PUBLIC UNIVERSITY OF NAVARRE

DEPARTMENT OF ENGINEERING

**Evaluation and development of techniques for measuring, predicting and mitigating soil erosion and sediment export for a more sustainable agriculture**

by

**Iñigo Barberena Ruiz**

Supervisors:

**Javier Casalí Sarasibar**

**Miguel Ángel Campo Bescós**

**Rafael Muñoz Carpena**

A dissertation submitted to the Doctoral Programme in Science and Industrial Technologies in fulfillment of the requirements for the degree of DOCTOR OF PHILOSOPHY

2025

# PRESENTATION OF A NEW GRAPHICAL USER INTERFACE FOR THE USE OF VEGETATIVE FILTER STRIPS MODELING SYSTEM (VFSMOD)

## 6.1 Introduction

The Clean Water Act (CWA) serves as the primary legal framework regarding water pollution control in the United States. The CWA was first passed in 1948 and significantly amended in 1972 to incorporate a more systematic approach toward safeguarding the chemical, physical, and biological aspects of the waters in the United States. One of the primary regulatory features added during the 1972 amendments was the Total Maximum Daily Load or TMDL program, which sets the maximum allowable pollutant load a waterbody can assimilate without breaching water quality standards(U.S. Congress, 1972). Under this program, which arose from Section 303(d) of the CWA, states, territories, and authorized tribes must identify waters that do not meet set standards and establish TMDLs to restore these waters. These plans, which guarantee TMDLs are based on sound science and legislatively appropriate under the Act, are supported and approved by the Environmental Protection Agency (EPA) (EPA, 2024a, 2024b).

A TMDL includes contributions from both point sources, such as discharge pipes from treatment plants or industrial facilities, and nonpoint sources, such as runoff from agricultural land, roads, or urban areas. Because nonpoint source pollution is diffuse and highly variable in space and time, its management represents a major challenge in the context of TMDL implementation. As a result, there is a strong need to adopt and promote Best Management Practices (BMPs), which encompass a wide array of strategies intended to reduce pollutant loads from the landscape before they reach water bodies. BMPs may include structural measures, like constructed wetlands or sedimentation basins, as well as non-structural approaches, such as changes in land management, crop rotation, or conservation tillage.

One of the most researched and applied BMPs under agriculture, urban and transportation settings are Vegetative Filter Strips (VFS). Vegetative Filter Strips are zones of planted or native vegetation located downslope from agricultural, urban or other pollutant-generating areas. Their purpose is to intercept and treat surface runoff, reducing the transport of sediments, nutrients, pesticides, and pathogens to nearby streams, rivers, or lakes (Dillaha et al., 1989a). The vegetation slows the water that is flowing through the strip; it increases surface roughness; promotes infiltration and facilitates settling of suspended solids and related contaminants (Barfield et al., 1979; Barling & Moore, 1994; Bolton et al., 1991; Clinnick, 1985; Muñoz-Carpena et al., 1999). These processes reduce not only total surface runoff volume but also pollutant load before entering adjacent surface water bodies. VFS, when properly designed, installed and managed are a cost-effective method for improving surface water quality with positive environmental sustainability benefits.

VFS have been used for decades around the world and have consistently demonstrated their effectiveness in reducing nonpoint source pollution (Chen et al., 2024; Magette et al., 1989). Their simplicity, affordability, and additional ecological benefits such as erosion control, habitat creation, and landscape aesthetics make them especially attractive to farmers, landowners, and environmental managers. In the United States, they have been promoted by federal and state agencies as part of voluntary and regulatory conservation programs (EPA, 2002). In Europe as well, vegetative buffers are being increasingly adopted under agri-environmental schemes, in alignment with the growing focus on green infrastructure and sustainable agriculture (Brown et al., 2012; Klein et al., 2023). Despite their demonstrated utility, the design and implementation of VFS still require careful consideration of site-specific factors such as slope, soil type, rainfall patterns, vegetation cover, and land use upstream. Their performance is not uniform and can vary significantly depending on local conditions and management practices (Barling & Moore, 1994; Dillaha et al., 1989b; Kuo et al., 2005; Muñoz-Carpena, Vellidis, et al., 2007; Reichenberger et al., 2007).

In addition to TMDLs, it is essential to underscore the critical role of **VFS** as a quantitative mitigation measure for pesticide runoff within the regulatory environmental risk‑assessment framework, given their notable environmental and economic benefits. The estimated cost to bring a new pesticide to market, as of 2014–2019, averaged around **US $301 million**, with the **registration phase alone accounting for approximately 13.9 %** of that total (AgbioInvestor, 2024). Given such high development costs, regulators frequently mandate VFS as a mitigation measure, incorporating them directly into legally binding label instructions to protect non‑target organisms and water bodies. Empirical research (Chen et al., 2024) reinforces the efficacy of VFS in significantly reducing pesticide runoff, emphasizing their value both as practical conservation tools and as quantifiable elements in regulatory risk mitigation strategies. Meanwhile, VFS are generally recognized as low‑cost yet effective best management practices for intercepting pesticides before they reach surface waters.

Given their central role in reducing nonpoint source pollution and their inclusion in many watershed management plans, particularly those aimed at achieving TMDL goals, VFS continue to be a key tool in integrated water resource management. However, maximizing their effectiveness requires tools that support not only their promotion but also their proper dimensioning and evaluation under realistic field conditions.

Although Vegetative Filter Strips (VFS) have shown to be effective tools for mitigating nonpoint source pollution, their performance is highly dependent on proper design. For a VFS to meet pollutant reduction goals, such as those defined by a TMDL, it must be adapted to the specific physical, hydrological, and chemical conditions of the site where it is installed (Muñoz-Carpena & Parsons, 2004). A range of factors influences the trapping efficiency of VFS, including soil type, slope, vegetation structure, rainfall intensity and duration, inflow sediment concentration, and filter length and flow concentration, shallow water table conditions and others(Barling & Moore, 1994; Dillaha et al., 1989; Flanagan et al., 1989; Fox et al., 2010; Lauvernet & Munõz-Carpena, 2018; Muñoz-Carpena, Vellidis, et al., 2007; Reichenberger et al., 2007). These variables interact in very complex ways so one cannot rely on empirical rules or simple rule-of-thumb-type guidelines alone to go for effective filters. Early efforts often involved manual design based on expert knowledge, which, although practical, introduced large uncertainties and inconsistencies (Dillaha et al., 1985).

To move beyond these limitations, mechanistic models have been developed to simulate the physical processes governing flow, sediment transport, and pollutant filtering within VFS. These models provide a means to test design scenarios, evaluate performance and comprehend the processes behind VFS. Mechanistic VFS modeling must cover significant components from overland flow, and infiltration to sediment deposition and pollutant transport (Barling & Moore, 1994; Muñoz-Carpena et al., 1999). However, like all environmental models, these tools are subject to uncertainty from model structure, parameter estimation or variability in input data (C. T. Haan, 1989). Neglect of uncertainty leads to overconfidence in model predictions and worse design decisions.

Evaluating the uncertainty associated with BMP performance is critical for making informed, risk-aware decisions, especially in the context of TMDL compliance (Muñoz-Carpena et al., 2007; Shirmohammadi et al., 2006). Quantifying uncertainty helps decision-makers understand the range of possible outcomes and the likelihood that a given design will meet pollutant reduction targets. It also supports adaptive management by identifying which parameters contribute most to model variability and where additional data collection could reduce uncertainty (Reckhow, 1994a, 1994b). Sensitivity analysis can identify the most influential input parameters, while uncertainty analysis propagates variability in those parameters through the model to evaluate its impact on outcomes (Saltelli et al., 2004). These two approaches are complementary and, when applied jointly, provide a robust assessment of model behavior and reliability also in the context of VFS (Fox et al., 2010; Muñoz-Carpena et al., 2018, 2019; Muñoz-Carpena, Zajac, et al., 2007; Muñoz‐Carpena et al., 2010; Reichenberger et al., 2023).

To design a VFS that meets specific TMDL objectives, a systematic modeling approach is recommended. According to Parsons & Muñoz Carpena (2001), the process begins by (1) identifying and ranking input parameters by their sensitivity, (2) assigning probability distributions to the most influential parameters, and (3) using Monte Carlo simulations to generate probabilistic output distributions, such as sediment trapping efficiency. The probabilistic output distribution can then be applied to identify confidence intervals and exceedance probabilities to assess if the VFS is likely to deliver the performance required under operational conditions. This process not only accounts for system variability but also improves transparency in design evaluation.

A number of mechanistic models have been proposed for simulating the behavior of VFS. Among the earliest is GRASSF, developed at the University of Kentucky to simulate sediment filtration by artificial grass media (Barfield et al., 1978, 1979; Hayes et al., 1982), and later integrated into SEDIMOT II (Wilson et al., 1981). Another commonly used model is CREAMS (Knisel, 1980), which simulates surface runoff and sediment yield, although it lacks detailed representation of VFS hydrology and the dynamic influence of filter properties on flow. The most comprehensive and widely adopted model for VFS performance is VFSMOD (Muñoz-Carpena et al., 1999; Muñoz-Carpena & Parsons, 2023), a field-scale, storm-based, mechanistic model designed specifically to simulate the routing of inflow hydrographs and sedimentographs through vegetative filters. VFSMOD improves upon earlier models by offering (a) state-of-the-art descriptions of surface flow through the filter, (b) dynamic changes in flow due to sediment deposition, (c) time-dependent infiltration modeling, (d) support for complex and variable storm patterns, and (e) the ability to account for heterogeneities in slope and vegetation along the filter strip(Muñoz-Carpena et al., 1999), (f) mechanistic description of complex chemical fate and transport processes through VFS (trapping, sorption distribution, leaching, degradation, remobilization, etc.) (Muñoz-Carpena et al., 1999, 2019; Reichenberger et al., 2023).

Because VFSMOD includes dedicated modules for surface water balance, overland flow depth, sediment filtration, and pollutant transport, it is especially well-suited for evaluating BMPs in the context of TMDL planning and the regulatory risk analysis and management of complex runoff pollutants like pesticides. It enables the design of filters based on quantitative objectives, facilitates the integration of uncertainty analysis, and provides a platform for calibrating model performance when field data are available. Importantly, when properly calibrated and tested, such models can reduce the need for expensive field trials and enable robust scenario testing across a range of site conditions (Muñoz-Carpena, Zajac, et al., 2007). In this context, VFSMOD serves not just as a simulation tool, but as an essential decision-support platform for evaluating and designing effective and reliable vegetative filter strips.

Although VFSMOD is a well-established and widely used model for simulating the hydrological and pollutant-filtering behavior of vegetative filter strips (Chen et al., 2024), the original Microsoft Windows graphical user interface (GUI) (Muñoz-Carpena & Parsons, 2020), VFSMOD-W, had several limitations that affected its usability, particularly in newer computing environments. One of the most critical issues was its incompatibility with recent versions of Microsoft Windows, which hindered its adoption and required users to implement manual workarounds or system modifications to run the software. This problem ultimately motivated the development of a new GUI (redesigned from the ground up) not only to solve compatibility issues but also to offer a cross-platform user-friendly, robust and flexible interface with more functionalities.

The new GUI has been built in Python, allowing for cross-platform functionality and eliminating the need for external dependencies. Unlike its predecessor, which was limited to Windows and required users to install additional MATLAB libraries, the new GUI can run on Windows, macOS, and Linux without the need for initial Python installation or any prior software installation on the user’s system. This significantly improves accessibility and broadens the potential user base, including technical staff in agencies and industry, and non-expert users such as farmers or local government personnel.

Beyond platform compatibility, the new GUI introduces a wide range of functional enhancements. One major improvement is the expansion of sensitivity and uncertainty analysis capabilities. The interface now allows users to include model variables across all water, sediment and pollutant modules rather than a predefined subset in previous versions. In addition, it supports global methods such as the variance-decomposition Sobol sensitivity analysis (Sobol, 1993), which was not available in the previous version. The sensitivity analysis module is now fully integrated into the GUI, removing the previous dependence on external tools like SimLab. Results are presented through an interactive visualization system that helps users interpret the impact of each input variable on model outputs more intuitively.

The uncertainty analysis module has been significantly enhanced as well. Similarly, all model variables can now be included in the pseudo-Monte Carlo uncertainty propagation process, and the resulting distributions and exceedance probabilities are presented using interactive plots. This makes it easier for users to evaluate whether a given VFS design is likely to meet its pollutant reduction goals under realistic variability in a rigorous statistical way.

Another key advancement is in the calibration tools. The new GUI no longer depends on MATLAB libraries, making the calibration process simpler and more accessible. Users can now calibrate the model using a wider range of data inputs. In addition to traditional hydrograph and sedimentograph calibration, the new interface supports calibration using single values such as total runoff volume, which are often easier to obtain in field conditions. Calibration can also be performed using pesticide-related outputs, and be objectively evaluated using bootstrapping and hypothesis testing techniques (Ritter & Muñoz-Carpena, 2013). Furthermore, the system includes a parameter identifiability analysis module, which helps users determine which parameters most influence the calibration results. This information can guide the setup of the calibration process and reduce uncertainty.

One of the most notable additions is the ability to perform design under uncertainty. Whereas the previous GUI supported only deterministic design based on average conditions, the new version allows users to determine VFS dimensions that meet TMDL goals under varying environmental conditions and input uncertainties. This capability is particularly important when managing agricultural pollution at the catchment scale or in regulatory pesticide risk assessment scenarios, where variability is inherent and high.

Additionally, the new GUI supports the parallel execution in regular PCs of computationally intensive tasks such as sensitivity analysis, uncertainty analysis, design under uncertainty, and parameter identifiability analysis. This parallelization dramatically reduces computation time and enables users to explore a larger number of scenarios more efficiently. Other improvements include the ability to simulate and analyze new VFSMOD processes, such us degradation of multiple products simultaneously, including their degradation into secondary compounds, which is particularly relevant for regulatory risk assessment. Lastly, the overall size of the program has been reduced by eliminating unnecessary dependencies, making it lighter and more portable.

In this chapter, the new graphical user interface for VFSMOD that overcomes the limitations of the first version is presented. In addition to the compatibility issues, the new interface adds a number of new features intended to improve usability, flexibility and analysis capability. These improvements combine to increase the usefulness of VFSMOD as a mechanistic (process-based) decision-support tool, for designing, evaluating and optimizing vegetative filter strips for any given set of environmental and regulatory constraints.

## 6.2 Proposed new paradigm for VFS design under uncertainty

Ultimately, the objective of using VFSMOD is often to design a vegetative filter strip that achieves specific pollutant reduction targets, whether in terms of sediment, runoff volume, or pesticide loads. To meet these targets, it is necessary to explore different system configurations until a solution is found that satisfies the desired criteria. This iterative process is referred to as the **design phase**, and it was already implemented in the previous graphical user interface, where the model would identify the minimum filter length required to achieve a set of deterministic goals.

In the old GUI version, the design was strictly deterministic: all system parameters were fixed, and only the length of the filter was adjusted iteratively to meet the performance threshold. However, it is widely recognized that input parameters in environmental models are often uncertain. This uncertainty can significantly influence the outcome of the design process, leading to under- or overestimation of filter performance if not properly accounted for.

To address this, the new GUI introduces a **design under uncertainty** functionality. Instead of relying on a single set of input values, this approach uses probabilistic input distributions to perform the design process across a range of plausible conditions. The result is no longer a single "optimal" filter length, but rather a range of filter lengths associated with different levels of confidence or exceedance probabilities. This provides users with a more robust basis for decision-making, enabling them to select a design that not only meets pollutant reduction targets but also maintains performance under realistic variability.

When combined with the statistically-grounded calibration workflow described earlier, this new design capability enables a complete, objective methodology for vegetative filter design. Users can first calibrate the model using observed data and rigorous goodness-of-fit evaluation, and then proceed to design the filter with full consideration of the system’s inherent uncertainty. This integrated approach strengthens the reliability of model-based decisions and enhances the utility of the new graphical interface as a comprehensive decision-support tool for environmental management.

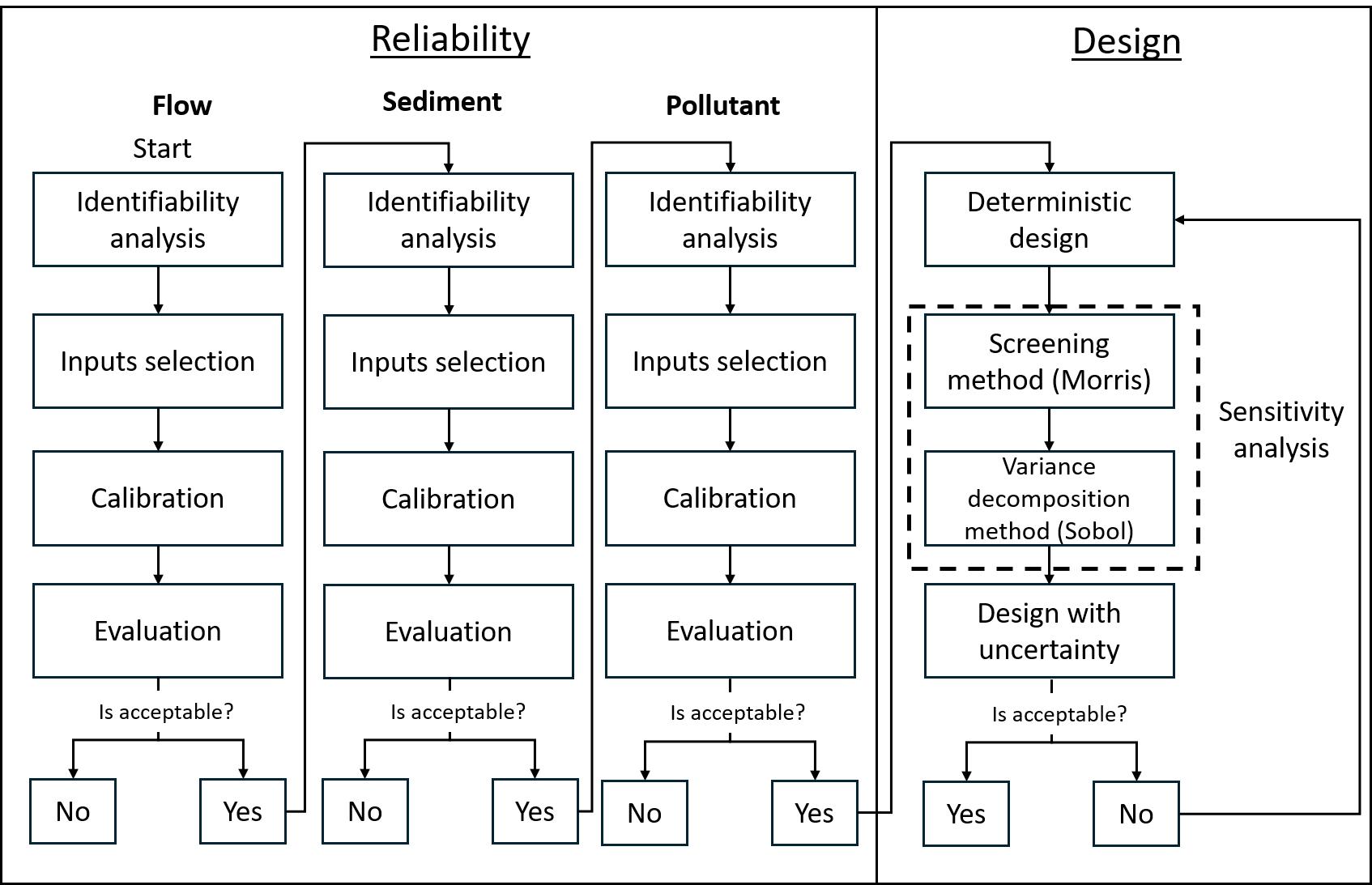


Figure 6.1. **Proposed flowchart for the design of a vegetative filter strip.** The process begins with a reliability phase, where the model is calibrated. Parameter values are estimated sequentially, starting with flow, followed by sediment, and finally the pollutant, as sediment and contaminant transport depend on flow behavior. During each calibration step, identifiability analysis can be performed, and the resulting calibration can be evaluated using advanced metrics. Once calibration is completed, the design phase follows. This includes a deterministic design and a design under uncertainty, ultimately yielding the optimal filter length based on a calibrated model and accounting for system uncertainty.

The full methodology supported by the new graphical user interface can be summarized as follows: the process begins with model calibration to assess its reliability under observed conditions, and only after successful calibration is the design phase initiated. Calibration can be performed for up to three components—runoff, sediment, and pesticide transport—depending on data availability. Notably, pesticide calibration was not supported in earlier versions of the interface, and represents a significant enhancement in this new version.

Because sediment and pesticide transport are driven by overland flow, it is essential to begin the calibration process by optimizing parameters related to runoff. If sufficient observational data are available, an initial identifiability analysis can be conducted to identify which parameters are most suitable for calibration. This step helps prevent equifinality by focusing the optimization process on the most informative inputs. Once these inputs have been selected, the model is calibrated and evaluated using the expanded suite of tools described previously. If the runoff component meets predefined performance criteria, the same process is applied to the sediment module, and then, if applicable, to pesticide transport.

Following successful calibration, the process moves into the design phase. A useful first step is to perform a deterministic design, as it requires significantly less computational effort and provides an initial estimate of the required filter dimensions. This approach is particularly helpful when evaluating whether certain pollutant reduction targets, such as a specific sediment delivery ratio, can be met under fixed storm conditions.

Once a general understanding of system behavior has been obtained through deterministic design, the user can proceed to a more comprehensive design under uncertainty. A practical first step in this process is to apply a Morris sensitivity analysis, which requires relatively few model runs and can identify the most influential input parameters affecting the selection of the filter length. Based on the results, the uncertainty design can be focused on a reduced set of inputs with low variability, thus reducing computational demand while maintaining analytical rigor.

Using this refined input set, the interface can perform a variance-based design analysis. The results can then be used to construct confidence intervals for different filter lengths, allowing the user to determine, for example, the probability that a given filter configuration will meet pollutant reduction targets under realistic variability. Alternatively, the user can select the filter length that meets the target threshold in a specified percentage of simulated conditions. If the resulting confidence levels are not acceptable, the process can be iterated—for example, by returning to a deterministic design under a less severe storm scenario, followed by a new round of uncertainty-informed design.

This iterative, modular workflow enables users to design vegetative filter strips that are not only effective under average conditions, but also robust under uncertainty. By integrating model calibration, sensitivity and identifiability analysis, and uncertainty-based design, the new GUI offers a complete and transparent decision-support system for VFS planning.

## 6.3 Overview of the new GUI

The newly developed graphical user interface (GUI) can be understood as a bridge between the VFSMOD model and the user. Like most environmental simulation tools, VFSMOD operates through a set of structured input files that describe the system to be simulated, and it produces output files with the results of the simulation. The core function of the GUI is to manage these input and output files in a user-friendly and dynamic way and add additional model evaluation and analysis capabilities. In essence, all the capabilities offered by the GUI, ranging from model configuration to advanced analyses, are implemented through the automated modification of input files and the interpretation of output data. By handling this interaction seamlessly, the GUI shields users from the complexity of manual file editing and data processing, while enabling the full range of VFSMOD’s functionalities. Importantly, a core principle of the GUI design is to be separated from the VFSMOD command line engine, so that future versions can be used by swapping the engine. This also ensures compliance with existing VFSMOD versions used in other regulatory tools such as the Surface Water Assessment eNabler (SWAN) tool (EC-ESDAC, 2022), using the EU FOCUS (FOrum for the Co‐ordination of pesticide fate models and their USe) framework, and in North America, tools under development around the US-EPA Pesticide in Water Calculator (PWC) framework such as Canada's PMRA “VFSPipe”, California Department of Pesticide Regulation's (CDPR) Pesticide Registration Evaluation Model (PREM) and VFS4PWC, and others.

This new GUI supports a broad range of modeling operations: from simple single-run simulations to assess system response under fixed conditions, to multi-step workflows involving calibration of runoff, sediment, and pesticide components, evaluation of goodness of fit, identifiability analysis, deterministic and uncertainty-based design, and comprehensive sensitivity and uncertainty assessments. The GUI modules for these operations are organized in the main menu of the application (Fig. 6.2).

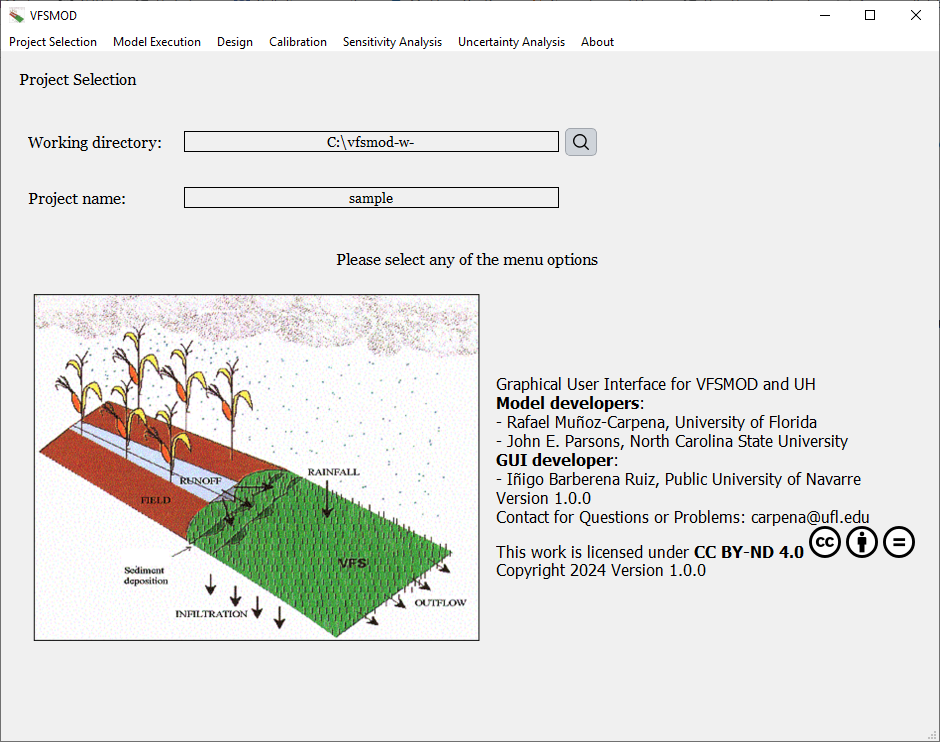


Figure 6.2. **Window displayed when opening the Graphical User Interface.** By default, the window that appears upon launching the Graphical User Interface is the one related to project selection, as shown here. At the top, a panel allows users to select the different functionalities available within the interface. The working directory and the project name can be specified directly below. In the central lower part, information about the model developers and the developer of the Graphical User Interface is displayed.

### 6.3.1 Upslope field inputs preparation module (UH)

To properly simulate the performance of a vegetative filter strip, VFSMOD requires detailed information about the upslope source area contributing runoff and pollutants. These input parameters include data such as the hyetograph (rainfall distribution), hydrograph (runoff flow rate over time), and sedimentograph (sediment load over time). However, acquiring this type of data is often challenging, especially at the level of temporal and spatial resolution required for mechanistic modeling. To address this limitation, the original VFSMOD package includes a complementary tool module known as UH (Muñoz-Carpena & Parsons, 2004), which generates the upslope source area inputs from general characteristics based on NRCS hydrological methods (USDA, 1986).

The UH module follows a structured sequence of steps to generate the necessary data and VFSMOD input files for the simulation. First, it reads input parameters describing the upslope source area, including total rainfall depth, rainfall distribution, storm duration, and the contributing drainage area, among others. Using this information, the module estimates the total runoff generated from the upslope area through the Curve Number (CN) method (USDA, 1986).

Once the total runoff is determined, the UH module calculates key hydrologic timing parameters such as time of concentration, peak runoff rate, and time to peak using the NRCS TR55 approach. With these parameters, the hydrograph is then constructed based on unit hydrograph theory. The hydrograph shape is influenced by the storm type selected by the user, following standard NRCS storm distributions (types I, IA, II, III), although users may also manually define custom rainfall accumulation patterns for greater flexibility and application in other world regions.

For sediment input, the module employs different versions of the event-based Universal Soil Loss Equation (USLE) such us (MUSLE, MUSS, Foster, and Williams forms) to estimate sediment yield from the storm event. The soil particle size distribution (soil texture) is provided by the user and used to obtain the USLE K factor. The average sediment concentration in runoff is subsequently estimated based on the total runoff volume and the sediment load, allowing for realistic inflow conditions to be simulated. The representative sediment particle characteristics of the sedigraph, i.e., effective diameter (d50), fraction of coarse particles, and particle density, can be supplied by the user or estimated internally in UH based on the user-supplied USDA texture of the source area topsoil (Whoolhiser et al., 1990). This estimate can be improved in VFSMOD later with a new dynamic algorithm based not only on soil but storm characteristics (Reichenberger et al., 2023). Finally, the results are used to create input files formatted ready for VFSMOD simulations.

Figure 6.3 summarizes the complete UH workflow:

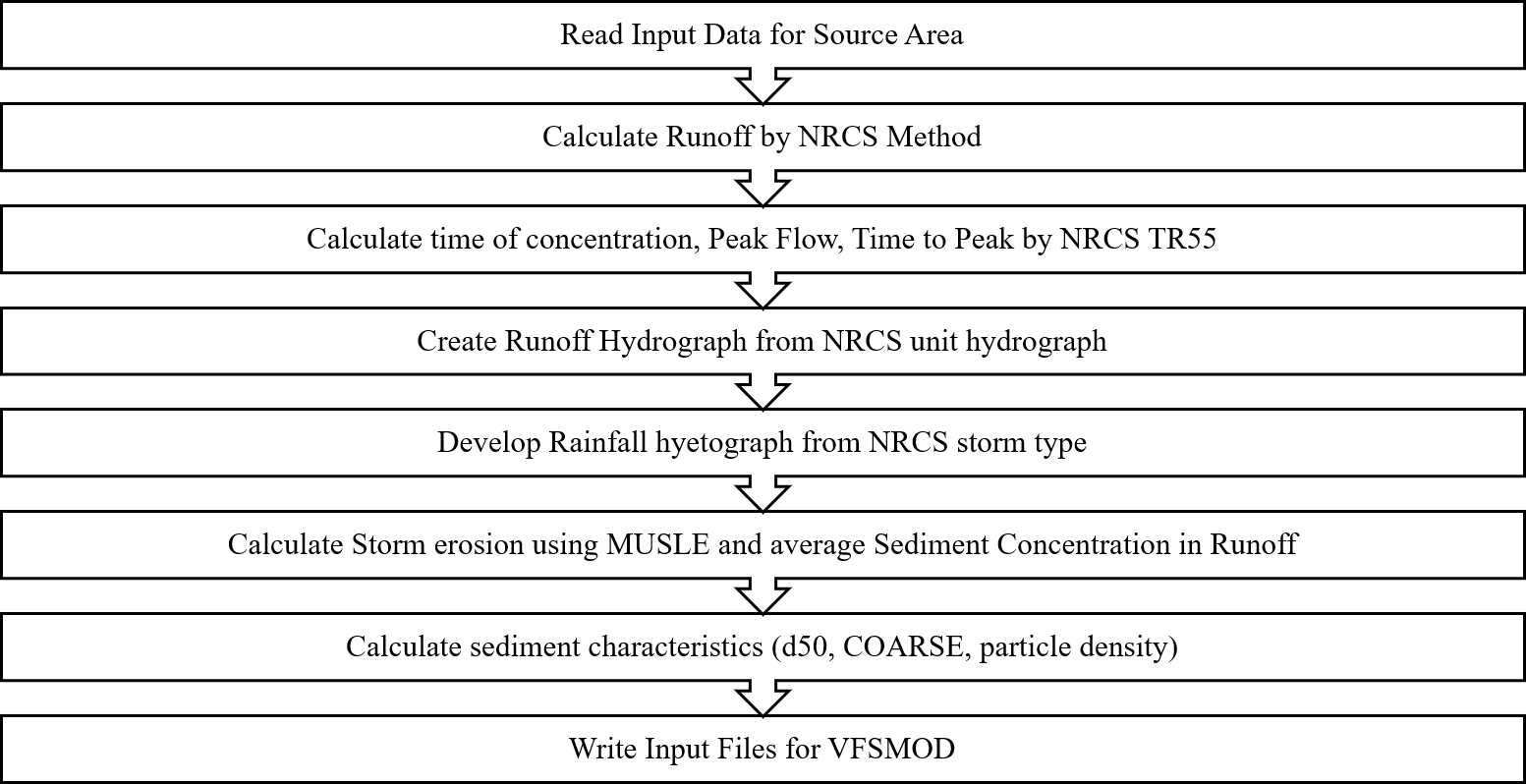


Figure 6.3. **Computational processes performed in the Unitary Hydrograph (UH) module**. (adapt. from *Muñoz-Carpena & Parsons (2020))*

The UH module produces three main outputs required by VFSMOD to simulate runoff and sediment transport processes: the hyetograph, the hydrograph, and the sediment loss file (sedimentograph). However, VFSMOD requires additional input data beyond those generated by UH, as not all relevant parameters are related to the upslope source area but relate to the specifics VFS and pollutant characteristics

### 6.3.2 VFS and pollutant characteristics and single VFSMOD execution

These additional VFS inputs are grouped into several categories: *Overland Flow*, which include the slope, width, and length of the buffer; *Infiltration-Soil*, which define how water infiltrates into the soil; *Buffer Vegetation*, such as vegetation height and Manning’s roughness coefficient; and *Water Quality*, which describe the types and concentrations of pesticides applied upstream. This last category is particularly important, as VFSMOD is capable of simulating complex pesticide transport and degradation through the filter strip, making it a powerful tool for evaluating pollutant mitigation strategies in regulatory and management applications.

Each of these input categories is managed in VFSMOD through different files with specific extensions, and each file contains multiple parameters required by the model. Together, these inputs allow VFSMOD to simulate a wide range of hydrological and pollutant transport processes under realistic conditions.

Figure 6.4 presents a general flow diagram of a single VFSMOD execution, showing the input data required to run UH, the outputs generated by UH, the additional inputs required directly by VFSMOD, and the final model outputs.

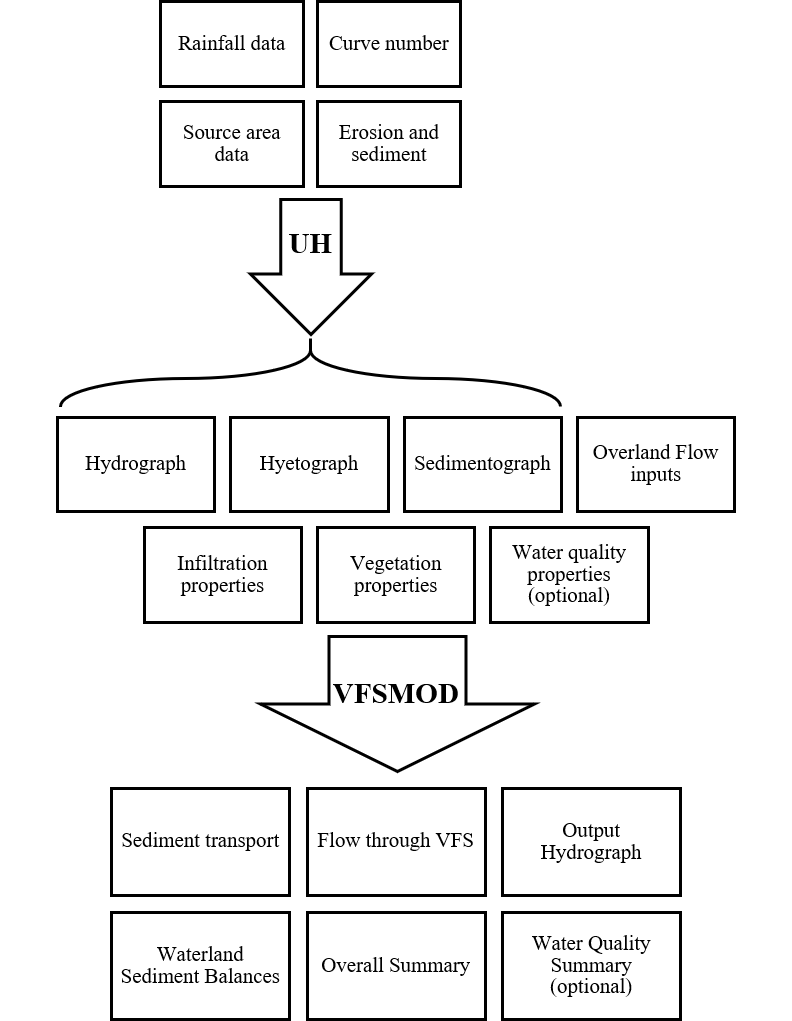


Figure 6.4. **Flowchart of the VFSMOD modeling process.** When observed inputs are not available the model can use UH to generate the required upstream data such as the hydrograph, hyetograph, and sediment-related inputs. These are created using the field or source area conditions. Once generated, VFSMOD uses these inputs to simulate and evaluate the performance of the vegetative filter strip.

The workflow described above applies to a single model execution scenario, where the objective is to evaluate the behavior of a specific system configuration based on a known set of input parameters. This type of execution, that can be referred as a deterministic run, allows users to analyze how a vegetative filter strip responds under particular environmental conditions. Such individual simulations can be readily performed using the new graphical user interface (GUI) presented in this work.

### 6.3.3 Automatic inverse model calibration against observed data

Model-based decision-making often requires more than just single-run simulations. A robust modeling approach must also include comprehensive model evaluation procedures to understand how input variability influences model outcomes and to assess the reliability of model predictions. As emphasized by Saltelli et al. (2004, 2005), model evaluation should be framed within a well-defined statistical framework that includes several interconnected components.

For a model to be reliably applied in environmental simulations, the parameter values used must accurately represent the hydraulic properties of both the soil and the filter (Ritter et al., 2007). The success of the modeling process depends critically on the representativeness of these inputs. One approach to achieve this is through parameter calibration, where model inputs are adjusted so that the model outputs (such as runoff, sediment yield, or pesticide concentrations) match observed data.

While manual calibration of VFSMOD against field observations is possible, it is often time-consuming and highly subjective. An alternative is inverse modeling, which involves iteratively adjusting parameters within predefined bounds to minimize the discrepancy between simulated and observed outputs. This optimization is typically guided by an objective function, which quantifies the error between model predictions and observations. This process is commonly referred to as model calibration.

To ensure robust and reliable use of environmental models, calibration must be performed objectively. A variety of statistical indicators can be used to evaluate the goodness of fit between observed and simulated values. However, relying on a single metric may lead to misleading conclusions. A more comprehensive evaluation should combine graphical analysis with multiple statistical indicators, including both absolute error statistics and normalized goodness-of-fit metrics. In addition, the statistical significance of the model fit should be assessed—an aspect that is often overlooked but can greatly reduce subjectivity in interpreting model performance.

To support this more rigorous approach, the new graphical user interface includes a dedicated calibration module. This module enables users to visualize model fit through graphical tools, assess performance using a range of error metrics, and evaluate statistical significance using probability distributions derived via bootstrapping, following the methodology proposed by (Ritter & Muñoz-Carpena, 2013). It also incorporates hypothesis testing to determine whether specific performance indicators exceed predefined thresholds. By integrating these features, the GUI offers a more transparent, statistically grounded calibration workflow that reduces user subjectivity and strengthens the credibility of model applications.

Calibrating or performing inverse optimization on all model parameters simultaneously is generally not recommended. Doing so increases the risk of encountering issues related to parameter non-identifiability or non-uniqueness, situations in which multiple combinations of parameter values result in equally good model performance. In such cases, it becomes impossible to determine which specific combination of parameters is truly correct, leading to what is known as equifinality(Beven & Freer, 2001). This phenomenon undermines the reliability of the calibration process and reduces confidence in the model’s predictive power. To mitigate these issues, calibration should focus only on those parameters to which the model is most sensitive. In practice, this means that the subset of parameters selected for optimization should be determined based on a preliminary sensitivity analysis. This approach reduces the dimensionality of the problem and enhances parameter identifiability.

It is important to distinguish between sensitivity analysis and identifiability analysis, as they address different aspects of the modeling process. Sensitivity analysis evaluates how variations in model inputs influence outputs—making it a characteristic of the forward or simulation problem. In contrast, identifiability analysis examines how changes in inputs affect the model’s goodness-of-fit to observed data (Gupta & Razavi, 2018)(Gupta & Razavi, 2018). This makes identifiability analysis a feature of the inverse problem, where the goal is to determine which parameters can be reliably estimated from available data. While identifiability inherently depends on model sensitivity, the reverse is not necessarily true: a parameter can be influential in the simulation of outputs without being identifiable through calibration, particularly when outputs are insensitive to unique combinations of that parameter. Therefore, although both analyses are interconnected, they provide different insights and must be treated as distinct components of a rigorous modeling workflow.

The new GUI supports this methodology by integrating tools for both sensitivity and identifiability analysis, ensuring that model calibration is not only efficient but also scientifically grounded. This reduces the risks associated with equifinality and enhances the credibility of model-based decision-making. Furthermore, the implementation of an easy-to-use and automated calibration module within the VFSMOD GUI is expected to greatly benefit modelers working on the design of vegetative filter strips, particularly when experimental data are available. By streamlining the calibration process and embedding it directly within the interface, the new approach helps reduce the uncertainty and subjectivity that often accompany water quality model calibration tasks.

### 6.3.4 Global sensitivity and Uncertainty Analysis Module

Sensitivity analysis refers to the process of quantifying the influence that each input parameter has on the model outputs. In other words, it examines how changes in individual inputs affect the results produced by the model. On the other hand, uncertaintyanalysis focuses on evaluating the overall uncertainty in the outputs that arises due to variability or lack of knowledge in the model inputs. While sensitivity analysis identifies which parameters matter most, uncertainty analysis characterizes how uncertain the model predictions are, given the uncertainty in those parameters.

First, it is necessary to identify a reduced subset of input variables that exert the most significant influence on output variability. Based on this reduced set of inputs, the next step involves a quantitative decomposition of the output variance, attributing portions of the total variability to both individual input effects and higher-order interactions among them. Finally, a model uncertainty analysis is performed by constructing probability density functions (PDFs) for the output variables, derived from the variance-based methods. This probabilistic representation enables a more nuanced understanding of model behavior and supports more informed, risk-aware decision-making.

In recognition of the importance of both processes in model evaluation, the new graphical user interface includes built-in modules for sensitivity and uncertainty analysis. These tools allow users not only to understand the internal behavior of the model, but also to explore the robustness of its predictions under realistic environmental variability.

### 6.3.5 VFS design module: deterministic and under uncertainty

This new GUI, as previously noted, enables both deterministic analysis, based on fixed input values, and uncertainty analysis. In the uncertainty analysis, the selection of the filter dimensions is carried out by explicitly accounting for system uncertainty and incorporating confidence margins.

The graphical user interface has been specifically designed to support the complete methodology described above (Figure 6.1), in a seamless and user-friendly way. As mentioned previously, it includes dedicated modules for sensitivity and uncertainty analysis, both of which are essential for model evaluation and design under uncertainty.

Because these analyses can be computationally intensive, especially when large numbers of simulations are required, several components of the workflow have been parallelized to improve efficiency. Tasks such as identifiability analysis, deterministic design, design under uncertainty, sensitivity analysis, and uncertainty analysis are all capable of taking advantage of multi-core processing. As a result, users can obtain results significantly faster, with total computation time depending on the number of processor cores available on their machine.

In the following sections, these GUI modules will be further documented demonstrated through practical examples, and the key improvements over the previous version of the GUI will be highlighted. It is important to note that the purpose of these examples is not to present real-world applications, but rather to illustrate the capabilities and functionality of the new interface.

## 6.4 Illustrative case study

The application of the model is illustrated using a benchmark experimental site located in the Piedmont region of North Carolina. The site that has been previously used to evaluate vegetative filter strip (VFS) performance (Muñoz-Carpena, 1993; Muñoz-Carpena et al., 1999; Shirmohammadi et al., 2006a). The study area includes an agricultural field situated upslope from the planned VFS. The dominant soil type is Cecil clayey, classified as a kaolinitic, thermic Typic Hapludult, with a silty-loam surficial horizon (Parsons et al., 1994). The field includes six runoff plots, each 4 meters wide and 37 meters long, with slope gradients ranging from 5% to 7%. To replicate worst-case erosion scenarios, the field rows were oriented parallel to the slope to maximize surface runoff and soil loss.

Surface runoff was monitored at the lower edge of two control plots with no filters installed. These were assumed to represent baseline runoff conditions for comparison with adjacent plots containing VFS. Two plots were equipped with 4.3-meter-long grass filter strips, while another two had 8.5-meter-long strips, corresponding to field-to-filter area ratios of 9:1 and 4.5:1, respectively. The filters consisted of a mixed stand of fescue, bluegrass, and bermuda grass.

Runoff volumes from all plots were measured using HS-type flumes (0.15 m depth), following the method described by Brakensiek, D.L. et al (1979). Runoff was directed from each plot into flumes via rain gutters and piping systems. Water levels within the flumes were recorded using a float–potentiometer system connected to a half bridge circuit with 2 V excitation. A Campbell Scientific CR10 datalogger was used on-site to record rainfall and runoff data and to control water quality sampling. Rainfall was measured at 5-minute intervals using a tipping bucket rain gauge, while water levels in the flumes were recorded every 30 seconds during storm events. Each plot was equipped with an automatic water quality sampler containing 24 one-liter bottles. These samplers were triggered whenever water levels in the flumes rose or fell by at least 5 mm. The inlets were positioned downstream in a plywood trough. Collected samples were analyzed for sediment concentration and particle size distribution, following Gee & Bauder (1986).

### 6.4.1 Single execution

The most basic functionality of the program is to perform a single execution of the VFSMOD model. That is, given a defined set of system conditions, the user can simulate and evaluate how the system behaves under those conditions. This section presents a simple execution example, including the dialogs available in the interface for setting up and running the model, and the type of results that can be obtained.

As previously mentioned, running VFSMOD requires certain input data, such as the hyetograph, hydrograph and sediment-related data. These data are often difficult to obtain directly from field measurements. To address this, the program provides the option to generate them using the UH module. If those inputs are not available, the user can first execute the UH component to create the required hyetograph and hydrograph.

The following figure (Figure 6.5) shows the interface dialog used to execute the UH model.

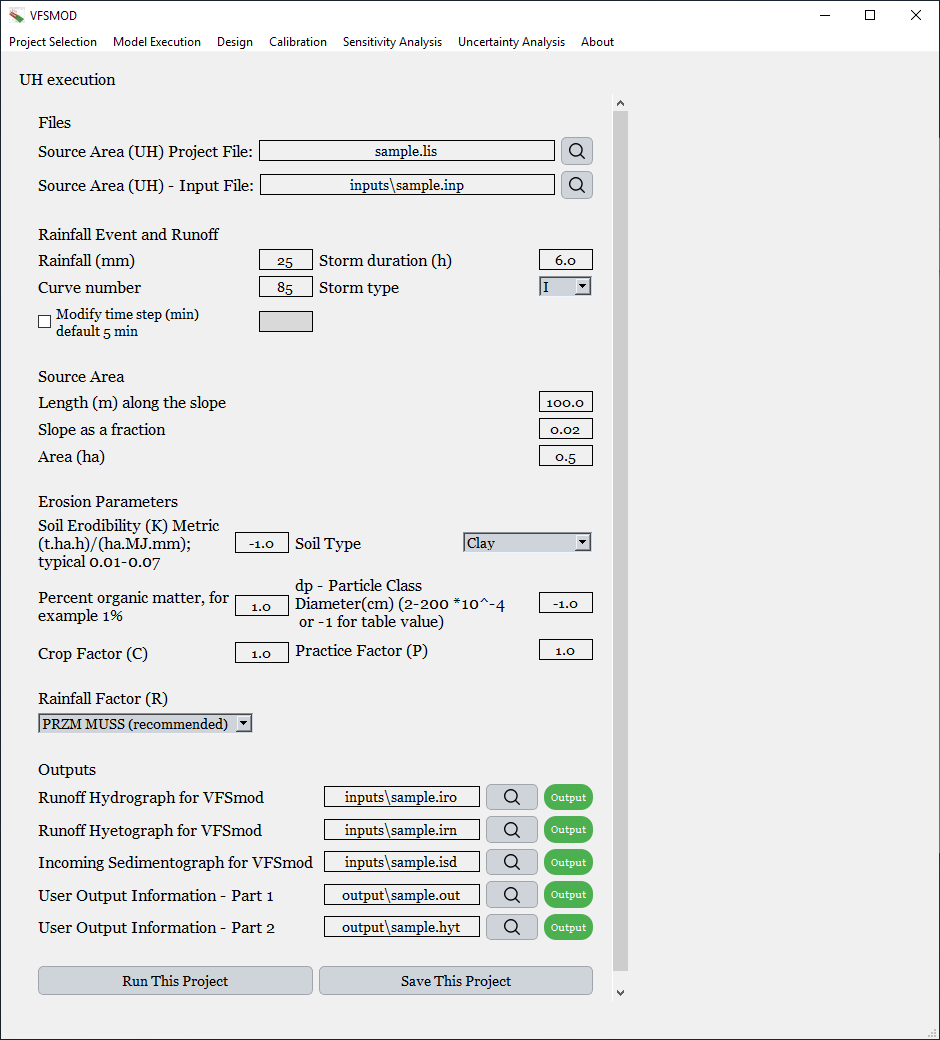


Figure 6.5. **Interface for executing UH.** This window shows the interface used to run UH, where users can select the files containing the necessary information for execution, input the required parameters, and define the output file locations.

As shown in the interface, the top line edits display the file paths required to run UH. These paths are shown in relative format whenever the files are located within the same working directory defined in the *Project Selection* tab, which is also visible in the image above.

The *.lis* file contains the paths to both the input and output files needed for the UH execution. Specifically, it includes the location of the *.inp* file, which stores all the necessary information for the run. Once the user enters values into the dialog, the program inserts them into the .*inp* file. UH is then executed using the .*lis* file, which provides access to the .*inp* file and defines where the resulting outputs will be saved.

The required input parameters include rainfall (mm), curve number, rainfall duration, storm type, slope length (along the slope), slope steepness (as a fraction), and source area (m²). It also includes erosion-related parameters used in MUSLE such as the K, C, and P factors, organic matter content, and soil texture (selectable via a dropdown menu), as well as particle diameter. The user also selects the method for computing the R factor (rainfall erosivity), again via dropdown.

Beneath the input fields, the output file paths generated by UH are listed. These paths are stored in the .*lis* file and are accompanied by status indicators: green when the file or path exists, and gray when it does not, in which case clicking the button has no effect. At the bottom of the dialog, two options are available: *Run This Project*, which saves the current configuration and immediately executes the UH model, and *Save This Project*, which simply stores the settings in the designated files without running the model.

When defining the storm event, particularly in terms of frequency, the user can choose from several predefined storm types provided by the NRCS. However, the interface also allows for full customization: users may manually define the storm by entering the normalized cumulative rainfall distribution over a 24-hour period directly in the dialog. Alternatively, a non-normalized storm can be selected, where the cumulative precipitation is assigned for each absolute time interval rather than being normalized. This enables greater flexibility for representing site-specific or custom-designed rainfall events. Both options can be seen in Figure 6.5.

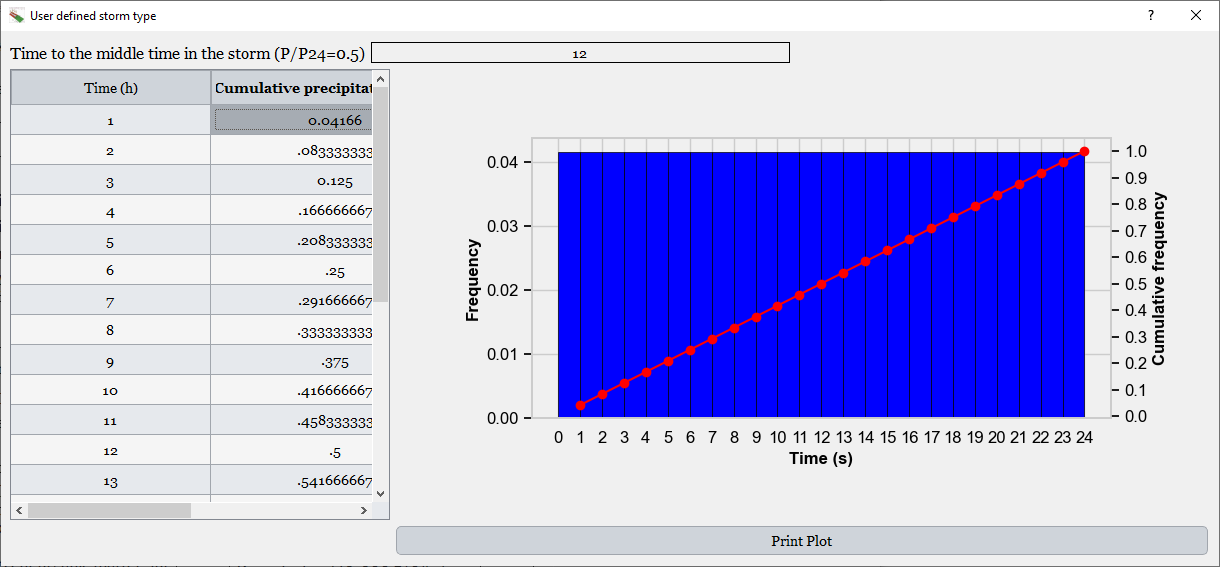


Figure 6.5. **Dialogs for entering the user-defined storm distribution**. In the upper figure, on the left, the table displays the accumulated precipitation data for a normalized 24-hour storm, while on the right, a graph updates automatically as the table values are modified. In the lower figure, the same layout is used to display a non-normalized storm, where the cumulative precipitation is specified for each absolute time interval.

In the previous image, which shows the precipitation frequency graph of the storm, a *Print Plot* button can be seen. This button allows the user to save the graph with higher resolution than a standard screenshot. When clicked, it opens a dialog window like the one shown next.



Figure 6.6. **Dialog showing the properties for saving the graph.** In this case, the user can specify the image resolution in DPI, choose whether to use a transparent background, enable tight bounding box to remove unnecessary whitespace, and define the padding around the image.

In this dialog, users can configure several options to customize the output image. The DPI (dots per inch) can be selected to define the resolution. There is also an option to make the background transparent, if desired. The *Tight Bounding Box* checkbox allows for automatic trimming of excess white space around the plot, ensuring the saved image focuses tightly on the content. Additionally, the *Padding Around the Image* setting controls the amount of space left around the figure's edges. These features are especially useful when preparing figures for publications or presentations.

Once the UH module has been executed, the outputs related to UH are generated. As previously mentioned, UH’s primary purpose is to produce the hydrograph, the hyetograph, and the sediment-related outputs that describe the conditions upstream of the filter.

By clicking the *Output* button shown in Figure 6.5, the program displays the textual content of the output file created by UH. This includes detailed numerical data, such as time series of runoff and sediment flow. An example of this dialog is shown in the next image, where the hydrograph file content is presented.

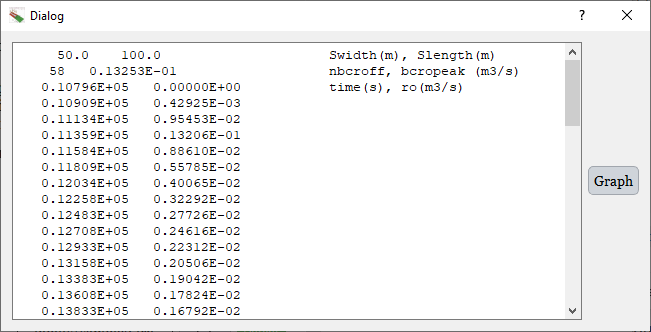


Figure 6.7. **Dialog displaying the contents of the file containing the hydrograph information generated by the UH execution.** The left side shows the raw text data from the file, while on the right, a button labeled *Graph* allows the user to visualize the hydrograph in graphical form.

On the right-hand side of the dialog, there is a *Graph* button. When this is clicked, the hydrograph is displayed graphically. This interactive graph allows users to hover over the plot, and as the cursor moves along the line, it dynamically shows the corresponding X value (time) and Y value (discharge).

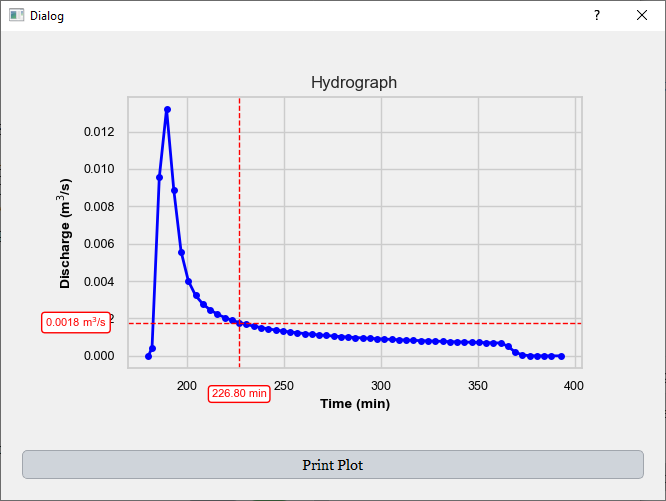


Figure 6.8. **Dialog showing the hydrograph plot.** When hovering over the graph, the X value (time) and Y value (discharge) are displayed interactively. As with other graphs in the Graphical User Interface, there is a "Print Plot" option available to export the figure with customizable settings.

The same functionality is available for the file containing the **hyetograph**. On one side, the content of the file is displayed in plain text, and on the other, it can also be visualized **graphically**, as shown in the next figure. This allows users to better interpret the temporal distribution of rainfall used as input.

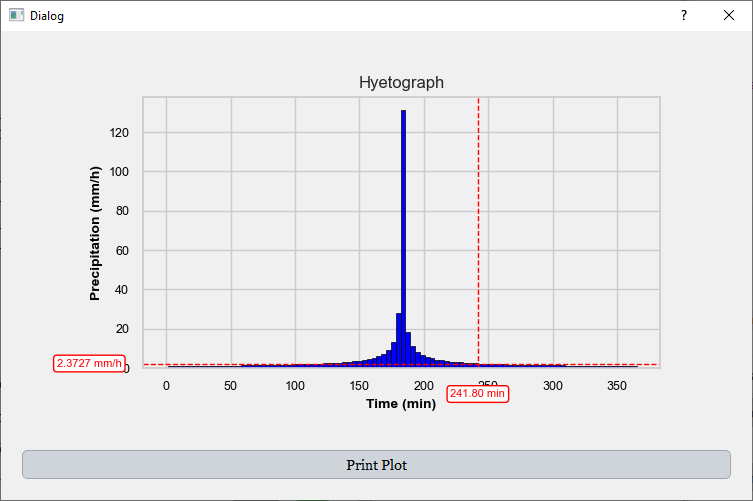


Figure 6.9. **Dialog showing the hyetograph plot.** As with the hydrograph, hovering over the graph displays the X value (time) and the Y value (precipitation). The *Print Plot* option is also available for exporting the figure with customizable resolution and formatting.

In contrast, no graphical representation is provided for sediment-related information, since UH does not generate a time series of sediment transport, that is, it does not produce a sedimentograph. Instead, it only outputs aggregate sediment characteristics, without temporal resolution. Similarly, other outputs generated by UH do not include graphical visualizations.

The next step in evaluating the performance of a vegetative filter strip (VFS) is to run the VFSMOD model itself, now that the upstream flow, hyetograph and sediment inputs have been established using UH. To do this, a separate execution dialog is used, which is shown in the following figure.

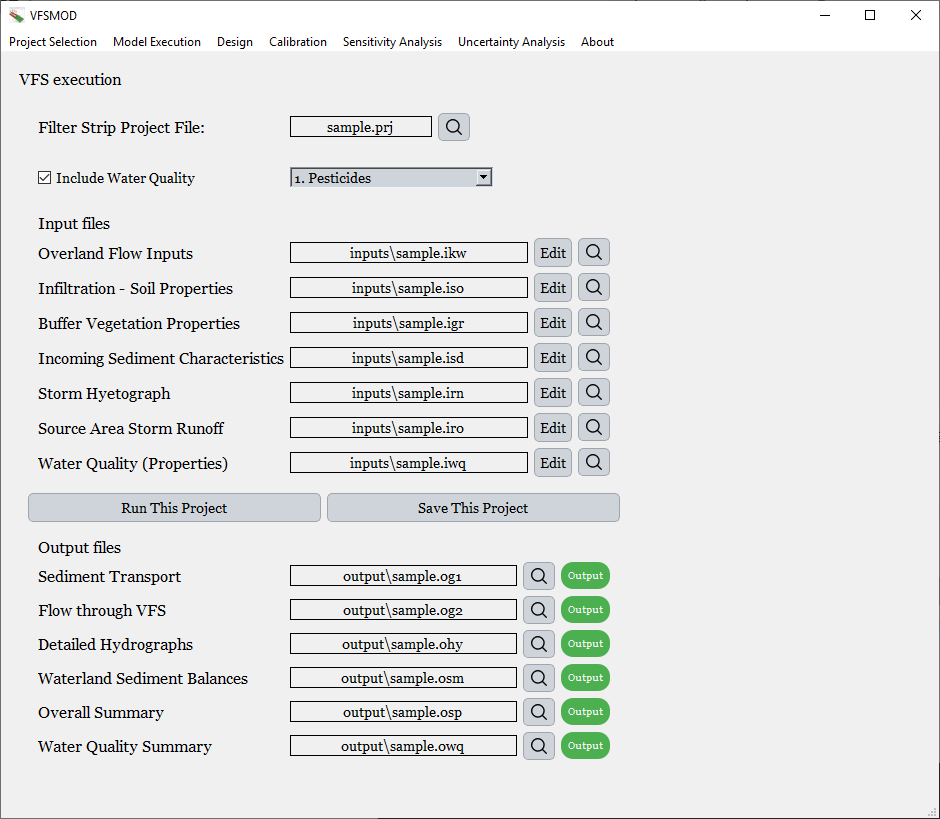


Figure 6.10. **Dialog for executing VFSMOD.** At the top, the interface displays the line edit showing the path to the .*prj* file, which contains the directories of both input and output files. The rest of the dialog presents the specific file paths for the input data required to run VFSMOD and the output files where results will be stored.

In this interface, the .prj file contains all the necessary paths to the input and output files required to execute VFSMOD. As with the .*lis* file used previously for the UH module, the .*prj* file is automatically generated by the program, users do not need to create or edit it manually. A checkbox labeled **Include *Water Quality*** allows users to enable the pesticide simulation module. When selected, additional input fields related to pesticide properties become available. Two options are provided for execution: *R****un This Project***, which saves the current configuration and immediately executes the simulation, and ***Save This Project***, which stores the setup without initiating the run. Output file paths are also displayed in this dialog. If the water quality option is enabled, an additional output path corresponding to the ***Water Quality Summary*** file will appear, as shown in the previous figure.

The input dialogs for VFSMOD are organized by thematic categories. The first section, shown in the following figure, corresponds to the *Overland Flow Inputs*. In this dialog, users can define key parameters such as the length and width of the vegetative filter strip—critical for evaluating its hydraulic and pollutant removal performance. Additionally, more advanced parameters used in the model’s internal calculations are also displayed; however, these are typically pre-set and do not need to be modified by most users.

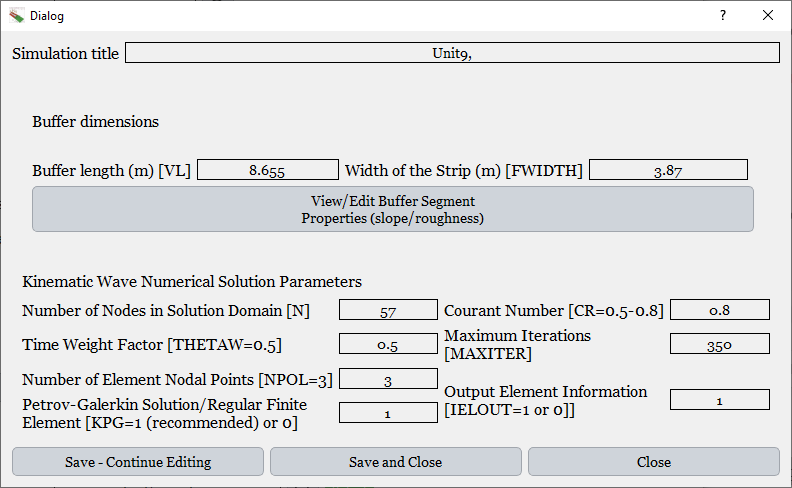


Figure 6.11. **Dialogs displaying the Overland Flow Inputs.** This section allows the user to input key information such as the buffer length and width. Additionally, a button is provided that opens a dialog to define the slope and Manning’s roughness for each segment of the filter strip.

One of the interactive features of this interface is the *View/Edit Buffer Segment Properties (slope/roughness)* option, which opens a new dialog where users can define multiple segments within the buffer. Each segment can be assigned a different length, Manning’s roughness coefficient, and slope. This capability reflects one of the notable characteristics of the VFSMOD model itself, which allows for spatial variability in surface roughness and slope within the filter strip. As the user edits the table of segment parameters, the associated graph is updated in real time to visually reflect the configuration, enhancing clarity and precision in model setup.

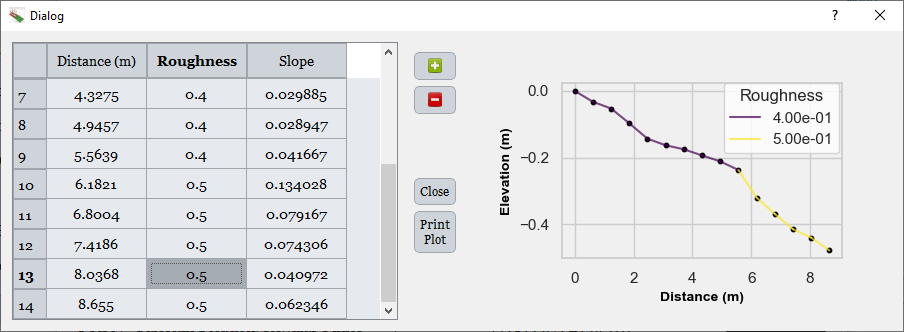


Figure 6.12. **Dialog for editing the segments of the vegetative filter strip.** The user is given the option to modify the characteristics of each segment of the filter, including Manning’s roughness and slope values, directly in the table on the left. On the right, a dynamic graph displays the **elevation profile** of the filter, which updates in real time as the values are edited.

Another available dialog focuses on S***oil Properties and Infiltration***. Within this section, users can input key parameters such as the **vertical saturated hydraulic conductivity**, the **initial water content**, and the **water table depth**, among others. The interface also offers the option to define whether the soil profile consists of a single layer or **multiple layers**. When multiple layers are specified, their properties are aggregated using a **depth-weighted average**, ensuring that deeper layers appropriately influence the overall soil behavior. Additionally, users can choose whether to include a **water table** in the simulation. This dialog also provides options for specifying **soil characteristic curves**, allowing for a more accurate representation of the relationship between water content and soil suction, which is essential for modeling infiltration dynamics.

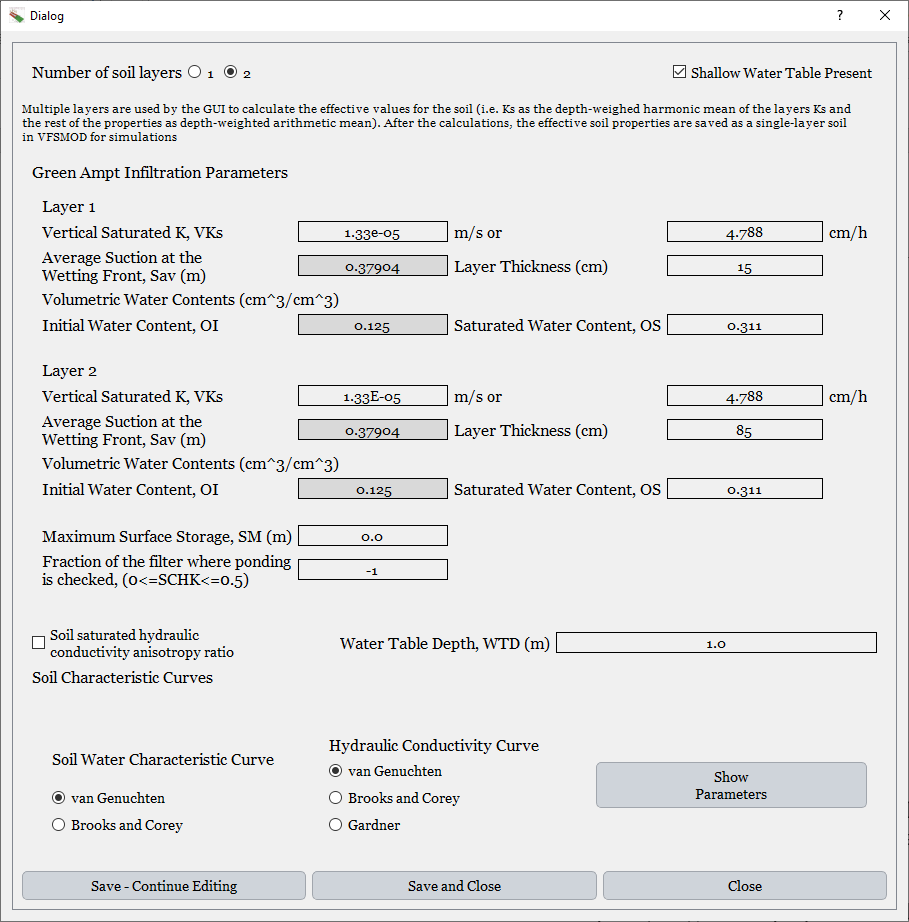


Figure 6.13. **Dialog for entering soil infiltration properties.** This interface allows the user to provide key data related to soil infiltration, such as the **water table depth, vertical saturated hydraulic conductivity**, and **initial water content**.

The next section of the interface allows users to specify vegetation properties and incomingsediment characteristics. For vegetation, parameters include the spacing between grass stems, the surface roughness (used in hydraulic calculations), and the vegetation height, all of which are critical for simulating flow resistance and filtering efficiency.

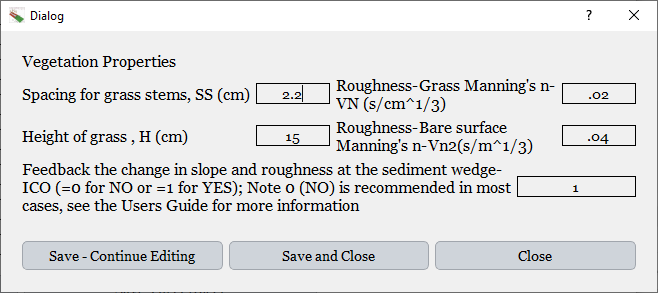


Figure 6.14. **Dialog showing vegetation properties.** This interface allows the user to define key characteristics of the vegetative buffer, such as the **spacing between grass stems**, the **height of the vegetation**, and the **surface roughness** introduced by the vegetation.

Regarding sediment properties, users can input details related to the incoming flow, such as the sediment concentration, the median particle diameter (D₅₀), and the porosity of the sediment. These properties define the behavior of sediment transport and deposition within the vegetative filter strip. It is important to note that the characteristics of the incoming sediment, such as the hydrograph, hyetograph, and sediment concentration, are generated by the UH module, which simulates the source area conditions upstream of the filter.

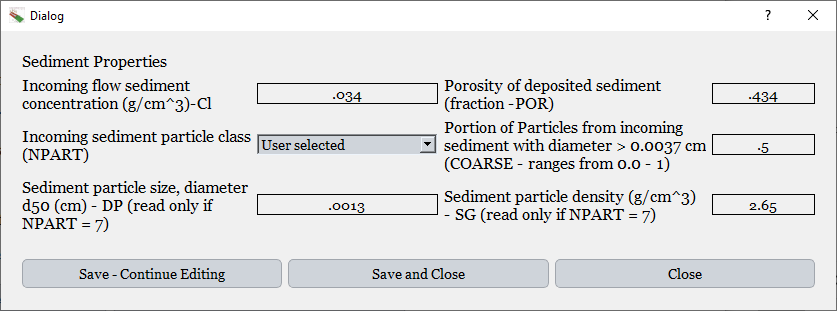


Figure 6.15. **Dialog showing sediment properties.** This interface displays the sediment characteristics used by VFSMOD, which are typically generated by UH during the upstream simulation. However, users may also enter or adjust this information manually if desired. Key parameters include the **incoming flow sediment concentration**, the **particle size (e.g., d50), sediment porosity**, and other related variables.

In the case of the **hyetograph** and **hydrograph**, which are also generated by the UH module, users have the option to manually modify these datasets if necessary. This flexibility allows for the replacement of simulated data with **observed measurements** or any custom-defined values created by the user. As with other components of the interface, the **graphs associated with the hyetograph and hydrograph are dynamically updated** whenever the data tables are modified, ensuring immediate visual feedback.

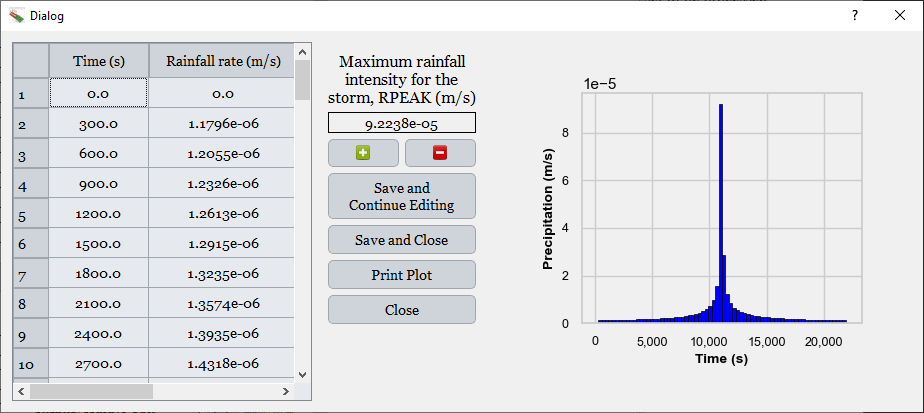


Figure 6.16. **Dialog showing the hyetograph generated by UH. T**he precipitation time series (hyetograph) created by UH is displayed in a tabular format on the left and is dynamically updated in the graph on the right. Although this data is automatically produced by UH, users also have the option to manually input or modify the values if observed or alternative rainfall data is available.

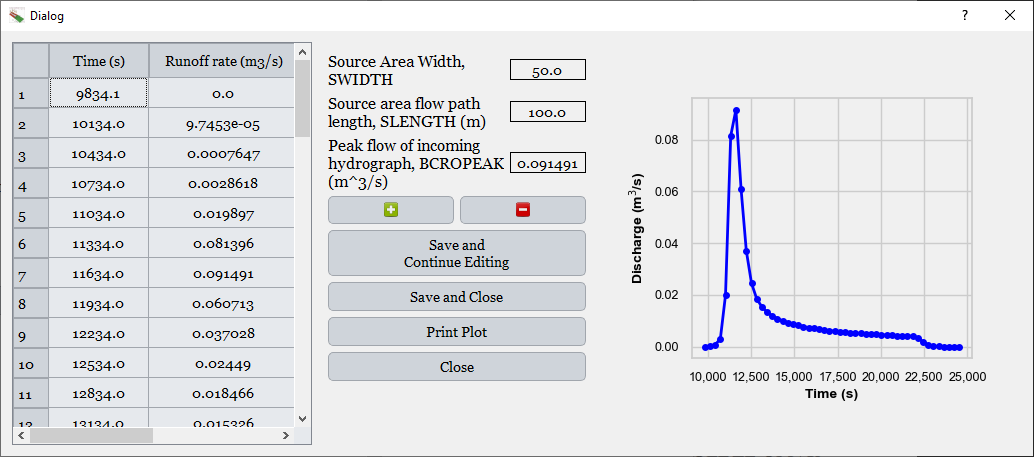


Figure 6.17. **Dialog showing the hydrograph generated by UH.** The hydrograph is displayed with a table on the left containing the time-discharge values, while the right side shows the corresponding graph that updates automatically when the table is edited. This allows users to visualize the discharge dynamics over time for the simulated rainfall event.

Finally, the last input section, optional but essential for water quality analysis, is the *Water Quality* dialog, where users can provide the necessary data related to pesticides. While the previous graphical user interface already supported the inclusion of pesticide data, several significant improvements have been implemented in this new version. In parallel with the development of the updated interface, the VFSMOD engine was enhanced to allow the simulation of multiple pesticides simultaneously. One of the key innovations is the ability to account for pesticide degradation, where a parent compound can degrade into another modeled pesticide. This introduces more complex mass balance dynamics, as degradation can result in both transformation into a known pesticide and complete breakdown.

The interface has been specifically designed to accommodate this complexity, enabling users to define degradation pathways, specifying the target degradation product and the proportions of transformation and total loss. When multiple pesticides are included, the model will generate separate outputs for each compound, such as *Pesticide Delivery Ratio* or *Leachate Depth*, allowing for a detailed evaluation of each substance individually. These individual outputs are fully integrated into the calibration and design modules, so that objectives or constraints can be set for a specific pesticide being modeled, ensuring targeted environmental performance.

Figure 6.22 shows the dialog that appears when selecting the option to enter *Water Quality* inputs. The required input fields are the same as those included in the previous graphical user interface. However, when multiple pesticides are modeled, input data must be provided separately for each compound. The dialog is designed to dynamically adjust when more than one pesticide is selected, expanding to display additional line edits where the corresponding values can be entered. Required inputs include, for instance, the initial concentration of pesticides in the runoff, the highlighted pesticide for reporting purposes, and any remobilized pesticide from previous events.

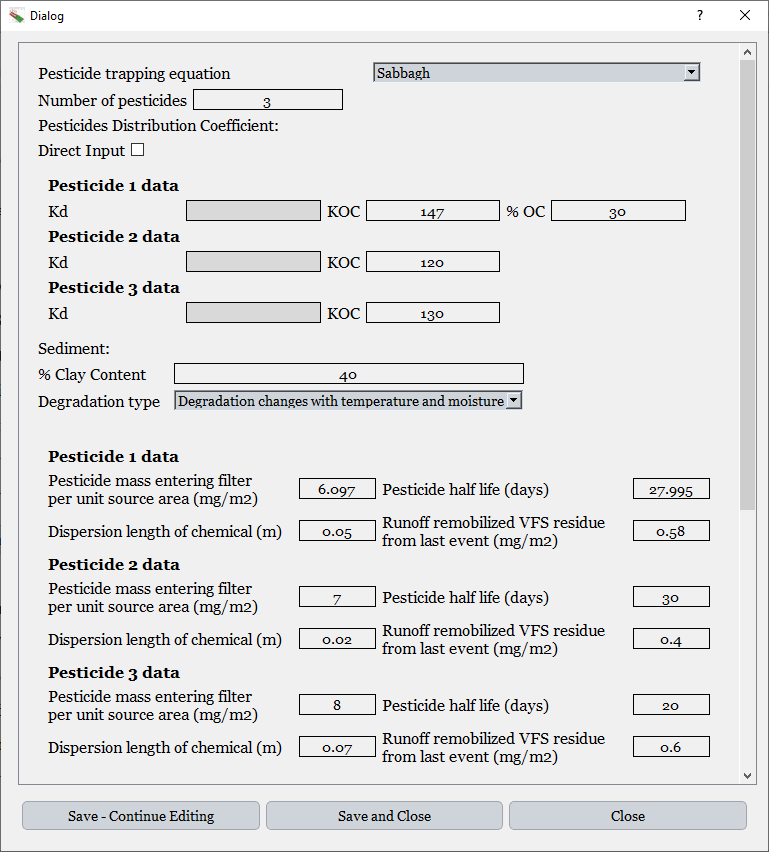
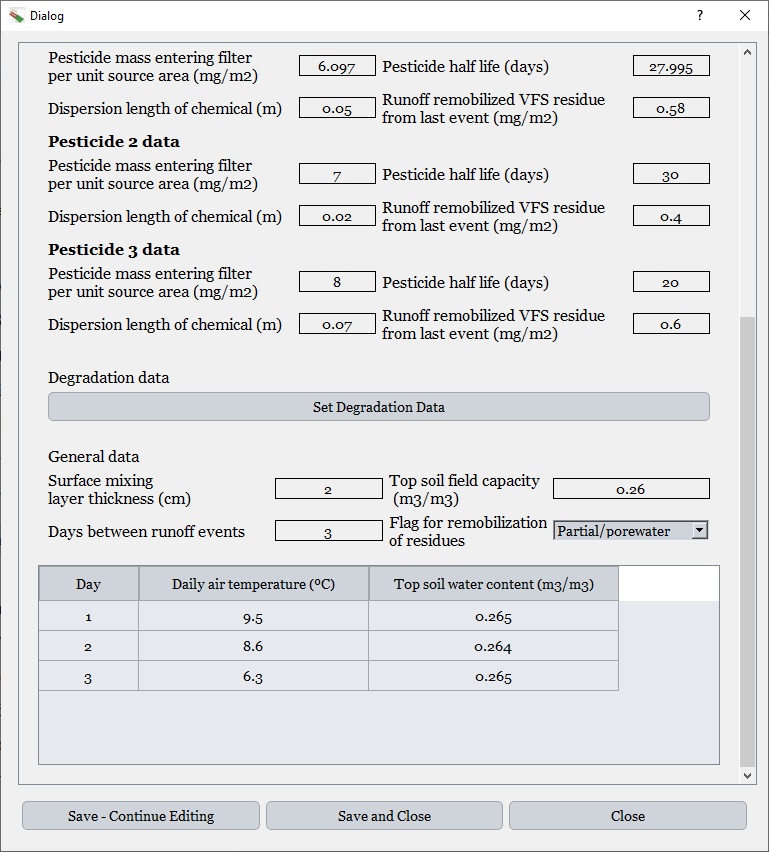
 

Figure 6.18. **Dialog for entering pesticide-related information (*Water Quality* inputs)**. This interface allows the user to input data for each pesticide individually. Key information includes the initial concentration of each pesticide entering the filter, the pesticide half-life, and parameters that influence degradation, such as temperature and moisture conditions between rainfall events (as shown in the lower degradation table). Additionally, the dialog provides a button labeled *Set Degradation Data*, which opens a separate window for defining inter-pesticide degradation relationships.

Figure 6.18 illustrates the case in which pesticides undergo degradation (“Degradation changes with temperature and moisture”). However, an option is also available to simulate scenarios without degradation. If the user selects this option, the parameters “Dispersion length of chemical” and “Pesticide half-life” are automatically disabled in the GUI, indicating that the user does not need to specify these values.

An additional feature is the *Set Degradation Data* button, which opens a window to define the pesticide degradation matrix. In this matrix, users can specify the percentage of a pesticide that degrades into another modeled pesticide, enabling complex degradation chains to be accurately represented. As illustrated in the following figure, when three pesticides are included, it is possible to define, for example, that 50% of pesticide 1 degrades into pesticide 2, and the remaining 50% into pesticide 3. Similarly, pesticide 2 may degrade 70% into pesticide 3, while the remaining 30% is assumed to degrade completely and is removed from the system.

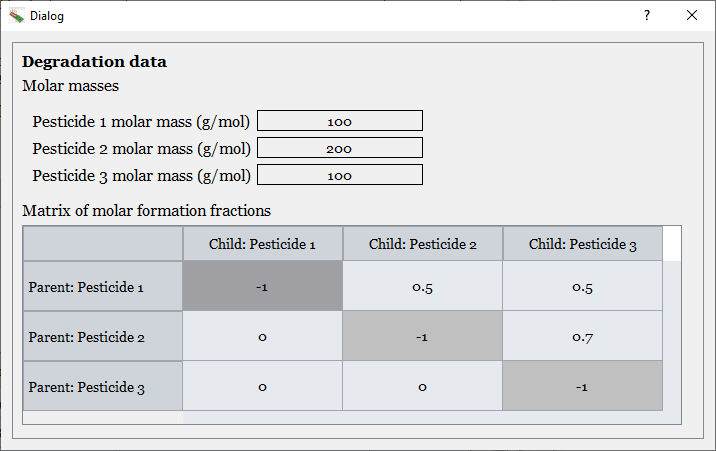


Figure 6.19. **Dialog showing pesticide degradation settings.** The table displays the molecular weight of each pesticide and a degradation matrix indicating the proportion of each compound that degrades into other pesticides.

Once all the input data have been provided, the model execution can be carried out. As with the UH module, for each type of output, the interface offers the option to view the text content of the corresponding output file. Additionally, in some cases, it is also possible to visualize the results through graphs.

Among the available outputs are the *Sediment Transport* results, the outputs related to the flow through the vegetative filter strip, and the *Detailed Hydrographs*, which provide a more comprehensive representation of hydrologic responses. When selecting the option to display graphs for the *Detailed Hydrographs*, the dialog shown below is presented.

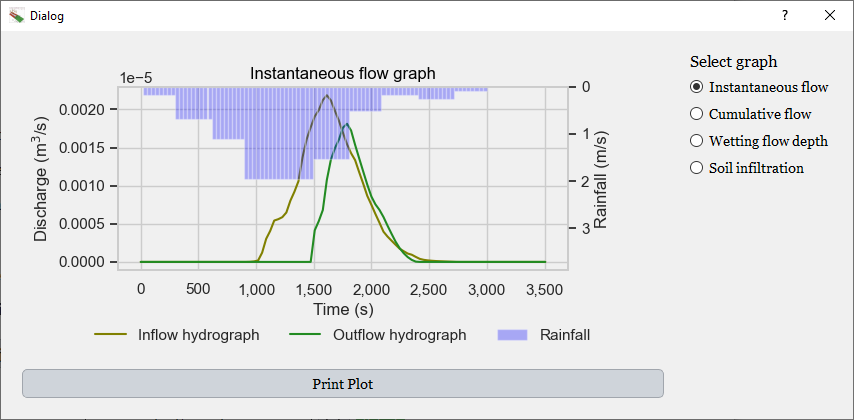


Figure 6.20. **Dialog showing the Detailed Hydrographs output section.** The central area displays the selected graph, while the right panel allows users to choose among different output types. In this case, the selected plot corresponds to Instantaneous Flow, showing both the inflow and outflow hydrographs.

This interface allows the user to choose among four different types of plots, each of which is automatically updated upon selection. The following figures display the remaining graphs available for this output category.

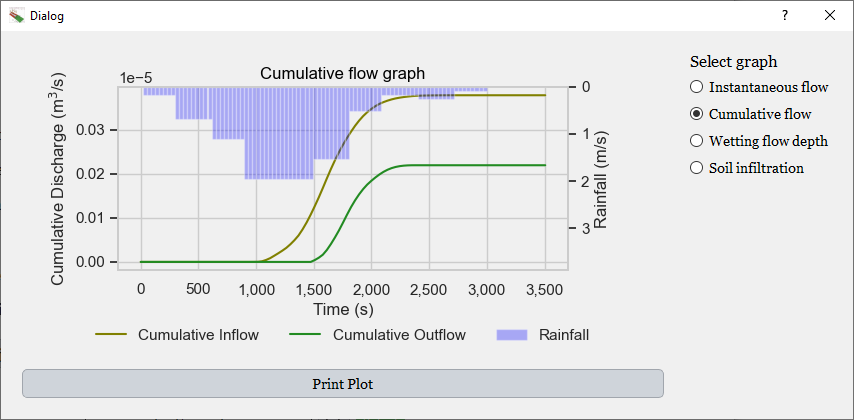


Figure 6.21. **Cumulative flow plot over time.** This graph displays the cumulative inflow and outflow through the filter, illustrating the total volume of water entering and exiting the system during the simulation period.

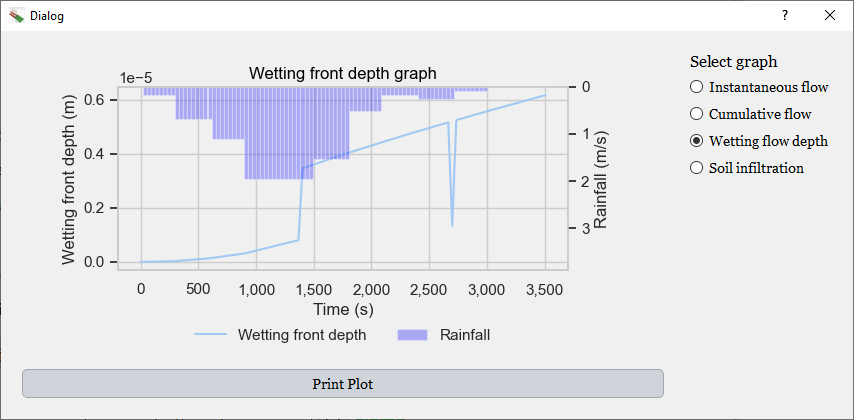


Figure 6.22. **Temporal evolution of wetting front depth.** The plot shows how the wetting front progresses over time in the vegetative filter strip, reflecting the depth reached by infiltrated water.

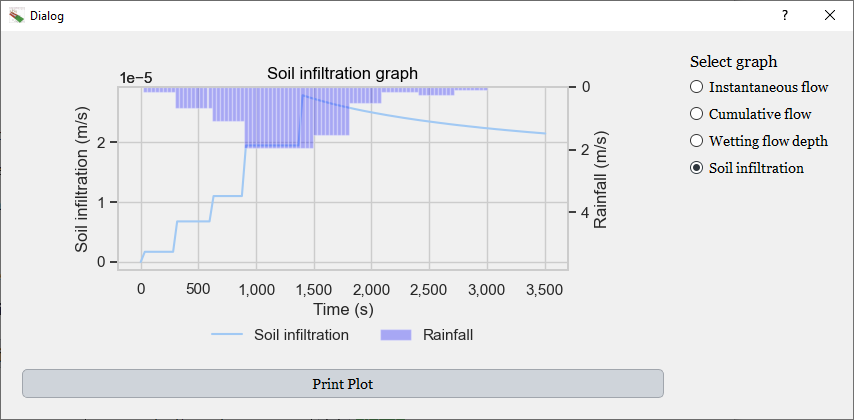


Figure 6.23. **Soil infiltration over time.** This figure presents the evolution of infiltration throughout the simulation period.

The model also generates results for ***Water and Sediment Balances*** and for the ***Overall Summary***. The ***Overall Summary*** provides key aggregated information regarding the performance of the filter and the input conditions specified by the user. In this case, the interface also includes the option to generate graphical representations to facilitate the interpretation of the system's behavior.



Figure 6.24. **Table showing general information on filter performance and input conditions.** This summary includes key indicators of the vegetative filter's behavior, as well as the main input values used for running the simulation.

These graphs illustrate the water balance by displaying the rainfall input, the runoff entering the filter, the infiltration, and the runoff exiting the filter. Similarly, sediment dynamics are visualized through charts that show the sediment load entering the system, the amount retained within the filter, and the sediment exiting downstream. Examples of these plots are shown in the following figures.

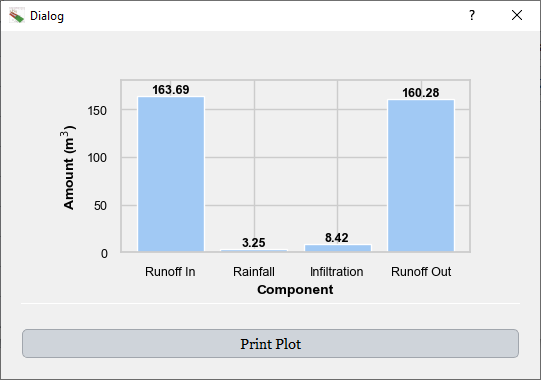


Figure 6.25. **Dialog showing surface water balance dynamics.** The chart displays the dynamics of water throughout the event, including runoff entering the filter, rainfall over the filter area, infiltration within the filter, and runoff exiting the filter.

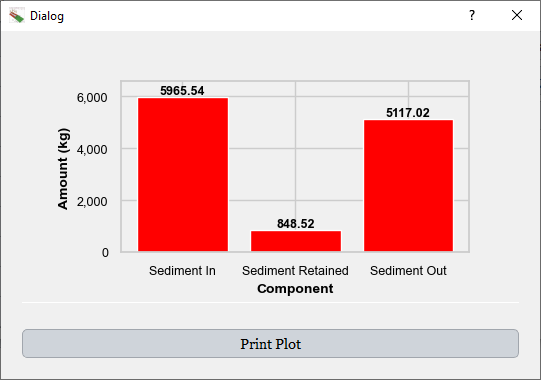


Figure 6.26. **Dialog showing sediment balance.** This chart presents the total sediment load entering the filter, the amount retained by the filter, and the sediment that exits the system.

Finally, if the *Water Quality* option was selected and the project was run, the user will have access to the *Water Quality Outputs*. By clicking the corresponding button, the interface displays a dialog showing the contents of the file containing the pesticide-related output data, along with graphical options for visualizing the results, as illustrated in the following figure.

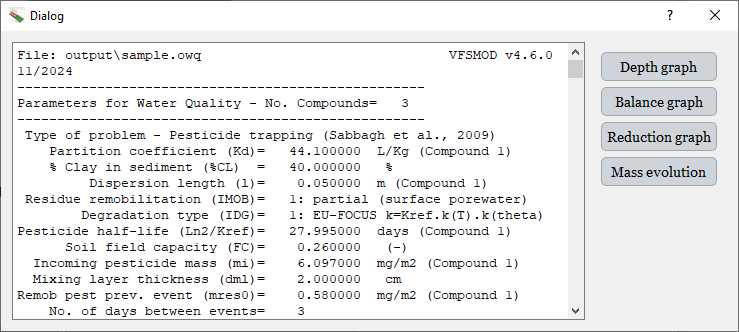


Figure 6.27. **Dialog displaying Water Quality output data.** On the left, the contents of the file containing the pesticide-related outputs are shown. On the right, buttons provide access to the different graphical visualizations available for this output.

Starting with the graphical representations, the *Depth graph* shows the distribution of pesticide concentration in the filter at various depths. It includes the pore water concentration, as well as the ratio between the solid and liquid phases. Users can select which pesticide to visualize, as each compound generates different results depending on its specific behavior in the filter.

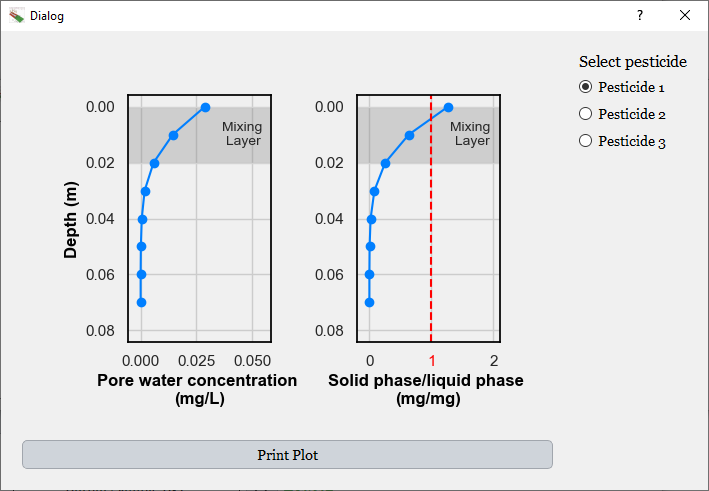


Figure 6.28. **Graph showing the evolution of *Pore water concentration* and the *solid-to-liquid phase ratio* across soil depth.** This information is displayed for each pesticide individually.

The *Balance graph* displays the final fate of each pesticide. It distinguishes between pesticides that have exited the filter (e.g., leached or transported in solution or sorbed to sediments) and those that remain within the system. For those retained, the graph categorizes them based on their location: sorbed in the mixing layer from the last event, sorbed or dissolved in the current mixing layer, or trapped with the sediment.

Another pie chart shows the distribution of the pesticide mass after degradation, based on the number of days between events defined by the user in the input dialog. The interface accounts for degradation processes and separates the remaining mass into two categories: dissolved surface residue after degradation and sorbed residue after degradation.

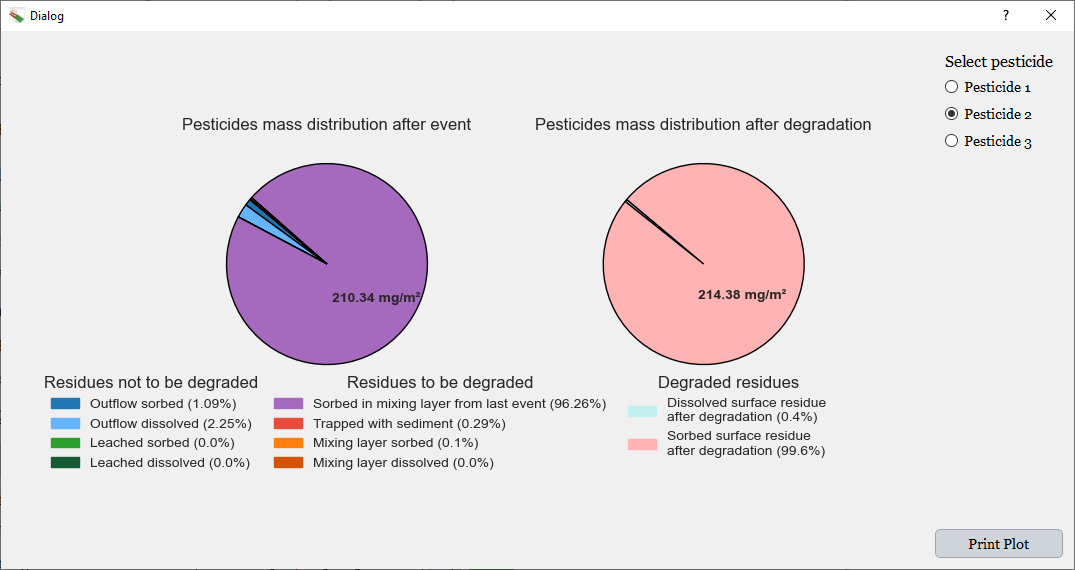


Figure 6.29. **Pesticide mass available after the event and after degradation.** In the case of the pesticides available after the event, there will be residues that remain undegraded and others that will undergo degradation. The degraded fraction results in transformed pesticide masses, which are shown in the pie chart on the right.

The *Reduction graph* displays the percentage reduction of various pollutants, including runoff, sediment, and pesticide concentrations. It quantifies the system's efficiency in reducing contaminant transport, as illustrated in the following figure.

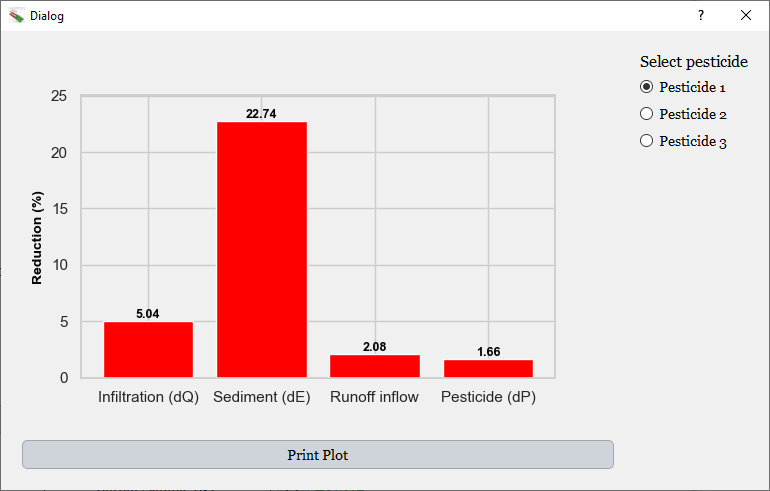


Figure 6.30. **General output summary for reductions in infiltration, sediment, runoff, and pesticide transport in the filter.** This dialogue displays the overall reductions achieved by the filter in terms of infiltration, sediment retention, runoff reduction, and pesticide transport mitigation.

The *Mass evolution* graph shows how the total mass of each pesticide evolves over time. For example, in the case of Pesticide 1, a steady decrease is typically observed due to degradation and transport processes. However, Pesticide 2 may exhibit an increasing trend. This occurs when a portion of Pesticide 1 degrades into Pesticide 2, as specified in the degradation matrix provided in the input dialog. If the rate at which Pesticide 1 transforms into Pesticide 2 exceeds the degradation rate of Pesticide 2 itself, the total mass of Pesticide 2 can increase over time.

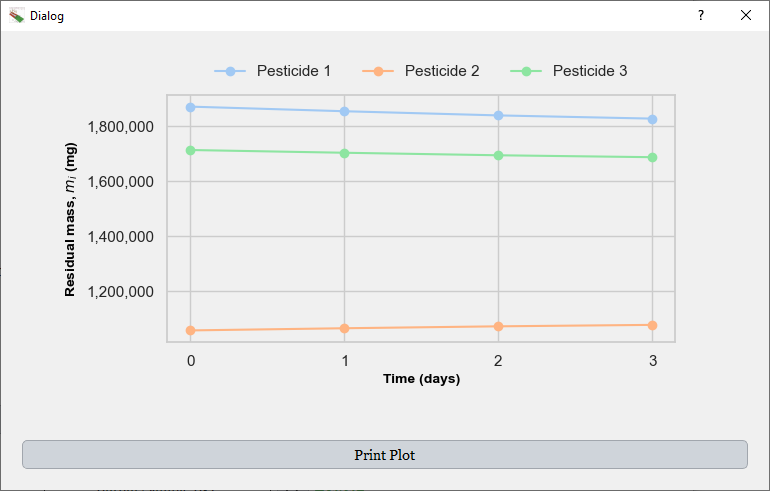


Figure 6.31. **Evolution of pesticide mass over time due to degradation processes.** In this example, the mass of Pesticide 2 increases as a result of the degradation of Pesticide 1, which partially transforms into Pesticide 2. For the remaining pesticides, the mass decreases over time due to degradation.

### 6.4.2 Calibration

When designing Vegetative Filter Strips (VFS), the modeler is faced with the challenge of identifying appropriate parameter values for a specific application. As previously discussed, this is typically addressed through calibration using observed data, which ensures that the model accurately reflects the real behavior of the system under given conditions.

To avoid the subjectivity and time-consuming nature of manual calibration, inverse optimization is used as a robust and objective alternative. This method involves iteratively adjusting model parameters to minimize the difference between simulated and observed outputs, based on a defined objective function.

For calibration to be reliable, it is widely recommended to combine different types of evaluation criteria. Specifically, model performance should be assessed using a **dimensional error metric,** a **dimensionless (normalized) goodness-of-fit indicator,** and **graphical representations** that allow visual inspection of the fit (Ritter & Muñoz-Carpena, 2013). This multi-criteria approach provides a more complete and robust evaluation than relying on a single metric.

In the calibration tools implemented in the new graphical user interface, the **Root Mean Square Error (RMSE)** is used as the dimensional indicator, providing an absolute measure of the average error between simulated and observed values. The **Coefficient of Efficiency (Ceff),** also known as the **Nash–Sutcliffe Efficiency (NSE),** is used as the normalized performance indicator. It has been widely applied to evaluate the performance of hydrologic models, as it compares the variance about the 1:1 line (representing perfect agreement) to the variance of the observed data (Legates & McCabe, 1999). Ceff values range from minus infinity to 1, where Ceff = 1 implies that the predicted values exactly match the observed ones. A value of 0 indicates that the model is only as accurate as the mean of the observations, while negative values suggest that the model performs worse than the observed average.

In addition to these numerical indicators, the GUI includes graphical outputs that compare observed and simulated values, providing users with a visual means to assess the quality of the calibration. Furthermore, the interface integrates **bootstrap resampling techniques** to estimate the statistical significance of the calibration metrics. By generating a distribution of Ceff values through bootstrapping, users can assess whether the performance exceeds predefined confidence thresholds, further reducing subjectivity in the model evaluation process.

The entire methodology for evaluating calibration performance implemented in the GUI follows the approach proposed by Ritter & Muñoz-Carpena (2013), which combines absolute and normalized statistical indicators, graphical interpretation, and statistical significance testing to ensure a comprehensive and objective evaluation.

All of these elements—objective optimization, multiple performance metrics, graphical tools, and significance testing—are integrated into the calibration module of the new interface, offering a complete and statistically grounded framework for parameter estimation in VFS design. The integration of bootstrap resampling and threshold-based significance testing was not available in the previous version of the GUI. Their inclusion in the new interface represents a major step forward, as they address a critical aspect of model evaluation that was previously overlooked, enhancing both the objectivity and statistical rigor of the calibration process.

As previously mentioned, the program allows for separate calibration of runoff, sediment, and pesticide components. Since both sediment and pesticide transport depend on overland flow, calibration typically begins with the optimization of runoff parameters. This is the approach followed in the example presented here.

The graphical user interface enables the user to specify the name of the project containing the system description, as well as the files that include the observed data needed for calibration. In the case of runoff, this corresponds to the hydrograph file; for sediment, the corresponding file is the sedimentograph.

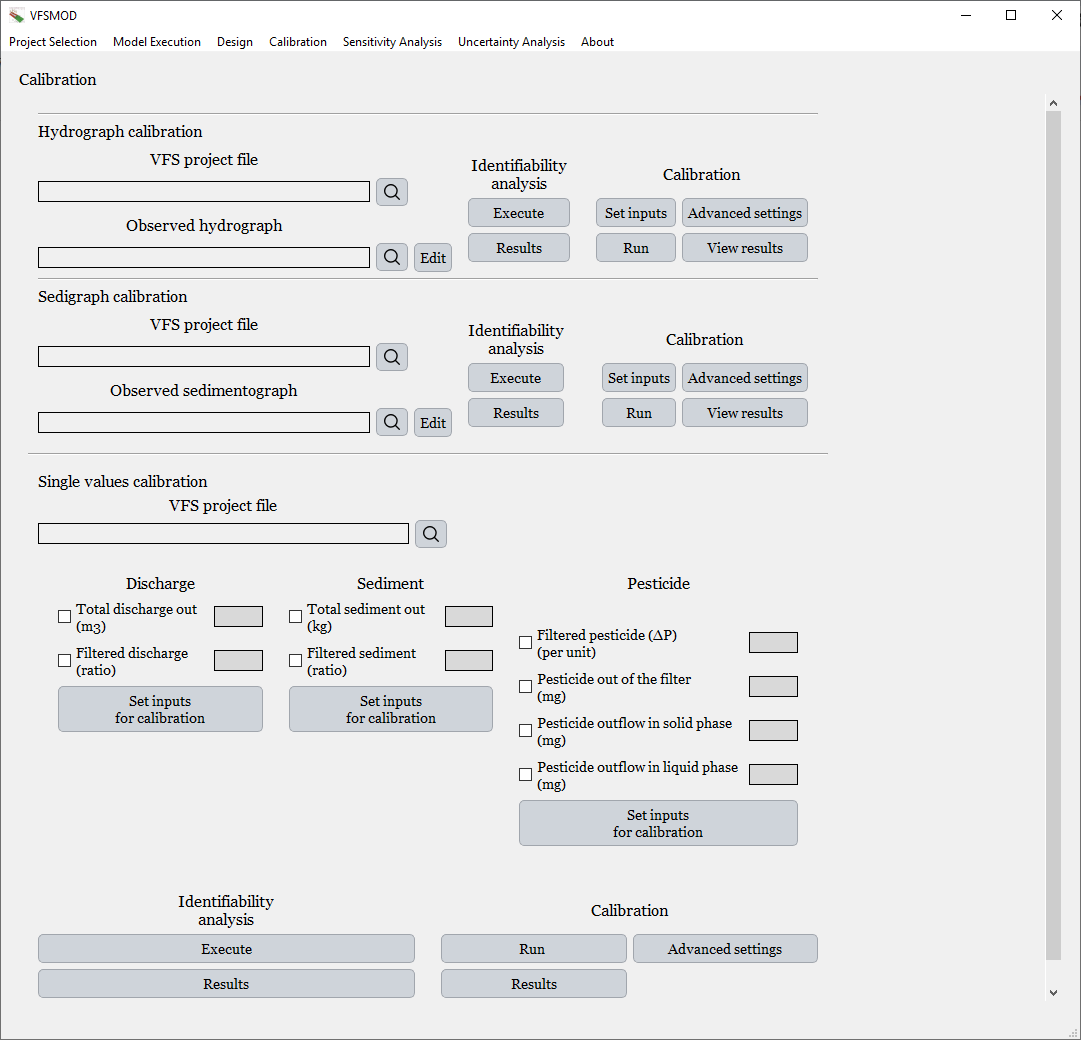


Figure 6.32. **General calibration dialogue.** At the top, the interface provides tools for calibrating the hydrograph, including identifiability analysis and parameter calibration. Below, it displays the section for calibrating the sedimentograph, and finally, the section for single-value calibration, which allows users to calibrate using individual values, including pesticide-related outputs.

In addition, the user can define the **parameter space** for each calibration, meaning the acceptable range of values over which each model parameter will be varied during the inverse optimization process. This setup ensures that the calibration remains within physically realistic bounds and provides a structured basis for model evaluation.

For this calibration, a rainfall event totaling 33.03 mm was selected. The storm corresponds to the Piedmont location, as previously mentioned, and the observed data were obtained from an experimental study published by Muñoz-Carpena (1993). The available observations included the inflow hydrograph, the outflow hydrograph, and the outflow sedimentograph, representing the flow and sediment exiting the vegetative filter strip.

Figure 6.37 shows the rainfall distribution associated with this event. The precipitation profile was generated using the tools provided within the graphical user interface, which allow users to visualize and validate storm inputs prior to running simulations or calibration procedures.

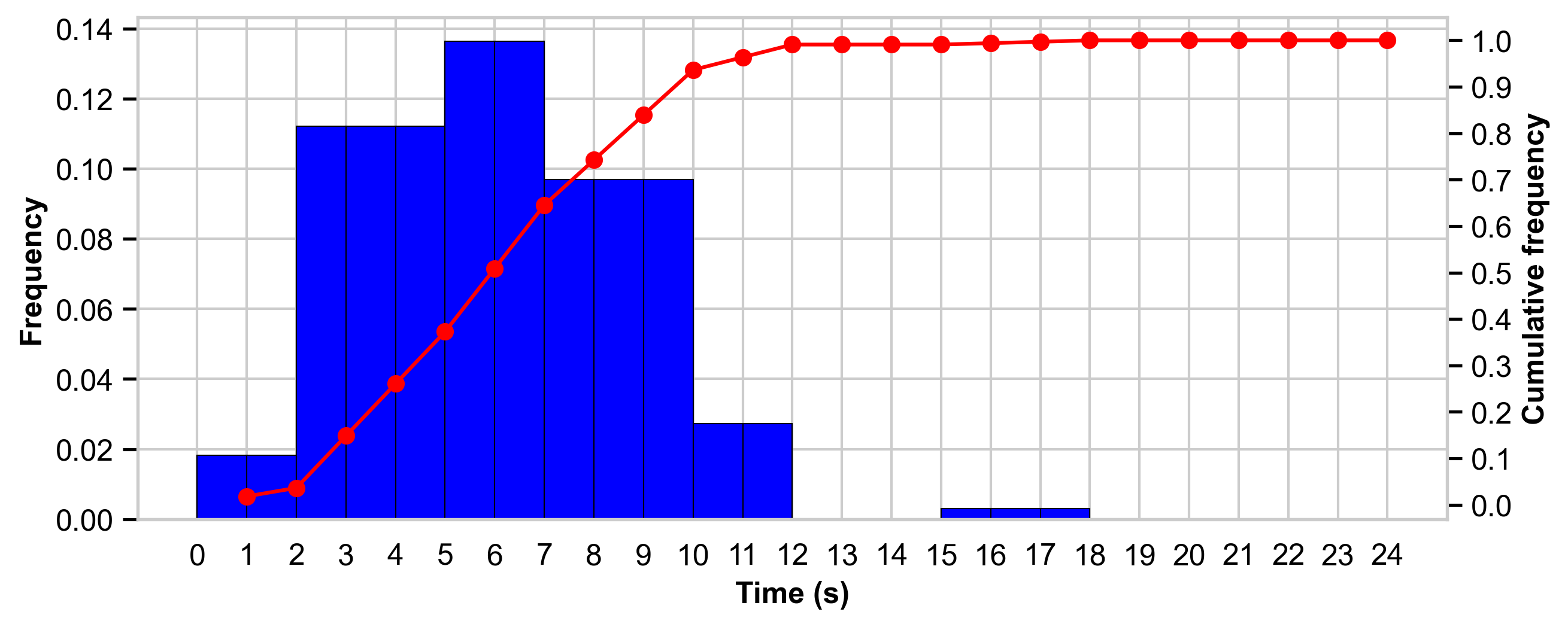


Figure 6.33. **Normalized 24-hour rainfall distribution used for calibration.** This precipitation profile represents the storm event applied during model calibration, distributed over a normalized 24-hour period.

Figure 6.37 displays the normalized precipitation distribution used for the simulation. Specifically, it shows the relative frequency of rainfall, expressed as the percentage of total precipitation occurring in each hour over a hypothetical 24-hour period. This standardized representation is how storm data are provided to the model. Although the actual storm had a duration of approximately 0.75 hours, the interface transforms it into a normalized 24-hour format to define the rainfall pattern, which is then scaled to match the real event duration internally.

In addition to the rainfall data, observational records were also available for the inflow and outflow hydrographs of the vegetative filter strip. These are shown in Figure 6.38, where the dynamic response of the system to the rainfall event can be clearly observed.

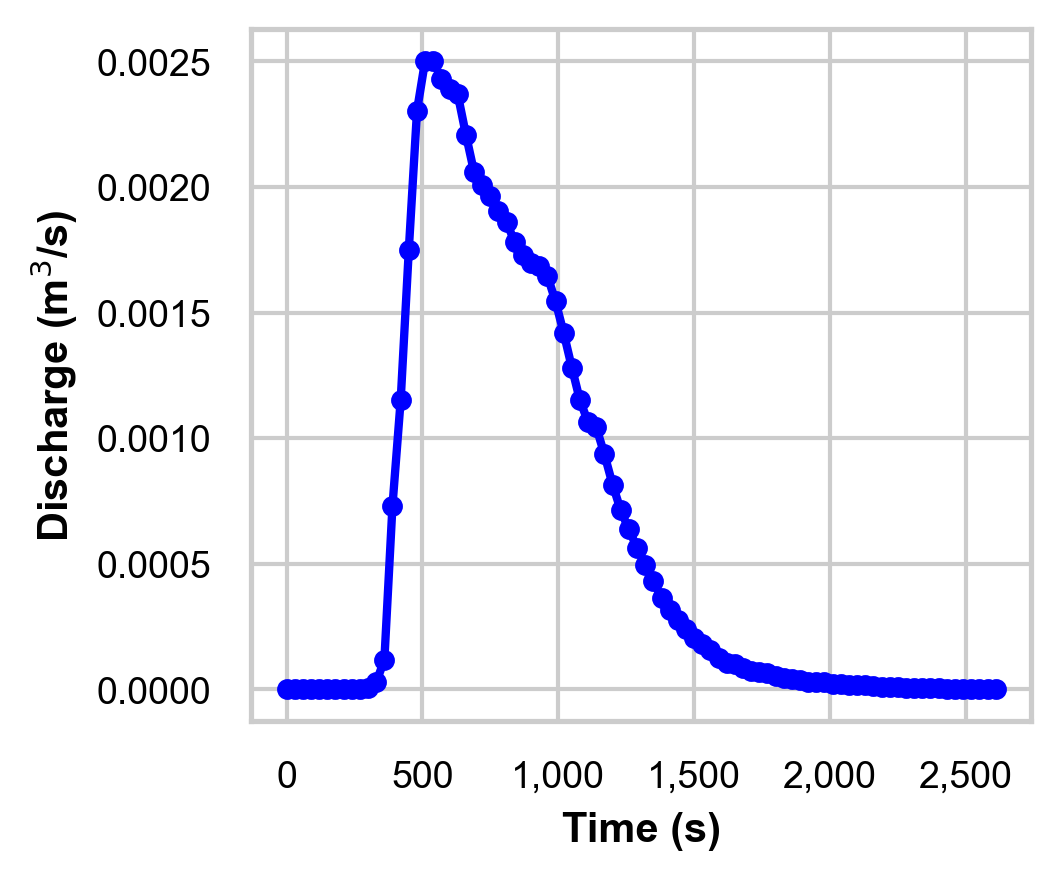
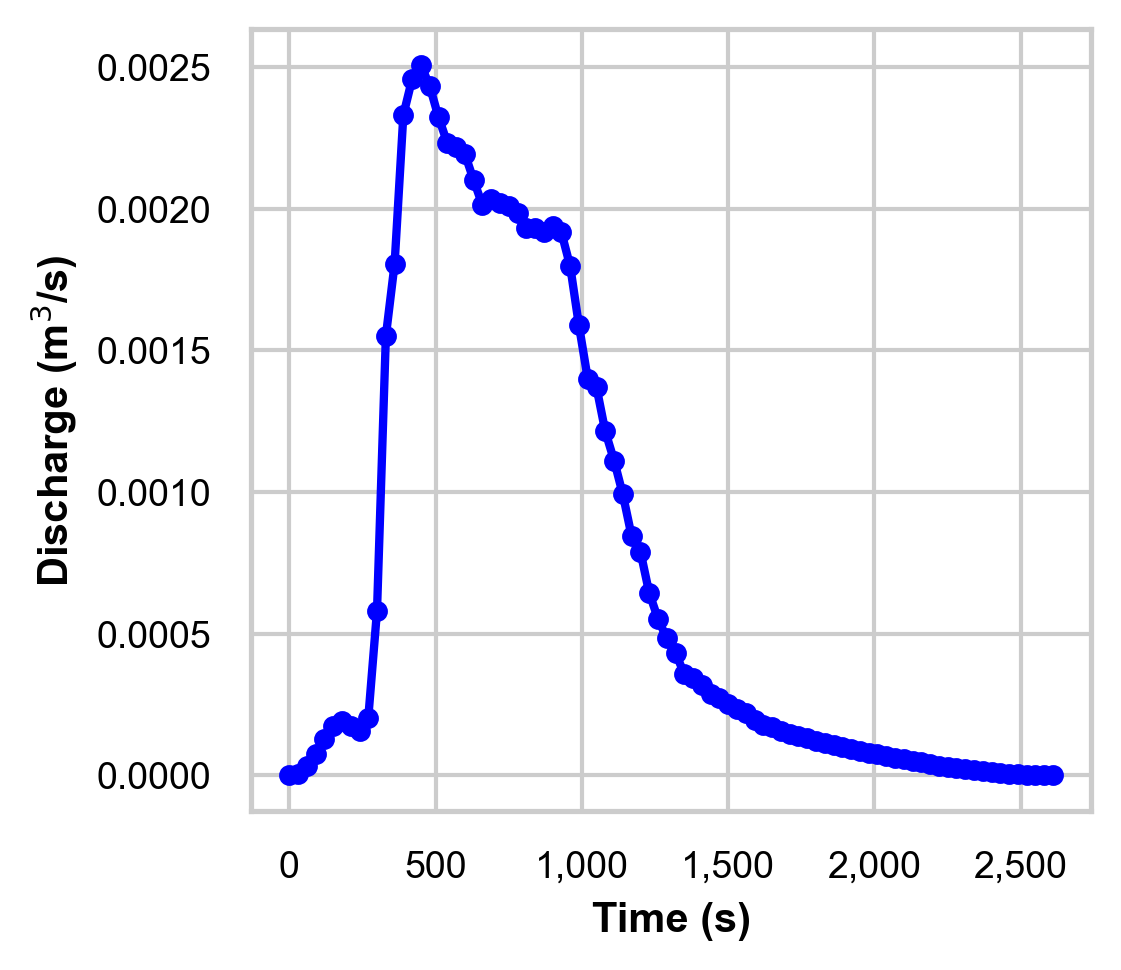


Figure 6.34. **Inflow and outflow hydrographs used in the calibration process.** The hydrograph on the left represents the inflow to the filter and is an input to the model, while the hydrograph on the right shows the outflow, which is the one used for calibration. Both graphs were generated using the Graphical User Interface.

In the calibration process, the outflow hydrograph is used as the observed data to which the model output is compared. The objective is for the simulated hydrograph to closely match the observed outflow hydrograph from the filter, minimizing the difference between them. The inflow hydrograph, by contrast, serves as a required input to the model, representing the amount of water entering the vegetative filter strip over time.

As mentioned earlier, inflow hydrographs are typically generated using the UH module when observational data are not available. However, in this case, observed data were available for the inflow, and therefore UH was not used. This reduces the overall uncertainty in the simulation, since the inflow hydrograph is known and fixed, allowing the calibration process to focus exclusively on the parameters that govern the transformation of inflow into outflow through the filter. In other words, the only source of variation between the inflow and outflow hydrographs is attributed to the filter’s characteristics, not to upstream variability.

The parameters selected for this calibration, along with their minimum and maximum values, were based on the study by Shirmohammadi et al. (2006), which analyzed the same Piedmont location used in this work. As the environmental and site-specific conditions are consistent with those examined in that study, the parameter ranges adopted here are considered appropriate for representing the hydraulic behavior of the filter strip at this site.

Table 6.1. **Parameters used in the calibration process and their minimum and maximum values.** The table displays the lower and upper bounds defined for each parameter during the calibration.

|  |  |  |
| --- | --- | --- |
|  | **Minimum value** | **Maximum value** |
| Vertical Saturated K (m/s) | 0.0000005 | 0.0005 |
| Initial water content (m3/m3) | 0.05 | 0.25 |

Once the parameters to be calibrated and their respective ranges have been defined, following the structure shown in Figure 6.1, the next step may involve performing an identifiability analysis. As previously discussed, this step can help prevent issues related to equifinality and also reduce the risk of convergence failures during optimization, particularly when dealing with a large number of parameters. Identifiability analysis is used to determine which input parameters have the greatest influence on the model's goodness-of-fit metric, thereby guiding the selection of parameters to be optimized.

In the present case, only two parameters are involved in the calibration, making the identifiability analysis unnecessary from a practical standpoint. However, for demonstration purposes, an identifiability analysis is carried out.

The graphical user interface includes the option to perform this analysis using the Morris screening method, which is computationally efficient and requires relatively few model runs. The analysis is conducted using the same input distributions that will later be applied during the calibration process, ensuring consistency between the identifiability results and the subsequent optimization. Moreover, this process has been parallelized within the interface to significantly reduce execution time.

The output of the identifiability analysis consists of two key metrics: the **mean of the elementary effects**, and the **standard deviation of the elementary effects** for each parameter. The mean reflects the overall influence of a parameter on the calibration metric (i.e., its identifiability), while the standard deviation indicates the presence of interaction effects or non-linearities.

Figure 6.39 displays the results of the identifiability analysis, as generated by the graphical user interface.

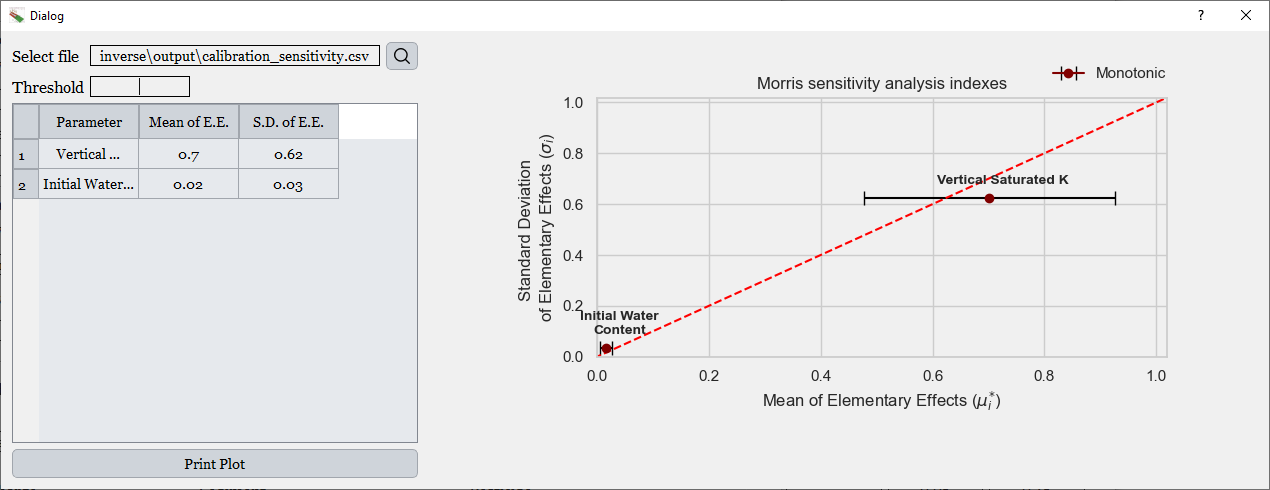


Figure 6.35. **Dialog displaying the results of the identifiability analysis.** At the top left, a line edit shows the path to the file containing the identifiability analysis data. Below it, another line edit allows the user to set a threshold to define which parameters have a Mean of Elementary Effects above that value. A table displays both the Mean of Elementary Effects and the Standard Deviation of Elementary Effects. On the right side, a Morris plot visualizes these values graphically.

In this example, the results of the identifiability analysis show that the Vertical SaturatedHydraulic Conductivity has a significantly greater influence on the calibration metric than the Initial Water Content. This indicates that the Vertical SaturatedHydraulic Conductivity is more identifiable under the given conditions, that is, changes in this parameter lead to more pronounced variations in the model’s goodness-of-fit. In contrast, the Initial Water Content exhibits a much lower influence on the fit, suggesting that it is less informative for calibration in this case.

If more input parameters were involved, the interface would allow users to apply a threshold to the Mean of the Elementary Effects (i.e., the Morris μi\* value), automatically selecting only those parameters whose influence exceeds a user-defined cutoff. This thresholding tool helps streamline the calibration process by focusing only on the most identifiable parameters. Even though the Initial Water Content has a negligible effect in this example, it is still included in the calibration workflow here for demonstration purposes.

Once the parameters to be calibrated and their distributions have been defined (as shown in the previous table), the calibration process can begin. The graphical user interface allows users to choose from three available objective functions: the Root Mean Square Error (RMSE), the Nash–Sutcliffe Efficiency (NSE), and the Normalized Nash–Sutcliffe Efficiency (NNSE).

The NNSE is defined as:

and can be useful in certain contexts because it transforms the NSE scale into a metric that is bounded between 0 and 1 for NSE values above zero, enhancing interpretability and comparability across different modeling applications. This transformation also reduces the exaggerated penalization of moderately poor fits, making it especially relevant when dealing with noisy environmental data.

In addition to selecting the objective function, users can define a convergence tolerance, which determines the sensitivity of the optimization algorithm to changes in the objective function. A smaller tolerance value requires the algorithm to search more precisely before declaring convergence, potentially improving accuracy at the cost of longer computation time. The maximum number of iterations can also be specified, placing an upper limit on how long the inverse optimization is allowed to run.

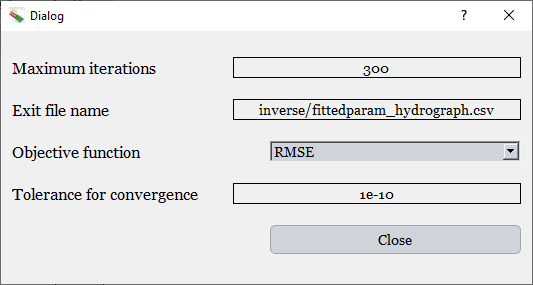


Figure 6.36. **Dialog for selecting Advanced Settings for calibration.** This interface allows the user to define parameters such as the maximum number of iterations, the path to the file that will store the calibration results, the objective function, which can be either RMSE, NSE or NNSE and the convergence tolerance.

In this case, the **Root Mean Square Error (RMSE)** was selected as the objective function, with a convergence tolerance of 1×10⁻¹⁰ and a maximum of 300 iterations, as shown in the previous figure. It is important to note that the objective function is used to guide the optimization process during calibration: it quantifies the discrepancy between simulated and observed values, and the algorithm works to minimize this discrepancy by adjusting the model inputs. Although a single goodness-of-fit metric is used as the objective function, for example, RMSE, **the evaluation of the calibration is not based solely on this metric**. Once the calibration is completed, the interface provides **a broader set of performance indicators**, both dimensional and dimensionless, to allow for a more comprehensive assessment of model fit.

Although many optimization algorithms exist, each with their own advantages and disadvantages, the previous version of the VFSMOD graphical interface relied on the Global Multilevel Coordinate Search/Nelder-Mead method. However, this approach was implemented in MATLAB, which required users to install large and heavy MATLAB libraries with several gigabytes in size. In contrast, the new interface is fully developed in Python, allowing for the integration of optimization methods that are native to Python environments.

For this reason, the **Differential Evolution (DE)** algorithm was selected in the new interface. DE offers several advantages: it is easy to implement, well-suited for non-linear and non-differentiable objective functions, and it performs well in high-dimensional search spaces. Additionally, DE is resistant to becoming trapped in local minima and can handle noisy or discontinuous objective functions, features that make it particularly appropriate for environmental modeling and inverse problems.

During the calibration process, the DE algorithm iteratively modifies the input parameters and evaluates the objective function (in this case, the RMSE) at each step. Based on this feedback, the algorithm updates its search strategy to improve model fit. Throughout the optimization, the interface displays the current value of the objective function, and includes a live-updating plot that visually compares observed and simulated data. The simulated values are updated in real time at each iteration, allowing users to visually track the calibration process and assess convergence.

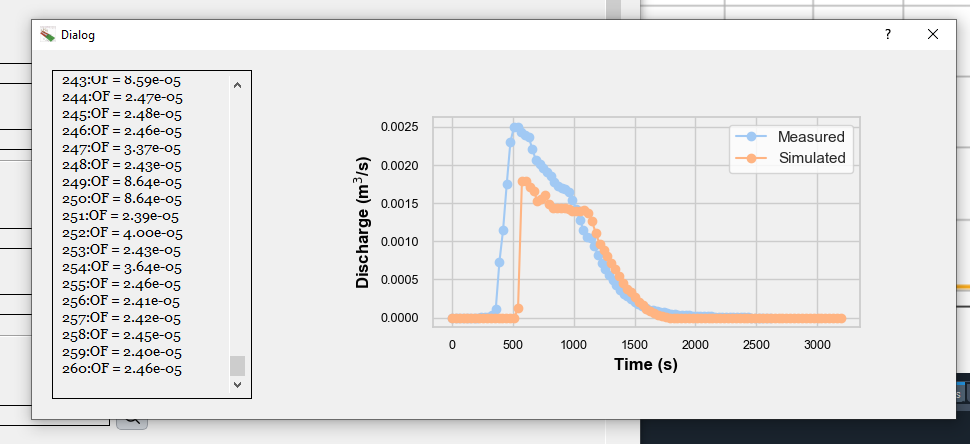


Figure 6.37. **Dialog displayed during calibration execution.** On the left, the values of the selected objective function are shown for each iteration of the optimization process. On the right, the plot illustrates the evolution of the simulated hydrograph compared to the observed one.

The results displayed after the calibration allow for a comprehensive evaluation of the model performance. The program generates a dedicated file containing all relevant calibration outcomes. This file is automatically read by the interface and presents, first, a summary of the numerical indicators used to assess the quality of the calibration. The following goodness-of-fit metrics are reported: Nash–Sutcliffe Efficiency (NSE), Root Mean Square Error (RMSE), Mean Absolute Error (MAE), Index of Agreement (IOA), Relative Index of Agreement (IOA-r) and Modified Nash–Sutcliffe Efficiency relative to simulated values (Ceff\_r):

As mentioned earlier, graphical evaluation is a key component of model calibration. To support this, the interface provides two types of visual outputs. First, a **time series plot** displays both observed and simulated discharge values over time, allowing users to visually assess how well the model reproduces the dynamics of the observed event. Second, a **scatter plot** presents simulated values against observed values, accompanied by a 1:1 reference line. This plot highlights the degree of agreement at each time step and reveals potential biases or systematic deviations between the two datasets.

Together, these visual tools complement the numerical indicators described previously, offering an intuitive and effective way to evaluate calibration performance.

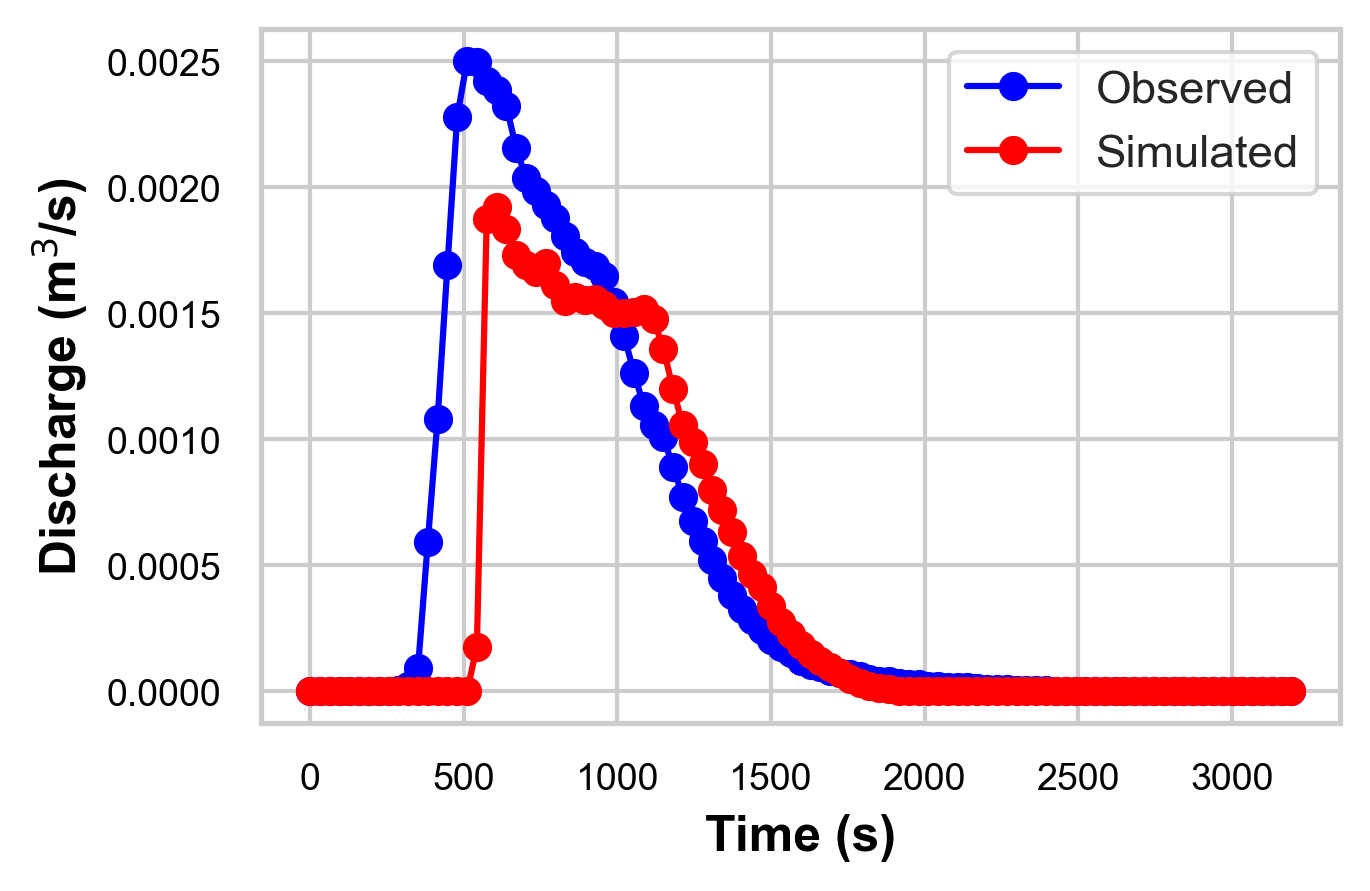


Figure 6.38. **Calibration evaluation: comparison of hydrographs.** One of the available plots for assessing model calibration. This figure shows the temporal comparison between the observed and simulated hydrographs.

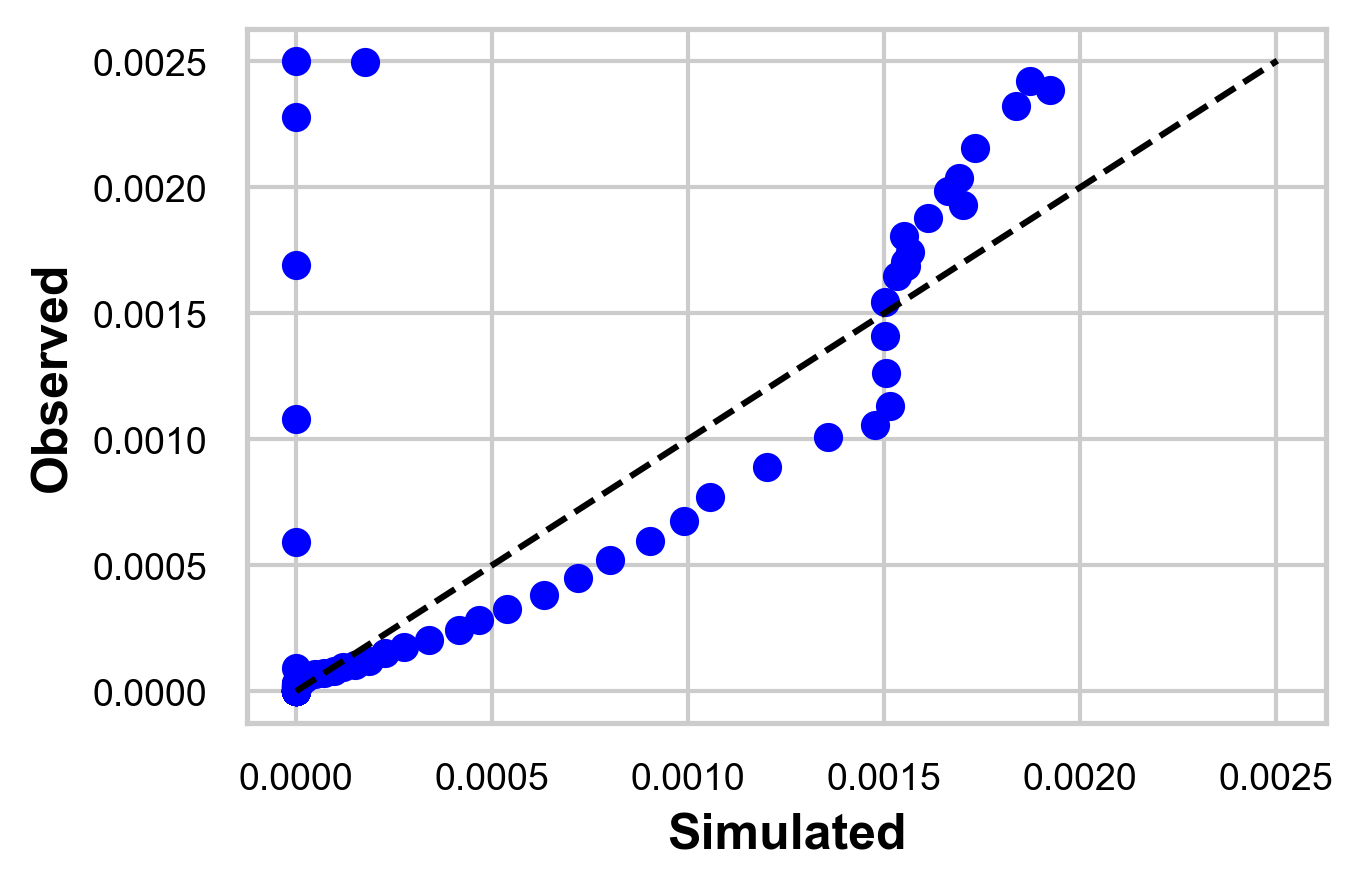


Figure 6.39. **Calibration evaluation: simulated vs. observed.** Plot used to evaluate model calibration. It displays the simulated values versus the observed ones, allowing for a visual assessment of the model fit.

Following the approach proposed by Ritter & Muñoz-Carpena (2013), a **bootstrap resampling** procedure was applied to estimate the uncertainty associated with the Nash–Sutcliffe Efficiency (NSE) obtained from the calibration. To determine the appropriate block length for the bootstrap process, the method described by Politis & White (2004) was used.

Figure 6.44 shows the results of the bootstrapping analysis for the previously discussed calibration. This probabilistic evaluation provides a more robust understanding of the reliability of the calibration metric by quantifying the variability in NSE across multiple resampled datasets.

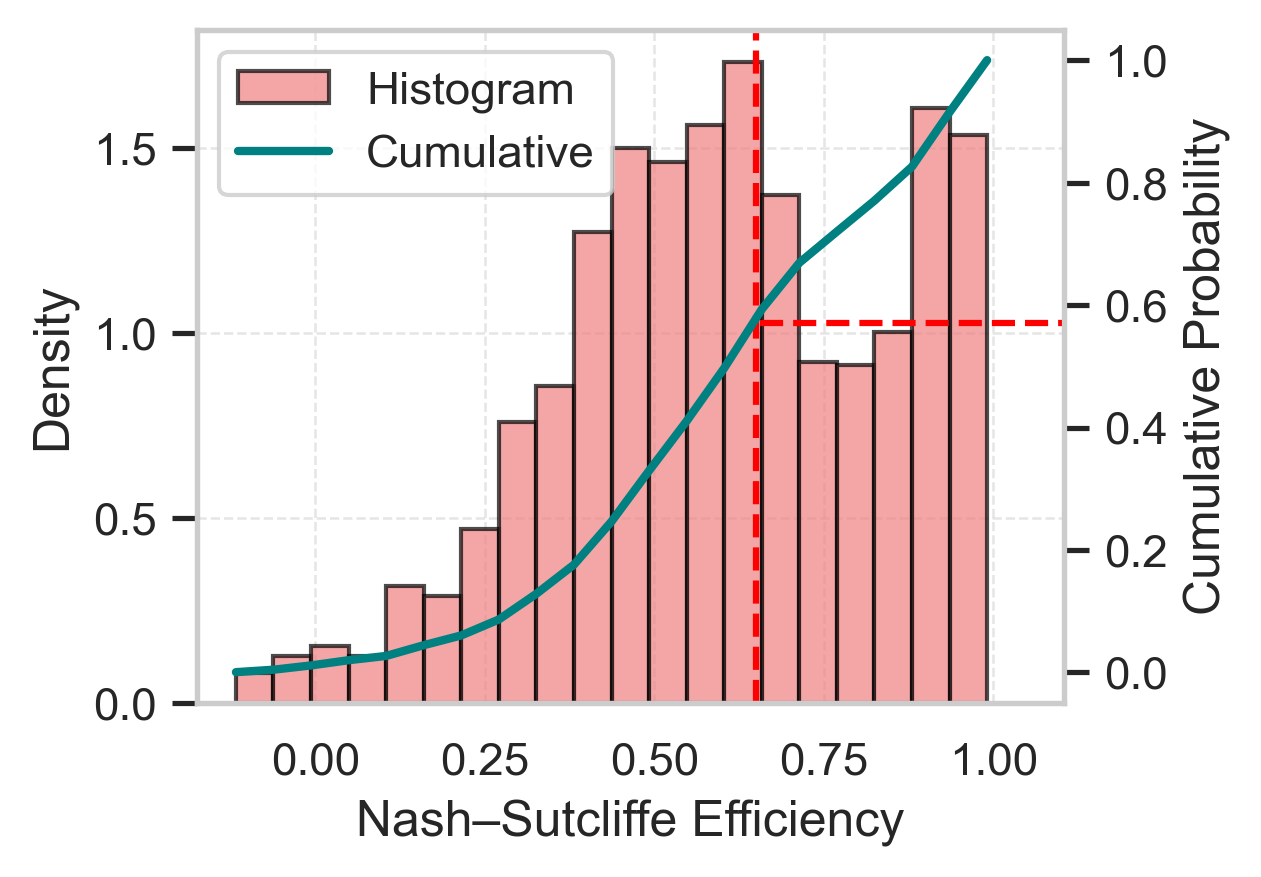


Figure 6.40. **Bootstrapping results for Nash–Sutcliffe efficiency.** This figure displays the probability distribution and cumulative distribution of Nash–Sutcliffe efficiency values obtained through bootstrapping. The vertical red line represents a user-defined threshold, intended to assess the proportion of simulations that exceed the selected efficiency level.

The dialog displayed after the bootstrap analysis is shown in Figure 6.45. On the left, the interface presents the same distribution plot of the Nash–Sutcliffe Efficiency (NSE) results shown previously. On the right, it provides tools for hypothesis testing and interpretation.

The user can specify a target NSE value, and the interface automatically calculates the probability that the true NSE is lower than the specified threshold. Additionally, the dialog displays the estimated NSE and RMSE values corresponding to the 2.5th, 50th (median), and 97.5th percentiles of the bootstrap distribution. This feature allows users to assess the confidence intervals of the calibration metrics and make more informed decisions regarding model performance.

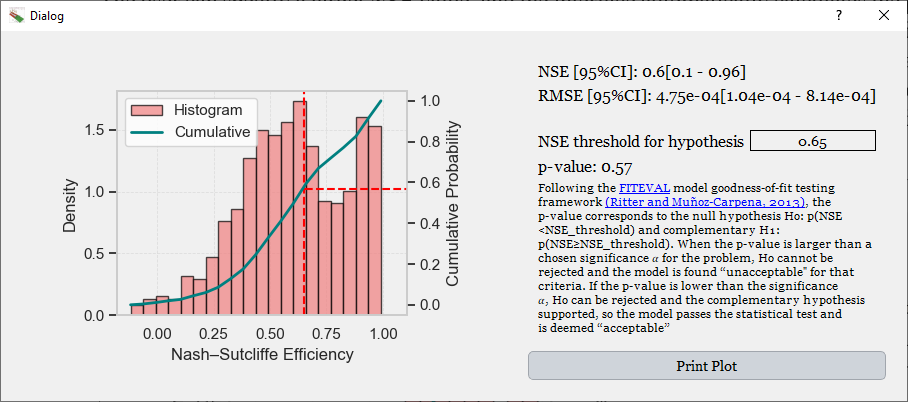


Figure 6.41. **Bootstrap results dialogue.** On the left, the probability and cumulative distribution plot of the efficiency index is shown. On the right, a line edit allows users to enter a custom threshold, with the associated non-exceedance probability automatically updated below.

Once the hydrograph calibration is completed, the values of the calibrated parameters become available. These values are displayed in the dialog window, and in this case can also be seen in Figure 6.46. As shown, the **Vertical Saturated Hydraulic Conductivity** was calibrated to 1.5 x 10-5, while the **Initial Water Content** was estimated at 0.25.

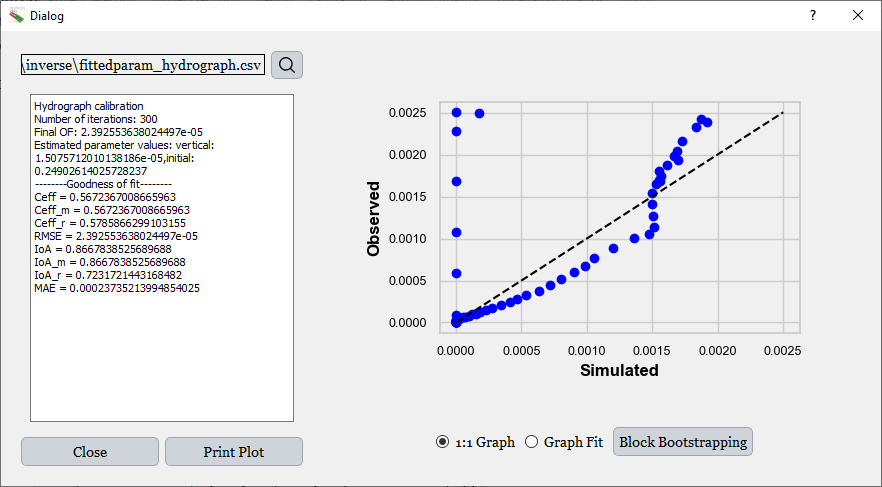


Figure 6.42. **Dialog showing the hydrograph calibration results.** On the left side, a Line Edit box shows the selected parameter values and associated goodness-of-fit indicators. On the right, the user can visualize the corresponding plots for model performance evaluation. In the lower right corner, a button allows the user to perform a bootstrapping procedure to estimate the uncertainty of the Nash–Sutcliffe Efficiency indicator.

Beyond simply displaying the results to the user, the interface automatically creates a new project in which the calibrated parameter values are updated and saved. This ensures continuity and reproducibility in the workflow. The newly created project is then automatically loaded into the interface and positioned at the next step in the calibration sequence, specifically, the calibration of sediment-related parameters. This seamless transition helps streamline the multi-stage calibration process and reduces the need for manual intervention between phases.

Following the calibration workflow diagram, the next step is the **sediment calibration**, provided that observed data are available. In this case, sediment data are indeed available and were obtained from the same study by Muñoz-Carpena (1993). For the same rainfall event used in the runoff calibration, a **sedimentograph** was recorded, capturing the evolution of sediment load over time.

Figure 6.47 shows the sedimentograph corresponding to this event, which serves as the observed dataset for the sediment calibration process.

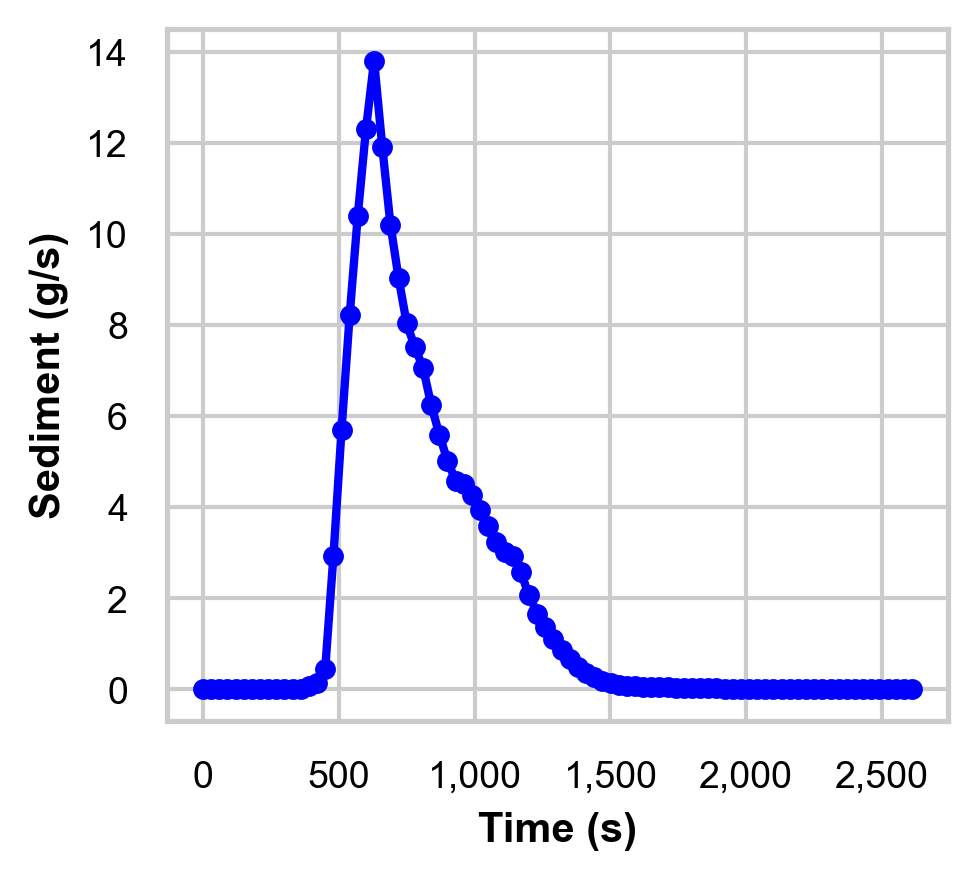


Figure 6.43. **Sedimentograph used for calibration.** This figure displays the sediment discharge leaving the filter, which is used during the sediment calibration step.

The sediment-related parameters to be calibrated, along with their minimum and maximum values, are listed in the table below.

Table 6.2. **Parameters used for sediment calibration.** Minimum and maximum values defined for each parameter involved in the sediment calibration process.

|  |  |  |  |
| --- | --- | --- | --- |
|  | **Minimum value** | | **Maximum value** |
| Incoming Flow sediment concentration (g/cm3) | 0.07 | 0.27 | |
| Sediment particle class diameter d50 (cm) | 0.003 | 0.01 | |

As in the previous calibration step, a **parallelized identifiability analysis** can be performed to determine which parameter has the greatest influence on the selected goodness-of-fit indicator.

Figure 6.48 presents the results of this identifiability analysis, showing the relative impact of each parameter on model performance.

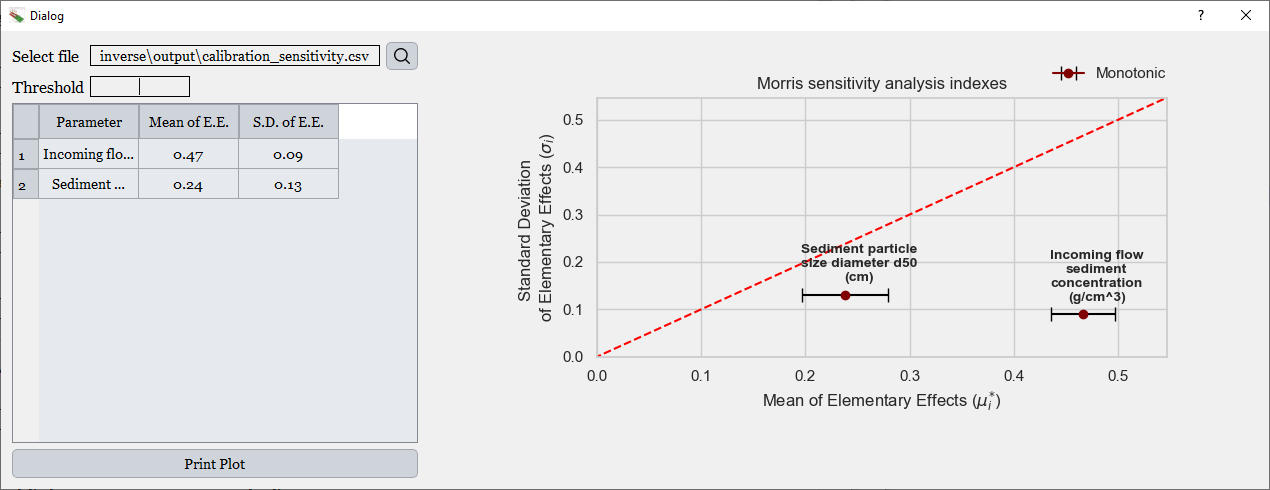


Figure 6.44. **Figure showing the results of the identifiability analysis for the calibration of the sediment load.** On the left, the table displays the values of the mean and standard deviation of Morris elementary effects. On the right, the corresponding Morris plot is shown.

In this case, both the particle diameter of the sediment and the sediment concentration in the inflow were found to influence the variation in the goodness-of-fit indicator. Among the two, the sediment concentration exhibited a greater effect on model performance.

Nevertheless, for demonstration purposes, both parameters were selected for calibration. The calibration procedure was executed, and the resulting graphical outputs are shown below. These include the time series comparison of observed and simulated sediment loads, as well as the corresponding scatter plot with the 1:1 reference line.

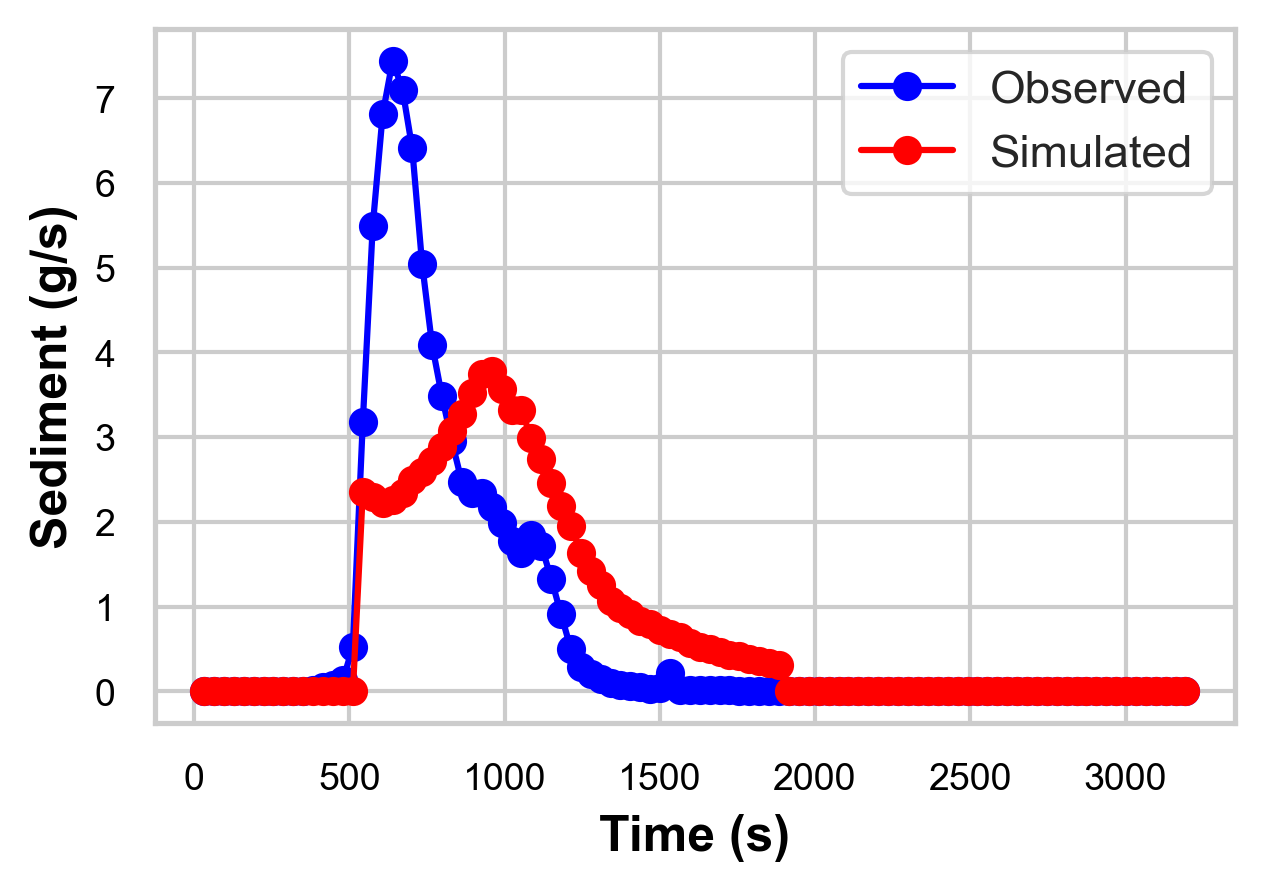


Figure 6.45. **Comparison between observed and simulated sediment load over time.** This figure displays how the simulated sediment load aligns with observed data after the calibration process.

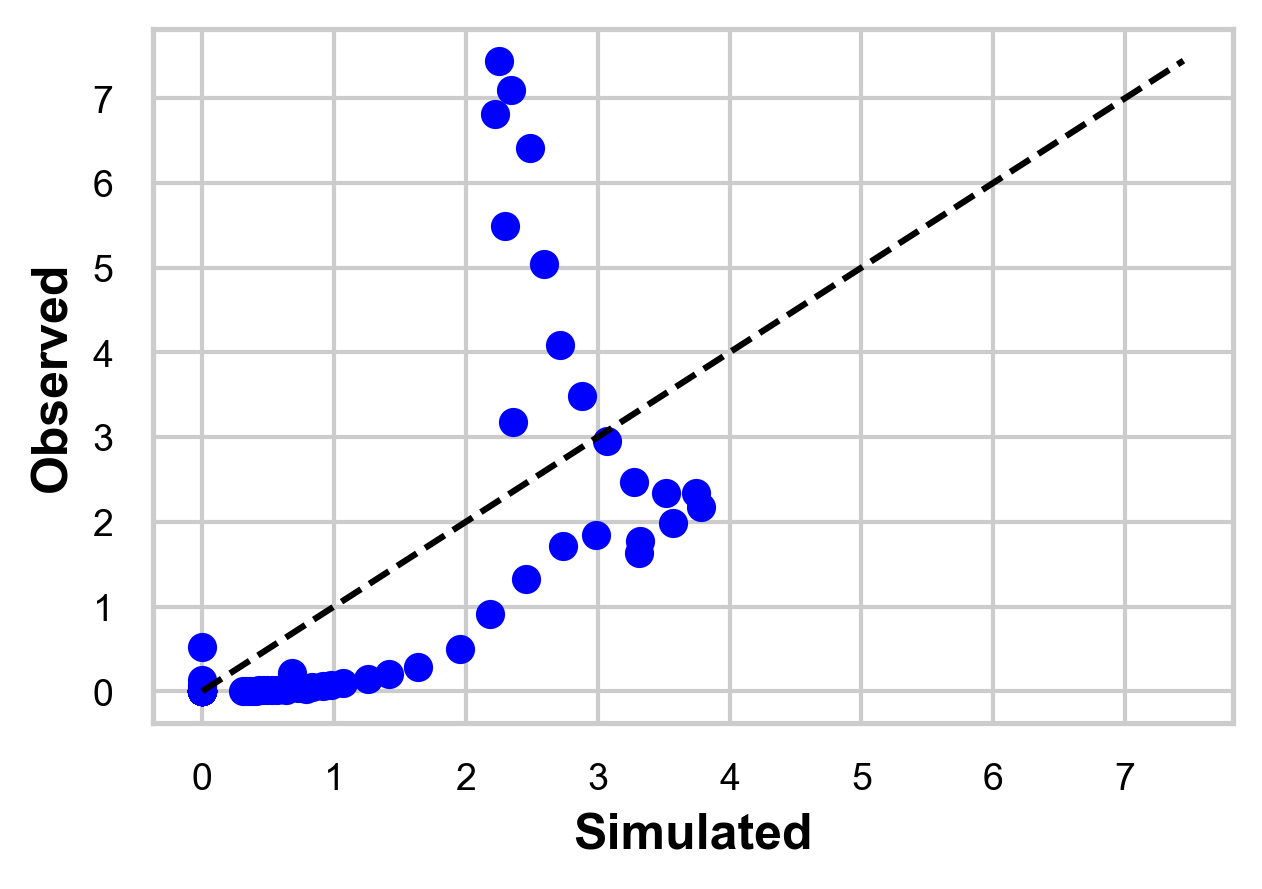


Figure 6.46. **Simulated versus observed sediment load after the calibration process.** This scatter plot allows a direct visual comparison of the agreement between observed and modeled sediment load values.

As in the previous step, the calibrated parameter values from the sediment calibration are automatically saved in a newly generated project. This allows the user to directly proceed either to the next calibration phase (pesticide) or to the design stage.

In this case, however, the sediment calibration was not successful, as shown in the figures above: the simulated sedimentograph does not adequately match the observed data. To address such situations, the new graphical user interface introduces a functionality not available in the previous version, the ability to perform **calibration using single-value metrics** instead of full time series. This option is available for runoff, sediment, and pesticide components.

This single-value calibration feature enables the user to match observed cumulative values or characteristic ratios, which are often easier to obtain than high-resolution time series. The following single-value indicators are available for calibration:

* **Runoff**:
  + Total outflow volume (m³)
  + Proportion of runoff volume filtered by the strip
* **Sediment**:
  + Total sediment mass exiting the filter (kg)
  + Proportion of sediment retained by the strip
* **Pesticide**:
  + Proportion of pesticide retained by the strip
  + Pesticide mass exiting the filter (mg)
  + Pesticide in the solid phase exiting the filter (mg)
  + Pesticide in the dissolved phase exiting the filter (mg)

These metrics provide a practical alternative when detailed time series data are not available. Moreover, the new GUI allows users to calibrate all three components (runoff, sediment, and pesticide) **simultaneously or independently**, depending on data availability and modeling objectives.

In the present example, due to the poor performance of the time-series-based sediment calibration, a new calibration is performed using the **total sediment mass exiting the filter** as the observed metric. Additionally, the **total outflow volume** is included again in this step for demonstration purposes, to illustrate how multiple components can be calibrated concurrently using single-value data.

Both observed values for runoff and sediment were obtained from the same study by Muñoz-Carpena (1993), corresponding to the previously used rainfall event.

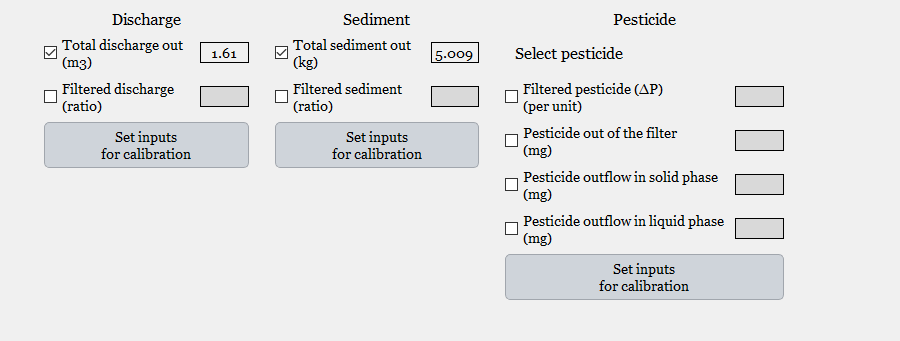


Figure 6.47. **Dialog used for the calibration of single-value outputs.** Unlike time-series outputs such as hydrographs or sediment load curves, single-value outputs represent aggregated quantities. For flow calibration, available outputs include Total Discharge Out and Filtered Discharge. For sediment, the outputs are Total Sediment Out and Filtered Sediment. For pesticide calibration, available outputs include Filtered Pesticide, Pesticide Out of the Filter, Pesticide Outflow in Solid Phase, and Pesticide Outflow in Liquid Phase. By checking the box next to each single-value output, users can provide an observed value for calibration.

In this case, two observed single-value metrics were used: a total outflow volume of 1.61 m3 and a total sediment mass of 5.009 kilograms, both measured at the outlet of the filter during the selected rainfall event. These values serve as the reference for evaluating model performance using a simplified error metric.

Unlike the time-series calibration, which allows the user to select from multiple objective functions, the single-value calibration applies a fixed error metric. The calculation involves comparing each simulated value with its corresponding observed value and computing the squared relative error. Specifically, for each output, the difference between simulated and observed values is normalized by the observed value, squared, and then rooted. Finally, the value is divided with the number of observed values used in the calibration.

The formula used is as follows:

Where S is the simulated value, O is the corresponding observed value and N is the number of output variables used in the calibration.

This approach provides a straightforward measure of relative deviation between simulated and observed single-value metrics, allowing for efficient and objective calibration when time series data are unavailable or unreliable.

The calibration produced a simulated **total outflow volume** of **1.448 m³,** compared to the observed value of **1.61 m³,** and a simulated **total sediment mass** of **5.197 kg**, compared to the observed **5.009 kg**. Using the error formulation described in the previous section, this calibration yielded an error of **0.0537**.

Figure 6.52 shows the dialog displayed after the calibration. The interface presents the comparison between simulated and observed single-value metrics, as well as the calibrated parameter values obtained from the optimization.

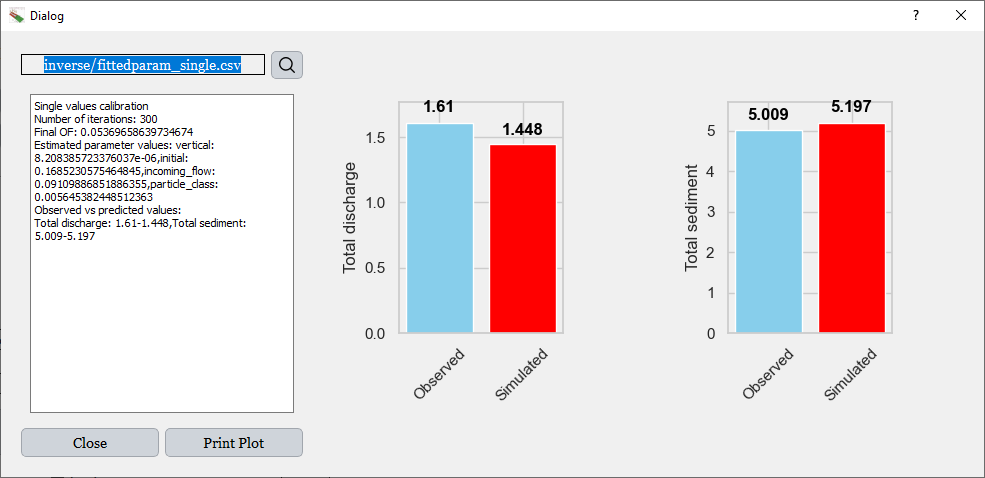


Figure 6.48. **Dialog displayed after the calibration of single-value outputs.** On the left, the interface displays the number of iterations, the value of the objective function, the estimated parameter values after calibration, and the comparison between observed and simulated values. On the right, a graphical comparison is shown between observed and simulated for each selected output, allowing visual assessment of calibration accuracy.

The final calibrated values for this step are listed below:

Table 6.3. **Estimated parameter values obtained from the calibration of single-value outputs.** This figure presents the values of the parameters resulting from the optimization process focused on matching observed single-value outputs such as flow and sediment-related quantities.

|  |  |
| --- | --- |
|  | **Obtained value** |
| Vertical Saturated K (m/s) | 0.00000082 |
| Initial water content (m3/m3) | 0.17 |
| Incoming Flow sediment concentration (g/cm3) | 0.091 |
| Sediment particle class diameter d50 (cm) | 0.0056 |

In this particular case, no pesticide data were available for the project, and therefore the pesticide module was not included in the calibration process. However, as previously described, the new **Graphical User Interface** fully supports pesticide calibration, an important **enhancement** over the previous version of the tool.

The interface allows the user to **select a specific pesticide** for calibration, enabling targeted adjustment of the model parameters associated with that compound. This functionality was not available in the earlier version of the tool and represents a key step forward in model flexibility and applicability.

With the runoff and sediment calibration completed, and the **estimated parameter values** obtained, the system is now sufficiently characterized to move on to the **design phase** of the vegetative filter strip, as outlined in the workflow diagram shown in Figure 6.1.

### 6.4.3 Design

#### 6.4.3.1 Deterministic design

Once the model has been calibrated, the values of key parameters are known and can be used to proceed to the next step: the design phase. This phase aims to determine the appropriate dimensions of the vegetative filter strip that meet specific environmental performance objectives defined by the user.

The design involves two adjustable variables: the length of the filter (in meters) and the vegetation spacing (in centimeters). The user can define a range of values for these variables, and the program will iteratively simulate the system’s behavior for each configuration. The results are then interpolated to identify the combinations of filter dimensions that satisfy the defined targets. Prior to this step, precipitation depths associated with selected return periods, along with area, runoff, and erosion data, are processed through the UH utility.

This design process is especially relevant in the context of Total Maximum Daily Load (TMDL) programs or other environmental regulations, which often require a specified percentage reduction in pollutant loads, such as sediment or pesticide runoff, from a disturbed area. In such cases, the goal is to identify the minimum filter dimensions that achieve compliance with these regulatory thresholds.

The user may specify the storm events under which the system should perform. These can be input as specific rainfall values or as a range of precipitation intensities. For example, in the case shown below, the system is configured to test storm events from 50 mm to 200 mm, with an increment of 30 mm. Similarly, the filter length is varied from 1 meter to 51 meters, with a step of 15 meters.

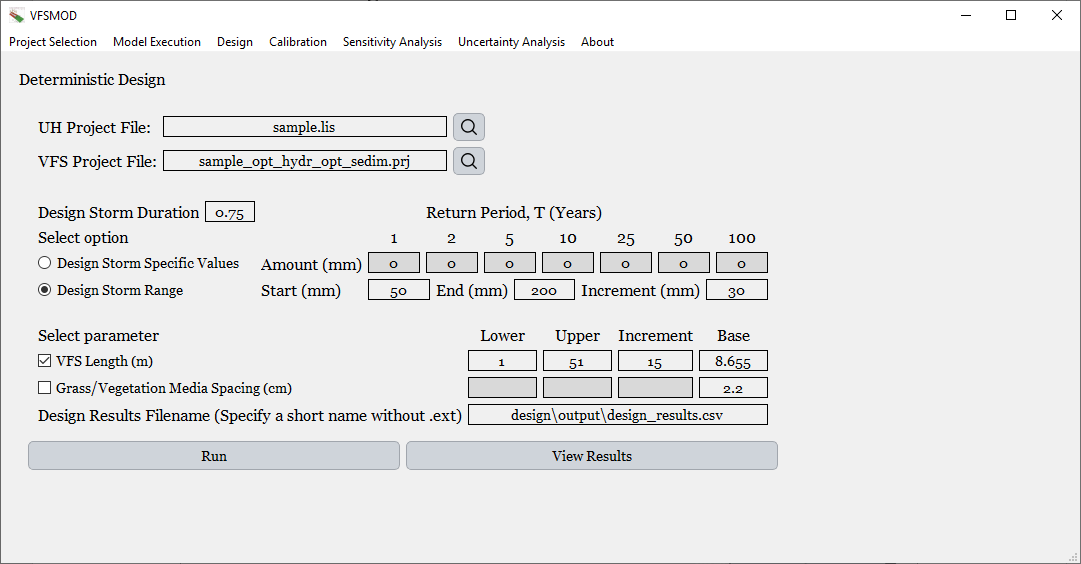


Figure 6.49. **Dialog for deterministic design.** At the top of the interface, the file paths for the UH and VFSMOD execution inputs are displayed. Below that, the storm duration is specified, followed by the section where the user can select the design storms, either by entering fixed values or defining a range. At the bottom, the user sets the range of the design parameter to be tested, such as the ***filter strip length*** or the ***grass/vegetation median spacing***. In this case, a minimum, maximum, and increment are provided to iterate over the ***filter strip length***.

When the user initiates the simulation, the program begins iterating through all combinations of storm events and filter sizes defined in the setup. In this particular case, the filter dimension being varied is the filter length.

For each rainfall event specified, the model computes a series of output variables across the different filter lengths. This process is repeated for all selected precipitation events, ensuring that every combination of rainfall intensity and filter size is evaluated.

In this initial stage, the objective is to calculate and store the results of all possible combinations. The interface then displays a table summarizing the outputs corresponding to each configuration, as shown in the following figure.

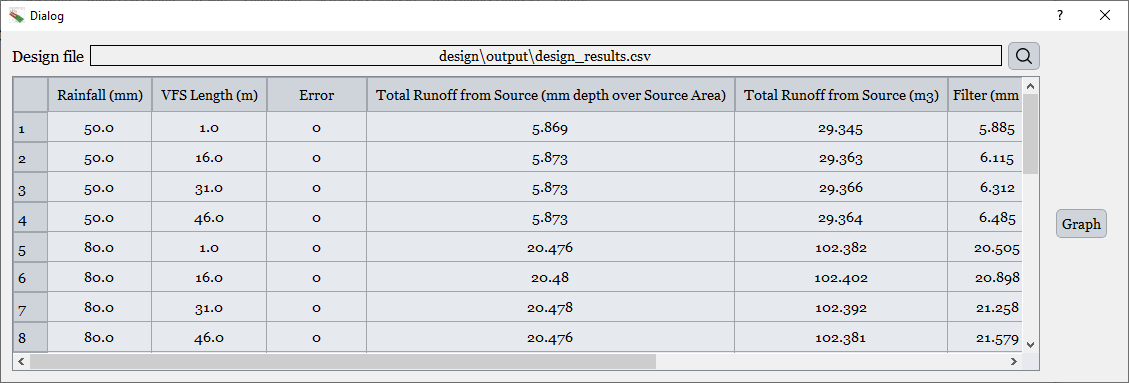


Figure 6.50. **Dialog showing the results obtained from the deterministic design.** This interface displays a table that reads data from a file generated after the design phase. The table includes the rainfall values used in the design, the filter strip dimension being iterated (in this case, the ***filter length***), and a column indicating execution errors. A value of **1** in this column alerts the user of a failed simulation, often due to an inappropriate range of input parameters. The remaining columns present the outputs computed by the model for each configuration. On the right side of the interface, a button allows for graphical visualization of these results.

To facilitate interpretation, the program includes a graphical visualization tool. The figure below shows the dialog that appears when the user selects the option to display the design results graphically.

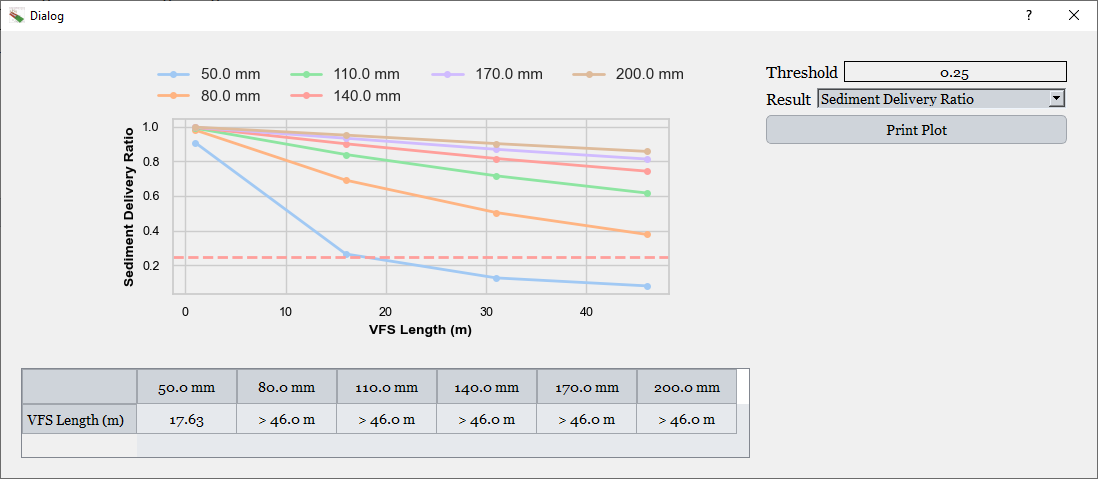


Figure 6.51. **Dialog displaying the graphical results of the deterministic design.** On the central left side of the interface, a graph shows **response curves** corresponding to different rainfall events and varying filter strip lengths. To the right, a threshold can be defined by the user, which dynamically updates a horizontal line across the plot. Below, a dropdown menu allows selection of the output variable of interest. The table at the bottom displays the minimum **filter length** required to meet the specified threshold (e.g., an *Sediment Delivery Ratio* of 0.25) for each rainfall event.

In this dialog, a plot is presented showing the response curves for each rainfall event, corresponding to the different filter lengths tested. On the right side of the dialog, under the *Result* section, the user can select which output variable to analyze. When a different output is selected, the graph is updated accordingly to display the selected variable.

The user can also specify a threshold value that the selected output must meet. For example, in the case shown, the output being analyzed is the *Sediment Delivery Ratio* (SDR), and the user has defined a target threshold. When the threshold is entered or modified, a red horizontal line appears on the graph to indicate this value.

Below the plot, a table is displayed showing the minimum filter length required to meet the threshold for each rainfall event. If the threshold is not satisfied within the tested range, the table will indicate that the required length exceeds the maximum value tested. In the example shown, the highest filter length evaluated was 46 meters. As seen in the results, for rainfall events above 80 millimeters, the 0.25 threshold for *Sediment Delivery Ratio* is not met, and the required length is reported as greater than 46 meters.

In addition to the *Sediment Delivery Ratio* shown in the previous example, the user can also select from a wide range of other outputs for visualization. These include:

* Total runoff from the source area (mm depth over the source area)
* Total runoff from the source area (m³)
* Total runoff exiting the filter (mm depth over the source and filter area)
* Total runoff exiting the filter (m³)
* Total infiltration within the filter
* Mass of sediment entering the filter
* Sediment concentration in runoff from the source area
* Mass of sediment exiting the filter
* Sediment concentration in runoff exiting the filter
* Sediment Delivery Ratio
* Runoff Delivery Ratio
* Pesticide Delivery Ratio

Among these, the most relevant for decision-making are typically the *Sediment Delivery Ratio*, *Runoff Delivery Ratio*, and *Pesticide Delivery Ratio*, as they directly reflect the filter's performance in reducing pollutant transport. In some cases, runoff can also be used as a proxy for nutrient losses. In the case of pesticide modeling, where multiple compounds may be evaluated, the interface allows users to select the *Delivery Ratio* for each individual pesticide.

These three indicators are particularly insightful because they are highly sensitive to changes in the filter dimensions. In contrast, other outputs such as the sediment concentration in runoff entering the filter tend to remain constant, as they are determined by upstream conditions and are unaffected by the size of the vegetative filter strip.

The SDR is defined as the ratio between the mass of sediment exiting the filter and the mass of sediment entering it (i.e., SDR = sediment out from filter / sediment into the filter), and it serves as a key performance indicator for evaluating the effectiveness of vegetative filter strips. In this case, it is assumed that the objective is to achieve a 75% reduction in sediment output, corresponding to a *Sediment Delivery Ratio* of 0.25. This target is established to comply with requirements set by the Total Maximum Daily Load program. As illustrated in Figure 6.55, for a rainfall event of 50 mm, under the same precipitation distribution and storm duration described in the calibration section, a minimum vegetative filter strip length of 17.63 meters would be required to meet the SDR threshold.

This represents a deterministic design, where input parameters are fixed and no uncertainty is considered. While this approach is computationally efficient and useful for verifying that a particular design target (e.g., SDR = 0.25 for a 50 mm event) is achievable under specific conditions, it does not account for variability in input parameters.

To incorporate uncertainty into the design, reflecting the natural variability and potential imprecision in inputs, the next section will present a probabilistic design approach, which is more computationally demanding but provides a more robust basis for decision-making.

#### 6.4.3.2 Design with uncertainty

Uncertainty is defined as the estimated amount by which an observed or calculated value may deviate from the true value (Lapedes, 1978), and it plays a critical role in policy, regulatory, and environmental decision-making. Uncertainty in TMDL models is a real issue and should be taken into consideration not only during the TMDL assessment phase, but also in the design of BMPs during the TMDL implementation phase (Shirmohammadi et al., 2006b).

Predictive models and measured data almost always carry some degree of uncertainty. This is especially critical when model inputs are based on averages derived from sparse or inconsistent measurements, which fail to represent the inherent variability of natural systems. The use of default or poorly characterized parameters, such as soil properties, climate data, or management practices, can significantly impact model reliability.

Given these challenges, stochastic methods like Monte Carlo Simulation (MCS) offer a valuable alternative to deterministic approaches. While they do not reduce the inherent uncertainty of the system, they allow for a more realistic representation of input variability and its influence on model outputs, leading to more informed and robust decision-making. In this context, the integration of uncertainty into the design phase helps to determine the optimal filter length not as a fixed value, but as a function of input variability and confidence intervals, ultimately supporting more resilient water quality management strategies.

In this section, the same design procedure described previously, namely, the deterministic design, will be repeated, but incorporating input uncertainty. As before, the same project and example will be used, with the goal of achieving a *Sediment Delivery Ratio* of 0.25 for a precipitation event of 50 mm. As established in the previous section, this target was met with a filter length of 17.63 meters under deterministic conditions.

The uncertainty-based design follows the same principle as the deterministic approach: for a given precipitation, the model iteratively tests different filter lengths to identify the minimum one that meets the target. However, in this case, the procedure is repeated multiple times to account for uncertainty. Specifically, the model generates several plausible input scenarios based on user-defined uncertainty ranges, and in each scenario the design process is executed independently. This results in a distribution of required filter lengths across the ensemble of input conditions.

In this example, the uncertainty-based design is performed for a single rainfall event, 50 mm, and only one design variable is considered: the filter length. To execute this process, the user must define (i) the target SDR, (ii) the rainfall event, (iii) a system with fixed parameters, and (iv) the parameters to which uncertainty will be applied.

Any of the parameters used in VFSMOD can be selected to include uncertainty in the analysis, allowing users to explore the impact of parameter variability on model outputs. The following parameters have been selected to include uncertainty in this case study:

Table 6.4. **Parameters and distributions used for the uncertainty-based design.** This table summarizes the parameters involved in the uncertainty-based design procedure along with their associated probability distributions. These inputs define the uncertainty space over which the model simulations are performed to estimate the variability of the output response.

| **Parameter** | **Distribution Type** | **Range / Peak Value** |
| --- | --- | --- |
| Curve Number | Triangular | Min: 70, Peak: 75, Max: 80 |
| Organic Matter (%) | Uniform | Min: 1, Max: 2 |
| Crop Factor | Uniform | Min: 0.8, Max: 1 |
| Practice Factor | Uniform | Min: 0.8, Max: 1 |

In this study, all input parameters are considered to be independent from one another. The same assumption applies to the parameters selected in the following sections. These input distributions reflect plausible variability based on expert knowledge or field data. The simulation will use them to generate multiple realizations, allowing for the estimation of the probability that a given filter length will meet the desired performance threshold under uncertainty.

In this case, due to the significantly higher computational demand of the uncertainty-based design, the procedure is limited to a single rainfall event per execution. As in the deterministic design, the filter length will vary from 1 to 51 meters in increments of 15 meters. This setup ensures consistency with the previous approach while allowing the model to efficiently explore the impact of uncertainty on the filter performance.

To generate different scenarios based on the uncertainty provided by the user, and to quantify both the variability of the output and the influence of each uncertain input on the optimal filter length, the program offers three global sensitivity analysis methods: Morris, FAST, and Sobol. These methods allow for a dual purpose: quantifying output uncertainty and identifying the relative impact of each uncertain input on that variability.

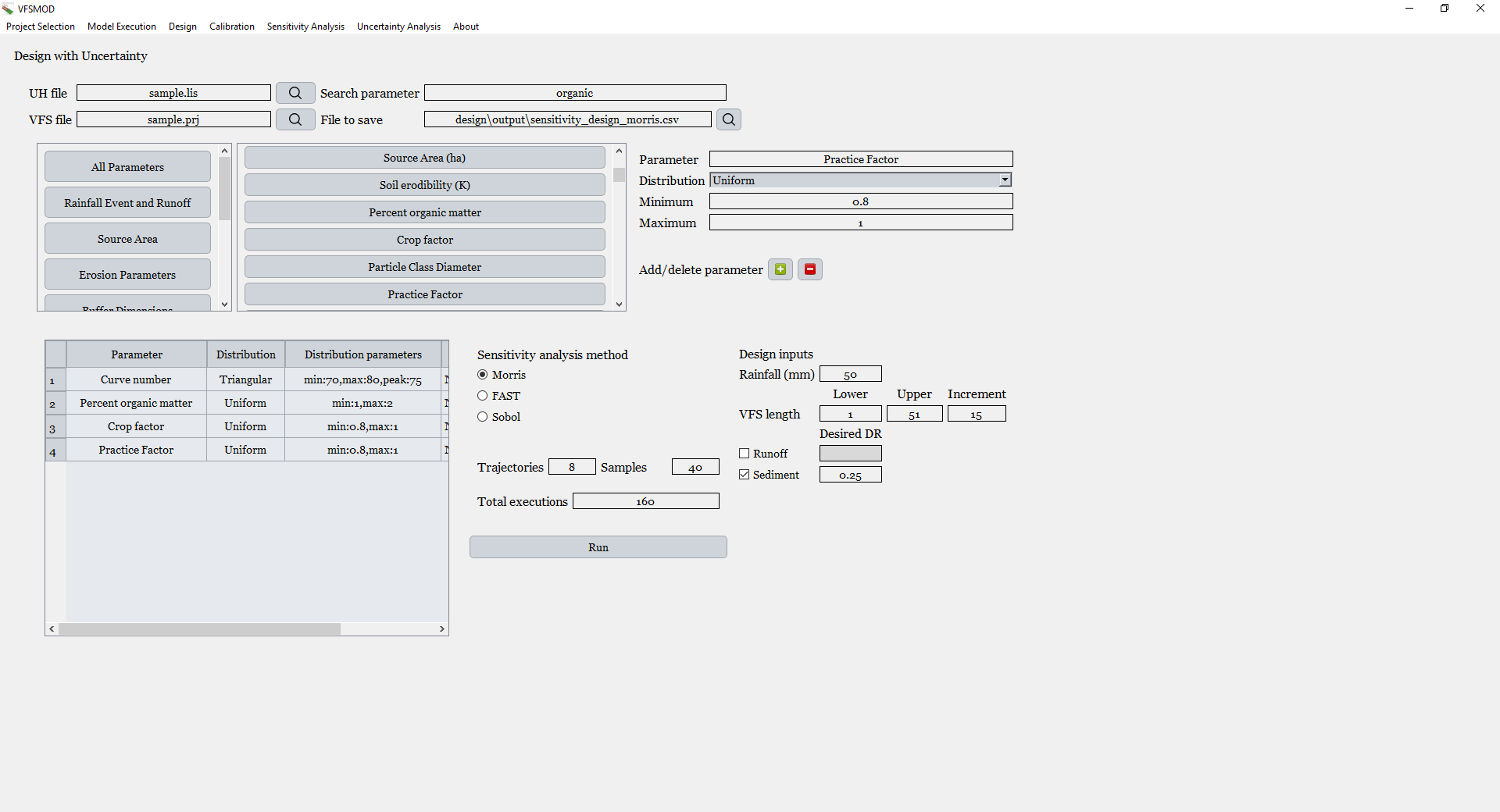


Figure 6.52. **Dialog window for running the uncertainty-based design.** At the top, the interface displays the file paths for the UH and VFS files. Below, all available parameters can be selected, with a section to the right for specifying the probability distribution associated with each selected parameter. In the lower-left corner, a table summarizes the selected parameters, their assigned distributions, and corresponding distribution parameters. To the right, the user can define the type of sensitivity analysis to be performed during the uncertainty-based design, as well as the sampling method and sample size. Further right, the user can specify the total precipitation to be considered in the design scenario, the iteration step for filter length testing, and the target performance threshold (e.g., a *Sediment Delivery Ratio* of 0.25). If pesticides are present in the project, the dialog automatically provides the option to input a *Pesticide Delivery Ratio* threshold for each pesticide included.

In this example, all three methods, Morris, FAST, and Sobol, have been applied for illustrative purposes. However, it is not necessary to perform all three analyses. A common and efficient approach is to begin with the Morris method, which requires fewer model evaluations and helps identify the most influential variables. Then, based on the results, a subset of the most impactful variables can be selected for a more detailed analysis using FAST or Sobol, which provide a more robust decomposition of output variance but at a higher computational cost.

This process has been parallelized, as it is computationally demanding and well-suited for parallel execution. In this example, the system was able to perform approximately six model runs per second using the computer on which the simulations were carried out. Each run involves modifying the input data files, executing both the UH and VFS models, and collecting the resulting outputs.

For the Morris method, eight trajectories were used, where each trajectory corresponds to a one-at-a-time perturbation path across the parameter space. This resulted in a total of 160 simulations. For the FAST and Sobol methods, 256 samples (or base points) were selected for each. Based on the structure of these methods, FAST required 4,096 simulations, and Sobol required 10,240 simulations.

In terms of computational time, the Morris analysis took approximately 27 seconds, FAST completed in around 11.5 minutes, and Sobol in approximately 29 minutes.

The following figure shows the interface dialog after the completion of these analyses.

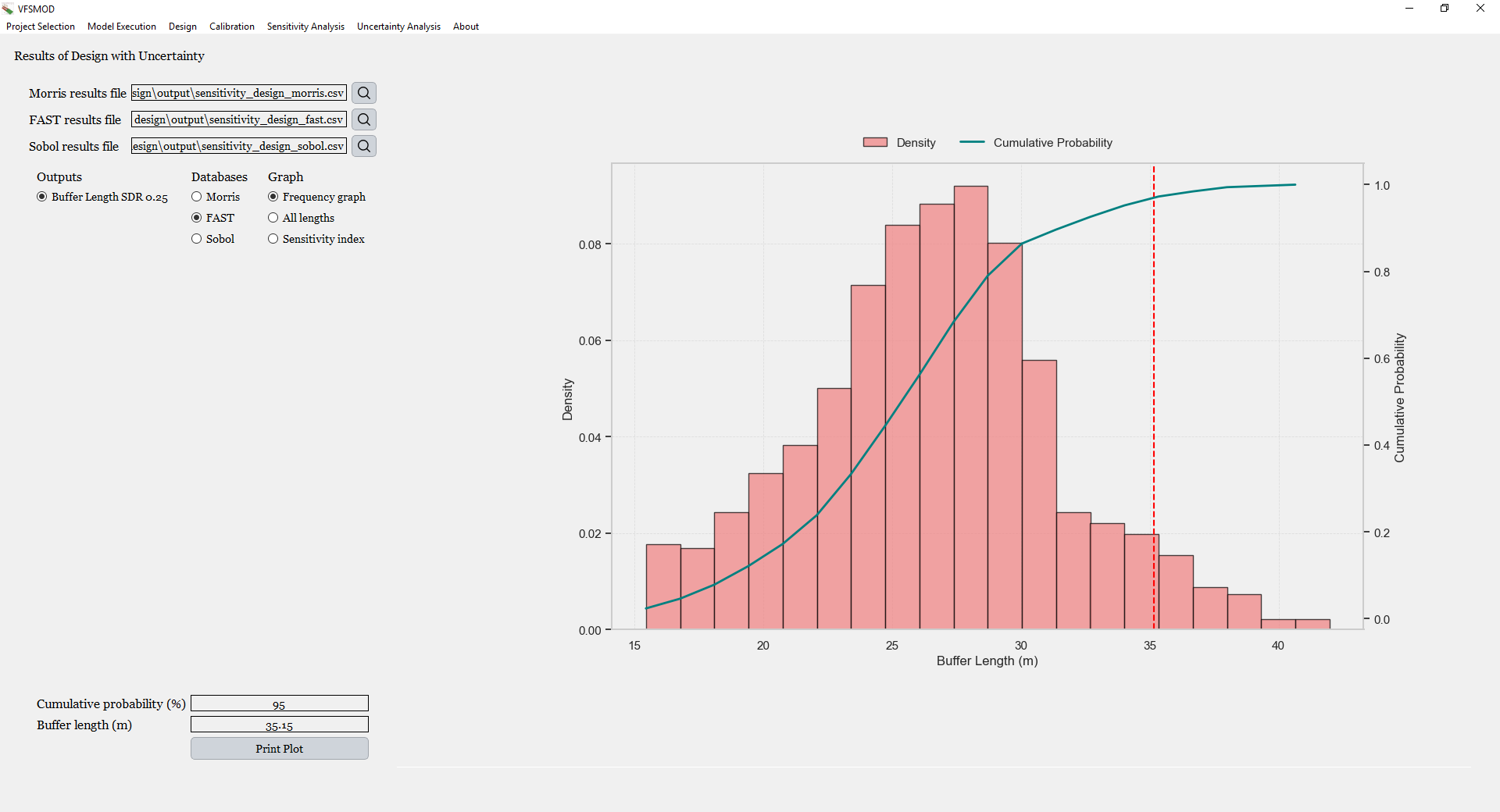


Figure 6.53. **Dialog displayed after performing the uncertainty-based design.** In the upper-left section, the path to the files generated during the uncertainty-based design is shown. Depending on the sensitivity analysis method applied, the results may originate from ***Morris***, ***FAST***, or ***Sobol*** methods. Below, users can select the **target output** to evaluate, the specific file to extract data from, and the **graph type** to be displayed, such as Frequency Graph, All Lengths, or Sensitivity Index. The selected graph is displayed on the right side. At the bottom-left, the **cumulative probability** is shown, while below it, the **corresponding buffer length** is displayed. Modifying either value automatically updates the other and refreshes the graph accordingly.

The uncertainty-based design module includes two dialog windows. The first one, previously described, allows the user to enter the necessary input data to configure and run the uncertainty design. Once the execution is complete, a file is automatically generated containing all the results of the analysis.

This file is then used to populate the second dialog, which is dedicated to visualizing the outcomes of the uncertainty-based design. The file path of the newly created result file is automatically inserted into the corresponding input field of this dialog.

As shown in the figure, the interface includes three input fields (line edits), one for each uncertainty analysis method: Morris, FAST, and Sobol. Each of these fields is automatically filled if the corresponding method has been executed.

Additionally, the dialog provides a column for selecting the outputs to be displayed. This allows the user to view the results in relation to the threshold(s) defined during the design setup. In the example shown, a threshold of 0.25 for the Sediment Delivery Ratio has been specified. However, since multiple thresholds can be evaluated simultaneously, whether for *Sediment Delivery Ratio*, *Runoff Delivery Ratio*, or the Delivery Ratios of different pesticides, the user can choose which output and threshold combination to analyze. The plot will automatically update based on the selected combination.

The second section of the dialog, labeled ***Databases***, allows the user to select which sensitivity analysis results to load and display. Users can choose between the available sensitivity studies, such as those from Morris, FAST, or Sobol to analyze the corresponding outputs.

The third section, labeled ***Graph***, offers three different plot options. Each time a radio button is selected, the corresponding graph is automatically updated and displayed. The three available visualizations are as follows:

1. **Frequency Plot** – This chart shows the distribution of the optimal filter length across all simulations. It helps visualize how the uncertainty in the input parameters translates into variability in the required filter size.
2. **Design Curve** – This graph displays how the selected output changes with respect to filter size, illustrating the range of performance associated with each filter length under uncertainty.
3. **Sensitivity Index Plot** – This plot presents the contribution of each input parameter to the variability in the required filter size, based on the chosen sensitivity analysis method.

Finally, the interface allows the user to select a cumulative probability value, and it will automatically display the smallest filter length that exceeds the target performance in that percentage of simulations. For instance, if a cumulative probability of 95% is entered, the dialog will return a filter length, such as 35.15 meters in this case, that achieves the *Sediment Delivery Ratio* threshold (e.g., 0.25) in 95% of the uncertainty scenarios.

This feature also works in reverse: the user can input a specific filter length, and the interface will immediately show the cumulative probability associated with that length. In both cases, the corresponding vertical line is automatically updated in the frequency plot, clearly indicating the selected filter length in the *Frequency Plot* graph and in the *Design Curve* graph, as illustrated in Figure 6.58 and 6.59 respectively.

Here is an example of each type of graph. The first one is the *Frequency Plot*, which displays the distribution of optimal filter lengths obtained from the design under each scenario. In this case, the results correspond to the uncertainty analysis conducted using the Sobol method. The red vertical line represents the filter length associated with a cumulative probability of 95%.

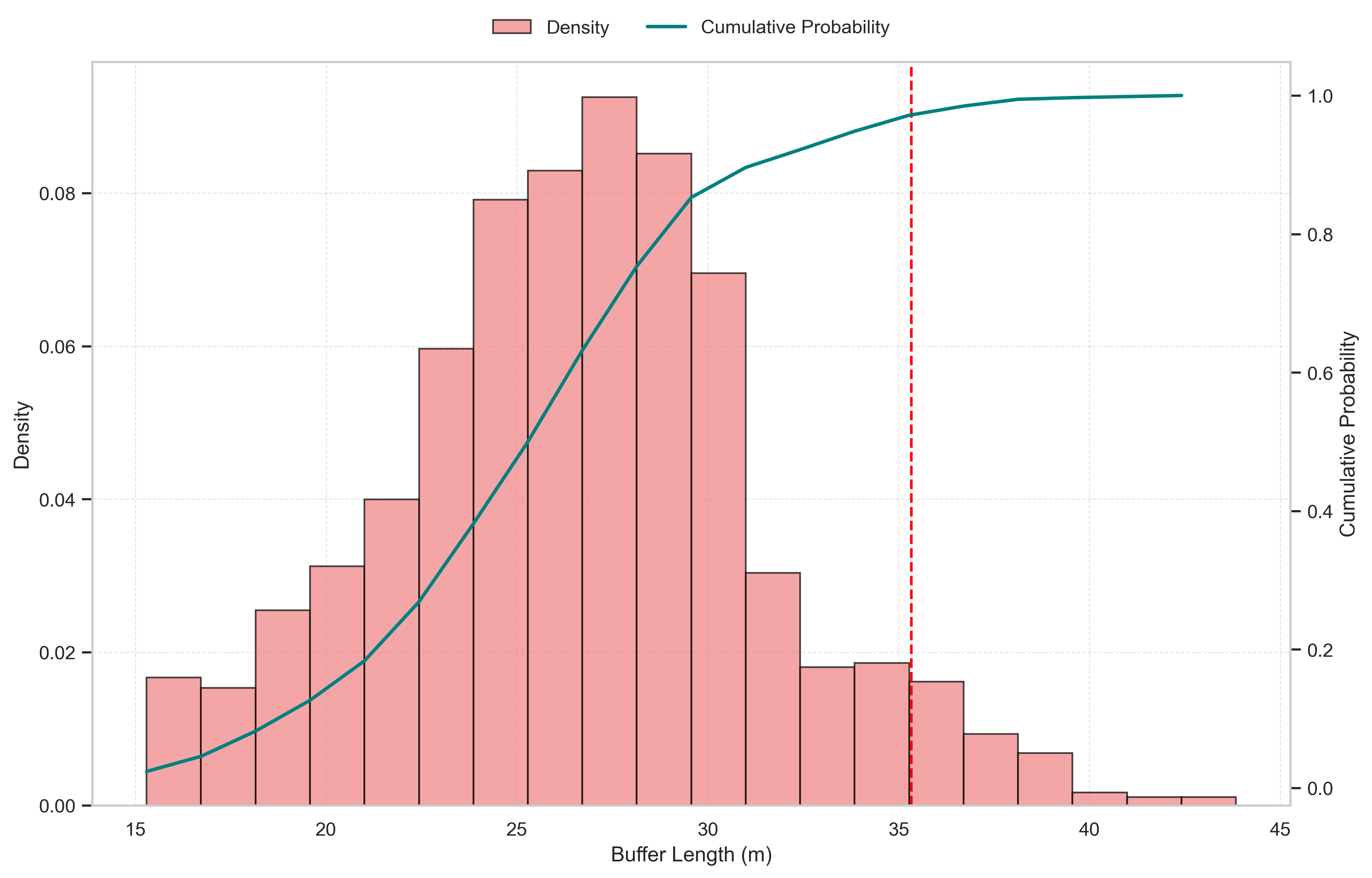


Figure 6.54. **Distribution of required buffer lengths under user-defined uncertainty**. This figure shows the frequency distribution of buffer lengths required to achieve the target output (e.g., a *Sediment Delivery Ratio* of 0.25) under all plausible system scenarios defined by the user's uncertainty setup. The cumulative distribution is also plotted. The red dashed vertical line indicates the buffer length corresponding to the selected confidence level (e.g., 95%), meaning that this buffer size achieves the target in 95% of the simulated scenarios.

The next graph also corresponds to the Sobol execution, which involved the highest number of simulations and therefore provides more refined results, allowing for a more comprehensive exploration of the uncertain input space. This plot is similar to the one shown in the deterministic design section. However, in this case, for each simulated filter length (ranging from 1 to 51 meters in 15-meter increments), the variability in the resulting *Sediment Delivery Ratio* is represented.

A 95% confidence interval is shown in gray for each length, illustrating the range of SDR values obtained under uncertainty. The solid black line indicates the median value (50th percentile), while the red dashed horizontal line represents the target SDR threshold defined by the user. The vertical red dashed line marks the filter length corresponding to the selected cumulative probability, 35.33 meters in this case. Additionally, the confidence interval for that optimal filter length is displayed, ranging from 16.8 to 37.06 meters.

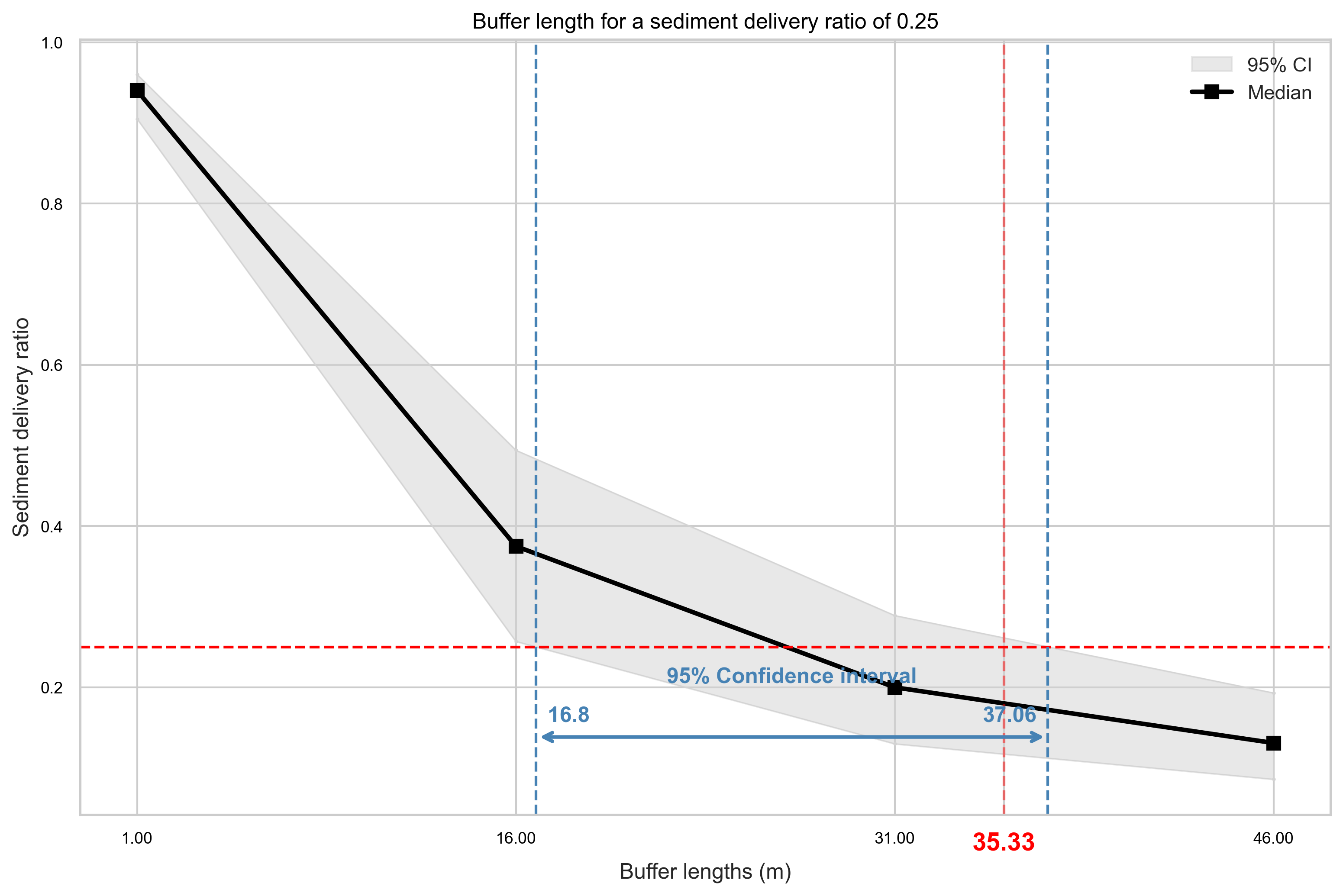


Figure 6.55. **Output response curve under uncertainty for the selected storm event.** This graph displays the variation of the selected output (e.g., *Sediment Delivery Ratio*) across different buffer lengths, similar to the deterministic design response curve. Unlike the deterministic case, here the shaded area represents the confidence interval at each length, capturing the uncertainty-induced variability. The blue band at the bottom highlights the **range of buffer lengths** within which the target condition is satisfied in 95% of the scenarios. Additionally, the red dashed vertical line indicates the **buffer length** that meets the user-defined cumulative probability.

Finally, the last graph displays the sensitivity indices obtained from each type of sensitivity analysis, Morris, FAST, and Sobol. These indices allow the user to understand the relative influence of each input parameter on the determination of the optimal filter length. The results provide insight into which variables contribute most to the uncertainty in the design outcome. The corresponding indices are shown below.

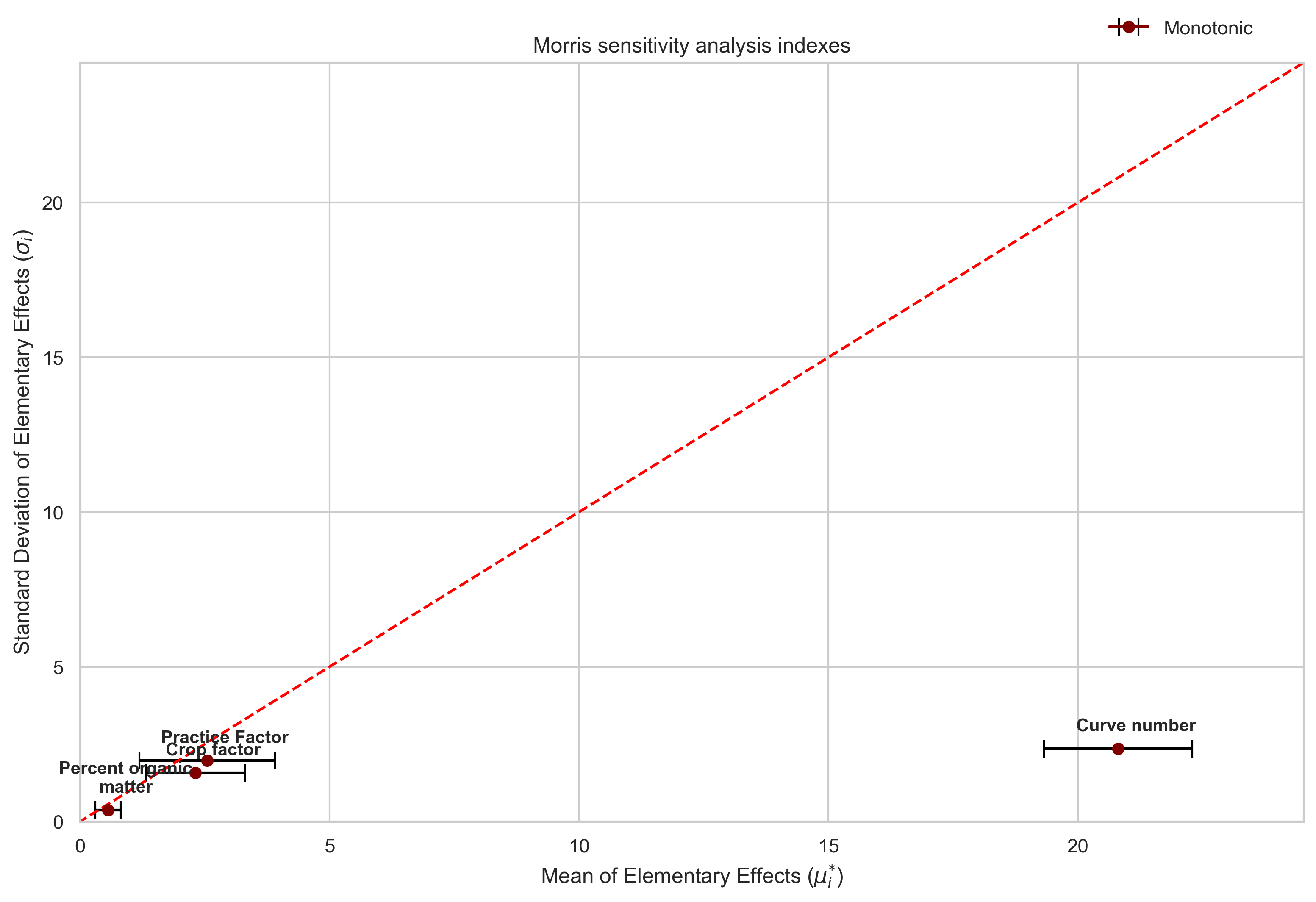


Figure 6.56. **Sensitivity indices from the Morris method**. This plot shows the mean of the elementary effects and the standard deviation for each input parameter, indicating both the overall influence of each parameter on the final buffer length (mean) and the degree of interaction or non-linearity (standard deviation). These results reflect how each parameter contributes to achieving the target *Sediment Delivery Ratio* of 0.25 under the uncertainty framework defined by the user.

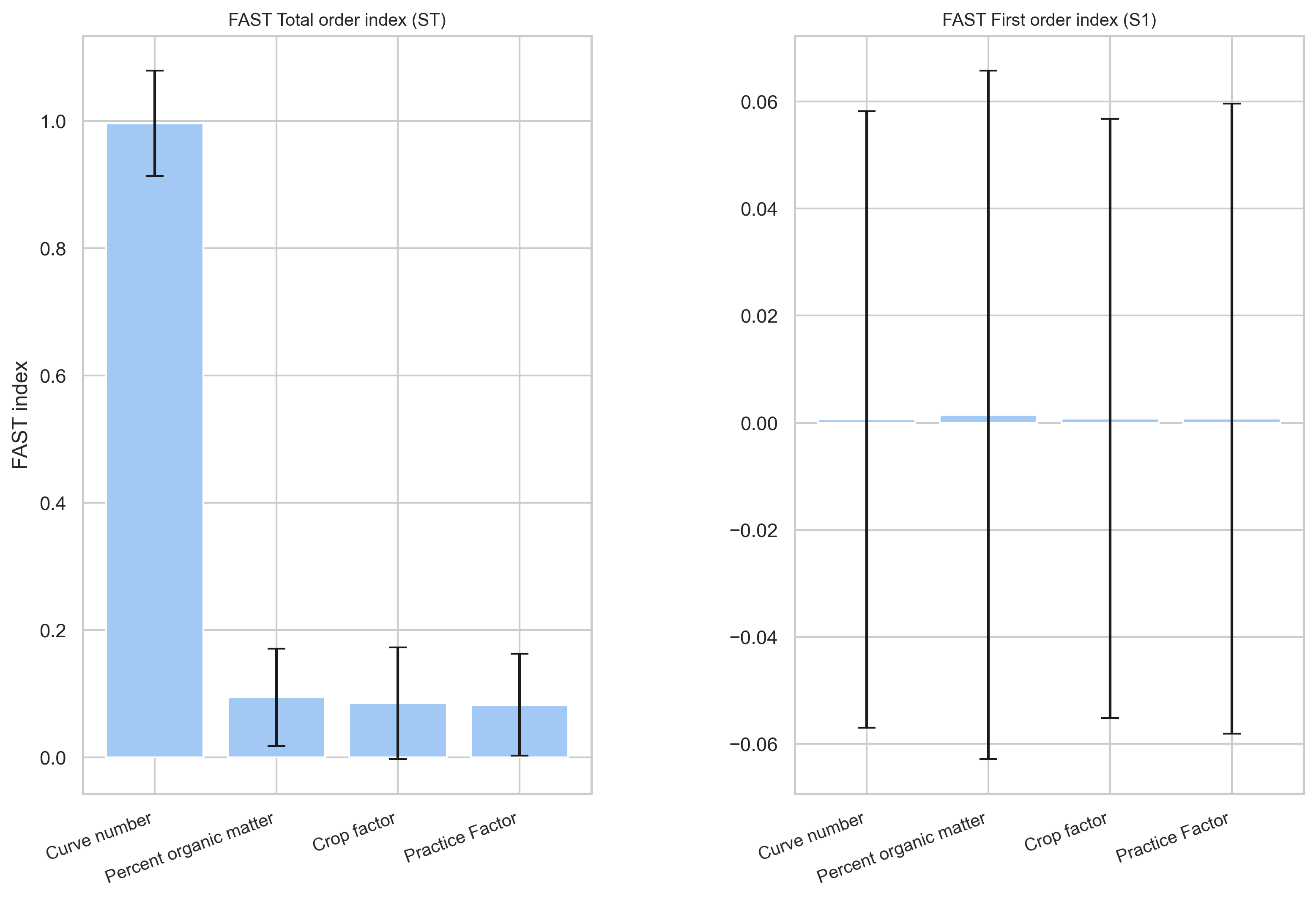


Figure 6.57. **Sensitivity indices from the FAST method.** This figure presents the ***Total-Order Index and the First-Order Index*** computed using the Fourier Amplitude Sensitivity Test (FAST). The first-order index measures the direct contribution of each parameter to the output variance, while the total-order index accounts for both direct and interaction effects. These indices reveal the relative importance of each parameter in determining the buffer length required to meet the sediment reduction target.

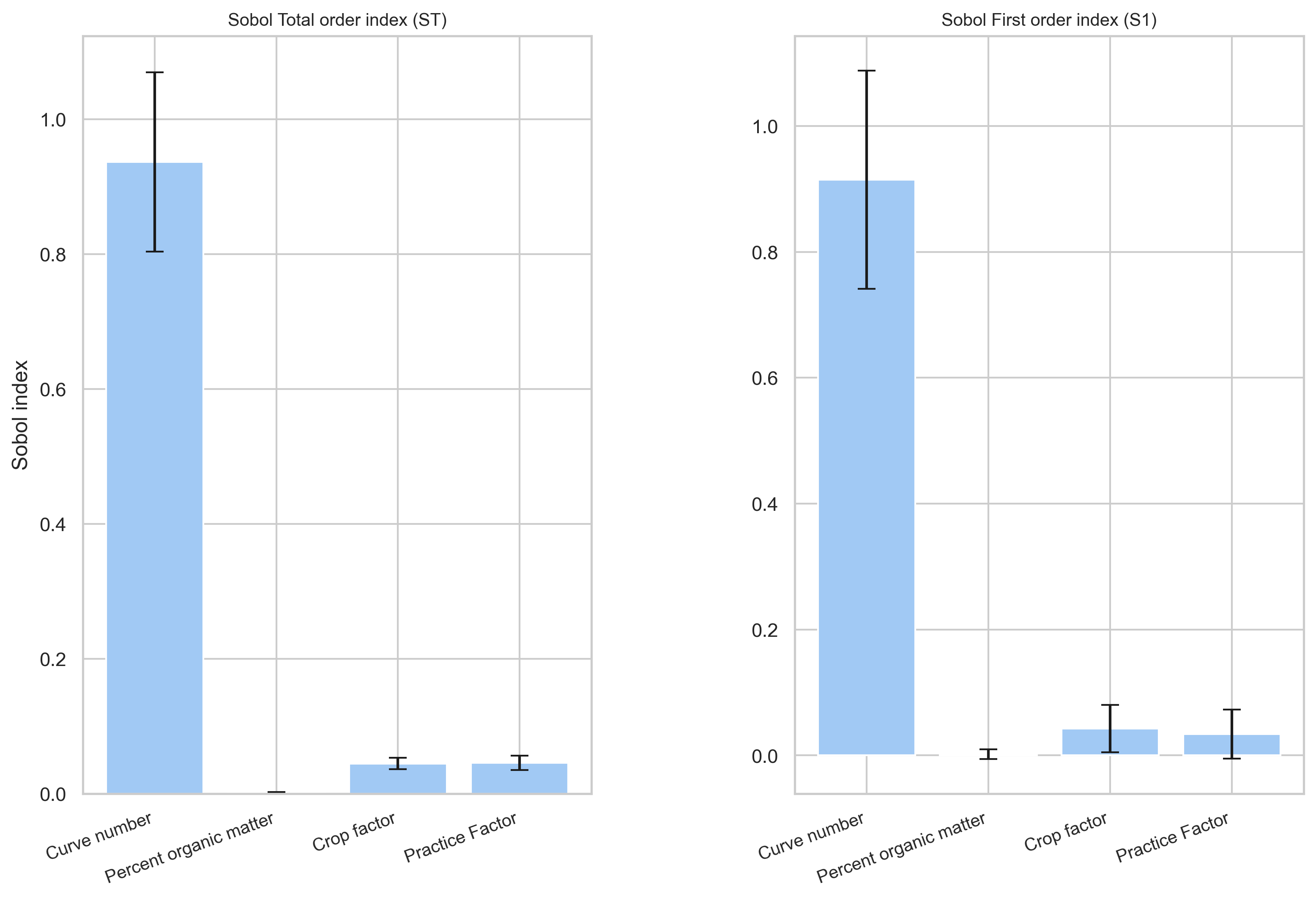


Figure 6.58. **Sensitivity indices from the Sobol method.** The plot displays the ***Total-Order Index*** and the ***First-Order Index*** from the Sobol sensitivity analysis. As with the FAST method, the first-order index captures the individual effect of each parameter, while the total-order index includes higher-order interactions.

In this case, it can be observed that the *Curve Number* is the factor that has the greatest influence on the optimal filter length across the different uncertainty scenarios. This is consistently reflected as the most significant input in all three indices Morris, FAST, and Sobol. As is well known, both FAST and Sobol allow not only for the estimation of the total effect index but also for the decomposition of interaction effects.

In this example, the error bars for both FAST and Sobol, particularly for FAST, appear to be quite large. This is likely due to the use of a base sample size of 256, which is the minimum recommended value for such analyses.

With this information, the user can choose to focus efforts on better characterizing the parameters that most influence the filter length, especially the Curve Number. Doing so would reduce overall system uncertainty, enabling more constrained and accurate input data and potentially avoiding the overdesign of the vegetative filter strip.

As previously mentioned, this design approach can also be applied to runoff reduction or pesticide reduction objectives. Although pesticide-related design was not performed in this particular example, the program fully supports it.

In the case of pesticides, the design procedure can be carried out simultaneously for all pesticides included in the project. However, it is important to note that each pesticide will produce its own set of results. This means that for every pesticide, the program will generate separate graphical outputs, including the frequency distribution of the optimal filter length, the response curve across filter lengths, and the corresponding sensitivity indices. Each pesticide may respond differently due to its specific properties, so the analysis is conducted independently for each one.

### 6.4.4 Sensitivity analysis

A proper sensitivity analysis enables a comprehensive evaluation of the model from multiple perspectives. It can help detect technical flaws in the model, identify critical regions within the input space (including potential interactions), establish research priorities, simplify complex models, verify whether different policy options lead to distinguishable outcomes, and anticipate possible falsifications or misinterpretations of the analysis results.

As explained in the introduction, one of the proposed methodologies for model evaluation involves combining two types of sensitivity analysis. The first step consists of a qualitative ranking of input parameters to identify which ones most influence the outputs. This can be done using the modified method of Morris (Campolongo et al., 2005; Morris, 1991). The Morris method is particularly useful because, with a relatively small number of simulations, it can rank input factors according to their relative effect on the model output and also provide an indication of possible interactions. Based on the results of this preliminary screening, a second, more computationally intensive method based on variance decomposition, such as the Fourier Amplitude Sensitivity Test (FAST) (Saltelli, 1999) or the Sobol method (Sobol, 1990), can be applied, focusing only on the most influential parameters identified previously.

One of the key advantages of the current interface is that the entire sensitivity analysis workflow can now be performed entirely within the program. This includes the sampling of input parameters, the execution of simulations, and the analysis and visualization of results. In contrast, in the previous graphical user interface, this process was not fully integrated: the sampling and the analysis of sensitivity results had to be carried out using external software such as SimLab. As a result, the previous workflow required multiple programs and manual data exchanges, whereas the new interface provides a seamless, all-in-one environment for conducting sensitivity analysis.

This statistical framework for model evaluation is applicable to a wide range of models, but it is especially efficient when dealing with computationally expensive models or when a large number of parameters must be evaluated simultaneously (Muñoz-Carpena et al., 2007). For this reason, the current graphical user interface offers the possibility to perform sensitivity analyses using Morris, FAST, and Sobol methods, as well as a simpler one-at-a-time (OAT) analysis in which only one parameter is varied at a time. In addition to removing the need for external tools such as SimLab, a key improvement over the previous interface is the ability to run a Sobol analysis, which was not previously supported, as only FAST was available as a variance-based method.

Starting with the local sensitivity analysis, which consists of individually modifying the value of a single parameter while keeping the others fixed, one notable improvement over the previous graphical user interface for VFSMOD is the significant increase in the number of parameters available for sensitivity analysis. In the earlier version, only 8 parameters, split between UH and VFSMOD inputs, could be assessed. In contrast, the current interface allows up to 57 parameters to be evaluated, representing a substantial expansion of the model's analytical capacity.

Although the One-At-A-Time (OAT) method involves varying one parameter at a time while keeping all others constant, the interface provides a way to streamline the process when analyzing multiple parameters. Instead of requiring the user to run separate simulations manually for each parameter variation, the interface allows the user to select all the parameters of interest at once. The program then automatically generates and runs all the necessary simulations in a single batch, where each simulation modifies just one parameter. This greatly improves efficiency and usability while preserving the logic of the OAT method.

In this example, the following parameters have been selected for the local sensitivity analysis:

Table 6.5. **Parameters used in the One-at-a-Time sensitivity analysis.** This table displays the input parameters selected for the One-at-a-Time (OAT) sensitivity analysis, including their **base values**, **minimum and maximum bounds**, and the **increment step** applied during the iterative evaluation.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Parameter** | **Base** | **Minimum** | **Maximum** | **Increment** |
| Rainfall (mm) | 100 | 20 | 200 | 20 |
| Curve number | 81 | 70 | 90 | 2 |
| Width of the Strip (m) | 3.87 | 3 | 10 | 1 |
| Filter Manning's n | 0.4 | 0.2 | 0.5 | 0.05 |

For each selected parameter, the user must specify a base value, which is typically set as either the mean or the median value and is generally the value used in standard model simulations. Additionally, the user must define the minimum and maximum values of the parameter, as well as the increment step. The model will then iterate over this defined range, modifying only the selected parameter while keeping all others fixed.

During this process, the program automatically adjusts the input files accordingly. If the parameter affects only the filter area and not the source area, the program will skip the execution of the UH component and only run VFSMOD. This optimization reduces computation time. For instance, modifying the Width of the Strip does not influence any outputs from UH, and therefore UH execution is omitted in that case.

Once all simulations have been completed, the results are saved in a CSV file. These results can be reviewed through the ***Local Results of Sensitivity Analysis*** tab, where the user can select any CSV file containing One-At-A-Time sensitivity analysis data. If the analysis has just been completed, the path to the newly created file will be automatically displayed. However, the user can also manually browse and select previously generated result files.

The following figure shows the interface as it appears after running a One-At-A-Time sensitivity analysis for the current example.

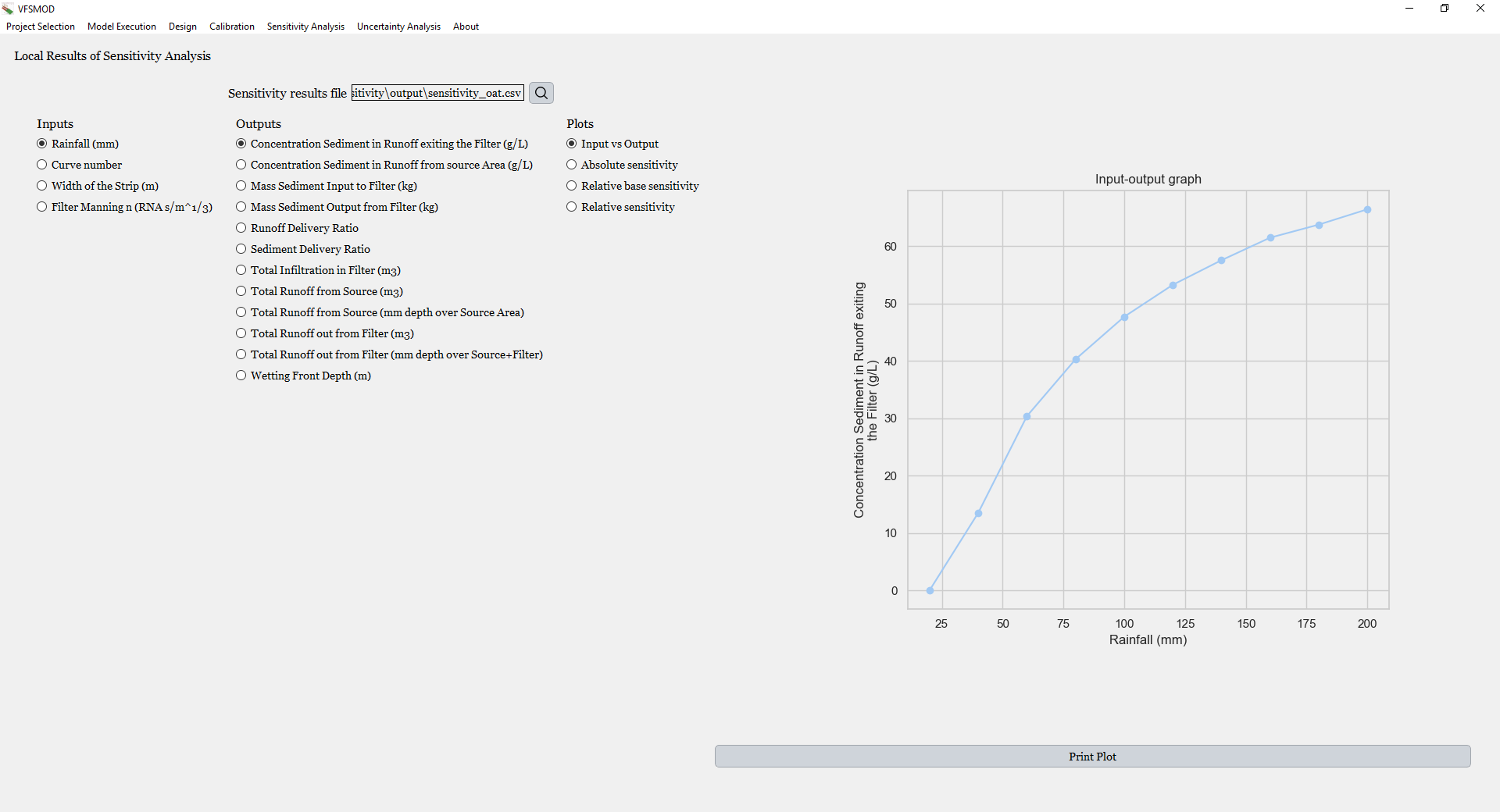


Figure 6.59. **Interface displaying the results of the One-at-a-Time sensitivity analysis.**  
At the top, the interface shows the file path corresponding to the results generated from the OAT sensitivity analysis. Below, on the left panel, the user can select the **input parameter**, the **model output**, and the **type of plot** to visualize. Available plot types include Input vs Output, Absolute Sensitivity, Relative Base Sensitivity, and Relative Sensitivity. The selected plot is dynamically displayed on the right.

The dialog shown in Figure 6.63 allows users to visualize the results of the One-At-A-Time sensitivity analysis. On the left-hand side, there is a column labeled *Inputs*, where users can select which input variable they want to analyze. As mentioned earlier, multiple inputs can be included in a single One-At-A-Time execution, even though each input is analyzed individually.

Next to it is the *Outputs* section. For each variation of an input parameter, the program records the corresponding values of all available outputs. When an output is selected from the list, the graph automatically updates to reflect its variation across the selected input range.

The right section of the dialog displays several types of plots. These are the same visualization types that were available in the previous version of the graphical user interface:

* **Input vs Output**: Shows the direct relationship between the input parameter and the selected output.
* **Absolute Sensitivity**: Displays the change in output with respect to the change in the input.
* **Relative Base Sensitivity**: Computed as (Output−BaseOutput)/(Input−BaseInput)×(BaseInput/BaseOutput). This reflects the relative change in output normalized to the base values of input and output.
* **Relative Sensitivity**: Computed as  
  (ΔOutput/ΔInput)×(Input/Output). This version is similar to the previous one but without relying on a predefined base value.

The following four figures illustrate the results for the Width of the Strip. Each plot corresponds to one of the four sensitivity analysis visualizations described previously: *Input vs Output*, *Absolute Sensitivity*, *Relative Base Sensitivity*, and *Relative Sensitivity*.

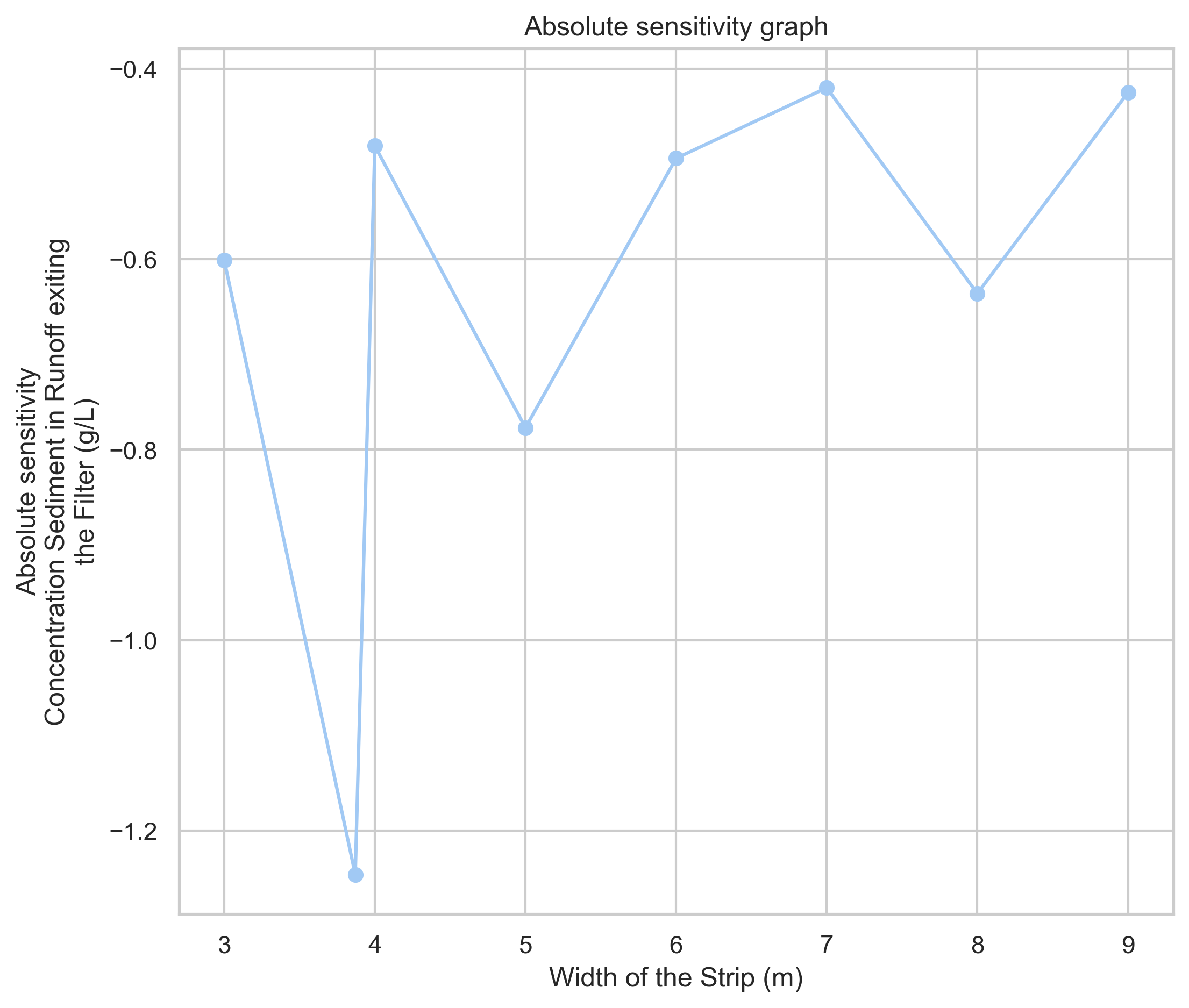
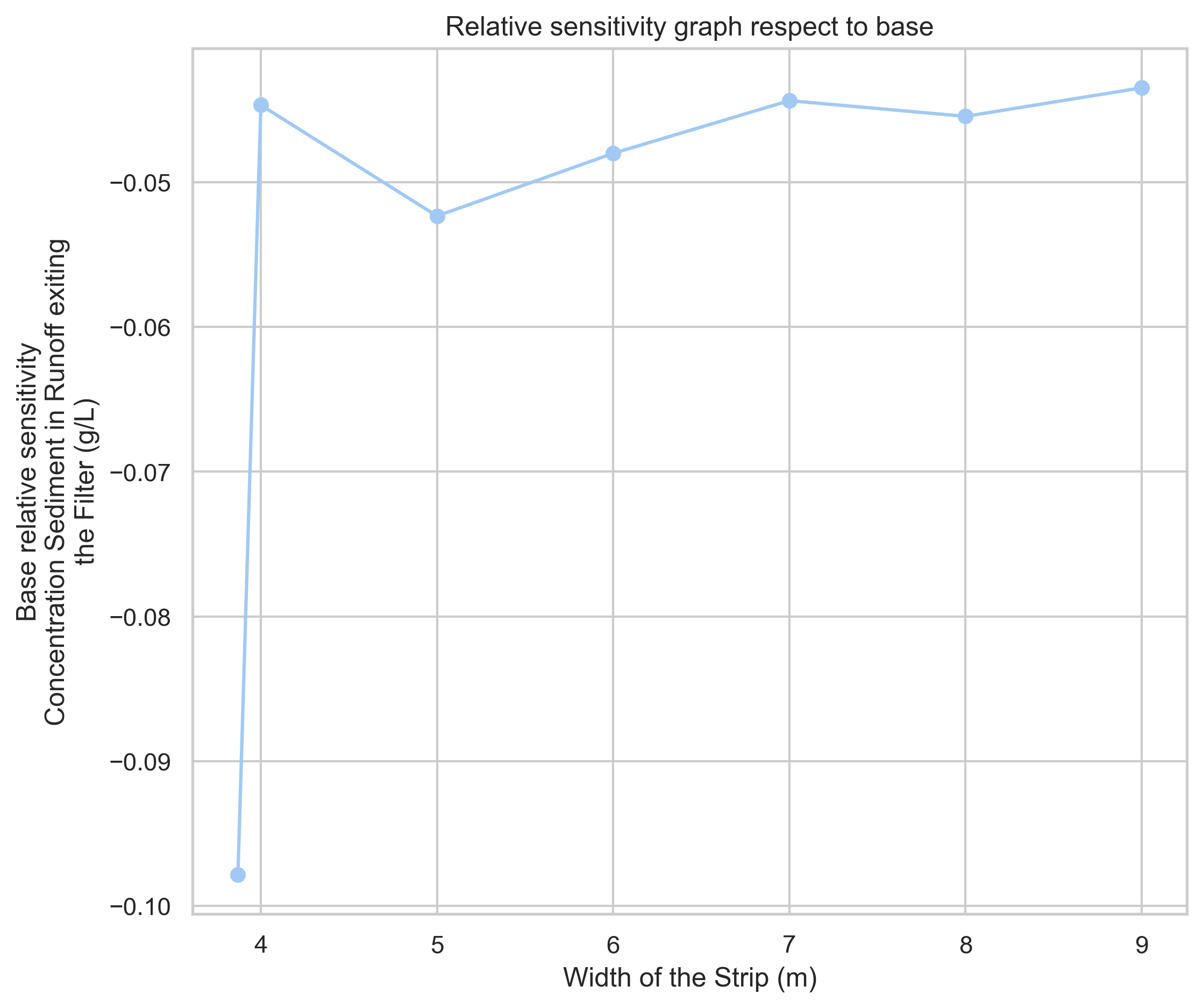
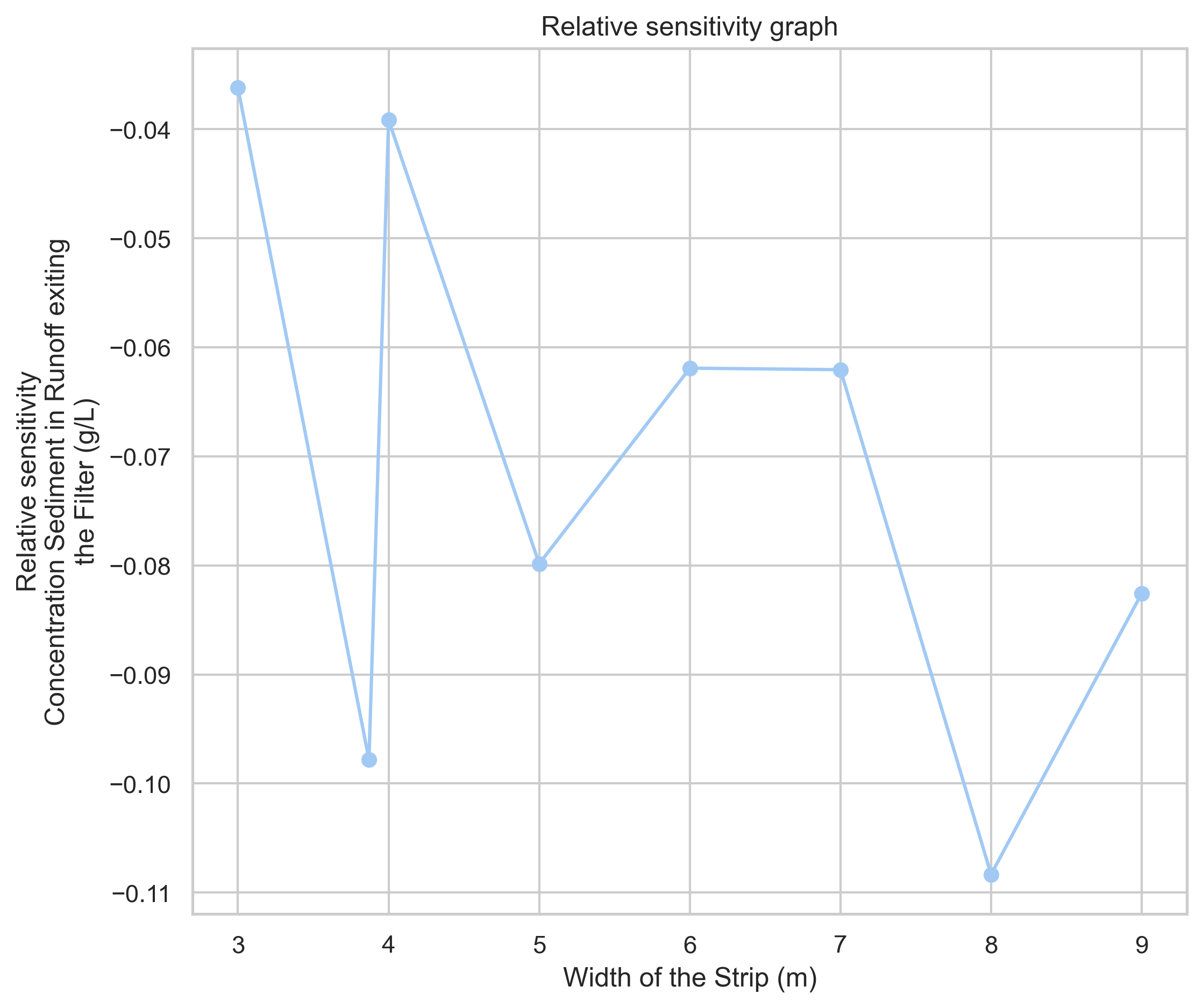
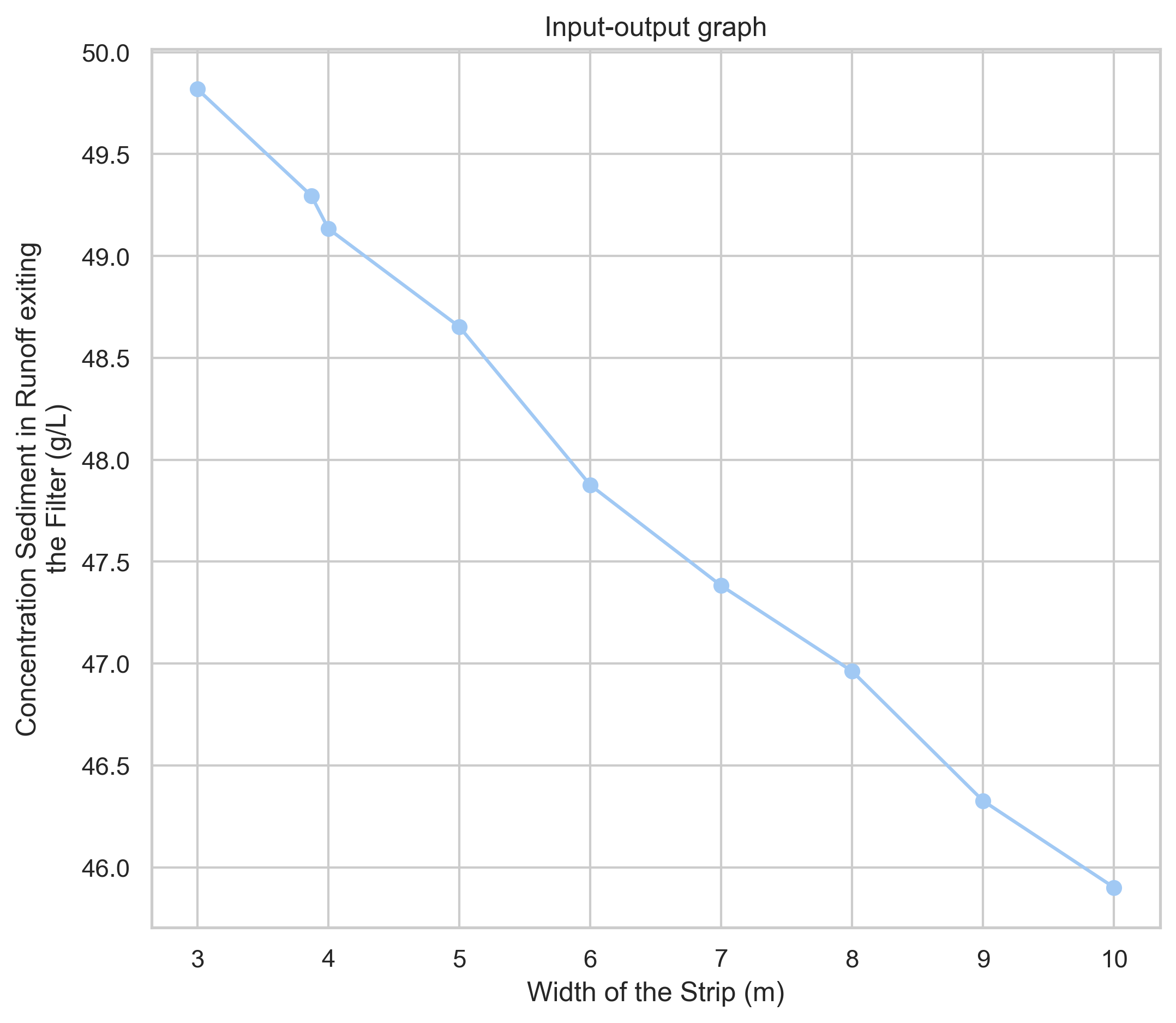


Figure 6.60. **Sensitivity analysis results for the parameter *Width of the Strip* using the One-at-a-Time method.** The figure displays the four available plots for evaluating the influence of the parameter ***Width of the Strip*** on the output variable Concentration of Sediment in Runoff exiting the filter. Starting from the top left and moving clockwise: Input vs Output, which shows the direct relationship between input variation and model response;  
Relative Sensitivity, which expresses the variation of the output relative to changes in the input;  
Relative Base Sensitivity, which measures sensitivity with respect to the base value of the input;  
Absolute Sensitivity, representing the absolute change in output per unit change in input.

Global sensitivity analysis offers a more comprehensive understanding of how model inputs influence outputs by exploring the entire input space rather than focusing on local behavior. Unlike traditional local methods, which typically vary one parameter at a time (OAT) and are limited to narrow ranges around a fixed point, global methods can capture both first-order effects and higher-order interactions, making them particularly effective for nonlinear or complex models (Saltelli et al., 2005).

A global sensitivity analysis can also be performed using the **Morris, FAST (Fourier Amplitude Sensitivity Test)**, and **Sobol** methods. In this example, all three have been conducted for illustrative purposes. The Morris method was executed using 30 trajectories, while both FAST and Sobol were configured with a sample size of 256. This resulted in a total of **150 model runs** for Morris, **1,024 runs** for FAST, and **2,560 runs** for Sobol.

The parameters included in the analysis are listed below, along with their corresponding probability distributions:

Table 6.6. **Parameters and distributions used in the global sensitivity analysis.**  
This table summarizes the parameters included in the global sensitivity analysis, along with their associated **probability distributions**. The information includes the **type of distribution** and the distribution parameters.

|  |  |  |
| --- | --- | --- |
| **Parameter** | **Distribution Type** | **Distribution parameters** |
| Rainfall (mm) | Uniform | Min: 20, Max: 200 |
| Curve number | Triangular | Min: 70, Peak: 85, Max: 90 |
| Width of the Strip (m) | Normal truncated | Min: 3, Max: 10, Mean: 5, Stdv: 0.1 |
| Filter Manning's n | Uniform | Min: 0.2, Max: 0.5 |

The user interface displayed after completing the global sensitivity analysis is shown below. In this case, up to three types of sensitivity analysis results can be generated, one for Morris, one for FAST, and one for Sobol. The interface includes three line edits to input the file paths corresponding to each of these methods. Once the execution is completed, the program automatically creates a file containing the results of the selected sensitivity analysis and inserts the path to this file in the corresponding field of the dialog for immediate access and visualization.

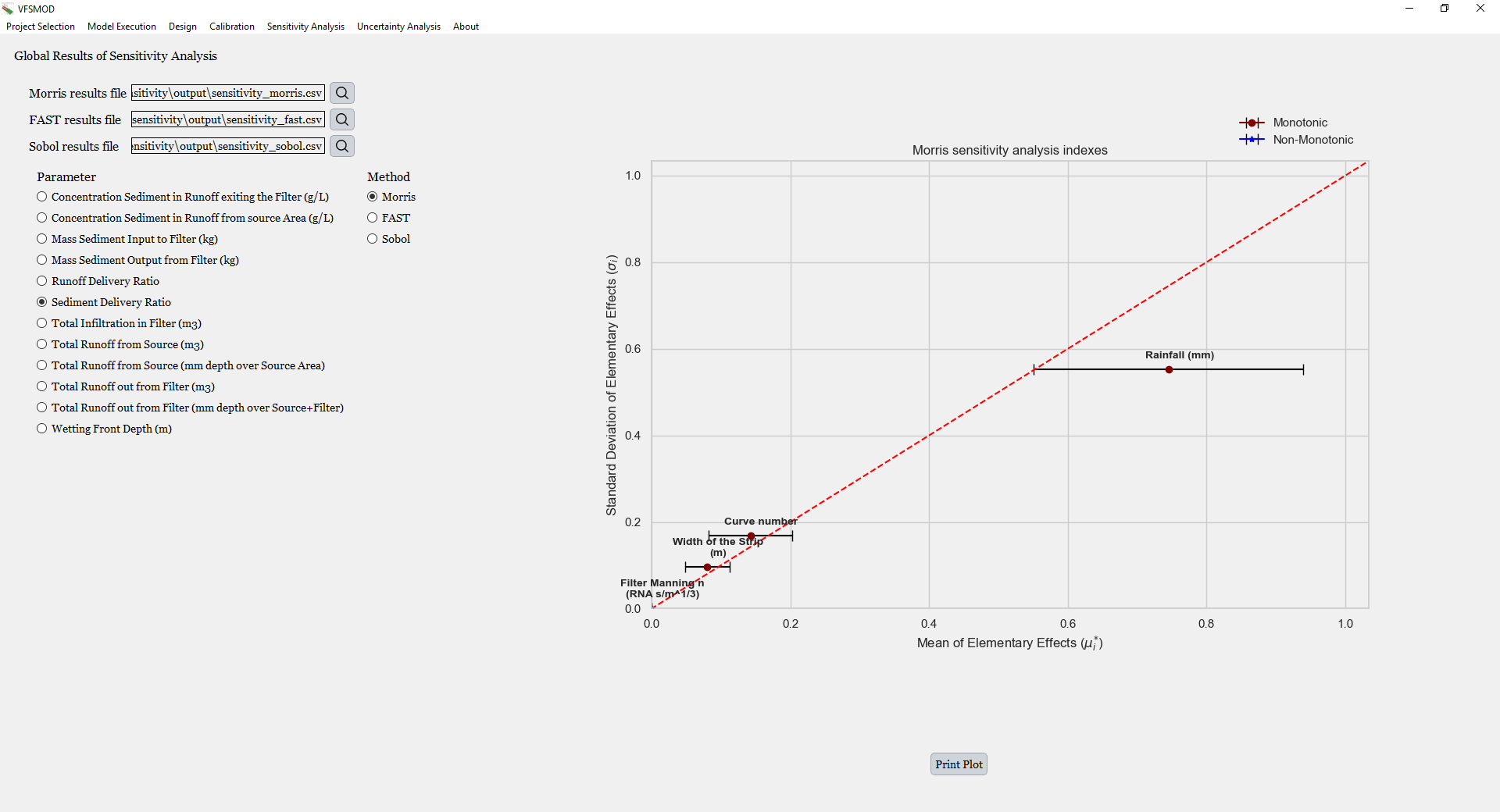


Figure 6.61. **Dialog after performing the global sensitivity analysis.** The interface displays the **file paths** corresponding to the results of each sensitivity method: ***Morris***, ***FAST***, and ***Sobol***, which can be loaded by specifying the path in the corresponding input field. Below, the user can **select the output variable** and the **sensitivity method** to visualize. On the right, the corresponding **sensitivity plot** is shown. In the example, the ***Sediment Delivery Ratio*** output is displayed for the **Morris method**. The plot updates dynamically depending on the selected method and output.

In the first column, the names of the parameters are displayed. When a parameter is selected, the corresponding sensitivity index graph updates automatically. The number and type of parameters shown in this panel may vary depending on whether the *Water Quality* module is enabled. If pesticide modeling is included, additional outputs such as the Pesticide Delivery Ratio and the Leachate Depth will be available **for each individual pesticide** being evaluated.

The interface also includes a selector to choose which sensitivity analysis method’s results to display. Since in this example the three methods (Morris, FAST, and Sobol) were all executed, the user can switch between them to view the corresponding sensitivity indices. Upon selection, the appropriate plot with the sensitivity indices for the selected method is shown. These indices are precomputed and stored in a CSV file, so whenever a new output or sensitivity method is selected, the interface simply retrieves the data from the file, and no additional calculations are required.

An example of the sensitivity index plots for the methods Morris, FAST, and Sobol is shown below. These results correspond to the execution performed for the output ***Total Infiltration in Filter***.

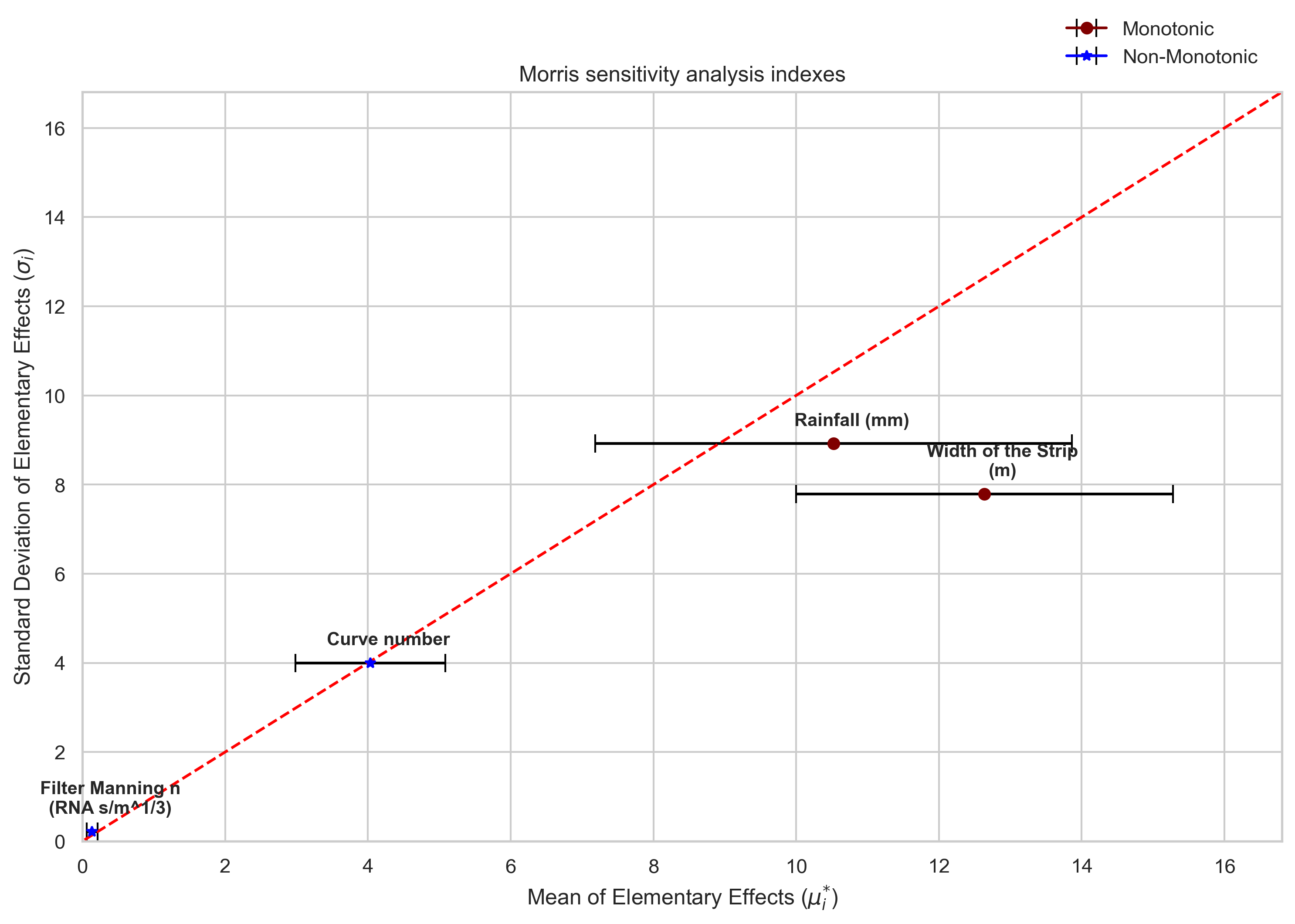


Figure 6.62. **Sensitivity analysis results for the output variable *Total Infiltration in the Filter***.  
This figure shows the impact of different model parameters on total infiltration using the Morris method. The x-axis represents the mean of elementary effects. The y-axis shows the standard deviation of elementary effects. Confidence intervals around the mean help evaluate the robustness of each estimate. Additionally, the plot highlights whether the sensitivity is monotonic or non-monotonic, offering insights into the stability of the parameter influence.

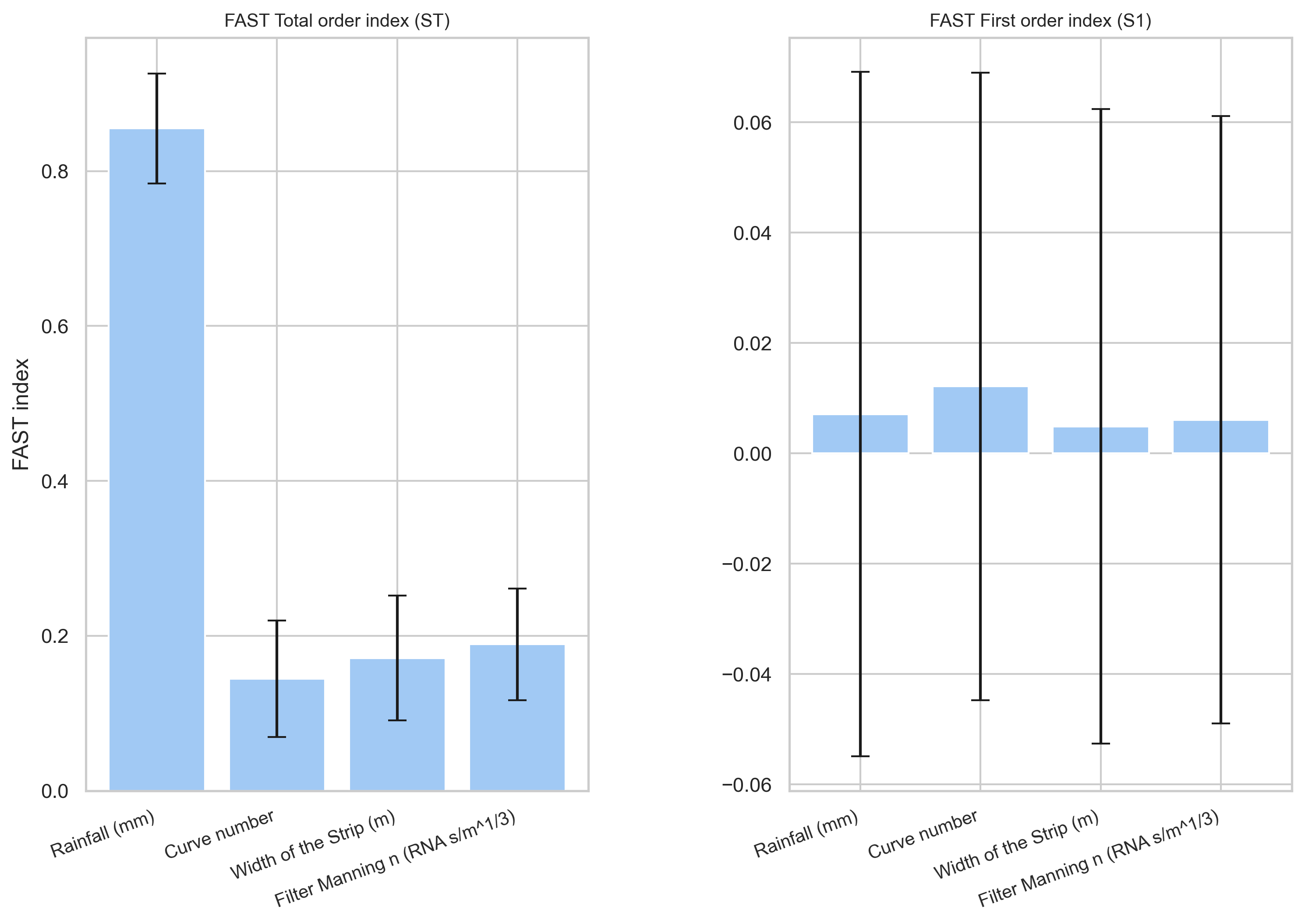


Figure 6.63. **FAST sensitivity indices for the output variable Total Infiltration in the Filter.**  
The figure shows the ***Total Order Index*** and ***First Order Index*** for each parameter, including their associated confidence intervals.

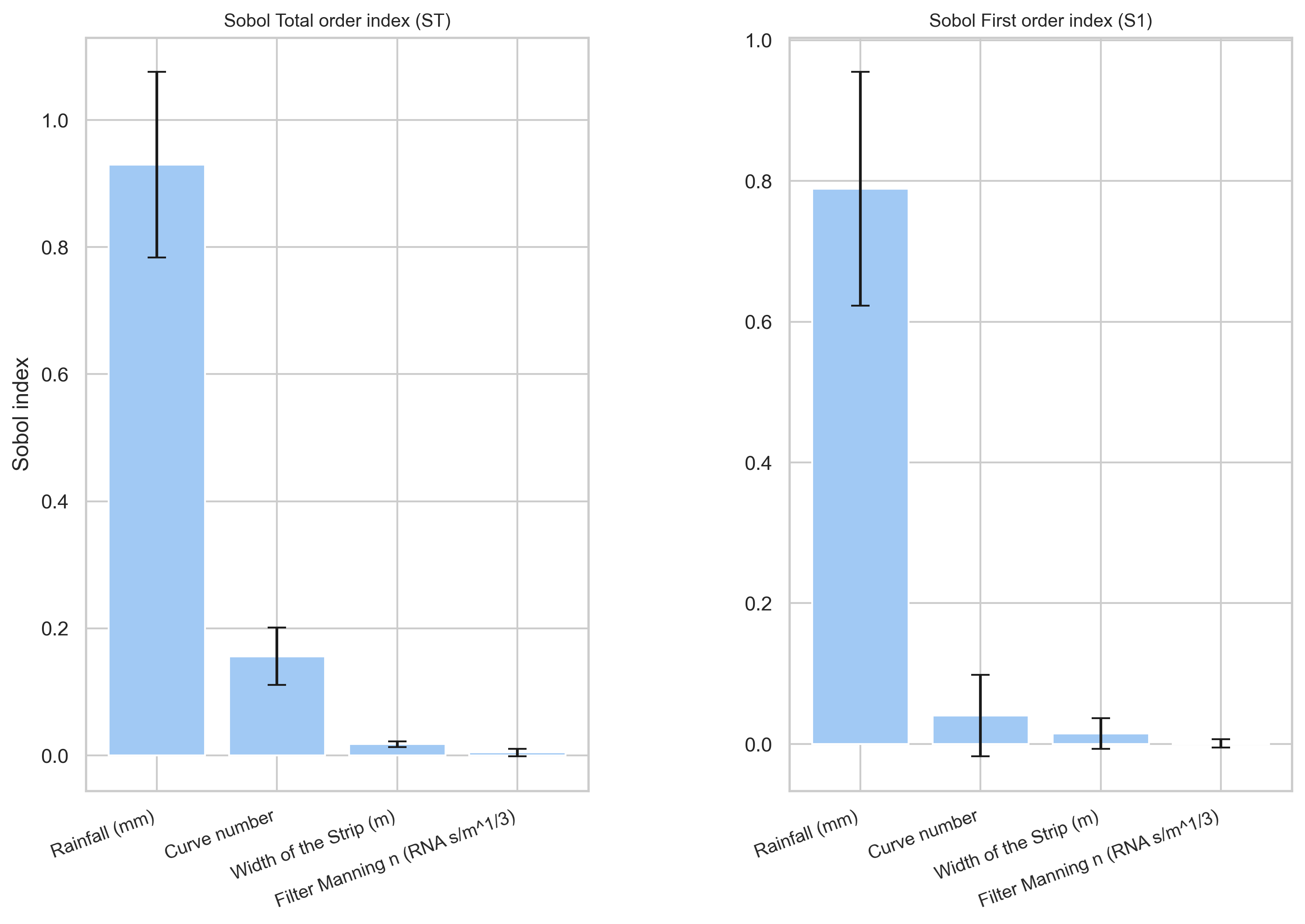


Figure 6.64. **Sobol sensitivity indices for the output variable Total Infiltration in the Filter.**  
This figure presents the ***Total Order Index*** and ***First Order Index*** for each parameter, with confidence intervals shown for each estimate.

In the case of the Morris method, two sensitivity measures are shown for each parameter. The first is the **mean of the absolute values of the elementary effects (**μ\***),** which provides an estimate of the overall influence of the parameter on the selected output. The second is the **standard deviation of the elementary effects (σ)**, which reflects higher-order effects such as non-linear behavior or interactions with other parameters. Since the effect of a parameter on the output may not always be monotonic, Campolongo et al. (2005) recommended using the **absolute mean (**μ\***)** to avoid potential cancellation of opposing effects due to sign changes. This approach is adopted here. To indicate whether a parameter behaves monotonically or not, the difference between the standard mean of elementary effects and the absolute mean is examined. If this difference exceeds 5%, the parameter is considered **non-monotonic**. In the plot (see Figure 6.66), monotonic and non-monotonic parameters are distinguished using different symbols.

In this study, the Morris method has been primarily used to **rank parameters** by importance due to its relatively low computational cost. However, the FAST and Sobol methods provide more **reliable quantitative results**, as they are based on more simulations and a **less structured sampling scheme**, which allows better exploration of the input space (Saltelli et al., 2004).

For the Sobol analysis, the **first-order effects** account for more than **80% of the total variance**, indicating that the model exhibits a **high degree of additivity**. This also suggests that the model could be effectively calibrated through **individual parameter adjustments**. Furthermore, the results show that **rainfall alone is responsible for over 80% of the variance** in the output ***Total Infiltration in Filter***, underscoring its dominant influence. Compared to the One-at-a-Time (OAT) approach, these global sensitivity analyses offer a deeper understanding of the model behavior, without requiring prior assumptions about model additivity or linearity (Muñoz-Carpena, Zajac, et al., 2007).

### 6.4.5 Uncertainty analysis

The most effective way to characterize model uncertainty is through the **Probability Density Functions (PDFs)** of the model outputs (Haan et al., 1995). One common approach to estimate these distributions is the **First Order Approximation (FOA),** which allows the expected value and variance of the output to be approximated using prior knowledge of model linearity and the variances and covariances of the input variables (Morgan & Henrion, 1990).

However, when this prior information is not available, an alternative is the use of **Monte Carlo simulations**, which involve multivariate sampling from the input distributions and using the resulting simulations to construct the output PDFs. Unlike FOA, Monte Carlo methods do not rely on assumptions of model linearity, but they do require that the probability distributions of the inputs are specified.

As mentioned in the introduction, these output PDFs can be obtained from the same model evaluations used for sensitivity analysis. That is, the interface allows uncertainty analysis to be performed either **directly from previously generated sensitivity analysis files,** or **independently via a dedicated Monte Carlo simulation.**

The user interface for uncertainty analysis is virtually identical to that of sensitivity analysis, with the key difference being that no sensitivity method needs to be selected. The only required input is the number of simulations to be performed. Once the runs are completed, the corresponding **results dialog** is automatically populated and made available for visualization. In this case, the same input parameters used in the sensitivity analysis were employed for the uncertainty analysis, along with their corresponding probability distributions. As with the sensitivity analysis, the user can include up to 57 different parameters in the evaluation. This represents a substantial improvement over the previous version of the program, where only 8 parameters, split between UH and VFSMOD inputs, could be used.

The following dialog is displayed after the execution of the uncertainty analysis.

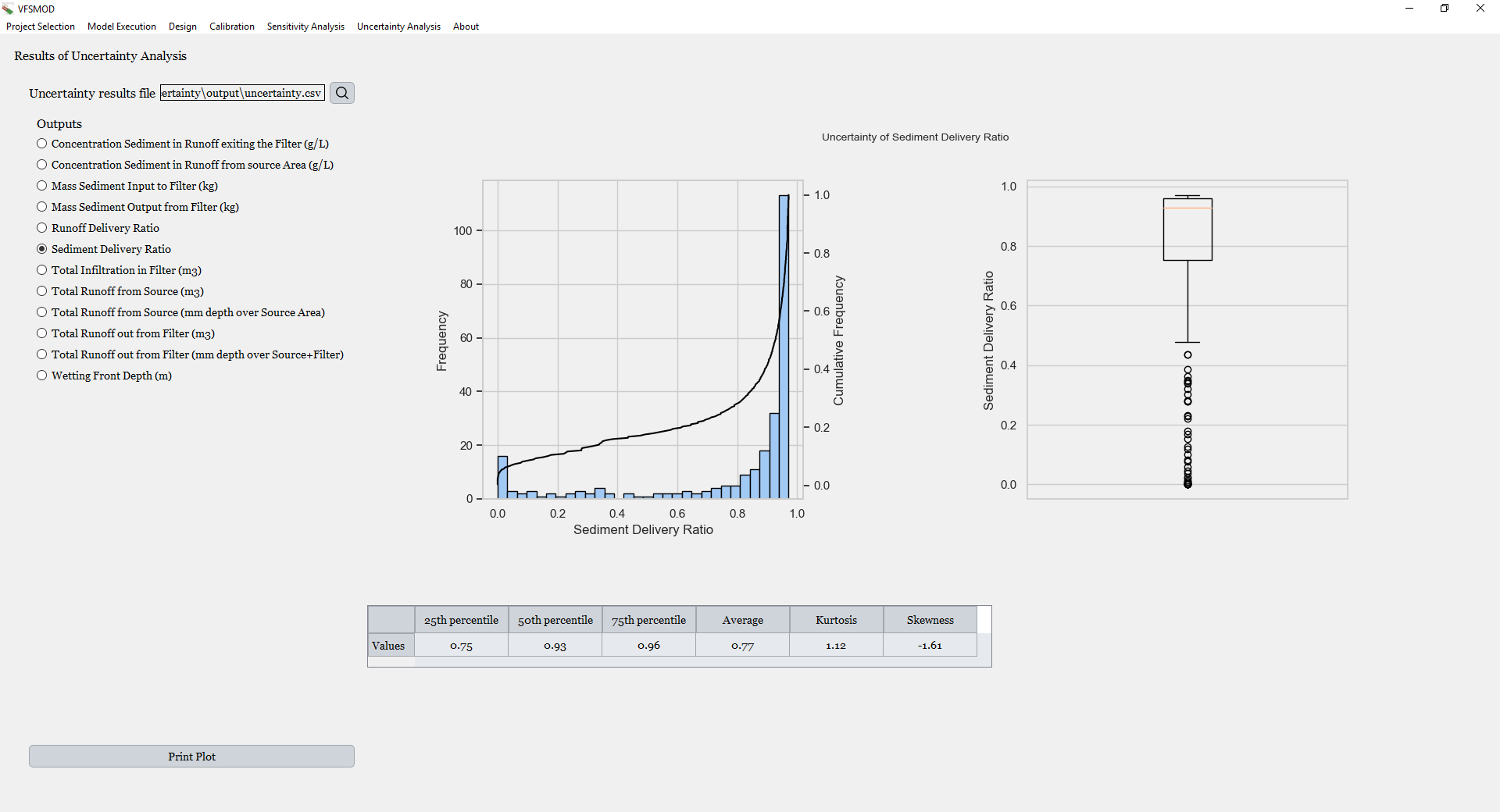


Figure 6.65. **Dialog showing the results of the uncertainty-based design.** At the top, the path to the file containing the uncertainty results is displayed. Below, the user can select the output variable to visualize. On the right, two graphs are shown: a **frequency and cumulative frequency plot**, and a **box-and-whisker diagram**. At the bottom, a table summarizes key statistical metrics: the 25th, 50th, and 75th percentiles, the mean, kurtosis, and skewness.

As shown, there is a line edit that displays the file path of the output file created during the execution. This file is automatically generated and loaded once the analysis is completed.

Below that, the available **outputs** can be selected. As in the sensitivity analysis interface, if pesticides are included in the project, additional outputs, such as the *Pesticide Delivery Ratio* and *Leachate Depth* for each pesticide, will also appear in the list. When an output is selected, both the **graphs** and the **summary table** at the bottom update automatically.

The first graph on the left displays both the **frequency distribution** and the corresponding **cumulative distribution** of the selected output. On the right, a **box-and-whisker plot** illustrates the statistical distribution of the same output, providing a clear visual representation of its variability.

At the bottom, a **summary table** is shown, which includes key statistical descriptors such as the 25th, 50th, and 75th percentiles, the **mean**, as well as the **kurtosis** and **skewness** of the distribution.

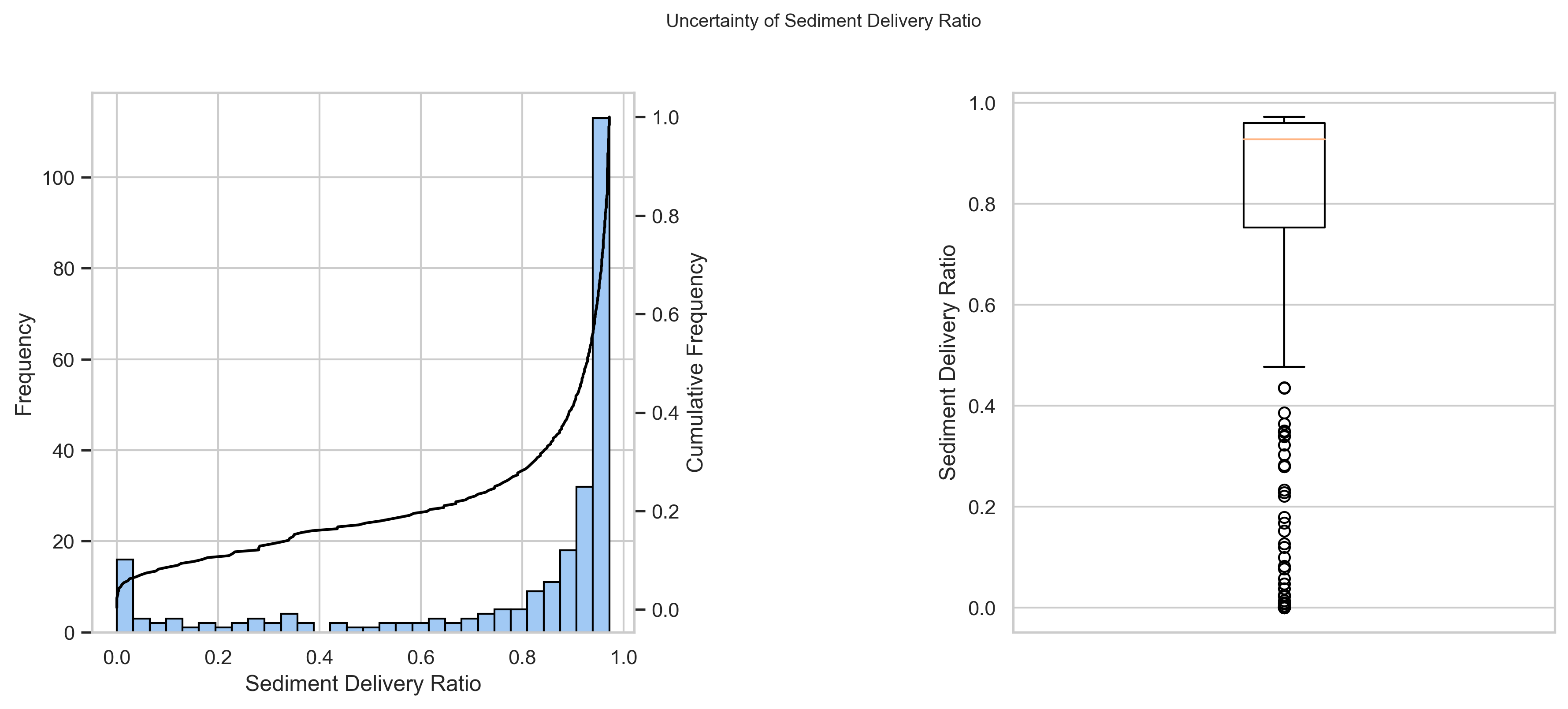
For further detail, the following example illustrates the case of the *Sediment Delivery Ratio*. 

Figure 6.66. **Visualization of uncertainty-based design results for the output variable *Sediment Delivery Ratio*.** On the left, a frequency and cumulative frequency plot illustrates the distribution of *Sediment Delivery Ratio* values under uncertainty. On the right, a **box-and-whisker plot** summarizes the spread of all *Sediment Delivery Ratio* values obtained from the simulations.

The uncertainty analysis shares some similarities with the uncertainty-based design described in Section 6.6.2. In both cases, the aim is to assess how uncertainty in the system’s inputs affects model outputs. However, the focus and implications of each approach differ.

In uncertainty-based design, the goal is to determine how the optimal filter length changes across a range of plausible input scenarios. For each scenario, defined by a specific combination of uncertain input values, the model performs an iterative process to identify the filter length that satisfies a predefined target, such as a specific *Sediment Delivery Ratio*.

In contrast, the uncertainty analysis explores how a given model output (e.g., SDR) varies under input uncertainty, typically assuming a fixed filter configuration. That is, the model does not search for an optimal design in each scenario, but rather shows how the performance metric responds when uncertain parameters vary.

In short, **uncertainty-based design** is useful for determining the **filter length required** to meet a performance objective under system uncertainty and within a given confidence interval. **Uncertainty analysis**, on the other hand, helps evaluate the **variability of a key output** (such as the Sediment Delivery Ratio) **given a selected filter length**, providing insight into the uncertainty or confidence interval of that output under the assumed system variability.

## 6.5 Conclusions

The new **VFSMOD** graphical interface enhances the design of vegetative filter strips by integrating statistical calibration and uncertainty analysis into the workflow. Previously, the design was purely deterministic: system parameters were fixed, and only the filter length was iteratively adjusted to meet pollutant reduction targets. In the new approach, the process begins with rigorous model calibration using observed data—first optimizing runoff parameters and then, if available, extending calibration to sediment and pesticide transport. Once calibration criteria are satisfied, users can perform a deterministic design to obtain an initial estimate of filter length, followed by a more advanced design under uncertainty that accounts for probabilistic input distributions. Sensitivity analysis helps identify the most influential parameters, and the resulting uncertainty-based design provides confidence intervals or exceedance probabilities for different filter lengths. This iterative, modular methodology enables the design of filter strips that are not only effective under average conditions but also robust under realistic variability, making the updated interface a comprehensive, reliable, and transparent decision-support tool for environmental management.

The primary objective of this chapter was to demonstrate the full range of functionalities integrated into the newly developed Graphical User Interface (GUI) for the VFSMOD modeling system. To this end, the chapter has systematically presented each component of the tool, beginning with the basic execution of the model and continuing with the calibration process, which constitutes the core of the model reliability phase. Beyond the reliability stage, the GUI also supports advanced functionalities such as the deterministic and uncertainty-based design of vegetative filter strips, as well as comprehensive tools for sensitivity and uncertainty analysis. While some of these features were present in the earlier version, the current interface introduces significant improvements and new modules that enhance both the user experience and the scientific robustness of the modeling workflow.

One of the most significant advancements of this Graphical User Interface is its integration of system uncertainty into the design of VFS, which is one of the core objectives of the VFSMOD modeling framework. Unlike previous versions, the current interface supports uncertainty-based design, enabling the user to account for variability in model parameters, inputs, and responses. This development is particularly relevant given the increasing emphasis on uncertainty in the context of TMDL modeling. Uncertainty has critical implications for policy, regulatory decisions, and environmental management. In the TMDL framework, uncertainty is not only a key consideration in pollutant load assessments but must also be addressed in the design and evaluation of BMPs. The ability of the GUI to present modeling results in terms of probabilities, such as the likelihood of exceeding a performance threshold, enhances the transparency and utility of the outputs. This probabilistic perspective provides decision makers and other stakeholders with more informative and robust support for making science-based decisions, particularly under uncertain environmental and operational conditions.

In addition to incorporating uncertainty into the design phase, the new Graphical User Interface significantly expands the capability for uncertainty assessment by allowing users to conduct uncertainty-based design with all input parameters available in the model, something that was not possible in the previous version of the GUI. This enhancement enables a more comprehensive and flexible exploration of the model’s behavior under variable conditions.

Furthermore, this functionality is now accompanied by more intuitive, detailed, and user-friendly visualizations. The improved graphical outputs allow users to more easily interpret the distribution of results, confidence intervals, and design thresholds, which makes the analysis of uncertainty not only more accessible but also more informative. Compared to the previous interface, this new environment facilitates better understanding and communication of model behavior and design implications under uncertainty.

Alongside the new functionalities developed to incorporate uncertainty in the design and evaluation of VFS, this updated GUI introduces a broad range of improvements over the previous version, making it a highly valuable tool for VFS design**.** One of the most significant enhancements is that the entire interface has been built and compiled in Python as a standalone application, meaning that users are not required to install Python themselves. This eliminates compatibility issues across different Windows versions and has also enabled the GUI to be successfully ported to macOS and Linux.

Other notable advancements, as highlighted in the introduction, include the ability to use all available model input variables in the sensitivity and uncertainty analyses, unlike the previous GUI, which only allowed a limited subset. Additionally, Sobol’s method has been incorporated as a robust alternative to FAST for global sensitivity analysis. Furthermore, the removal of dependencies on external software (such as Simlab) now allows all sensitivity computations to be performed entirely within the same application environment.

The calibration module has also been significantly enhanced. It now supports a full evaluation framework that includes bootstrapping, exceedance probabilities and p-value computation. Calibration can be performed not only using full time-series data such as hydrographs and sedimentographs, but also based on single-value outputs. This was not possible in the previous version and is now implemented without any reliance on MATLAB, avoiding the need for heavy external libraries and installations.

Another major addition is the inclusion of a parameter identifiability analysis prior to calibration, helping users focus on the most influential parameters for model fitting. Furthermore, all computationally intensive tasks, such as deterministic design, uncertainty-based design, identifiability analysis, sensitivity analysis, and uncertainty analysis, can now be executed in parallel. The application automatically detects the number of available cores and distributes the tasks accordingly, significantly speeding up execution time without requiring manual configuration.

A further innovation is the ability to simulate multiple pesticides in parallel, which is fully integrated across all modules, including design, calibration, sensitivity analysis, and uncertainty analysis. The overall interface has also been modernized with smoother graphical transitions, real-time updates, and interactive plotting, while maintaining a significantly lower system resource usage compared to its predecessor.

Together, these advancements make the new Graphical User Interface a powerful and efficient platform for the design and evaluation of vegetative filter strips, fully aligned with the needs of both researchers and decision-makers.

**CHAPTER 7**

# 8. REFERENCES

AgbioInvestor. (2024). *Time and Cost of New Agrochemical Product Discovery, Development and Registration A Study on Behalf of Crop Life International*.

Barfield, B. J., Tollner, E. W., & Hayes, J. C. (1978). *The use of grass filters for sediment control in strip mining drainage. Vol. I: Theoretical studies on artificial media*.

Barfield, B. J., Tollner, E. W., & Hayes, J. C. (1979). Filtration of Sediment by Simulated Vegetation I. Steady-State Flow with Homogeneous Sediment. *Transactions of the ASAE*, *22*(3), 0540–0545. https://doi.org/10.13031/2013.35057

Barling, R. D., & Moore, I. D. (1994). Role of buffer strips in management of waterway pollution: A review. *Environmental Management*, *18*(4), 543–558. https://doi.org/10.1007/BF02400858

Beven, K., & Freer, J. (2001). Equifinality, data assimilation, and uncertainty estimation in mechanistic modelling of complex environmental systems using the GLUE methodology. *Journal of Hydrology*, *249*(1–4), 11–29. https://doi.org/10.1016/S0022-1694(01)00421-8

Bolton, S. M., Ward, T. J., & Cole, R. A. (1991). Sediment‐Related Transport of Nutrients from Southwestern Watersheds. *Journal of Irrigation and Drainage Engineering*, *117*(5), 736–747. https://doi.org/10.1061/(ASCE)0733-9437(1991)117:5(736)

Brakensiek, D.L., Osborn, H.B., Rawls, W. J. (1979). *Field manual for research in agricultural hydrology*. US Department of Agriculture.

Brown, C., Balderacchi, M., van Beinum, W., Capri, E., & Trevisan, M. (2012). *Definition of vegetative filter strip scenarios for Europe*.

C. T. Haan. (1989). Parametric Uncertainty in Hydrologic Modeling. *Transactions of the ASAE*, *32*(1), 0137–0146. https://doi.org/10.13031/2013.30973

Campolongo, F., Cariboni, J., Saltelli, A., & Schoutens, W. (2005). Enhancing the Morris Method. In K. M. Hanson & F. M. Hemez (Eds.), *4th International Conference on Sensitivity Analysis of Model Output* (pp. 369–379).

Chen, H., Carley, D. S., Muñoz-Carpena, R., Ferruzzi, G., Yuan, Y., Henry, E., Blankinship, A., Veith, T. L., Breckels, R., Fox, G., Luo, Y., Osmond, D., Preisendanz, H. E., Tang, Z., Armbrust, K., Costello, K., McConnell, L. L., Rice, P., Westgate, J., & Whiteside, M. (2024). Incorporating the benefits of vegetative filter strips into risk assessment and risk management of pesticides. *Integrated Environmental Assessment and Management*, *20*(2), 454–464. https://doi.org/10.1002/IEAM.4824;SUBPAGE:STRING:FRAMEDPDF;REQUESTEDJOURNAL:JOURNAL:15513793;PAGE:STRING:ARTICLE/CHAPTER

Clinnick, P. F. (1985). Buffer strip management in forest operations: a review. *Australian Forestry*, *48*(1), 34–45. https://doi.org/10.1080/00049158.1985.10674421

Dillaha, T. A., Reneau, R. B., Mostaghimi, S., & Lee, D. (1989a). Vegetative Filter Strips for Agricultural Nonpoint Source Pollution Control. *Transactions of the ASAE*, *32*(2), 0513–0519. https://doi.org/10.13031/2013.31033

Dillaha, T. A., Reneau, R. B., Mostaghimi, S., & Lee, D. (1989b). Vegetative Filter Strips for Agricultural Nonpoint Source Pollution Control. *Transactions of the ASAE*, *32*(2), 513–0519. https://doi.org/10.13031/2013.31033

Dillaha, T. A., Sherrard, J. H., Lee, D., Mostaghimi, S., & Shanholtz, V. O. (1985). SEDIMENT AND PHOSPHORUS TRANSPORT IN VEGETATIVE FILTER STRIPS: PHASE I, FIELD STUDIES. *Paper - American Society of Agricultural Engineers*, 7294. https://www.scopus.com/record/display.uri?eid=2-s2.0-0022184399&origin=recordpage

EC-ESDAC. (2022). *Surface Water Assessment eNabler (SWAN). European Commission JRC European Soil Data Centre (ESDAC), FOCUS DG Santé, Surface Water*. https://esdac.jrc.ec.europa.eu/projects/swan

EPA. (2002). *Considerations in the Design of Treatment Best Management Practices (BMPs) to Improve Water Quality*.

EPA. (2024a). *Clean Water Act Section 303(d): Impaired Waters and Total Maximum Daily Loads (TMDLs)*. https://www.epa.gov/tmdl

EPA. (2024b, June 12). *Summary of the Clean Water Act*. https://www.epa.gov/laws-regulations/summary-clean-water-act

Flanagan, D. C., Foster, G. R., Neibling, W. H., & Burt, J. P. (1989). Simplified Equations for Filter Strip Design. *Transactions of the ASAE*, *32*(6), 2001. https://doi.org/10.13031/2013.31254

Fox, G. A., Muñoz-Carpena, R., & Sabbagh, G. J. (2010). Influence of flow concentration on parameter importance and prediction uncertainty of pesticide trapping by vegetative filter strips. *Journal of Hydrology*, *384*(1–2), 164–173. https://doi.org/10.1016/J.JHYDROL.2010.01.020

Gee, G. W., & Bauder, J. W. (1986). Methods of Soil Analysis, 2nd. Ed., Part I. . In A. Klute (Ed.), *Methods of Soil Analysis, Part 1: Physical and Mineralogical Methods*. John Wiley & Sons, Ltd. https://doi.org/10.2136/SSSABOOKSER5.1.2ED.C15

Gupta, H. V., & Razavi, S. (2018). Revisiting the Basis of Sensitivity Analysis for Dynamical Earth System Models. *Water Resources Research*, *54*(11), 8692–8717. https://doi.org/10.1029/2018WR022668;PAGEGROUP:STRING:PUBLICATION

Hayes, J. C., Barfield, B. J., & Barnhisel, R. I. (1982). *The use of grass filters for sediment control in strip mine drainage. III. Empirical verification of procedures using real vegetation.*

Klein, M., Klein, J., Flade, J., Großmann, D., Türkowsky, D., O’Connor, I., Spycher, S., Reichenberger, S., Sittig, S., Multsch, S., & Thomas, K. (2023). Risk mitigation measures for pesticide runoff: How effective are they? *Pest Management Science*, *79*(12), 4897–4905. https://doi.org/10.1002/PS.7691

Knisel, W. G. (1980). *CREAMS: A Field-Scale Model for Chemicals, Runoff, and Erosion from Agricultural Management Systems. Conservation Res Report No. 26.*

Kuo, Y.-M., Muñoz-Carpena, R., Li, Y., Campbell, K. L., & Parsons, J. E. (2005). Using Vegetative Filter Strips to Reduce Phosphorus Transport from the Phosphorus Mining Areas in Central Florida. *2005 Tampa, FL July 17-20, 2005*. https://doi.org/10.13031/2013.19809

Lapedes, D. N. (1978). *McGraw-Hill Dictionary of Scientific and Technical Terms* (McGraw-Hill (ed.); Second edi).

Lauvernet, C., & Munõz-Carpena, R. (2018). Shallow water table effects on water, sediment, and pesticide transport in vegetative filter strips-Part 2: Model coupling, application, factor importance, and uncertainty. *Hydrology and Earth System Sciences*, *22*(1), 71–87. https://doi.org/10.5194/HESS-22-71-2018,

Legates, D. R., & McCabe, G. J. (1999). Evaluating the use of “goodness-of-fit” measures in hydrologic and hydroclimatic model validation. *Water Resources Research*, *35*(1), 233–241. https://doi.org/10.1029/1998WR900018;CTYPE:STRING:JOURNAL

Magette, W. L., Brinsfield, R. B., Palmer, R. E., & Wood, J. D. (1989). Nutrient and Sediment Removal by Vegetated Filter Strips. *Transactions of the ASAE*, *32*(2), 0663–0667. https://doi.org/10.13031/2013.31054

Morris, M. D. (1991). Factorial Sampling Plans for Preliminary Computational Experiments. *Technometrics*, *33*(2), 161. https://doi.org/10.2307/1269043

Muñoz-Carpena, R. (1993). *Modeling hydrology and sediment transport in vegetative filter strips*. North Carolina State University.

Muñoz-Carpena, R., Fox, G. A., Ritter, A., Perez-Ovilla, O., & Rodea-Palomares, I. (2018). Effect of vegetative filter strip pesticide residue degradation assumptions for environmental exposure assessments. *Science of The Total Environment*, *619*–*620*, 977–987. https://doi.org/10.1016/J.SCITOTENV.2017.11.093

Muñoz-Carpena, R., & Parsons, J. E. (2004). A DESIGN PROCEDURE FOR VEGETATIVE FILTER STRIPS USING VFSMOD-W. *Transactions of the ASAE*, *47*(6), 1933–1941. https://doi.org/10.13031/2013.17806

Muñoz-Carpena, R., & Parsons, J. E. (2020). *VFSMOD-W Vegetative Filter Strips Modelling System MODEL DOCUMENTATION & USER’S MANUAL* (6.x).

Muñoz-Carpena, R., & Parsons, J. E. (2023, March 21). *VFSMOD: Vegetative Filter Strip Modeling System*. https://abe.ufl.edu/faculty/carpena/vfsmod/index.shtml

Muñoz-Carpena, R., Parsons, J. E., & Gilliam, J. W. (1999). Modeling hydrology and sediment transport in vegetative filter strips. *Journal of Hydrology*, *214*(1–4), 111–129. https://doi.org/10.1016/S0022-1694(98)00272-8

Muñoz-Carpena, R., Ritter, A., & Fox, G. A. (2019). Comparison of empirical and mechanistic equations for vegetative filter strip pesticide mitigation in long-term environmental exposure assessments. *Water Research*, *165*, 114983. https://doi.org/10.1016/J.WATRES.2019.114983

Muñoz-Carpena, R., Vellidis, G., Shirmohammadi, A., & Wallender, W. W. (2007). Evaluation of Modeling Tools For TMDL Development And Implementation. *Watershed Management to Meet Water Quality Standards and TMDLS (Total Maximum Daily Load) Proceedings of the 10-14 March 2007, San Antonio, Texas*. https://doi.org/10.13031/2013.22425

Muñoz-Carpena, R., Zajac, Z., & Kuo, Y. M. (2007). Global Sensitivity and Uncertainty Analyses of the Water Quality Model VFSMOD-W. *Transactions of the ASABE*, *50*(5), 1719–1732. https://doi.org/10.13031/2013.23967

Muñoz‐Carpena, R., Fox, G. A., & Sabbagh, G. J. (2010). Parameter Importance and Uncertainty in Predicting Runoff Pesticide Reduction with Filter Strips. *Journal of Environmental Quality*, *39*(2), 630–641. https://doi.org/10.2134/JEQ2009.0300;PAGEGROUP:STRING:PUBLICATION

Parsons, J. E., Daniels, R. B., Gilliam, J. W., & Dillaha, T. A. (1994). *Reduction in sediment and chemical load in agricultural field runoff by vegetative filter strips*.

Parsons, J. E., & Muñoz Carpena, R. (2001). Impact of Uncertainty on the Design of Vegetative Filter Strips. *2001 Sacramento, CA July 29-August 1,2001*. https://doi.org/10.13031/2013.7397

Politis, D. N., & White, H. (2004). Automatic Block-Length Selection for the Dependent Bootstrap. *ECONOMETRIC REVIEWS*, *23*(1), 53–70. https://doi.org/10.1081/ETC-120028836

Reckhow, K. H. (1994a). Importance of scientific uncertainty in decision making. *Environmental Management*, *18*(2), 161–166. https://doi.org/10.1007/BF02393758

Reckhow, K. H. (1994b). Water quality simulation modeling and uncertainty analysis for risk assessment and decision making. *Ecological Modelling*, *72*(1–2), 1–20. https://doi.org/10.1016/0304-3800(94)90143-0

Reichenberger, S., Bach, M., Skitschak, A., & Frede, H. G. (2007). Mitigation strategies to reduce pesticide inputs into ground- and surface water and their effectiveness; A review. *Science of The Total Environment*, *384*(1–3), 1–35. https://doi.org/10.1016/J.SCITOTENV.2007.04.046

Reichenberger, S., Sur, R., Sittig, S., Multsch, S., Carmona-Cabrero, Á., López, J. J., & Muñoz-Carpena, R. (2023). Dynamic prediction of effective runoff sediment particle size for improved assessment of erosion mitigation efficiency with vegetative filter strips. *Science of The Total Environment*, *857*, 159572. https://doi.org/10.1016/J.SCITOTENV.2022.159572

Ritter, A., & Muñoz-Carpena, R. (2013). Performance evaluation of hydrological models: Statistical significance for reducing subjectivity in goodness-of-fit assessments. *Journal of Hydrology*, *480*, 33–45. https://doi.org/10.1016/j.jhydrol.2012.12.004

Saltelli, A. (1999). Sensitivity analysis: Could better methods be used? *Journal of Geophysical Research: Atmospheres*, *104*(D3), 3789–3793. https://doi.org/10.1029/1998JD100042

Saltelli, A., Tarantola, S., Campolongo, F., & Ratto, M. (2004). *Sensitivity Analysis in Practice* (John Wiley & Sons).

Shirmohammadi, A., Chaubey, I., Harmel, R. D., Bosch, D. D., Muñoz-Carpena, R., Dharmasri, C., Sexton, A., Arabi, M., Wolfe, M. L., Frankenberger, J., Graff, C., & Sohrabi, T. M. (2006a). Uncertainity in TMDL models. *Transactions of the ASABE*, *49*(4), 1033–1049. https://doi.org/10.13031/2013.21741

Shirmohammadi, A., Chaubey, I., Harmel, R. D., Bosch, D. D., Muñoz-Carpena, R., Dharmasri, C., Sexton, A. M., Arabi, M., Wolfe, M. L., Frankenberger, J., Graff, C., & Sohrabi, T. M. (2006b). UNCERTAINTY IN TMDL MODELS. *Transactions of the ASABE*, *49*(4), 1033–1049. https://doi.org/10.13031/2013.21741

Sobol. (1990). Sensitivity estimates for nonlinear mathematical models. *Matematicheskoe Modelirovani*, *2*, 112–118.

Sobol, I. (1993). *Sensitivity Estimates for Nonlinear Mathematical Models*.

USDA. (1986). *Urban Hydrology for Small Watersheds*.

Whoolhiser, D. A., Smith, R., & Goodrich, D. (1990). *KINEROS: A kinematic runoff and erosion model documentation and user manual*.

Wilson, Barfield, B. J., & Moore, I. D. (1981). *A hydrology and sedimentology watershed model, Part I, Modeling Techniques*.