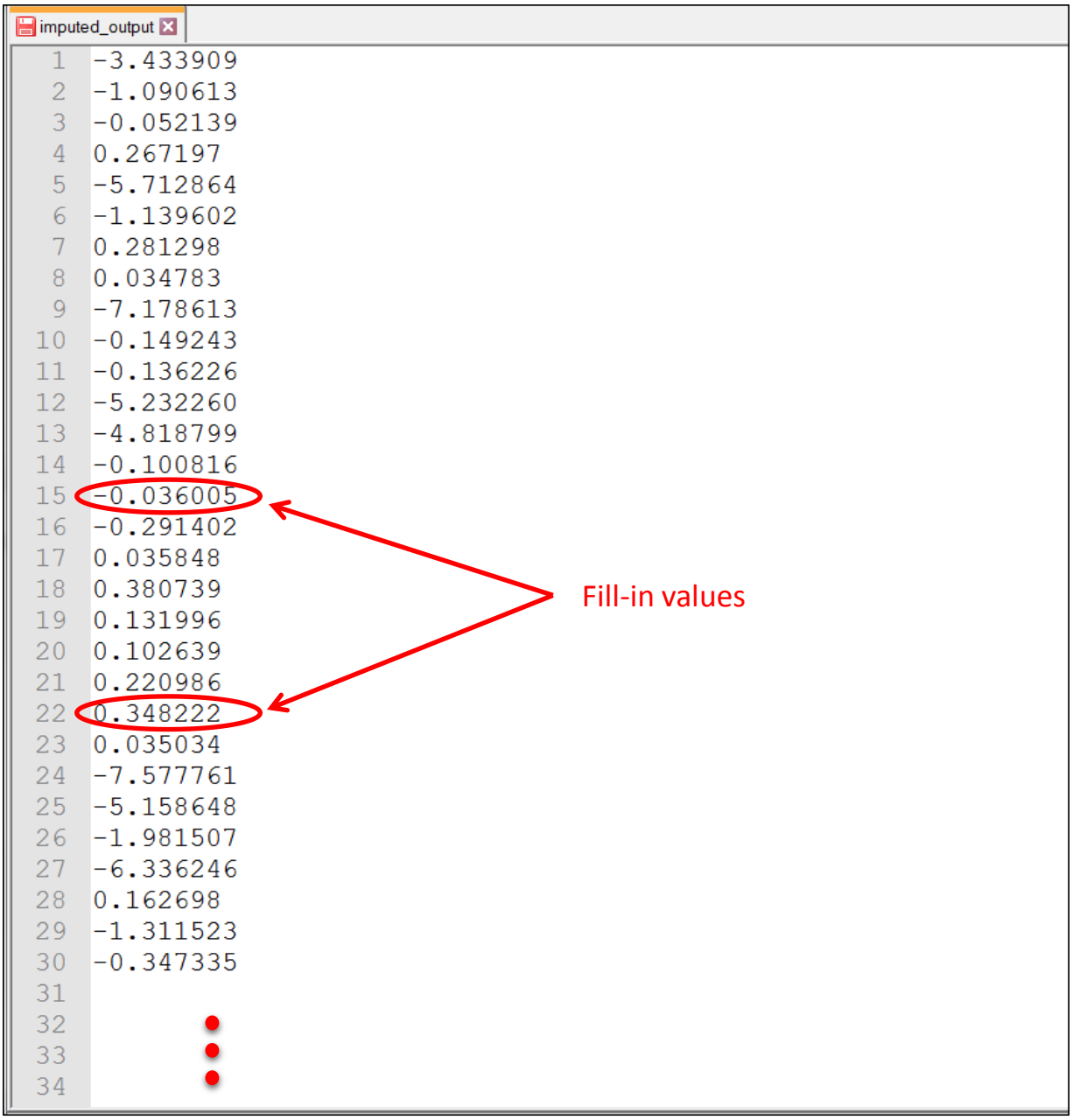


Figure 9.7. imputed_output.smp.

.

10. Example 1: VARS Off-line Mode with a Single-Output Model

This example shows how to run the STAR-VARS algorithm on the HBV-SASK hydrologic model, when of interest is to characterize the sensitivity of a model performance metric to the model parameters. The model performance metric is a scalar.

Step 1.
1a. In **VARS_inp.txt** (Figure 4.1), set line 18 to 1 (i.e., to set the algorithm to off-line mode), line 19 to 1 (i.e., to set the algorithm to stage 1 of the off-line mode), and line 22 to 1 (i.e., to set the algorithm to run for single-output models).

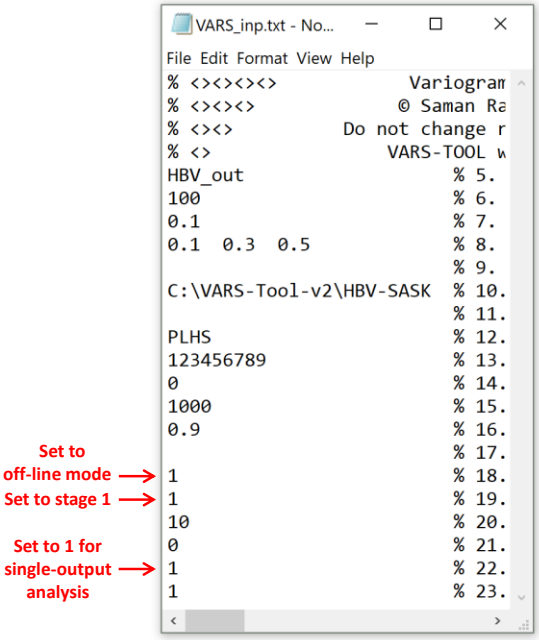


Figure 10.1. VARS_inp.txt in Example 1

1b. Run **main_VARS.m** and it generates all sample points (star centers through PLHS or another sampling strategy of interest, and star points through STAR) and stores them in **STAR_out.smp**.

As shown in the figure below, the arrangement of the sample points in the file is as follows: star center 1, star points of star 1 along dimension 1, star points of star 1 along dimension 2, …, star points of star 1 along dimension *D*, star center 2, star points of star 2 along dimension 1, star points of star 2 along dimension 2, …, star points of star 2 along dimension *D*, ……, and star center *m*, star points of star *m* along dimension 1, star points of star *m* along dimension 2,…, star points of star *m* along dimension *D*.

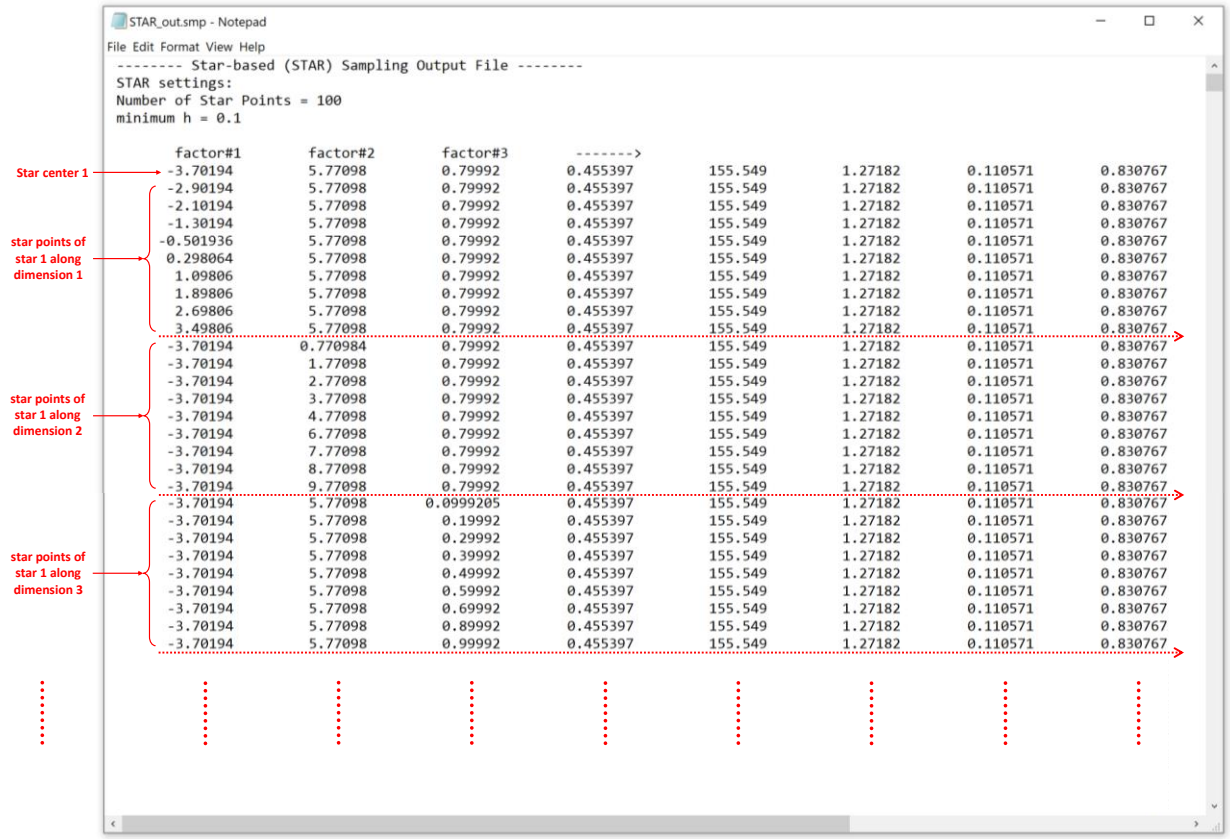


Figure 10.2. STAR_out.smp in Example 1

Step 2.

Run **extrnl_funEval_HBV.m** in the "VARS" folder. This function reads in the sample points (sets of parameter values) from **STAR_out.smp** (figure above), run the HBV-SASK model for all the parameters, and writes the results in **STAR_in.smp**.

2a. Make sure in **extrnl_funEval_HBV.m** that runType is set to 'single output'. This function will run **eval_HBV_SASK.m**, located in the "HBV-SASK" folder.

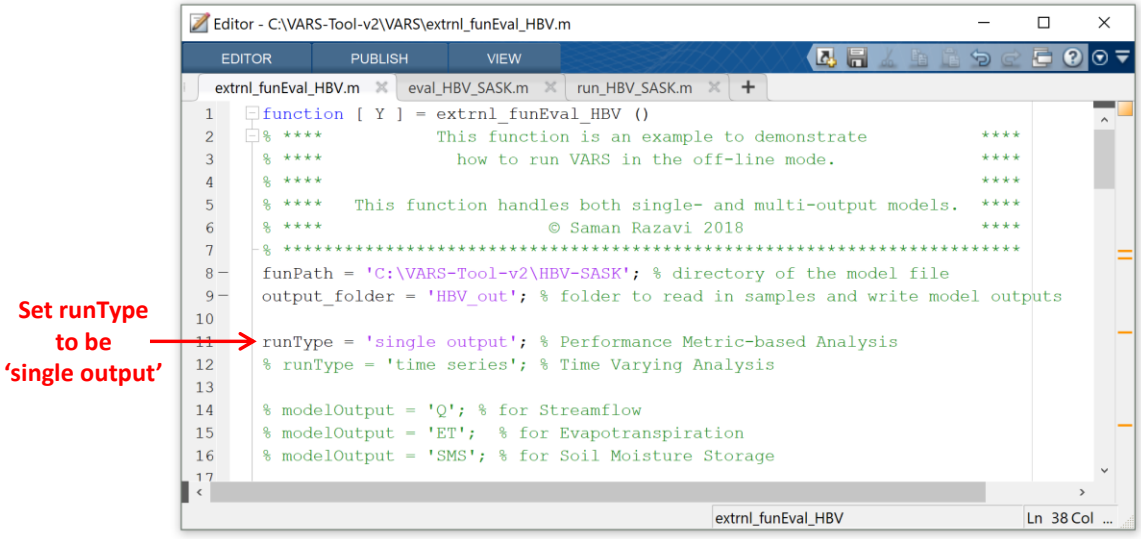


Figure 10.3. extrnl_funEval_HBV.m in Example 1

2b. In **eval_HBV_SASK.m**, located in the "HBV-SASK" folder:

- choose the case study of interest; VARS-TOOL includes two readily-available case studies, "Oldman Basin" and "Banff Basin".
- Choose the spin-up period: for no spin-up, set this length to zero.
- Choose the model performance (goodness-of-fit) metric: a range of options are available that provide the opportunity to carry out multi-criteria sensitivity analysis.

This function will run **run_HBV_SASK.m**, located in the same folder.

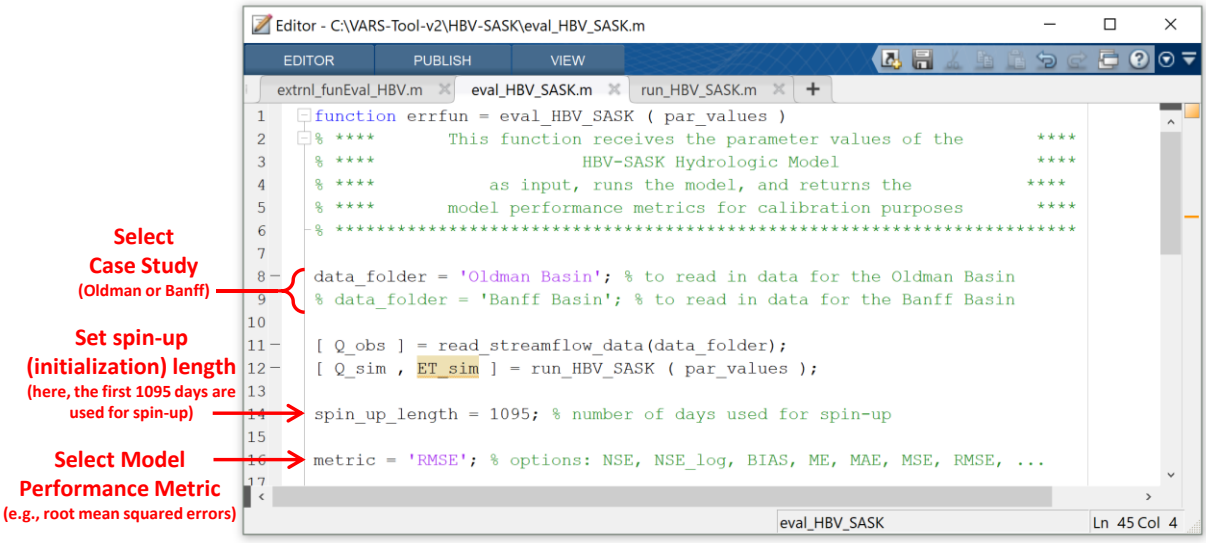


Figure 10.4. eval_HBV_SASK.m in Example 1

2c. In **run_HBV_SASK.m**, located in HBV-SASK folder, make sure that the case study is the same as the one selected in **eval_HBV_SASK.m**.

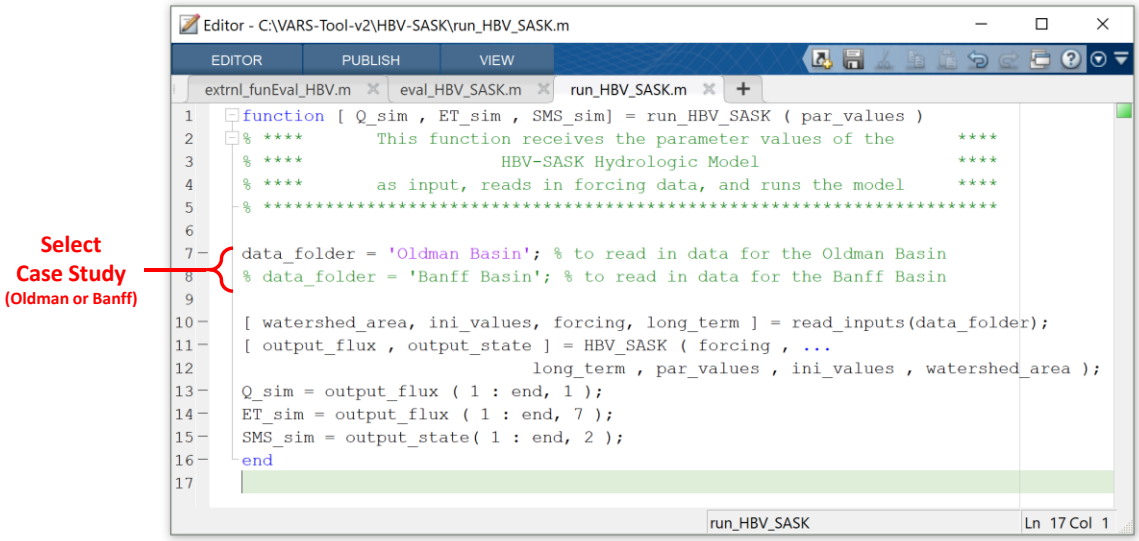


Figure 10.5. run_HBV_SASK.m in Example 1

2d. File **STAR_in.smp** will be generated automatically in **C:\VARS-Tool-v2\VARS\HBV_out**. This file contains the model response values for all the sample points, arranged in the exact same order of points in **STAR_out.smp**.

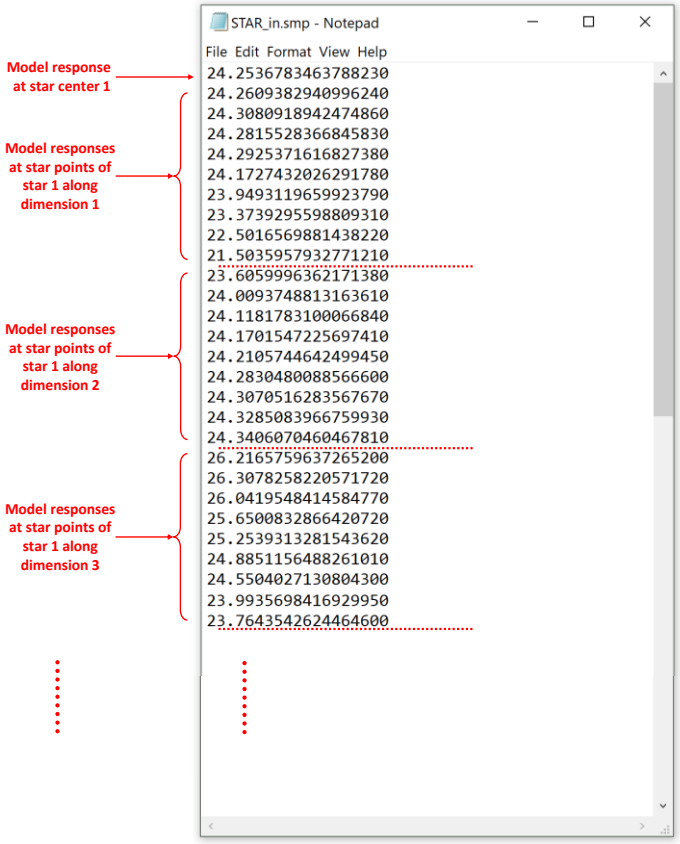


Figure 10.6. STAR_in.smp in Example 1

Step 3.
3a. In **VARS_inp.txt**, set line 19 to 2 (i.e., to set the algorithm to stage 2 of the off-line mode).

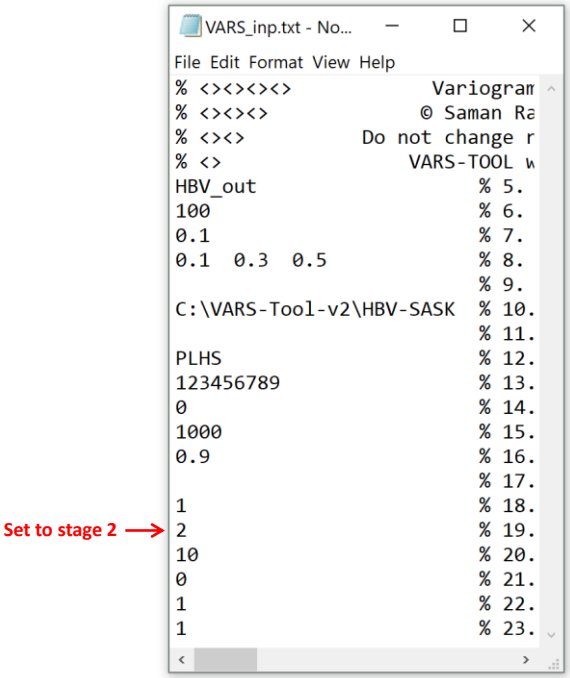


Figure 10.7. VARS_inp.txt in Example 1 – Stage 2

3b. Run **main_VARS.m** to carry out GSA. The results will be stored in **VARS_out_10.txt** through **VARS_out_100.txt**.

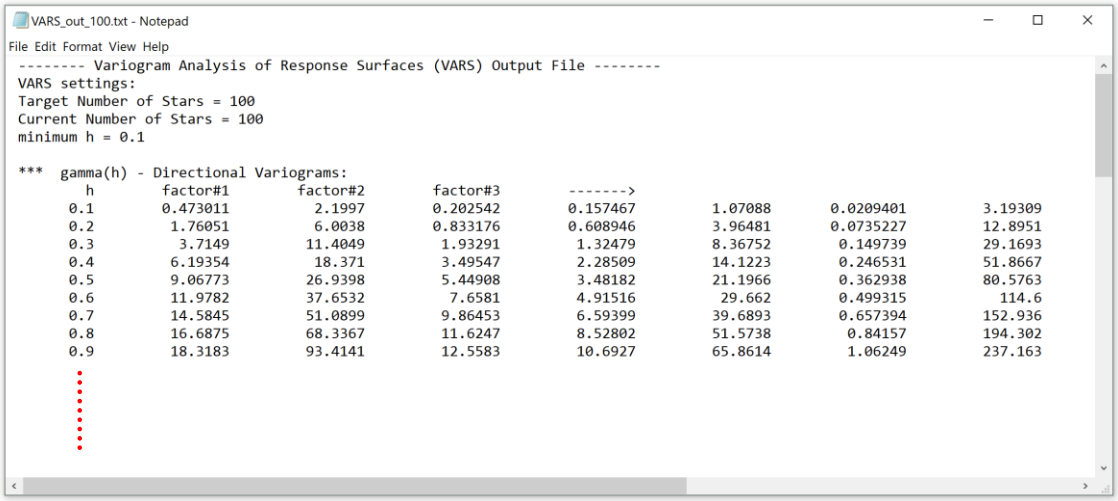


Figure 10.8. VARS_out_100.txt of Example 1

11. Example 2: Models with time series output via the Generalized Global Sensitivity Matrix (GGSM) approach

This example shows how to run the STAR-VARS algorithm on the HBV-SASK hydrologic model using the GGSM approach, to assess how the different model parameters influence the model outputs over time.

Step 1.
1a. In **VARS_inp.txt**, set line 18 to 1 (i.e., to set the algorithm to off-line mode), line 19 to 1 (i.e., to set the algorithm to stage 1 of the off-line mode), and line 22 to the length of model output time series. Here, for the Oldman Basin case study, this length is set to 9863 (days). The user may also set line 23 to 0, to save run time; otherwise, **main_VARS.m** will generate report files, separately for every time step t from 1 to T.

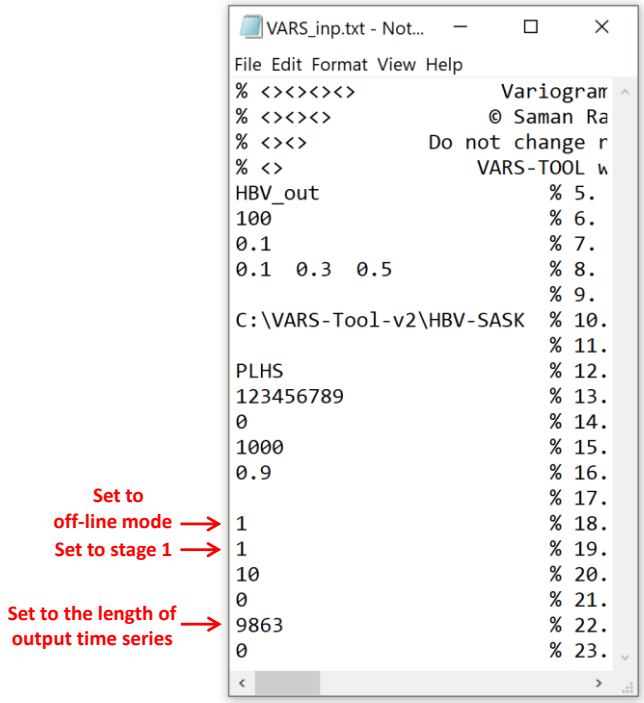


Figure 11.1. VARS_inp.txt in Example 2: Stage 1

1b. The same as Step 1b in Example 1.

Step 2.
Run **extrnl_funEval_HBV.m** in the "VARS" folder. This function reads in the sample points (sets of parameter values) from **STAR_out.smp**, run the HBV-SASK model for all the parameters, and writes the results in **STAR_in.smp**.
2a. Make sure in **extrnl_funEval_HBV.m** that **runType** is set to **'time series'**, that the model output of interest is chosen (e.g., soil moisture storage), and that the time period of interest in

chosen, as shown in the figure below. This function will directly run **run_HBV_SASK.m**, located in the "HBV-SASK" folder.

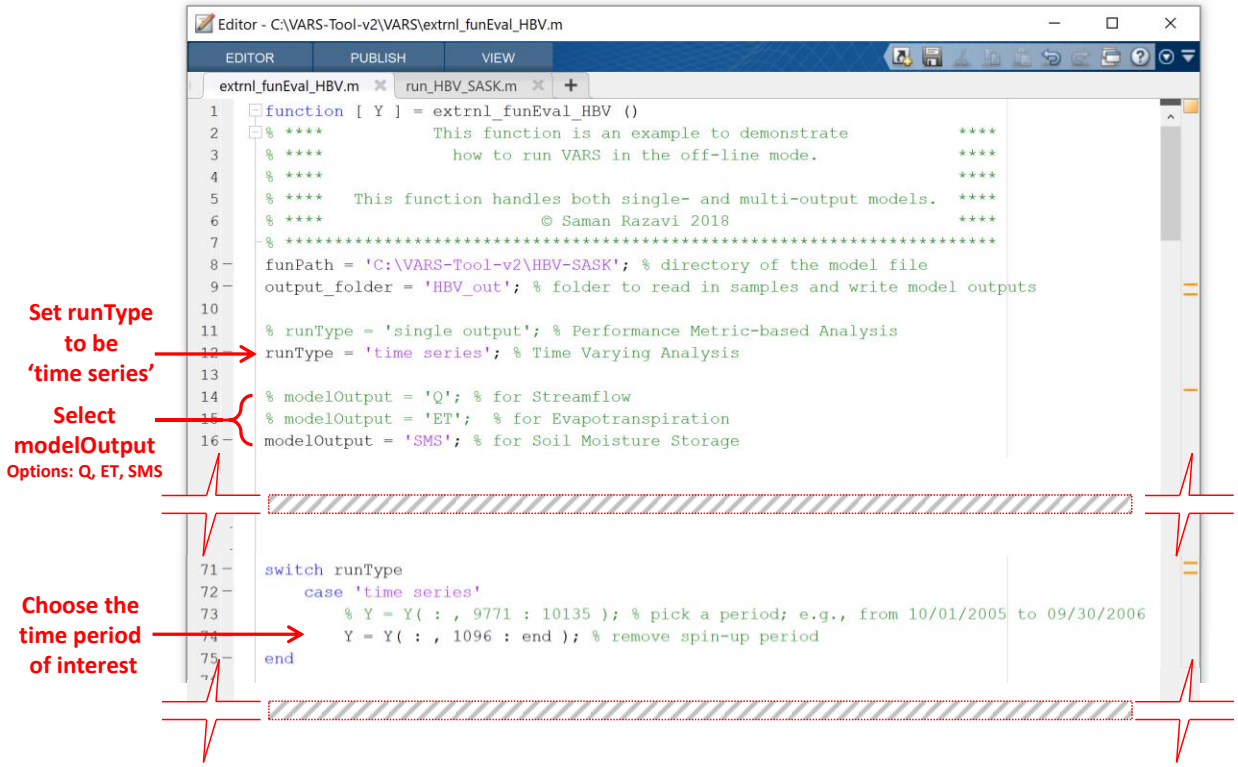


Figure 11.2. extrnl_funEval_HBV.m in Example 2

2b. None.

2c. In **run_HBV_SASK.m**, located in the "HBV-SASK" folder, choose the case study of interest. See Figure 10.5.

2d. A series of STAR_in files will be generated for the time period of analysis, for t=1 to T, *i.e.*, **STAR_in_t=1.smp** to **STAR_in_t=9863.smp**. These files will be stored in **C:\VARS-Tool-v2\VARS\HBV_out**. Each file contains the model response values *at the respective time step* for all the sample points, arranged in the exact same order of points in **STAR_out.smp**.

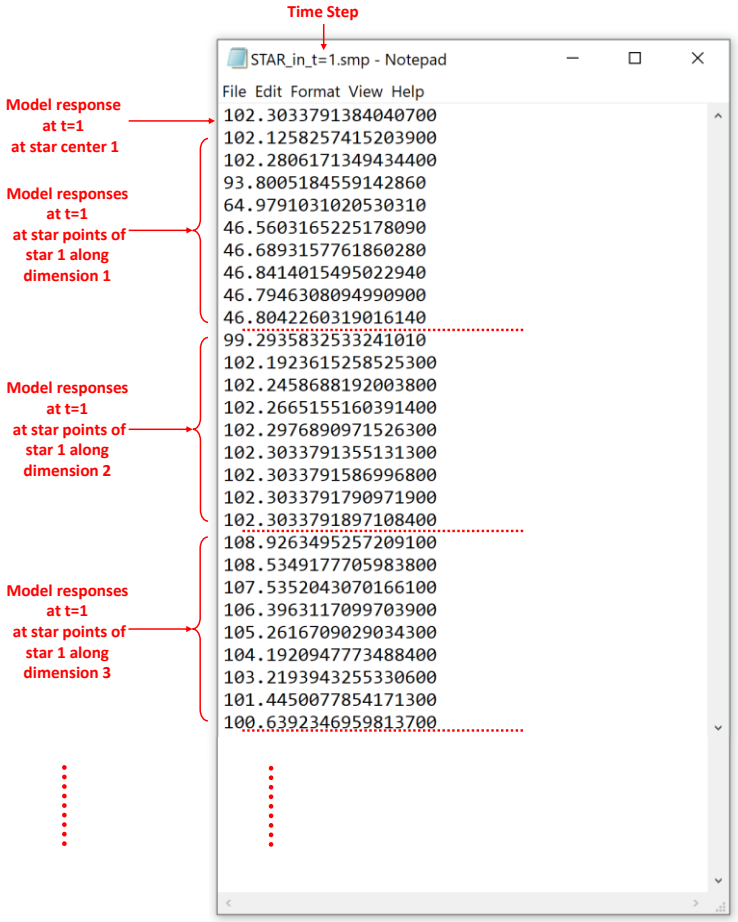


Figure 11.3. STAR_in_t=1.smp in Example 2

Step 3.

3a. In **VARS_inp.txt**, set line 19 to 2 (i.e., to set the algorithm to stage 2 of the off-line mode).

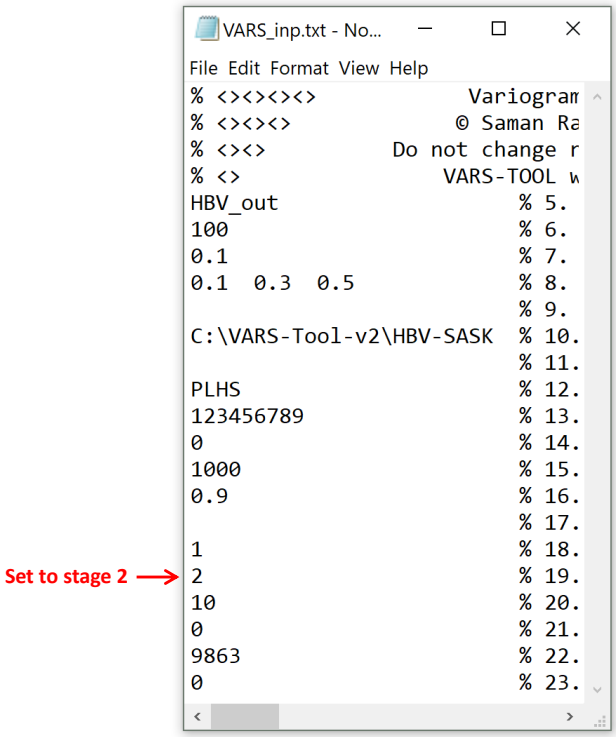


Figure 11.4. VARS_inp.txt in Example 2: Stage 2

3b. Run **main_VARS.m** and it will carry out GSA based on the GGSM approach. Upon completion of the run, all the time-varying and time-aggregate GSA indices will be returned in variable **VARS_out.RG** in MATLAB environment. Also, a summary of time-aggregate indices will be stored in files **VARS_RG_out_10.txt** through **VARS_RG_out_100.txt**.

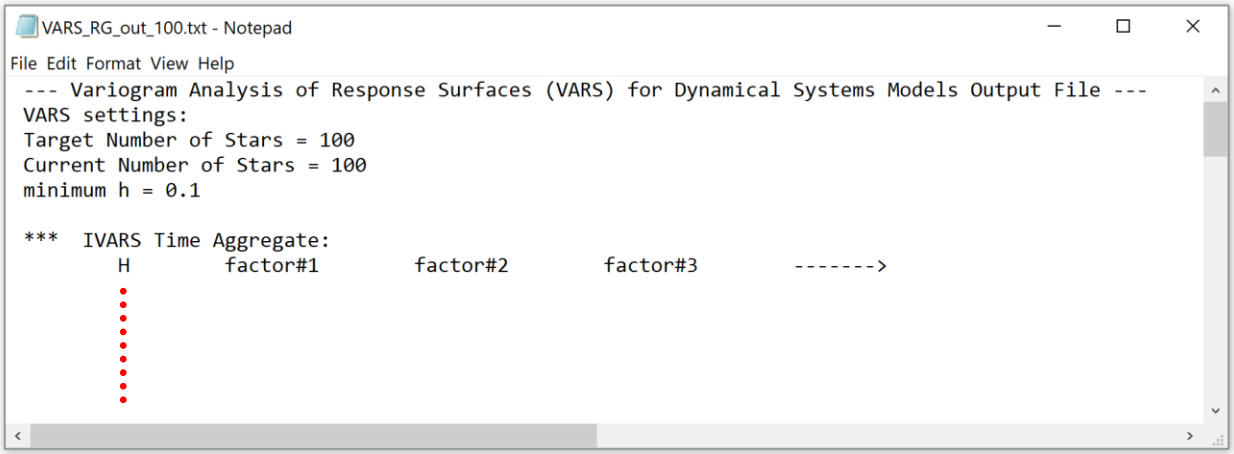


Figure 11.5. VARS_RG_out_100.txt in Example 2

Appendix A: The HBV-SASK Hydrologic Model

A1. Model Description

HBV-SASK is rainfall-runoff model based on an interpretation of Hydrologiska Byråns Vattenbalansavdelning model (Lindström et al. 1997). This model was coded at the University of Saskatchewan for educational purposes (Figure A 1). The code is modular, with each module simulating a different hydrologic process, thereby allowing the user to easily investigate the different model flux and state variables. Table A1 lists the model parameters, their description, and preferred ranges. The first 11 parameters are hydrologic process parameters, while parameter 12 is designed to account for bias (uncertainty) in precipitation. The specification of parameters 11 and 12 is optional, and if not supplied by the user, the model run with the default values.

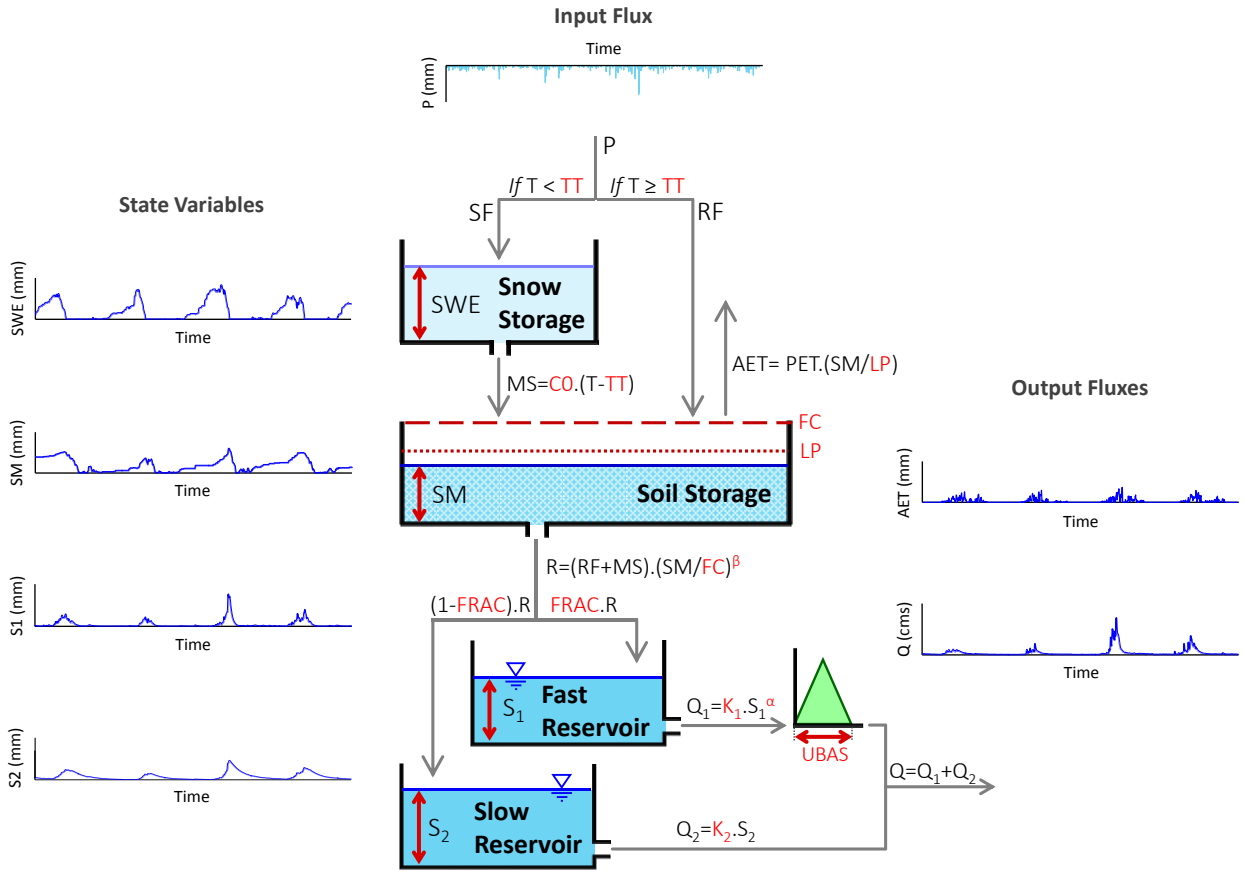


Figure A 1. The architecture of the HBV-SASK hydrologic model. The model works on a daily time step and reports all the flux and state variables. P: precipitation; T: temperature; SF: snowfall; RF: rainfall; SWE: snow water equivalent; MS: melted snow; AET: actual evapotranspiration; PET: potential evapotranspiration; SM: soil moisture; R: soil release; S1 and S2: storage in fast and slow reservoirs; Q1 and Q2: discharge from fast and slow reservoirs; Q: total watershed discharge.

While all the process equations are included in the above figure, PET (daily potential evapotranspiration) is calculated using the equation below based on average monthly temperature ($T_{monthly}$) and average monthly potential evapotranspiration ($PET_{monthly}$), supplied as input to the model.

$$PET = (1 + ETF * (T - T_{monthly})).* PET_{monthly} \quad \text{Equation (A1)}$$

Table A 1. The parameters of the HBV-SASK model

Number	*Parameter Name*	*Lower Bound*	*Upper Bound*	*Description*
1	TT	-4	4	Air temperature threshold in °C for melting/freezing and separating rain and snow
2	C0	0	10	Base melt factor, in mm/°C per day
3	ETF	0	1	Temperature anomaly correction in 1/°C of potential evapotranspiration
4	LP	0	1	Limit for PET as a multiplier to FC, i.e., soil moisture below which evaporation becomes supply limited
5	FC	50	500	Field capacity of soil, in mm. The maximum amount of water that the soil can retain
6	β (beta)	1	3	Shape parameter (exponent) for soil release equation (unitless)
7	FRAC	0.1	0.9	Fraction of soil release entering fast reservoir (unitless)
8	K1	0.05	1	Fast reservoir coefficient, which determines what proportion of the storage is released per day (unitless)
9	α (alpha)	1	3	Shape parameter (exponent) for fast reservoir equation (unitless)
10	K2	0	0.05	Slow reservoir coefficient which determines what proportion of the storage is released per day (unitless)
11	UBAS	1	3	Base of unit hydrograph for watershed routing in day; default is 1 for small watersheds
12	PM	0.5	2	Precipitation multiplier to address uncertainty in precipitation (unitless); default is 1.

References:

Lindström, G., Johansson, B., Persson, M., Gardelin, M. and Bergström, S. (1997) Development and test of the distributed HBV-96 hydrological model. Journal of hydrology 201(1-4), 272-288.

A2. How To Run HBV-SASK?

All the MATLAB files needed to run HBV-SASK are included in a folder named "HBV-SASK". The user may need to use/edit the following files to run the model:

run_HBV_SASK.m

This is a main MATLAB .m file that receives parameter values, reads in forcing and watershed data, calls the HBV-SASK engine (**HBV_SASK.m** explained below), and returns the simulated streamflow at the outlet and other state and flux variables. The user needs to set the data folder for the case study of interest at the beginning of this file.

HBV_SASK.m

This MATLAB .m file is the HBV-SASK engine that includes all the process equations. This file is called by **run_HBV_SASK.m** and may not be run directly by the user.

eval_HBV_SASK.m

This MATLAB .m file is to assess the performance of HBV-SASK against observed streamflow using a range of error metrics. The user needs to set the data folder (case study) at the beginning of this file and choose the error metric of interest (e.g., MSE for Mean Squared Errors).

factorSpace.txt

This text file incudes the parameter specifications and their ranges. This is only needed when the user needs calibrate the model parameters and conduct sensitivity and uncertainty analysis. If the last two parameters are to be set at default values, the users should remove their respective lines in this file.

calibrate_HBV_SASK.m

This file is designed to calibrate the parameters of HBV-SASK via optimization.

A3. Case Studies

HBV-SASK in VARS-TOOL comes with "ready-to-run" case studies on two watersheds with different properties, Bow and Oldman Basins. These rivers are located in the Rocky Mountains in Alberta, Canada, and flow into the Saskatchewan River Basin (Figure A 2). Banff and Oldman Basins with watershed areas of 2178.53 and 1434.73 km^2 have average annual precipitation (rainfall + snowfall) of 795 and 611 mm. The average streamflow of Bow and Oldman River basins are estimated to be 38.6 m^3/s and 11.7 m^3/s at gauges 05BB001 and 05AA023, respectively. The historical records included span periods 1950-2011 and 1979-2008 for the Bow and Oldman Basins. Figure A *3* shows the time series of these historical records. Based on these periods, the runoff ratios of Bow and Oldman Basins are estimated to be 0.7 and 0.42, respectively. The data required to run HBV-SASK for these two basins are included in the folders named "Banff Basin" and "Oldman Basin" within the folder "HBV-SASK".

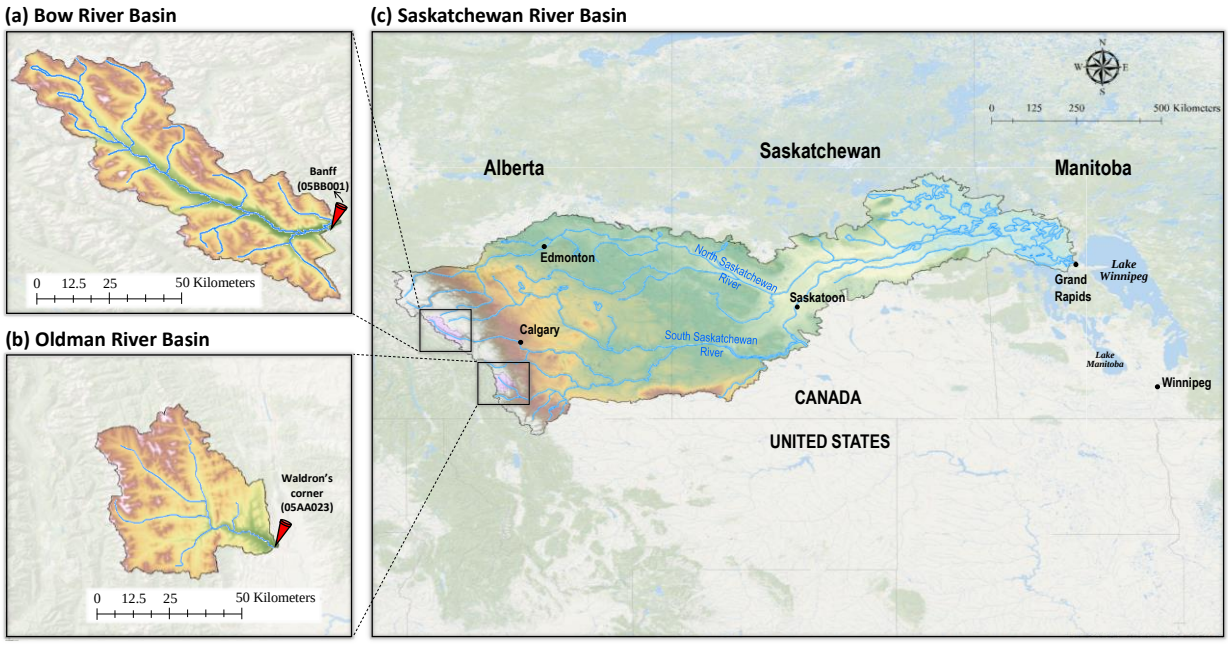


Figure A 2. Map of the Banff and Oldman Basins in Alberta, Canada

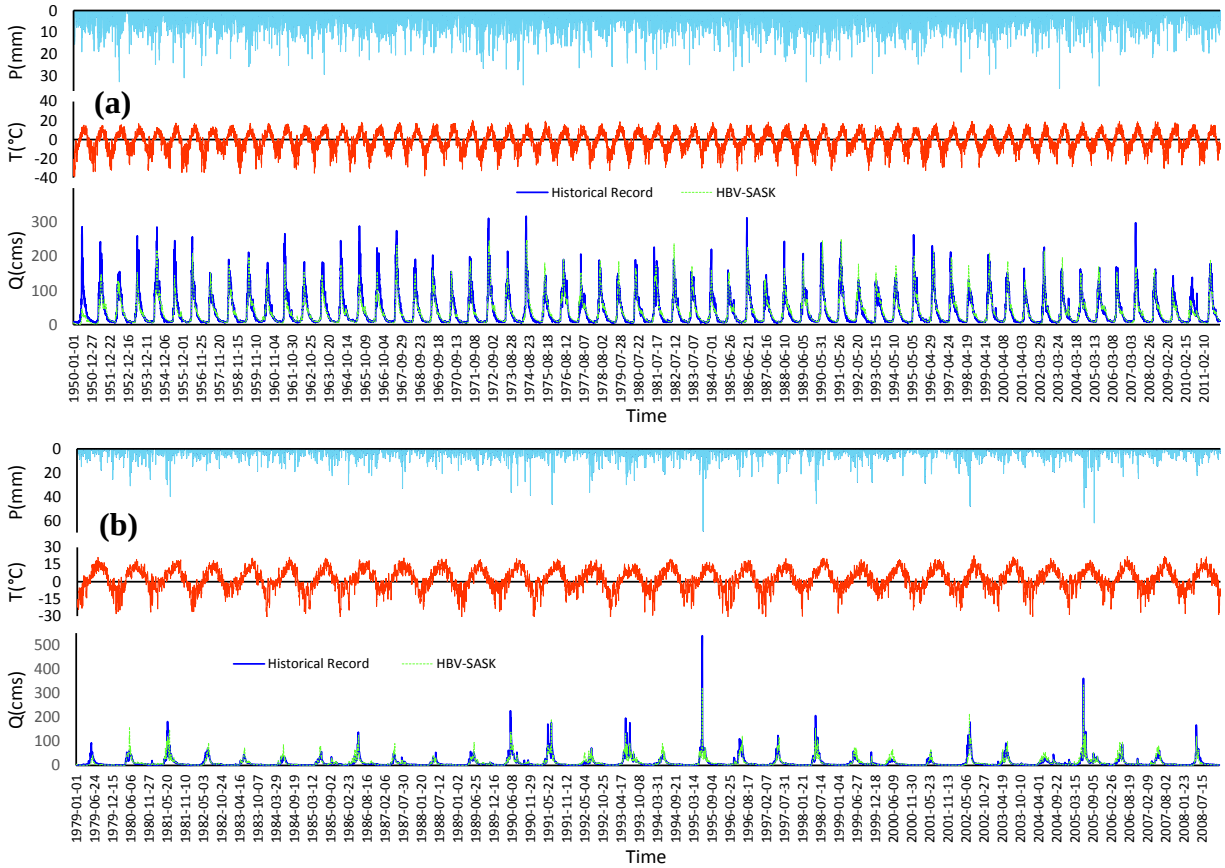


Figure A 3. Time series of precipitation, temperature, and streamflow of Banff (a) and Oldman (b) Basins. The simulated streamflow by HBV-SASK is based on the parameter values reported in Table A 2.

A4. Model Performance

The performance of the HBV-SASK model in reproducing observed streamflows in the Banff and Oldman basins, using different combinations (thousands) of parameter values, is shown in Figure A 4. The plots in this figure show the cumulative distrbution function (CDF) of mean squared errrors (MSE) and its normalized version (Nash-Sutcliffe Efficicney, NSE) across the parameter space. Table A *2* shows examples of well-performing paramter values for the two basins and their associated error metrics. These parameter values generate simulated streamflows that were already shown in Figure A 3.

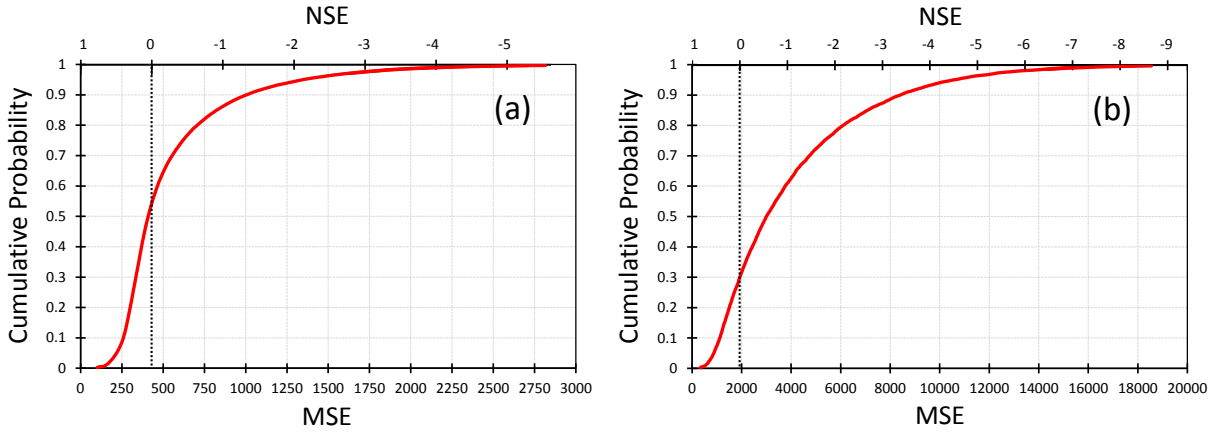


Figure A 4. Cumulative distribution functions of Mean Squared Errors (MSE) and respective Nash-Sutcliffe Efficiency (NSE) across the parameter space for (a) Banff and (b) Oldman Basins. For these plots, only the first 10 parameters of **Table A 1** were varied and parameters 11 and 12 were fixed at their default values.

Table A 2. Examples of well-performing paramter values for the Banff and Oldman Basins, based on NSE (Nash-Sutcliffe Efficiency), NSE-L (NSE on log-transformed flows), and %bias (volume bias)

Basin	TT	C0	ETF	LP	FC	beta	FRAC	K1	alpha	K2	UBAS	PM	NSE	NSE-L	%bias
Banff	1.878	2.810	0.001	1.000	435.398	1.001	0.624	0.051	1.000	0.006	1.020	1.025	0.824	0.903	0.287
Oldman	1.036	0.838	0.102	0.351	242.259	1.393	0.724	0.050	1.456	0.008	1.004	1.034	0.774	0.860	1.339

Appendix B: Example of external model runs

B1. Running models offline with OSTRICH

Here we illustrate how, in conjunction with the VARS-TOOL, thousands of external model simulations can be performed using OSTRICH toolkit (Matott, 2005), which is available for free at here. OSTRICH is a user-friendly model-independent optimization toolkit that also allows for Monte-Carlo type of simulations, where a large number of model simulations is required using sampled parameter sets. In addition to its regular (serial) version, OSTRICH supports Message Passing Interface (MPI)-based parallel processing on both Windows and Linux systems. In this example, we illustrate how to use OSTRICH in parallel mode to perform a large number of runs of a distributed physically-based model Modélisation Environmentale–Surface et Hydrologie (MESH, Pietroniro et al., 2007). In this case study, MESH (version 1.4.1037) is applied to Nottawasaga River Basin, Ontario, Canada as described in Haghnegahdar et al. (2017). Given the large computational burden, this example is specifically for use with OpenMPI parallel computing on high performance computing Linux systems, but can be modified to be used in Windows systems as well with the same structure. A brief description of the model and case study is presented at the end of this section.

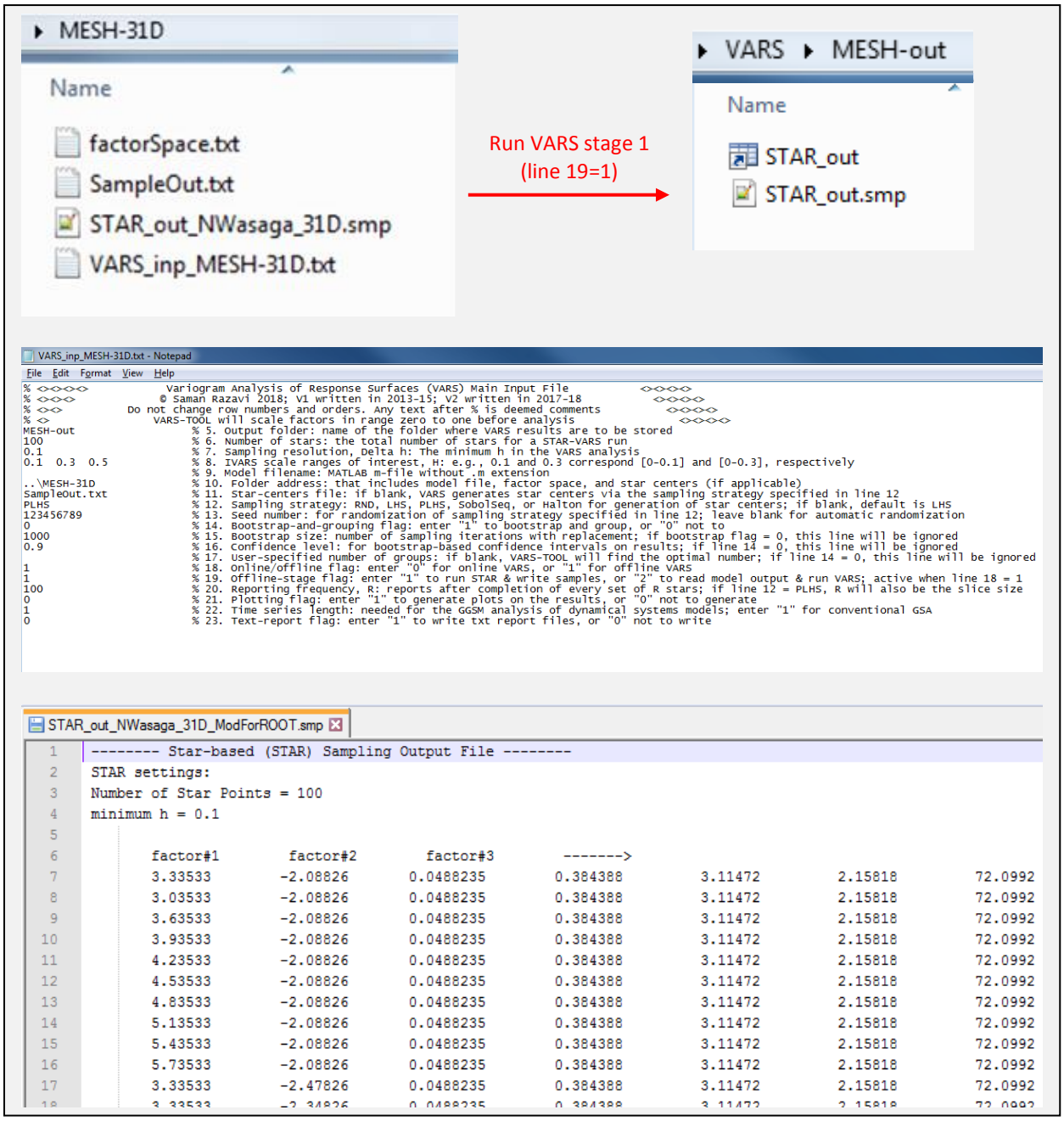
As described in section 5.5, three steps are involved: (1) Running VARS to generate sample parameter sets; (2) Run the model for all sampled parameter sets using OSTRCIH; and (3) Run VARS again by feeding back to it the model simulation results to generate sensitivity metrics. These steps are described below in details.

1) Running VARS to sample parameter sets

First we need to run VARS to generate sample parameter sets. The files required for this step are supplied in the MESH-31D folder seen below.

- Copy the VARS_inp_MESH-31D.txt in the main VARS folder and rename it to VARS_inp.txt;
- Run main_VARS.m; A folder, MESH-out, will be created and the associated STAR_out.smp (and also STAR_out.mat) will be generated in it.

As seen in the figure below, VARS is applied with 100 star centers (specified in SampleOut.txt file) and sampling resolution of 0.1, resulted in 28,000 parameter sets. As a verification, this generated STAR_out.smp file should be identical to the STAR_out_NWasaga_31D.smp (provided in the MESH-31D folder and seen in the snapshot below) when PLHS sampling is chosen because the same star-center file, SampleOut.txt, is specified on line 11 of the VARS_inp.txt file in this case. The VARS_inp.txt file can be changed as needed. The sample parameter sets generated here should then be fed into OSTRICH to run MESH simulations accordingly as described in the next step.



2) Running Model (MESH) offline using OSTRICH

This section describes how we can prepare OSTRICH and link it with our model (MESH here) to perform all model simulations and calculate required metrics associated with the parameter sets generated by VARS. If this step is already configured (i.e., model is linked with OSTRICH for this purpose), then in this step simply copy the parameter sets generated by VARS into the OstIx.txt file between the section identified by two key flags "`BeginInitParams`" and "`EndInitParams`", and then skip to step NO.3.

Otherwise, this step requires compiling OSTRCH and the model first, preparing OSTRCH files and folders, and executing OSTRICH to generate the response or metric needed by VARS to calculate sensitivity indices. This example is specifically designed for use with Open MPI parallel computing on Linux systems. The steps are described below.

2.1. Compiling MPI Ostrich and Model (MESH) in Linux

To run OSTRICH in parallel on a Linux machine or Linux-based cluster of machines, the Linux environment must provide MPI (Message Passing Interface) libraries. Any MPI implementation will suffice. Ensure the right modules are available in your system and loaded first.

To check the version of the Intel compiler available type:

```
module avail openmpi/intel
```

To load the desired module type, for example:

```
module load openmpi/intel/64/2.0.0
```

To compile MPI-OSTRICH in Linux using OpenMPI:

```
cd Path/to/OSTRCIH-source
make OMPI
```

NOTE: OSTRICH source code is normally included in the downloadable package. Otherwise, it can be obtained by contacting L. Shawn Matott via e-mail (lsmatott@buffalo.edu). Details on how to compile OSTRICH for each case can be found in its manual included in the downloaded package available at http://www.eng.buffalo.edu/~lsmatott/Ostrich/OstrichMain.html

Compiling MESH in Linux is the same as compiling other Fortran or C/C++ codes. It is advised to use the Intel compiler, as it is kept more up-to-date. Ensure the right modules are available and loaded first; "Intel Fortran" in this case. Note that when MESH is linked with MPI Ostrich, it is advised to compile it to run in serial – and not in MPI mode.

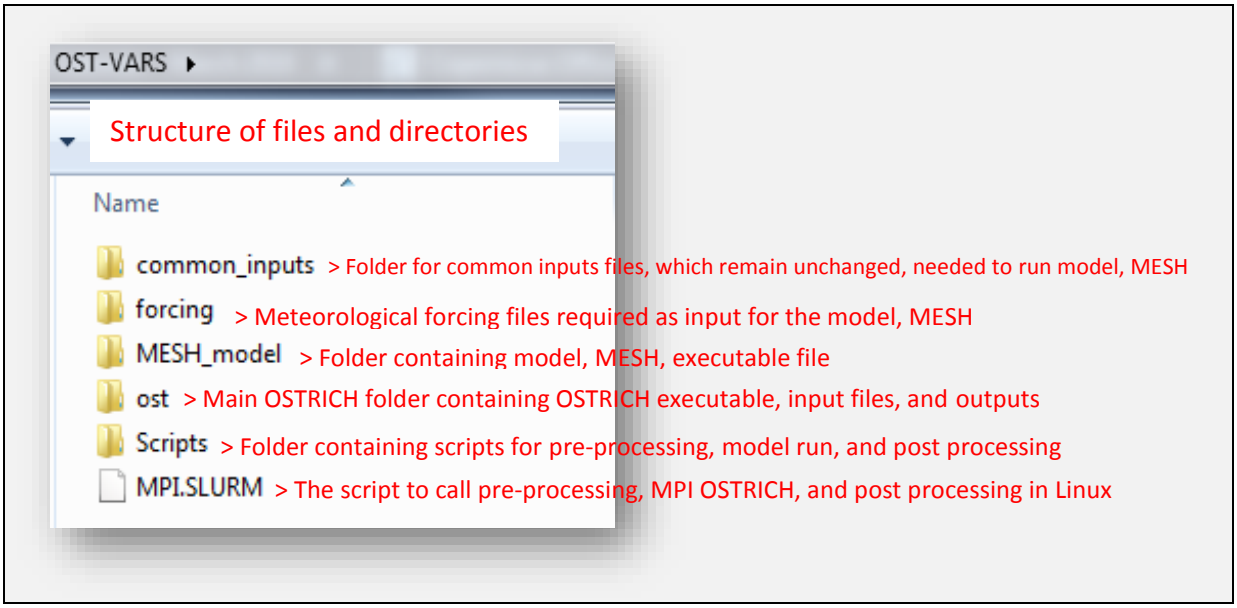
```
cd Path/to/MESH_Code
make -f makefile.ifort
make -f makefile.ifort clean
```

Note: Release notes and code for the latest version of MESH can be obtained here. Details of compiling MESH on Linux as well as on Windows are found on the MESH wiki page https://wiki.usask.ca/display/MESH/Compiling+Standalone+MESH

A compiled version of both MESH and OSTRICH is included in this package and might be used to run experiments on Linux systems. However, depending on the conditions, re-compiling might be required.

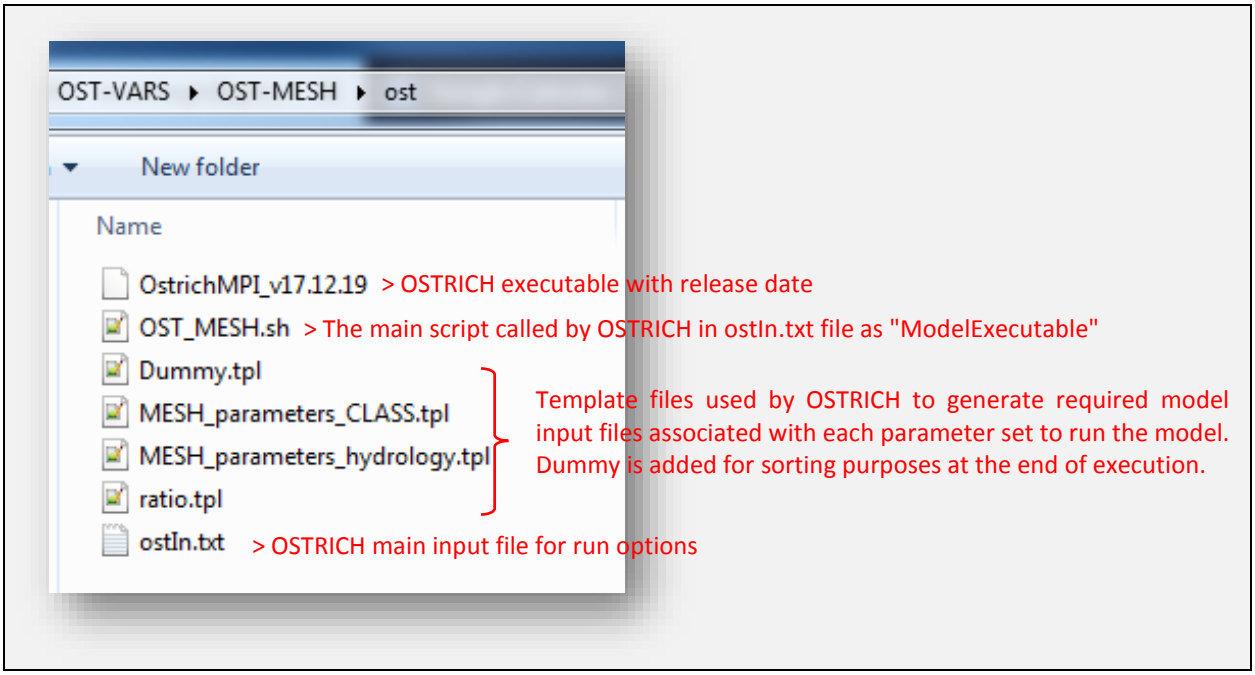
2.2. Arranging structure of files/directories for linking OSTRICH and the model

Users can customize the way they prefer to link a model to OSTRICH. In this example, we use the suggested structure shown below to link MESH with OSTRICH. Several required components are kept separate in this structure so they can be changed independently of one another when needed. They include some common files that remain unchanged throughout all the simulations; meteorological forcing files that also remain constant, MESH executable file; main OSTRICH folder; scripts for pre- and post-processing; and the main script to call MPI OSTRICH in Linux. Some instruction are provided inside the files of each folder.



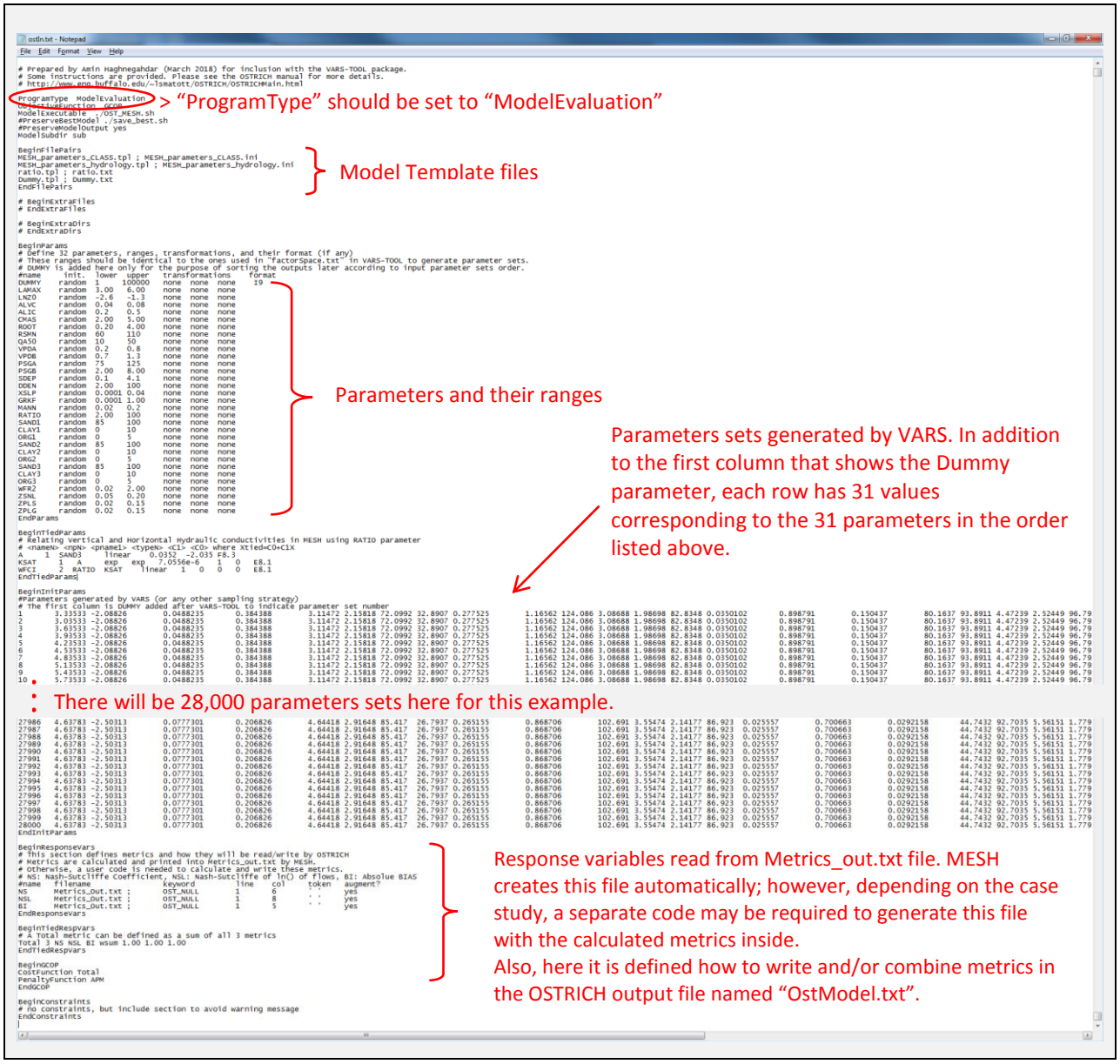
2.3. Preparing OSTRICH files

The "ost" directory is the engine to prepare and run OSTRICH; it contains OSTRICH executable and input files. OSTRICH output files will be written in this folder as well after execution. More instructions are included inside each file.



- **OstIn.txt** is the main OSTRICH controller file. This file is shown below and determines model executable files, OSTRICH program type, and contains parameter names and ranges (the number, order, and ranges of the parameters should be identical to those specified for VARS in factorSpace.txt file). It also should contain all parameter sets generated by VARS and to be simulated by the model. **Parameter sets generated by VARS in step 1 should be copied into the designated section in this file**. If the file gets too large, then codes or scripts can be written to perform this job. **Note that "`ProgramType`" in this file should be set to "`ModelEvaluation`"** for performing Monte-Carlo type simulations for the given parameter sets. The use of the "#" will comment out lines in this file. The number of metrics to be calculated and/or integrated as the final objective function, as well as, the way they will be printed out by OSTRICH is defined in this file.

- **XXX.tpl** files are template files used by OSTRICH to generate model input files associated with each parameter set to run the model. These files determine how the values for each parameter is mapped to the proper location in each input file.

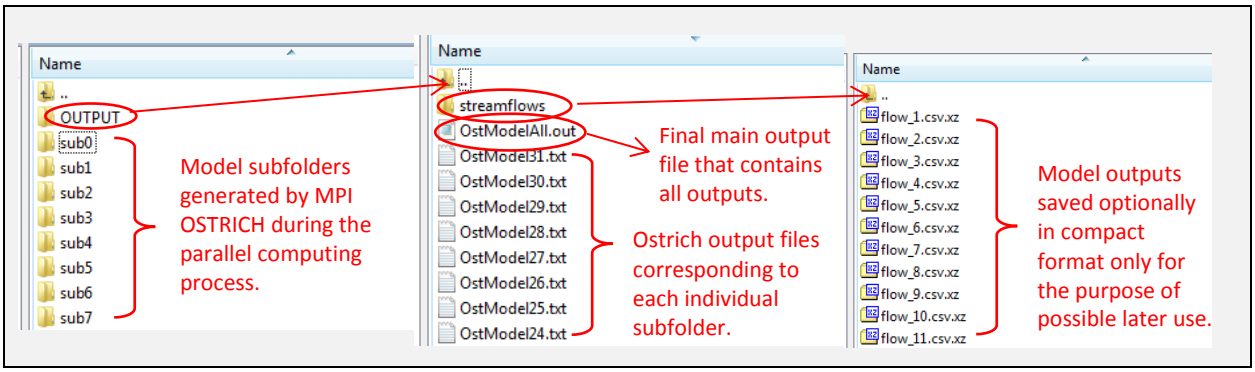


2.4. Submit the job to execute OSTRICH

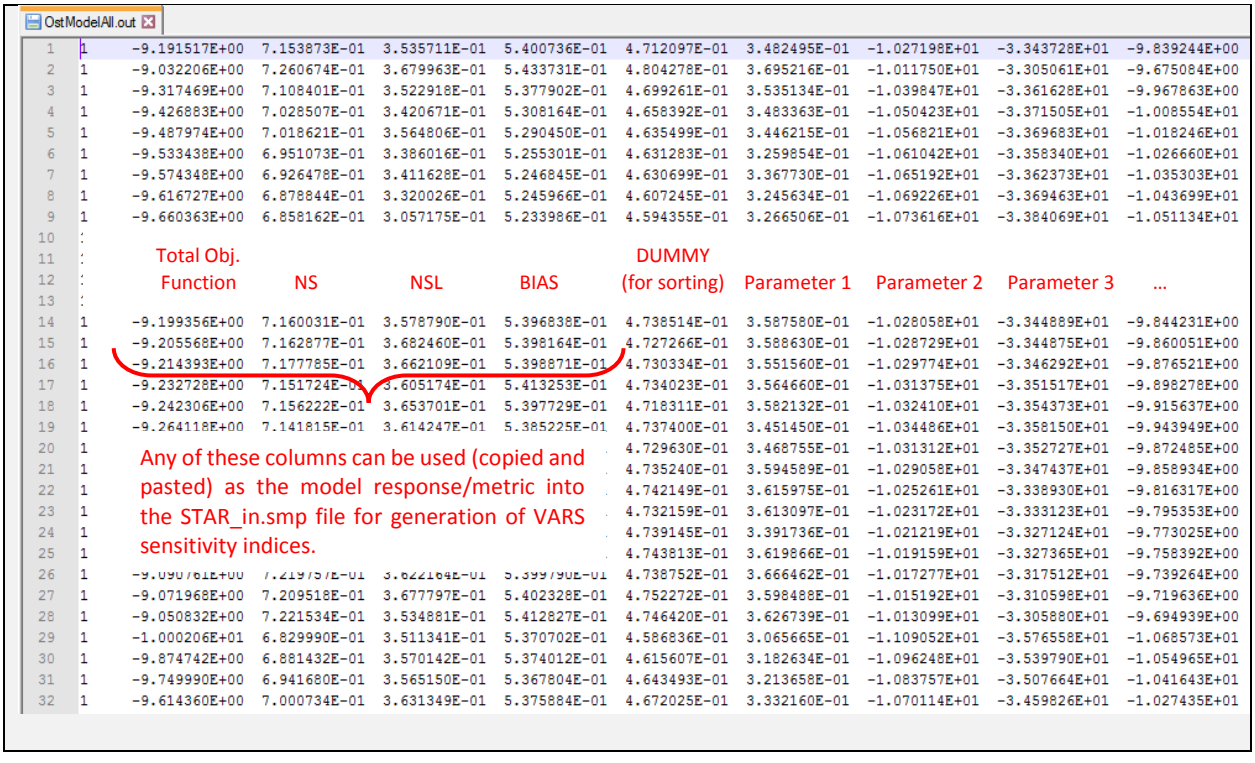
The batch file MPI.SLURM will take care of running MPI OSTRICH (along with pre- and post-processing) in Linux. This file is based on the Slurm job scheduler and instructions are available in it. The job is submitted by:

```
sbatch MPI.SLURM
```

Once the job is submitted, MPI OSTRICH will create model subfolders (sub1, sub2, …) and the "OUTPUT" directory in the main "ost" directory. The OUTPUT folder will contain all model outputs (if chosen to be saved) as well as OSTRICH output files as shown below.



Once all the simulations are complete, a final file, "OstModelAll.out", will be created that merges all OSTRICH output files from each subfolder together. The first 30 lines of this file is shown below for verification purposes. This case study was conducted with multiple metrics (NS=Nash-Sutcliffe, NSL=Nash-Sutcliffe of log of flows, and BIAS). Any of these metrics shown in this file can be copied into the STAR_in.smp for calculation of sensitivity indices by VARS as explained in the next step 3.



1	1	-9.191517E+00	7.153873E-01	3.535711E-01	5.400736E-01	4.712097E-01	3.482495E-01	-1.027198E+01	-3.343728E+01	-9.839244E+00
2	1	-9.032206E+00	7.260674E-01	3.679963E-01	5.433731E-01	4.804278E-01	3.695216E-01	-1.011750E+01	-3.305061E+01	-9.675084E+00
3	1	-9.317469E+00	7.108401E-01	3.522918E-01	5.377902E-01	4.699261E-01	3.535134E-01	-1.039847E+01	-3.361628E+01	-9.967863E+00
4	1	-9.426883E+00	7.028507E-01	3.420671E-01	5.308164E-01	4.658392E-01	3.483363E-01	-1.050423E+01	-3.371505E+01	-1.008554E+01
5	1	-9.487974E+00	7.018621E-01	3.564806E-01	5.290450E-01	4.635499E-01	3.446215E-01	-1.056821E+01	-3.369683E+01	-1.018246E+01
6	1	-9.533438E+00	6.951073E-01	3.386016E-01	5.255301E-01	4.631283E-01	3.259854E-01	-1.061042E+01	-3.358340E+01	-1.026660E+01
7	1	-9.574348E+00	6.926478E-01	3.411628E-01	5.246845E-01	4.630699E-01	3.367730E-01	-1.065192E+01	-3.362373E+01	-1.035303E+01
8	1	-9.616727E+00	6.878844E-01	3.320026E-01	5.245966E-01	4.607245E-01	3.245634E-01	-1.069226E+01	-3.369463E+01	-1.043699E+01
9	1	-9.660363E+00	6.858162E-01	3.057175E-01	5.233986E-01	4.594355E-01	3.266506E-01	-1.073616E+01	-3.384069E+01	-1.051134E+01
10	:									
11	:									
12	:									
13	:									
14	1	-9.199356E+00	7.160031E-01	3.578790E-01	5.396838E-01	4.738514E-01	3.587580E-01	-1.028058E+01	-3.344889E+01	-9.844231E+00
15	1	-9.205568E+00	7.162877E-01	3.682460E-01	5.398164E-01	4.727266E-01	3.588630E-01	-1.028729E+01	-3.344875E+01	-9.860051E+00
16	1	-9.214393E+00	7.177785E-01	3.662109E-01	5.398871E-01	4.730334E-01	3.551560E-01	-1.029774E+01	-3.346292E+01	-9.876521E+00
17	1	-9.232728E+00	7.151724E-01	3.605174E-01	5.413253E-01	4.734023E-01	3.564660E-01	-1.031375E+01	-3.351517E+01	-9.898278E+00
18	1	-9.242306E+00	7.156222E-01	3.653701E-01	5.397729E-01	4.718311E-01	3.582132E-01	-1.032410E+01	-3.354373E+01	-9.915637E+00
19	1	-9.264118E+00	7.141815E-01	3.614247E-01	5.385225E-01	4.737400E-01	3.451450E-01	-1.034486E+01	-3.358150E+01	-9.943949E+00
20	1					4.729630E-01	3.468755E-01	-1.031312E+01	-3.352727E+01	-9.872485E+00
21	1					4.735240E-01	3.594589E-01	-1.029058E+01	-3.347437E+01	-9.858934E+00
22	1					4.742149E-01	3.615975E-01	-1.025261E+01	-3.338930E+01	-9.816317E+00
23	1					4.732159E-01	3.613097E-01	-1.023172E+01	-3.333123E+01	-9.795353E+00
24	1					4.739145E-01	3.391736E-01	-1.021219E+01	-3.327124E+01	-9.773025E+00
25	1					4.743813E-01	3.619866E-01	-1.019159E+01	-3.327365E+01	-9.758392E+00
26	1	-9.090761E+00	7.219757E-01	3.622164E-01	5.399790E-01	4.738752E-01	3.666462E-01	-1.017277E+01	-3.317512E+01	-9.739264E+00
27	1	-9.071968E+00	7.209518E-01	3.677797E-01	5.402328E-01	4.752272E-01	3.598488E-01	-1.015192E+01	-3.310598E+01	-9.719636E+00
28	1	-9.050832E+00	7.221534E-01	3.534881E-01	5.412827E-01	4.746420E-01	3.626739E-01	-1.013099E+01	-3.305880E+01	-9.694939E+00
29	1	-1.000206E+01	6.829990E-01	3.511341E-01	5.370702E-01	4.586836E-01	3.065665E-01	-1.109052E+01	-3.576558E+01	-1.068573E+01
30	1	-9.874742E+00	6.881432E-01	3.570142E-01	5.374012E-01	4.615607E-01	3.182634E-01	-1.096248E+01	-3.539790E+01	-1.054965E+01
31	1	-9.749990E+00	6.941680E-01	3.565150E-01	5.367804E-01	4.643493E-01	3.213658E-01	-1.083757E+01	-3.507664E+01	-1.041643E+01
32	1	-9.614360E+00	7.000734E-01	3.631349E-01	5.375884E-01	4.672025E-01	3.332160E-01	-1.070114E+01	-3.459826E+01	-1.027435E+01

3) Running VARS to generate sensitivity indices

Here we run VARS again to read model outputs and obtain sensitivity indices:

- Navigate to the main VARS folder

- Change the stage flag on line 19 in VARS_inp.txt to 2
- Copy any of the metrics from "OstModelAll.out" file as a single column in STAR_in.smp
- Run main_VARS.m to obtain VARS output results in VARS_out.txt

Note on model crashes: Similar to other complex physically-based models, some combinations of parameter values can cause MESH simulations to crash. These crashes are identified by a "-9999" flag as the calculated metric value for the MESH example here. Before they can be fed back into VARS for sensitivity indices calculation, these "-9999" values should first be replaced by "NaN" values so their associated metric can be approximated using strategies available in VARS-TOOL as explained above.

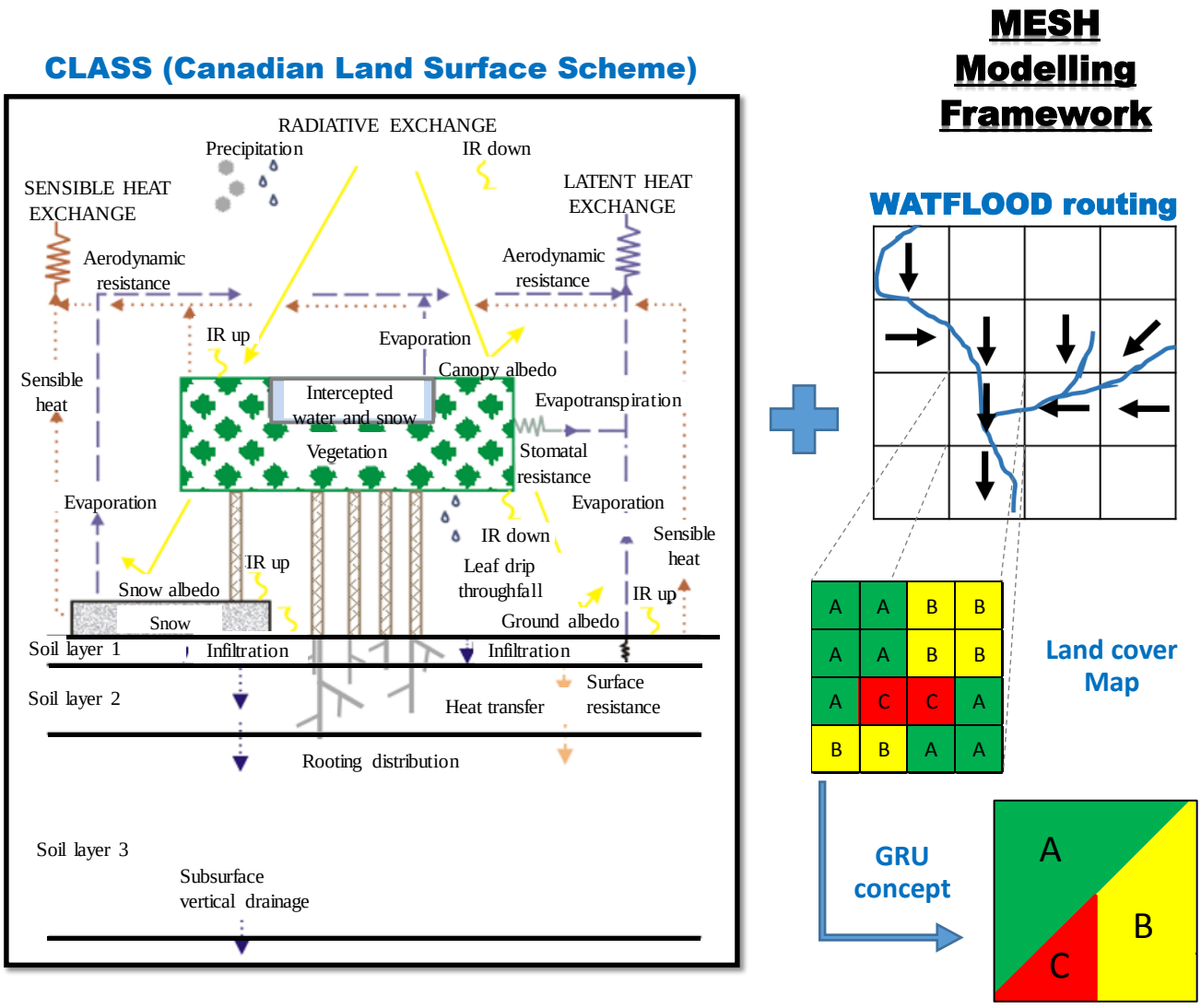
B2. Model description and case study

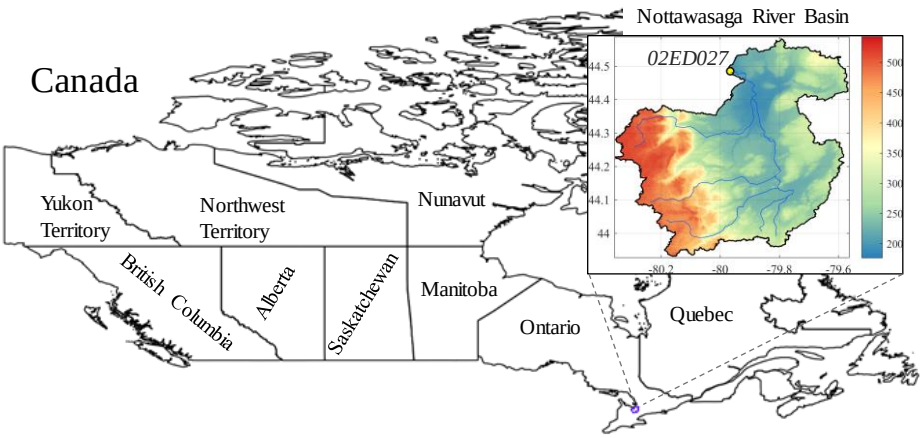
MESH is a semi-distributed coupled land surface-hydrology modelling system developed by Environment and Climate Change Canada (ECCC) for large-scale watershed modelling with consideration of cold region processes in Canada (Pietroniro et al., 2007).). It combines the highly parameterized Canadian Land Surface Scheme (CLASS, Verseghy, 1991; Verseghy et al, 1993) and the WATROUTE routing module of the WATFLOOD hydrological model (Kouwen, 1988).

MESH uses a grid-based modeling system and takes into account the sub-grid heterogeneity using the concept of Grouped Response Units (GRUs) (Kouwen et al., 1993). GRUs are the computational units in MESH often loosely defined to aggregate multiple attributes (e.g., soil and vegetation characteristics) into a single unit. The GRU concept is very useful for large scale distributed hydrological modelling because it largely increases computational efficiency by tying the majority of parameters to GRU types only.

The land surface scheme, CLASS, simulates the energy and water balances of the soil, snow and vegetation canopy for each GRU type in each grid cell. The amount of surface runoff (as well as other fluxes) is then calculated for each grid cell by the weighted area average of the existing GRUs within each cell. Then water is routed by WATFLOOD between the grid cells and across the river network for the entire basin. Figure below shows a schematic of the MESH model and the GRU concept.

In this example, the MESH model is applied to Nottawasaga River Basin, in Southwestern Ontario, Canada as seen below. Ranges used for the parameters are provided in the table below. For more details on the case study and model application please refer to Haghnegahdar et al. (2017).


CLASS (Canadian Land Surface Scheme)
RADIATIVE EXCHANGE
Precipitation
IR down
SENSIBLE HEAT EXCHANGE
LATENT HEAT EXCHANGE
Aerodynamic resistance
Aerodynamic resistance
IR up
Evaporation
Canopy albedo
Sensible heat
Intercepted water and snow
Evapotranspiration
Vegetation
Stomatal resistance
Evaporation
Evaporation
IR down
Sensible heat
IR up
Leaf drip throughfall
Snow albedo
IR up
Snow
Ground albedo
Soil layer 1
Infiltration
Infiltration
Surface resistance
Soil layer 2
Heat transfer
Rooting distribution
Soil layer 3
Subsurface vertical drainage
MESH Modelling Framework
WATFLOOD routing
Land cover Map
GRU concept
A
B
C



Parameters and their corresponding ranges used in VARS for the MESH model

Parameter	Description	Range
LAMAX	Annual maximum leaf-area index	(3, 6)
LNZ0	Natural logarithm of the roughness length	(-2.6, -1.3)
ALVC	Average visible albedo when fully-leafed	(0.04, 0.08)
ALIC	Average near-infrared albedo when fully-leafed	(0.2, 0.5)
CMAS	Annual maximum canopy mass (kg m^{-2})	(2, 5)
ROOT	Annual maximum rooting (m)	(0.2, 4)
RSMN	Minimum stomatal resistance (s m^{-1})	(60, 110)
QA50	Reference value of shortwave radiation (W m^{-2})	(10, 50)
VPDA	Vapour pressure deficit coefficient 'A'	(0.2, 0.8)
VPDB	Vapour pressure deficit coefficient 'B'	(0.7, 1.3)
PSGA	Soil moisture suction coefficient 'A'	(75, 125)
PSGB	Soil moisture suction coefficient 'B'	(2, 8)
SANDi[a]	Percent sand in the mineral soil of layer i (%)	(85, 100)[1] sandy soil (0, 45)[2] clay loam & silty clay loam
CLAYi	Percent clay in the mineral soil of layer i (%)	(0, 10)[1] sandy soil (30, 40)[2] clay loam & silty clay loam
ORGi	Percent organic matter in the mineral soil of layer i (%)	(0, 5)
SDEP	Soil permeable (Bedrock) depth (m)	(0.1, 4.1)
ZSNL	Threshold depth above which snow coverage is considered 100% (m)	(0.05, 0.2)
ZPLS	Maximum water ponding depth for snow-covered areas (m)	(0.02, 0.15)
ZPLG	Maximum water ponding depth for snow-free areas (m)	(0.02, 0.15)
DDEN	Drainage density (km km^{-2})	(2, 100)
XSLP	Estimated average slope of the GRU type	(0.0001, 0.04)
GRKF	Ratio of saturated horizontal hydraulic conductivity at a depth of 1 meter to the saturated horizontal hydraulic conductivity at the surface	(0.0001, 1)
MANN	Manning's roughness coefficient 'n'	(0.02, 2)
RATIO[b]	The ratio of horizontal to vertical saturated hydraulic conductivity	(2,100)
WFR2	Channel roughness factor	(0.02, 2)

[*] ranges are based on the crop vegetatoin type
[a] i indicates soil layer number (1 to 3)
[b] a parameter introduced by authors representing the horizontal saturated hydraulic conductivity