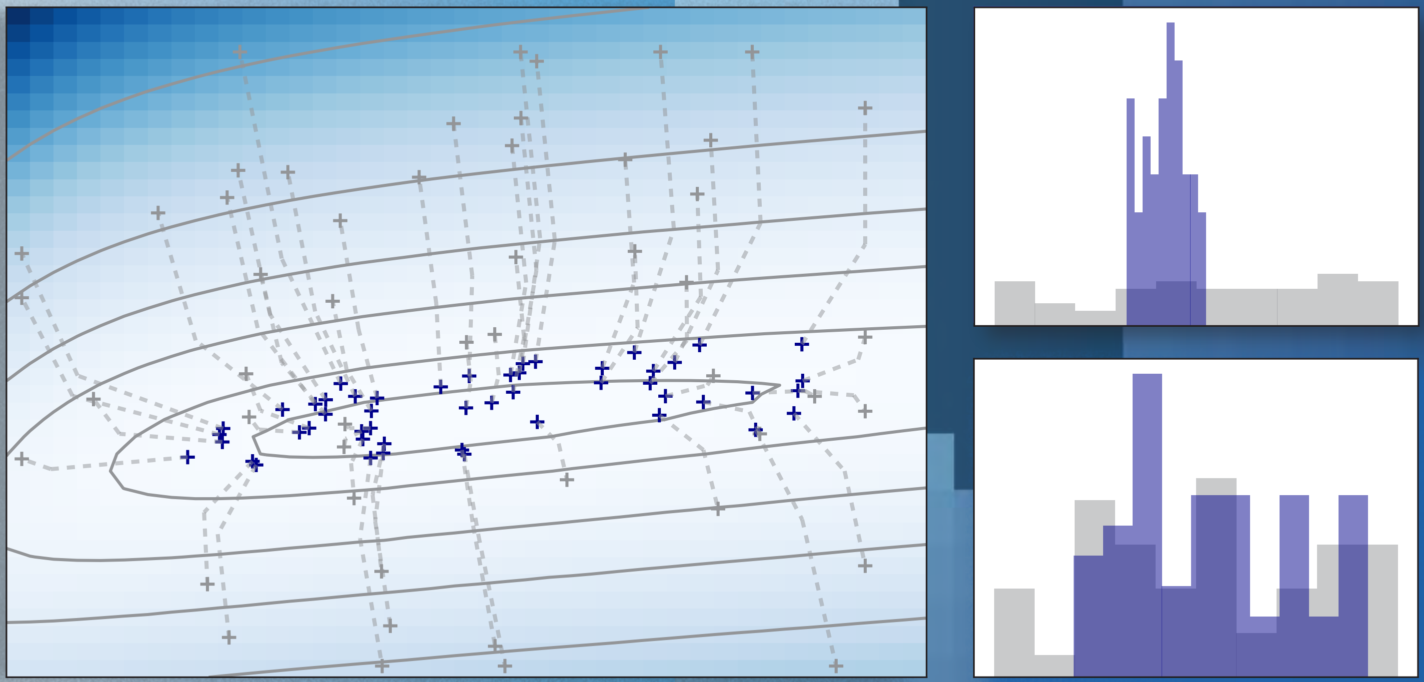
Water Availability and Use Science Program

Prepared in cooperation with the U.S. Environmental Protection Agency Great Lakes Restoration Initiative

Approaches to Highly Parameterized Inversion: PEST++ Version 5, a Software Suite for Parameter Estimation, Uncertainty Analysis, Management Optimization and Sensitivity Analysis



Techniques and Methods 7–C26

U.S. Department of the Interior
U.S. Geological Survey

Cover image. A depiction of an ensemble of realizations converging to a best fit (left), with before- and after-calibration parameter distributions (right).

Approaches to Highly Parameterized Inversion: PEST++ Version 5, a Software Suite for Parameter Estimation, Uncertainty Analysis, Management Optimization and Sensitivity Analysis

By Jeremy T. White, Randall J. Hunt, Michael N. Fienen, John E. Doherty

Water Availability and Use Science Program

Prepared in cooperation with the U.S. Environmental Protection Agency
Great Lakes Restoration Initiative

Techniques and Methods 7–C26

U.S. Department of the Interior
U.S. Geological Survey

U.S. Department of the Interior
DAVID BERNHARDT, Secretary

U.S. Geological Survey
James F. Reilly II, Director

U.S. Geological Survey, Reston, Virginia: 2020

Suggested citation:
White, J.T., Hunt, R.J., Fienen, M.N., and Doherty, J.E., 2020, Approaches to Highly Parameterized Inversion: PEST++ Version 5, a Software Suite for Parameter Estimation, Uncertainty Analysis, Management Optimization and Sensitivity Analysis: U.S. Geological Survey Techniques and Methods 7C26, 52 p., https://doi.org/10.3133/tm7C26.

ISSN 2328-7055 (online)

Acknowledgments

The authors would like to acknowledge support from the U.S. Environmental Protection Agency Great Lakes Restoration Initiative, USGS Water Availability and Use Science Program, USGS Mississippi Alluvial Plain Project, and New Zealand’s GNS Science. The authors would also like to acknowledge David Welter (South Florida Water Management District) for contributions to PEST++ Versions 1 through 3; this foundational work provided a base for portions of the Version 5 software described here. The authors would also like to acknowledge Wes Kitlasten, Mike Towes, Chris Nicols, Matthew Knowling, and Damien Merrick for contributions to PEST++. We would also like to thank Jake Knight, Matthew Knowling, Wes Kitlasten, and Paul Barlow for thoughtful and insightful comments that improved this report.

Contents

Figures

Tables

Conversion Factors

U.S. customary units to International System of Units

Multiply	By	To obtain
	Length	
inch (in.)	2.54	centimeter (cm)
foot (ft)	0.3048	meter (m)
mile (mi)	1.609	kilometer (km)
	Area	
acre	4,047	square meter (m^2)
square foot (ft^2)	0.09290	square meter (m^2)
square inch (in^2)	6.452	square centimeter (cm^2)
square mile (mi^2)	2.590	square kilometer (km^2)
	Volume	
gallon (gal)	0.003785	cubic meter (m^3)
million gallons (Mgal)	3,785	cubic meter (m^3)
cubic inch (in^3)	16.39	cubic centimeter (cm^3)
cubic foot (ft^3)	0.02832	cubic meter (m^3)
	Flow rate	
foot per second (ft/s)	0.3048	meter per second (m/s)
foot per day (ft/d)	0.3048	meter per day (m/d)
cubic foot per second (ft^3/s)	0.02832	cubic meter per second (m^3/s)
cubic foot per second per square mile [(ft^3/s)/mi^2]	0.01093	cubic meter per second per square kilometer [(m^3/s)/km^2]
gallon per minute (gal/min)	0.06309	liter per second (L/s)
gallon per day (gal/d)	0.003785	cubic meter per day (m^3/d)
million gallons per day (Mgal/d)	0.04381	cubic meter per second (m^3/s)
inch per year (in/yr)	25.4	millimeter per year (mm/yr)
	Hydraulic conductivity	
foot per day (ft/d)	0.3048	meter per day (m/d)

Abbreviations

GLM	Gauss-Levenberg-Marquardt
GSA	global sensitivity analysis
FOSM	first-order, second moment
HK	horizontal hydraulic conductivity
λ	Marquardt lambda
MAXSING	maximum number of singular values
OUU	optimization under uncertainty
PEST++ V5	Version 5 of the PEST++ software suite
PESTPP-GLM	PEST++ Gauss-Marquardt-Levenberg algorithm
PESTPP-IES	PEST++ iterative ensemble smoother algorithm
PESTPP-OPT	PEST++ management optimization algorithm
PESTPP-SEN	PEST++ global sensitivity algorithm
RCH	recharge
SFR	streamflow routing
Ss	specific storage
SVD	singular value decomposition
Sy	specific yield
TCP/IP	transmission control protocol/Internet protocol
USGS	U.S. Geological Survey
VK	vertical hydraulic conductivity
WEL	well extraction rate

Approaches to Highly Parameterized Inversion: PEST++ Version 5, a Software Suite for Parameter Estimation, Uncertainty Analysis, Management Optimization and Sensitivity Analysis

By Jeremy T. White,[1] Randall J. Hunt[1], Michael N. Fienen[1], John E. Doherty[2]

Abstract

PEST++ Version 5 extends and enhances the functionality of the PEST++ Version 3 software suite, providing environmental modeling practitioners access to updated Version 3 tools as well as new tools to support decision making with environmental models. Version 5 of PEST++ includes tools for global sensitivity analysis (PESTPP-SEN); least-squares parameter estimation with integrated first-order, second-moment parameter and forecast uncertainty estimation (PESTPP-GLM); an iterative, localized ensemble smoother (PESTPP-IES); and a tool for management optimization under uncertainty (PESTPP-OPT). Additionally, all PEST++ Version 5 tools have a built-in fault-tolerant, multithreaded parallel run manager and are model independent, using the same protocol as the widely used PEST software suite.

PEST++ Version 5 is consistent with PEST++ Version 3 conventions and design philosophy. The software's emphasis continues to target efficient and optimized algorithms that have proven beneficial in decision-support settings and can accommodate large, highly parameterized problems. Expanded and new capabilities are now available to express uncertainty using Monte Carlo and analytical uncertainty approaches and allow evaluation of thousands to millions of parameters. New management optimization capabilities in Version 5 also allow environmental models to be used to answer management questions using multiple societal constraints in a risk-based framework.

The PEST++ Version 5 software suite can be compiled for Microsoft Windows® and Unix-based operating systems such as Apple and Linux®; the source code is available with a Microsoft Visual Studio® 2019 solution; and CMake support for all three operating system is also provided. PEST++ Version 5 continues to build a foundation for an open-source framework capable of producing model-independent, robust, and efficient decision-support tools for large environmental models. The functionality of each of the PEST++ tools are demonstrated on a simple example problem. Implications of decisions used when using the PEST++ suite tools are also discussed.

Introduction

Environmental modeling is required for most quantitative resource management decisions (Anderson and others, 2015). Highly parameterized, physically motivated, process-based numerical models (for example, Doherty and Hunt, 2010) are routinely constructed to represent functions of a natural system so that the models can be used to evaluate the outcomes of management strategies or how the system responds to changes in conditions. In the resource-management context, environmental models are used as predictive tools rather than just explanatory tools (see Shmueli [2010] for a comprehensive discussion on this distinction). In the predictive setting, a quantitative framework for parameter estimation, data assimilation, uncertainty analysis, and risk-based management optimization provides decision makers with valuable insights regarding models used for decision making—information that cannot be developed otherwise. Uncertainty analysis can be particularly useful to support decision-making because putting simulated outcomes important for decisions in the context of uncertainty facilitates insight regarding limitations of the model's predictive capabilities. This information, in turn, provides the means to construct estimates of risk, a fundamental requirement for risk-based decision-making (for example, Freeze and others, 1990; Doherty and Moore, 2019).

Decision making typically comprises many aspects spanning a range of disciplines. As a result, the PEST++ tool development has been focused on making "universal" tools that communicate with, but function independently of, the underlying model (or models). Within environmental modeling, PEST (Doherty, 2015a, b) is the longest and most widely used universal parameter estimation and uncertainty analysis software suite. The need for more extensible and accessible

[1]U.S. Geological Survey

[2]Watermark Numerical Computing

software to augment PEST led to the development of PEST++ (Welter and others 2012 [Version 1]; Welter and others, 2015 [Version 3]). Earlier versions of PEST++ provided easier access to well-tested PEST approaches for highly parameterized models (linear uncertainty analysis, on-the-fly subspace dimensionality reduction; see Welter and others, 2012), as well as capabilities such as global sensitivity analysis (Welter and others, 2015). A more complete description of design goals and capabilities of Versions 1 and 3 are documented in Welter and others (2012) and Welter and others (2015), respectively. The highly parameterized context required for effective environmental modeling—which forms a foundation for PEST++ design—is discussed by Moore and Doherty (2005), Hunt and others (2007), Doherty and Hunt (2010), and Doherty (2015b), among others.

With PEST++ Version 5, the functionality of PEST++ is extended into algorithmic areas beyond Version 3 and is now grouped into several categories of decision-support analyses. Most of the capabilities of PEST++ Version 3 are now included in the PESTPP-GLM[1] executable. This includes minimum error variance parameter estimation combined with linear uncertainty analysis using first-order, second-moment (FOSM) methods, as well as new capabilities involving preconditioned Monte Carlo parameter and forecast uncertainty analysis. The PEST++ Version 3 global sensitivity capabilities are now contained in PESTPP-SEN, focusing on Sobol and Morris global sensitivity analyses (Saltelli and others, 2008). A third branch of capability includes PESTPP-IES, a new localized iterative ensemble smoother (White, 2018) that provides efficient handling of history matching and uncertainty analysis for very high-dimensional inverse problems. The last branch, PESTPP-OPT, is also new in Version 5, which facilitates management optimization by sequential linear programming under uncertainty (White and others, 2018). In addition to these codes, PEST++ Version 5 also includes a utility program to evaluate sets of parameter values in parallel (PESTPP-SWP). A tool for (multiobjective) particle-swarm optimization is available that uses parts of the PEST++ Version 5 code base and is documented elsewhere (PESTPP-PSO; Siade and others, 2019). The PEST++ Version 5 software suite can be compiled for Microsoft Windows®, Apple®, and Linux® operating systems; the source code is available in a Microsoft Visual Studio® 2019 solution; and CMake support is also provided. As documented here, PEST++ Version 5 continues to build a foundation for an open-source framework capable of producing robust and efficient parameter-estimation tools for large environmental models used in decision making.

Purpose and Scope

This report describes an expansion of the algorithms and tools included in PEST++ Version 3 documented by Welter and others (2012). PEST++ Version 5 replaces PEST++ Version 3[2], yet the overriding design concepts and approaches are consistent with those previously documented. This report focuses on documentation of enhancements and new capabilities added to PEST++ Version 5 (hereafter referred to as PEST++ V5); citations are given for the descriptions of functionality of previous versions of the software. However, input instructions and examples are documented here in such a way that a user can run PEST++ V5 without referring to documentation of the previous versions. The website for this publication (https://pubs.usgs.gov/tm/tm7c26) has hyperlinks for both the source code and executable of PEST++ V5.

The PEST++ V5 software suite includes updates to Version 3's algorithms for performing integrated FOSM-based parameter and predictive uncertainty analyses and Gauss-Levenberg-Marquardt (GLM) algorithm (PESTPP-GLM) along with global sensitivity analysis capabilities (PESTPP-SEN). However, PEST++ was designed to be extensible, which allows continual development to better address real-world decision-making problems (Welter and others, 2012). Thus, this report also focuses on the enhancements added to PEST++ Version 3. The following capabilities are now available in PEST++ V5: (1) an iterative ensemble smoother (PESTPP-IES) that allows computationally efficient history matching and uncertainty calculation for very highly parameterized models (White 2018); and (2) a tool for management optimization (PESTPP-OPT) under uncertainty (White and others, 2018).

The following report sections progressively build on a single example problem to demonstrate the decision-making applications that the PEST++ V5 software suite can address. Within each application of a PEST++ V5 tool, the theory supporting the use is briefly discussed if the theoretical basis for the given tool has not been discussed in a previous PEST++ report; citations to related supporting material are also provided throughout this report. In the style of presentation used here, a workflow for implementing the PEST++ V5 tools is also illustrated. The workflow moves through a step-wise approach to first evaluate model stability and identify primary relations between inputs and outputs (PESTPP-SEN), then undertake history matching and uncertainty analysis (PESTPP-GLM and [or] PESTPP-IES), and, finally, transfer the results of the uncertainty analysis to formal management

[1] The name "PEST++" reflects the C++ programming language used to develop the PEST++ software suite. However, filenames including "++"can be problematic for some compilers and operating systems. Therefore, compiled executables use the convention "PESTPP". Within this report, "PEST++" is used when referring to the overarching software suite and "PESTPP" is used to denote the compiled executable file used to implement PEST++ capabilities.

[2] Historically, PEST++ even-numbered versions are used for internal development; odd-number versions represent official releases. Therefore, Version 4 of PEST++ existed only as an unreleased code-development milestone and Version 5 represents updates to Version 3 codes documented in Welter and others (2015).

optimization under uncertainty (PESTPP-OPT). The main text then concludes with suggestions for application and summary. Appendix 1 documents the input instructions needed to apply these tools. All related terminology, concepts, file extensions, and so forth, follow the conventions and derivations presented and cited by Doherty (2015a, b), Doherty and Hunt (2010), Doherty and others (2010b), and Welter and others (2012, 2015) and are omitted here for brevity.

Example Problem Description

An enhanced version of a previously documented synthetic groundwater model (available at https://pubs.usgs.gov/tm/tm7c26), based on the models of Freyberg (1988) and Hunt and others (2019), is used to illustrate the capabilities of the PEST++ V5 software suite. Use of a synthetic model allows us to generate a “truth” that can then be used to evaluate PEST++ results, both in terms of ability to reproduce historical conditions and also to evaluate the ability of the uncertainty analysis processes in PEST++ to reproduce, in a statistical sense, the true forecast values.

The enhanced model has 3 layers, 40 rows, and 20 columns. The model simulates 25 stress periods: 1 steady-state stress period followed by 24 monthly, transient stress periods. The groundwater system is simulated using the MODFLOW 6 (Langevin and others, 2017) groundwater code.

Conceptually, the first 12 transient stress periods represent “true” historical conditions. A subset of simulated outputs of these first 12 stress periods are used as “observations” for history matching in the PESTPP-GLM and PESTPP-IES examples. The last 12 transient stress periods conceptually represent an “unmeasured” future condition. A subset of simulated outputs from the second 12 stress periods form the set of “forecasts” that are the focus of the uncertainty analyses performed using PESTPP-GLM and PESTPP-IES capabilities; some of these forecasts are also used as constraints in the final PESTPP-OPT management optimization example.

Across the 24 monthly transient stress periods, a seasonal signal is applied by modifying both the recharge and groundwater extraction. In addition, stress periods 10 and 22 are modified to represent appreciably drier periods characterized by increased groundwater extraction and decreased recharge. Stress periods 4 and 16 represent wet periods characterized by decreased groundwater extraction and increased recharge. Simulated recharge and groundwater extraction vary smoothly between the dry and wet times,

The surface-water system in the model consists of a straight stream (fig. 1), which is simulated with the Streamflow Routing (SFR) process; SFR reaches traverse the model domain from row 1 to row 40 in column 16; surface-water flow observations are monitored at reach 40 (the terminal reach; fig. 1). Six groundwater extraction wells are in layer 3 along with two groundwater level monitoring locations. Water enters the model domain as recharge and stream leakage in layer 1 and leaves through groundwater discharge to surface-water and groundwater extraction or through downgradient discharge to the general head boundary in row 40.

To simplify input file handling, a single, comprehensive control file was constructed on the basis of the enhanced Freyberg model, which is used to demonstrate each of the tools in the PEST++ V5 suite. Depending on the tool being demonstrated, different control file options—such as “tying” and “fixing” parameters, reassigning groups and changing observation weights—were used to demonstrate how a single control file can be used for a variety of decision-support analyses.

The enhanced Freyberg example problem uses the following parameterization, grouped by parameter type:

- horizontal hydraulic conductivity (HK): spatially varying, every active model cell;
- vertical hydraulic conductivity (VK): spatially varying, every active model cell;
- specific storage (Ss): spatially varying, every active model cell;
- specific yield (Sy): spatially varying, layer 1 active model cells;
- recharge (RCH): temporally varying, every stress period; and
- groundwater extraction rates (WEL): spatially and temporally varying, every well cell, every stress period.

When all parameters are adjustable, this parameterization scheme yields 8,175 parameters to represent model input uncertainty.

The following model outputs were included in the PEST model interface as “observations” for the purposes of history matching:

- surface-water flow at location sw_1: simulated outflow from SFR reach 40 (terminal reach), stress period 2 through 13; and
- groundwater levels at locations gw_1 and gw_2 in model layer 3: simulated groundwater level in layer 3 at these 2 locations, stress period 2 through 13.

The composite objective function was constructed by assigning groundwater level observations in stress periods 2–13 a weight of 15, which implies a standard deviation of 0.07 feet. The surface-water flow observations for stress periods 2–13 were assigned weights such that the coefficient of variation for each surface-water flow observation was 0.15. At initial parameter values, the objective function is equal to 120.

In PEST++, forecasts are included as zero-weighted observations that are also identified as forecasts for uncertainty analysis purposes using the PEST++ forecast option. For the example problem described here, the following three model outputs were also included as forecast “observations” in the control file:

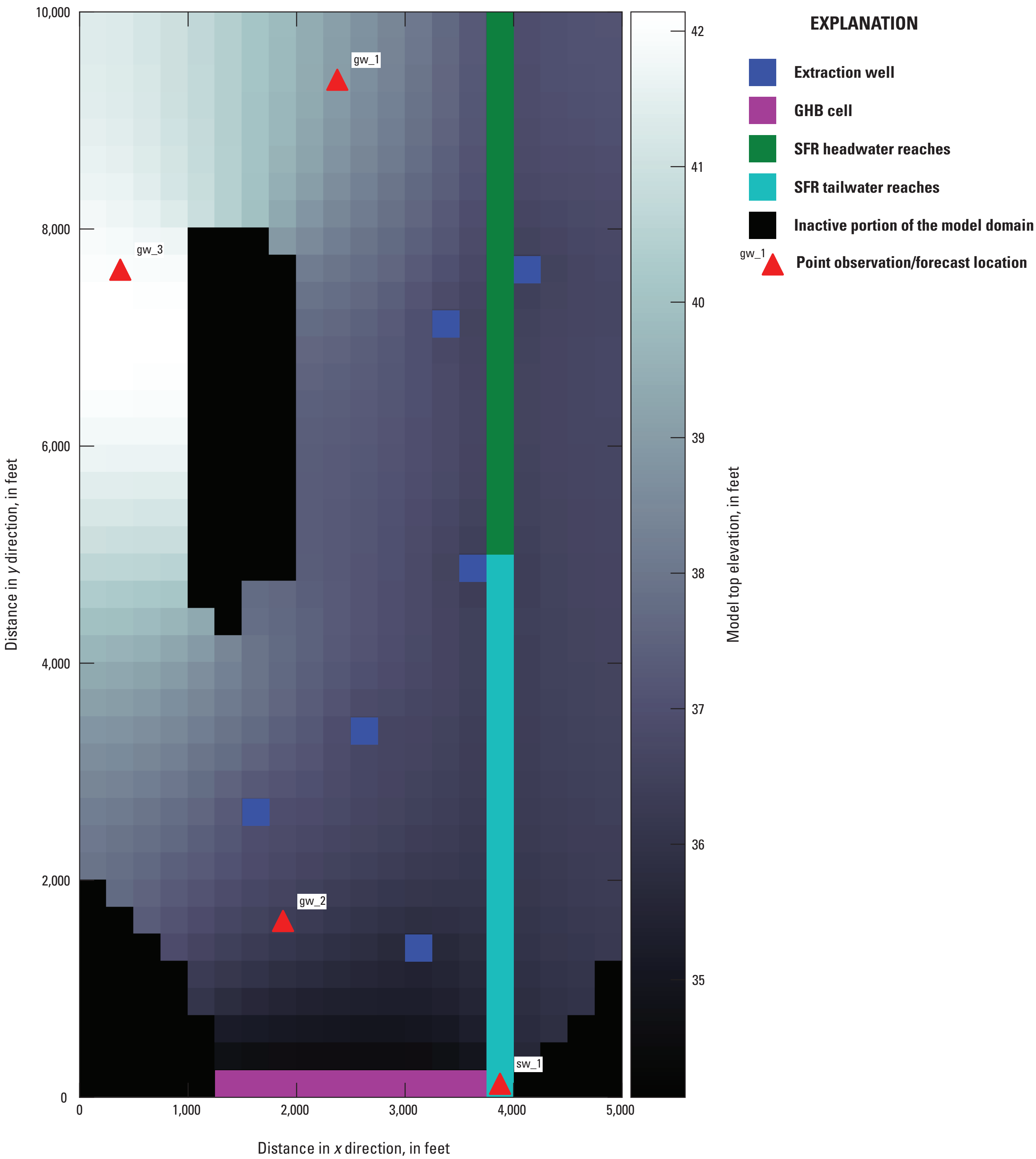


Figure 1. Enhanced Freyberg model boundary conditions and model extent. Observations from locations sw_1 (layer 1), gw_1, and gw_2 (layer 3) are used for history matching. Aggregated surface-water/groundwater exchange along the headwater and tailwater reaches for the last stress period (layer 1), as well as groundwater level at gw_3 (layer 1) for the last stress period, are used as forecasts. (GHB, general-head boundary; SFR, streamflow routing, ft, feet)

- aggregated surface-water/groundwater exchange for reaches in rows 1–20 (“headwater”) during stress period 22;
- aggregated surface-water/groundwater exchange for reaches in rows 21–40 (“tailwater”) during stress period 13; and
- groundwater level at location gw_3 in layer 1 at the end of stress period 22.

The forecasts are used for predictive uncertainty analyses in PESTPP-GLM and PESTPP-IES in a Bayesian framework. Within this framework, prior parameter uncertainty is defined by the judgement of the modeler, where “prior” equates to uncertainty that exists in parameters before history matching. The posterior uncertainty is the what remains after history matching. Both prior and posterior parameter uncertainty can be then propagated to forecasts (see Tarantola (2005) and Doherty (2015a, b) for background).

These forecasts were selected to represent model outputs that are informed, in varying degrees, by the observations selected for history matching. Outputs that are entirely constrained by observations are considered completely within the model solution space; outputs entirely unconstrained by observations are considered in the model null space (see Moore and Doherty [2005] and Doherty and others [2010b] for detailed explanation of a model’s solution and null spaces). The forecasts used here are neither entirely constrained nor unconstrained by the observations, thus have different null-space dependencies. As the null-space dependence of a given parameter increases, so too does the posterior parameter uncertainty because less information derived from observations used for history matching is available to constrain model parameters. Moore and Doherty (2005) and later Doherty and Welter (2010) documented that the ability to quantify forecast uncertainty in groundwater modeling is affected by null-space dependence.

The tailwater forecast (recorded at the end of the history matching period) is more informed by the history matching observations (smaller null-space dependence) because it integrates a large area of the model domain (Hunt and others, 2006). The headwater forecast (measured near the end of the forecast period during a seasonal dry time) is less informed by history matching observations (larger null-space dependence) that did not include dry periods. The groundwater-level forecast is expected to have a moderate null-space dependence because it is similar in character to the observations used for history matching but occurs during unknown future conditions.

The prior parameter distribution for the hydraulic conductivity field, well pumping multipliers, and recharge multipliers was conceptualized as a truncated multivariate Gaussian distribution (for example, Tarantola, 2005; Doherty, 2015a, b). Parameters HK, VK, Ss, and Sy were conceptualized as geostatistically correlated fields and described by an exponential variogram with a range of 1,000 feet and sill proportional to the square of statistical distance between parameter upper and lower bounds (table 1). Well extraction and recharge parameters were conceptualized as temporally correlated and described by an exponential variogram with a 3-month range and sill proportional to the square of statistical distance between the parameter upper and lower bounds. Figure 2 shows the realized parameter values for the “true parameter” realization.

As previously mentioned, the synthetic problem design used allowed selection of a “true” model to supply observations for history matching and to evaluate the outcomes of posterior forecast uncertainty analyses. To generate this “truth,” 300 realizations were drawn from the prior parameter distribution and evaluated with the model, yielding 300 values for each

Table 1. Prior parameter upper and lower bounds. Recharge and extraction rate parameter types have a range of upper and lower bounds to accommodate seasonal variability.

[HK, horizontal hydraulic conductivity; VK, vertical hydraulic conductivity; Ss, specific storage; Sy, specific yield]

Parameter type	Units	Lower bound	Upper bound
HK Layer 1	Feet per day	0.3	30
HK Layer 2	Feet per day	0.03	3
HK Layer 3	Feet per day	3	300
VK Layer 1	Feet per day	0.03	3
VK Layer 2	Feet per day	0.003	0.3
VK Layer 3	Feet per day	0.3	30
Ss Layer 1	Per foot	1e-7	1e-5
Ss Layer 2	Per foot	1e-7	1e-5
Ss Layer 3	Per foot	1e-7	1e-5
Sy Layer 1	none	0.001	0.1
Recharge (RCH)	Feet per day	1e-5 to 5e-5	3e-5 to 1.5e-4
Extraction Rate (WEL)	Cubic feet per day	15 to 73	100 to 510

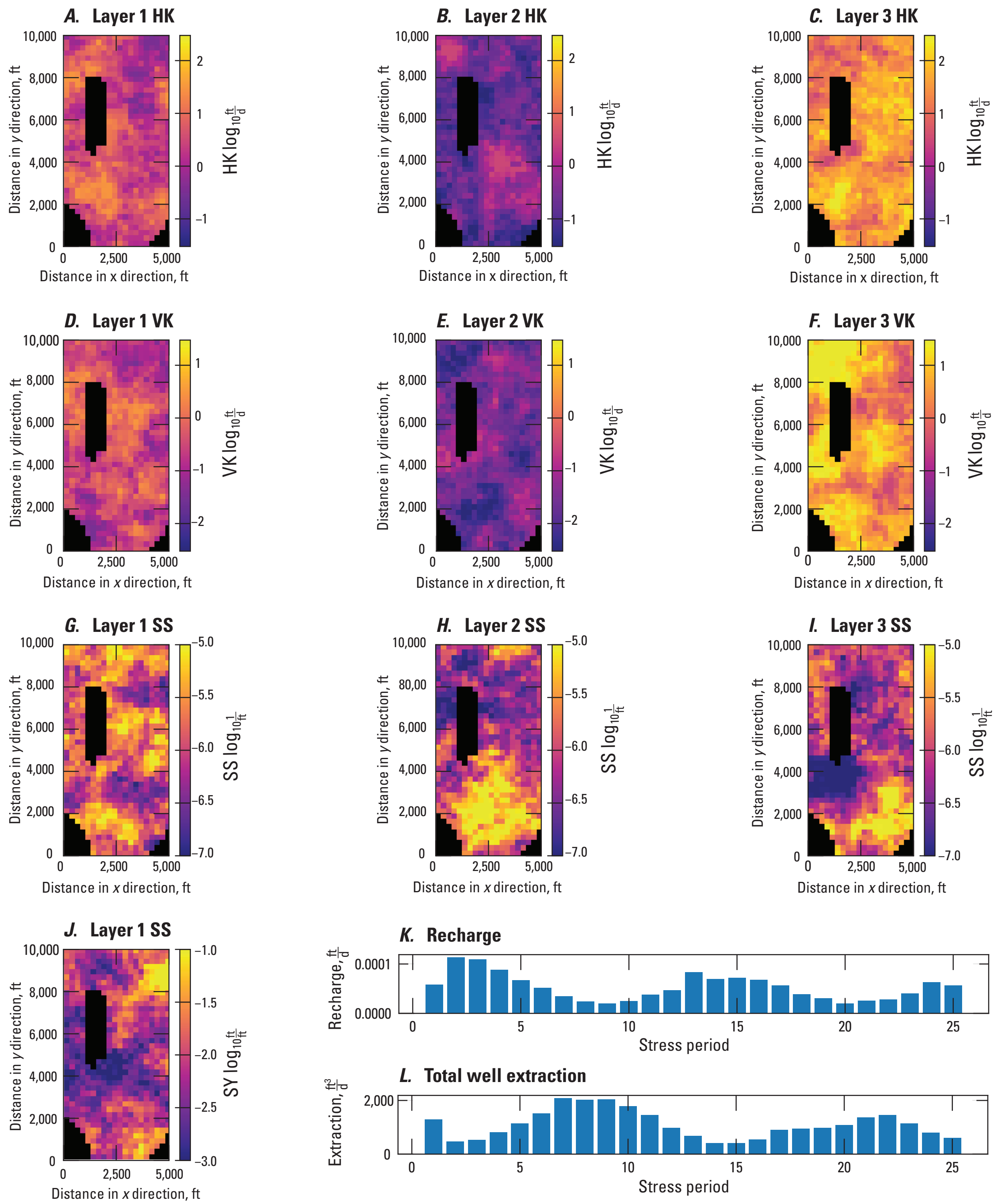


Figure 2. Plots of “true” parameter values. (HK, horizontal hydraulic conductivity; VK vertical hydraulic conductivity; SY, specific yield; SS, specific storage; ft, feet; d, day; ft^3, cubic feet)

observation and forecast. The objective function minimum, maximum, and mean across these 300 realizations was 82.6, 7104, and 978, respectively. From this set of 300 results, a realization among the lowest 95th quantile for the tailwater surface-water/groundwater exchange forecast was chosen for the "true" model (the objective function was 390 at initial parameter values); that is, the realization selected had 95 percent of the other realizations with higher amounts of flow at the tailwater forecast location. Recall that the tailwater forecast is from the end of the history matching period (stress period 13) and integrates most of the model domain and measured at the end of the history matching period, so it is expected to have small posterior uncertainty. The use of a statistically extreme realization characterized by small posterior uncertainty (because it is more informed by the observations) but with an extreme first moment makes the true tailwater quantity a difficult forecast to reproduce. The increased difficulty provides a robust comparison for the posterior uncertainty estimation of the PESTPP-GLM and PESTPP-IES analysis.

PESTPP-SEN Example

PESTPP-SEN implements two global sensitivity analysis (GSA) methods: Method of Morris (Morris, 1991) and Sobol (Sobol, 2001). Generally, GSA methods can provide a basis for diagnostic/screening analyses to identify important relations between model inputs (represented by parameters) and outputs (represented by the objective function, as well as the observations listed in the control file). Therefore, using global sensitivity analyses at the beginning of a modeling analysis workflow provides insight on the linearity of the inverse problem as well as parameter-to-forecast sensitivity, information that can be useful for implementing other PEST++ V5 tools. Readers are referred to Saltelli and others (2008) for an extensive and detailed explanation of global sensitivity analysis and to Welter and others (2015) for the PEST++ implementation.

GSA is computationally demanding, and, depending on the underlying forward model, may be limited to low dimensional parameter spaces (less than 100 parameters) to avoid excessive computational burden. The larger computational burden of GSA methods results from these techniques requiring repeated, systematic evaluation of the model at numerous points across parameter space. Moreover, they provide information on global sensitivity across the range of parameters, but the parameter estimation and uncertainty analyses often use a (more) local sensitivity with lower computational burden. Therefore, GSA methods are typically used as a diagnostic tool to identify important model input-output relations and to identify regions of parameter space that degrade model stability.

PESTPP-SEN Example Application

PESTPP-SEN was applied to the enhanced Freyberg example to investigate what parameter types affect both the objective function (important for history matching) and what parameters types affect the three forecasts of interest (important for predictive uncertainty analysis).

Input Approach

PEST++ options added to the control file (see the appendix 1 for a complete listing of PEST++ options):

- **Option name:** *tie_by_group,* **value:** *true*: use the PEST parameter "tying" construct to reduce the number of adjustable parameters (and therefore the number of model runs required to apply PESTPP-SEN) by treating all parameters in a group as a single parameter.

Employing this one PEST++ option reduced the number of adjustable parameters from 8,175 to 12. Note that tying parameters by group results in a larger grouped sensitivity than would be seen by individual, untied parameters. However, for screening and diagnostic purposes, the relative sensitivities among parameter types often still provide useful insights. Note that no global-sensitivity specific PESTPP-SEN options were added to the PEST++ control file. Instead, the analysis used internal default values for PEST++ options, which include selection of the computationally lighter Method of Morris analysis compared with Sobol, with sampling four locations that span the parameter space.

Results

The following PEST++ output files were evaluated:

- run record file (*.rec*), and
- Method of Morris results summary file (*.mio*).

With the default settings, the enhanced Freyberg model required 52 forward model runs for the global sensitivity evaluation: (12 parameters perturbation + 1 run with base parameters) x 4 locations in parameter space. If more than 4 locations in parameter space were needed, the user can override this default with *gsa_morris_r* PEST++_option added to the PEST++ control file; for example, adding ++*gsa_morris_r(6)* to the PEST++ control file would divide the parameter space into six increments. If the more computationally intensive Sobol global sensitivity test were desired, it would be invoked by the addition of the *gsa_method(sobol)* PEST++ option to the control file.

The results of this global sensitivity analysis are reported in the *.mio* output file as the mean sensitivity (fig. 3*A*) and coefficient of variation associated with the reported mean (fig. 3*B*). For our example, this information revealed that parameter types that most affect the objective function for history matching (layer 3 HK, fig. 3*A*), are not the same parameter

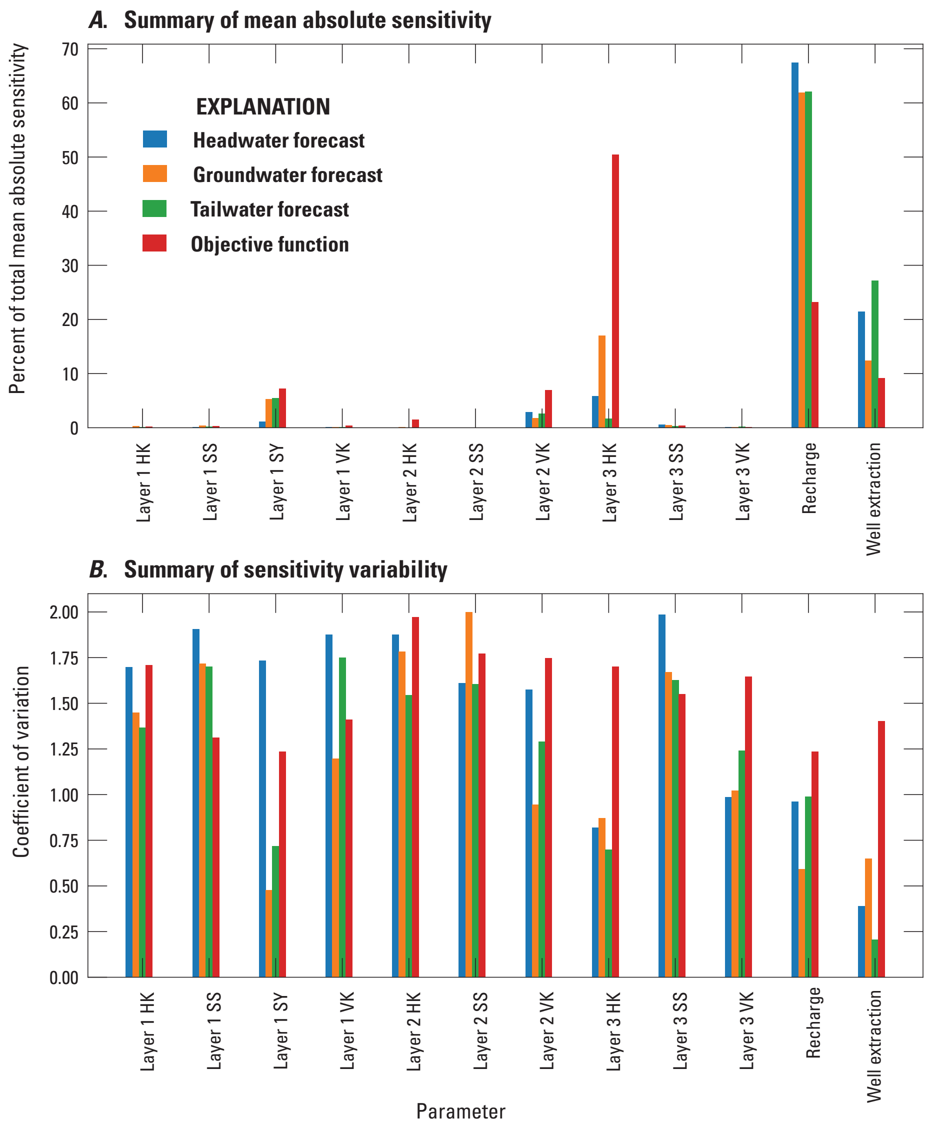


Figure 3. Summary of PESTPP-SEN results for Method of Morris global sensitivity analysis applied to the enhanced Freyberg model for the outputs of interest: the history-matching objective function and the three forecasts. *A*, the historical observations (encapsulated in the objective function) are most sensitive to Layer 3 HK, while the forecasts are most sensitive to recharge and, to a lesser extent, well extraction. *B*, the variation in parameter sensitivity across the four locations in parameter space where sensitivities were calculated; a linear relation between a given parameter type and a given output of interest would have a coefficient of variation of 0. Note: the legend of figure 3*A* also applies to figure 3*B*. (HK, horizontal hydraulic conductivity; VK vertical hydraulic conductivity; SY, specific yield; SS, specific storage)

types that are most influential to the three forecasts of interest (recharge; fig. 3*A*). Furthermore, the sensitivity analysis results indicate substantial variability in parameter sensitivity with respect to the history-matching objective function as parameters change across parameter space (fig. 3*B*), indicating a nonlinear relation between parameters and the objective function used for history-matching.

PESTPP-GLM Example

The code PESTPP-GLM includes functionality that was previously contained under the original PEST++ code (and Version 3's PESTPP executable). PESTPP-GLM retains the capabilities of the original code from PEST++ Version 3 in that it implements a subspace Gauss-Levenberg-Marquardt algorithm (Moré, 1978; Oliver and others, 2008)—the same algorithm implemented in PEST and PEST_HP (Doherty 2015a, b)—supplemented with automated and on-the-fly updated subspace parameter dimensionality reduction, an extension of the singular value decomposition (SVD) technique known as "SVD-Assist" (Tonkin and Doherty, 2005; Doherty and Hunt, 2010). PESTPP-GLM also continues to provide seamless FOSM (also known as "linear analysis") parameter and, optionally, forecast uncertainty analysis (see Doherty [2015b], Welter and others [2015], Anderson and others [2015], and White and others [2016] for a more detailed discussion of the theory and application of this family of uncertainty analysis).

In PEST++ V5, PESTPP-GLM has been enhanced to allow a user to access FOSM-based posterior Monte Carlo sampling and parallel ensemble evaluation, where these automated uncertainty analyses are available for each PESTPP-GLM iteration (through the *glm_iter_mc* PEST++ option) and at completion of the PESTPP-GLM iterations. "FOSM-based posterior Monte Carlo" refers to the workflow involving the approximation of the posterior parameter covariance matrix using FOSM, which is then used as a basis for performing sampling needed to construct a Monte Carlo run as well as subsequent evaluation of each sample or realization using the model. FOSM-based Monte Carlo serves as a way to "precondition" parameter realizations used for Monte Carlo analyses such that the resulting realizations are more likely to provide acceptable reproduction of the observed data (that is, fewer realizations need to be dropped after running because of overly poor fits). With this capability active, PESTPP-GLM automates a relatively complex workflow for nonlinear parameter and forecast uncertainty analysis with little or no user intervention and, in most cases, for a few additional model evaluations.

Theory: Gauss-Levenberg-Marquardt

PESTPP-GLM, like PEST and PEST_HP, implements the subspace Gauss-Levenberg-Marquardt algorithm for least-squares parameter estimation:

$$\Delta_\theta = -\left(J^T\Sigma_\varepsilon^{-1}J + \lambda * diag(J^T\Sigma_\varepsilon^{-1}J)\right)^{-1} J^T\Sigma_\varepsilon^{-1}(d_{sim} - d_{obs}) \quad (1)$$

where J is the Jacobian matrix, representing a first-order relation between parameters and observations, Σ_ε^{-1} is the inverse of observation noise covariance matrix, λ is the Marquardt lambda, d_{obs} is the vector of observations, d_{sim} is the vector of simulated equivalents to observations, and Δ_θ is a parameter change (for example, upgrade) vector. The quantity $J^T\Sigma_\varepsilon^{-1}J$ is known as the "normal" matrix and, under a quadratic assumption, is equivalent to the Hessian matrix in Newton's method (Oliver and others, 2008).

By varying the Marquardt lambda, a trust region is constructed between the Gauss-Newton direction (small lambda) and gradient descent (large lambda). During each iteration, PESTPP-GLM will, by default, solve equation 1 several times with different values of lambda and evaluate the model once for each resulting parameter change vector to find which value of lambda yields a d_{sim} vector that is most similar to the d_{obs} in a weighted least squares (in other words, minimize a residual $\mathscr{L}_2$ norm objective function, ϕ) sense. In addition to varying the lambda values, through the PEST++ option *lambda_scale_fac*, users can apply backtracking along each upgrade vector to test parameter upgrade locations within the GLM trust region. Backtracking is a technique to evaluate parameter values along the vector between the current parameter values and the GLM upgraded parameter values and can be an important consideration for highly nonlinear inverse problems. The full suite of lambda-backtrack parameter vectors is evaluated in parallel, and the vector that yields the minimum objective function value is selected as the next location to move to in parameter space. In this way, the GLM algorithm encapsulated in equation 1 is applied iteratively from an initial parameter vector θ, by first filling the Jacobian matrix at θ, then solving for and evaluating the optimal parameter change vector (with varied lambda values and optional backtracking) with equation 1, then accepting the minimum-objective function parameter upgrade vector by adding the upgrade vector to θ.

To fill the Jacobian matrix, the model must be evaluated once for each parameter with the given parameter in a perturbed state:

$$J[:, par_j] = \frac{\partial_{d_{sim}}}{\partial_{par_j}} \approx \frac{\Delta_{d_{sim}}}{\Delta_{par_j}} \quad (2)$$

where $J[:, par_j]$ denotes column par_j of the Jacobian matrix J, $\frac{\partial_{d_{sim}}}{\partial_{par_j}}$ is the partial first derivative of all observations with respect to parameter par_j, and $\frac{\Delta_{d_{sim}}}{\Delta_{par_j}}$ is the finite difference approximation of the partial first derivative. By running the model with parameter par_j perturbed and recording how the simulated equivalents to observations change compared to a base run (with no parameters perturbed), the Jacobian matrix can be filled column-wise. Because only one parameter is perturbed at a time and is thus independent of other runs, the process of filling the Jacobian matrix in this way is easily parallelized.

In most real-world environmental modeling settings, the number of parameters will vastly outnumber the number of available observations, meaning the normal matrix will be singular and cannot be directly inverted. In this case, a Moore-Penrose pseudoinverse can be used in place of a direct inverse (for example, Golub and Van Loan, 2012). Truncated SVD is used as a pseudo-inverse in PESTPP-GLM and PESTPP-IES. By default, PESTPP-GLM uses a randomized SVD solver (for example, Halko and others, 2011) to factorize the normal matrix of equation 1 to find a suitable pseudo-inverse. This solver is highly efficient, especially when the control variable MAXSING is used strategically because the SVD solver only needs to factorize the normal matrix to MAXSING components.

Similar to PEST and PEST_HP, PESTPP-GLM implements Tikhonov regularization via the * *prior information* and * *regularization* sections in the control file (Doherty and Hunt, 2010); these prior information equations are used to augment the Jacobian matrix with additional rows to inject expert knowledge into the inversion process and to stabilize the inversion process. However, compared to PEST and PEST_HP, finding the optimal regularization weight in PESTPP-GLM can be computationally burdensome, especially in high dimensions. This can add appreciably to the time to complete a PESTPP-GLM iteration and can affect parameter estimation performance. In recognition of this potential difficulty, PEST++ V5 includes a second option for including Tikhonov regularization that reflects the renewed focus of PESTPP-GLM on automating parameter and forecast uncertainty analysis. As such, within the combined parameter-estimation/uncertainty analysis PESTPP-GLM algorithm, the importance of a single, minimum error variance parameter set, governed by prior information equations, is important only in as much as this parameter set serves as the mean of the posterior parameter distribution for uncertainty analyses; that is, the distribution of parameter uncertainty around the mean value is of primary importance.

In recognition of this use, PESTPP-GLM provides the option for "regularized GLM" (Hanke, 1997; Oliver and others, 2008) as an alternative solution technique, by augmenting the normal matrix with (a scalar multiple of) the prior parameter covariance matrix:

$$\Delta_\theta = -\left(J^T\Sigma_\varepsilon^{-1}J + \lambda*(\Sigma_\theta^{-1})\right)^{-1}\Sigma_\theta^{-1}(\theta_0-\theta) + J^T\Sigma_\varepsilon^{-1}(d_{sim}-d_{obs}) \tag{3}$$

where Σ_θ is the prior parameter covariance matrix, θ_0 a vector of initial parameter values, θ is a vector of current parameter values and all other terms are as previously defined. Inspection of equation 3 indicates how regularization is injected into the GLM solution process through the prior parameter covariance matrix; this form of the GLM algorithm obviates augmentation of the Jacobian matrix with prior information equations and removes the need to explicitly solve for the optimal regularization weight factor. Instead, equation 3 relies on the prior parameter covariance matrix to enforce expert knowledge directly on the solution process, and, as a result, is more efficient than the dynamic regularization weight solution process.

The regularized GLM solution process is activated with the *glm_normal_form(prior)* option. In the regularized GLM solution scheme, singular value maximum (MAXSING) and Marquardt lambda (λ) become the primary mechanisms for users to regularize the solution, where the former is selected by the user to balance fit and adherence to soft knowledge and the latter is specified by the user through the *lambdas* PEST++ option.

Parameter and Forecast Uncertainty Estimation

After each GLM iteration and at the termination of all GLM iterations, PESTPP-GLM will solve for a FOSM posterior parameter covariance matrix, and, optionally, prior and posterior forecast uncertainty estimates, provided that the user identifies specific model outputs (for example, observations) as forecasts. The FOSM-approximate posterior covariance matrix is an application of Bayes' theorem (for details, see Fienen and others, 2010) and can be calculated as:

$$\overline{\Sigma}_\theta = \Sigma_\theta - \Sigma_\theta J^T(J\Sigma_\theta J^T + \Sigma_\varepsilon)^{-1}J\Sigma_\theta \tag{4}$$

where $\overline{\Sigma}_\theta$ is the posterior parameter covariance matrix and all other terms are as previously defined. Inspection of equation 4 reveals that all the components needed to solve for the posterior parameter covariance matrix are held in memory during each PESTPP-GLM iteration and at the termination of GLM iterations. If forecasts are included as zero-weighted "observations" in the control file and are identified through the *forecasts* PEST++ option, then PESTPP-GLM can estimate prior and posterior forecast uncertainty as follows (Doherty, 2015b):

$$\sigma_f^2 = y_f^T\Sigma_\theta y_f \tag{5}$$

$$\overline{\sigma}_f^2 = y_f^T\overline{\Sigma}_\theta y_f \tag{6}$$

where y_f is a vector of forecast sensitivities to adjustable parameters for forecast *f*, σ_f^2 is the prior variance of forecast *f*, $\overline{\sigma}_f^2$ is the posterior variance of forecast *f*, and all other terms are as previously defined. In this formulation, forecast sensitivity vectors are simply rows in the Jacobian matrix.

In addition to the computationally light FOSM-based parameter and forecast uncertainty estimates, PESTPP-GLM now has the ability to also perform nonlinear Monte Carlo uncertainty analysis. This is accessed by adding the *glm_num_reals* option to the PEST++ control file. PESTPP-GLM will then implement FOSM-based posterior Monte Carlo (also called Bayes-linear Monte Carlo) by drawing realizations from the assumed-Gaussian posterior parameter distribution defined by:

$$N(\overline{\mu}_\theta, \overline{\Sigma}_\theta) \tag{7}$$

where N is a normal distribution, $\overline{\mu}_{\theta}$ is a vector of estimated parameter values derived from the GLM estimation process and $\overline{\Sigma}_{\theta}$ is the posterior parameter covariance matrix. PESTPP-GLM will evaluate each of these realizations by running the model once for each realized parameter set (in parallel). If *glm_iter_mc* option is active, realizations will be drawn during each GLM iteration and evaluated with the GLM lambda (and optional backtracking) parameter upgrade vectors, which results in a FOSM-based posterior Monte Carlo uncertainty process be completed for each iteration of PESTPP-GLM. If forecasts are included as observations in the control file, this Monte Carlo process will yield an ensemble of forecast values. Conceptually, using the FOSM-based Monte Carlo to evaluate forecast uncertainty instead of relying on equation 6 relaxes the assumed linear relation between parameters and forecasts that underpins FOSM uncertainty analyses.

PESTPP-GLM Example Application

Our example problem is now used to demonstrate how PESTPP-GLM can perform efficient history matching and uncertainty analysis. We applied PESTPP-GLM in two different ways. First, we used PESTPP-GLM without any additional options to show how PESTPP-GLM can be run in a basic mode, relying entirely on internal default values for PEST++ options in the control file. Then we describe a second application that used a small number of PESTPP-GLM options to implement a more advanced PESTPP-GLM analysis.

Input Approach

Basic Analysis PEST++ options added to control file: None

Advanced Analysis PEST++ options added to control file:

- **option name:** *n_iter_base,* **value:** *−1*: number of base parameter iterations, where −1 indicates not to perform any parameter upgrade calculations using the adjustable base parameters in the control file and, instead, perform parameter upgrade calculations using superparameters;
- **option name:** *n_iter_super,* **value:** *4*: number of superparameter iterations before calculation of a new base parameter Jacobian matrix. In this case, specifying *n_iter_super* equal to four results in only superparameter iterations because we only request four total PESTPP-GLM iterations (controlled by the NOPTMAX variable);
- **option name:** *max_n_super,* **value:** *36*: use at most 36 superparameters during the super-parameter solution process;
- **option name:** *glm_normal_form,* **value:** *prior*: use the regularized GLM algorithm; and
- **option name:** *glm_num_reals,* **value:** *200*: draw and evaluate 200 FOSM-based posterior realizations.

PESTPP-GLM relies on a full-rank based Jacobian matrix that is filled by evaluating the model once for each adjustable parameter; we reduced the computational burden by reducing the number of adjustable parameters used in this example of the enhanced Freyberg model. First, we manually tied HK, VK, Ss, and Sy parameters into groups of 6 rows by 6 columns, effectively creating 26 blocks or zones per layer for each of these inputs. We also tied the well extraction rate parameters by stress period, yielding a total of 25 well extraction parameters. These tying operations effectively reduced the number of adjustable parameters from 8,175 to 330, while maintaining a representation of spatial and temporal parameter heterogeneity. Typically, pilot points (Doherty, 2003; Doherty and others, 2010a) would be used in place of the block-zone parameterization, as pilot points provide a smooth interpolation of parameters over the model grid. However, for demonstration purposes, we chose to represent spatial property heterogeneity using no external or additional utilities (implementation of pilot points requires external files that are typically constructed using graphical user interfaces in conjunction with PEST groundwater utilities [Doherty 2015a] or with the python module pyEMU [White and others, 2016]).

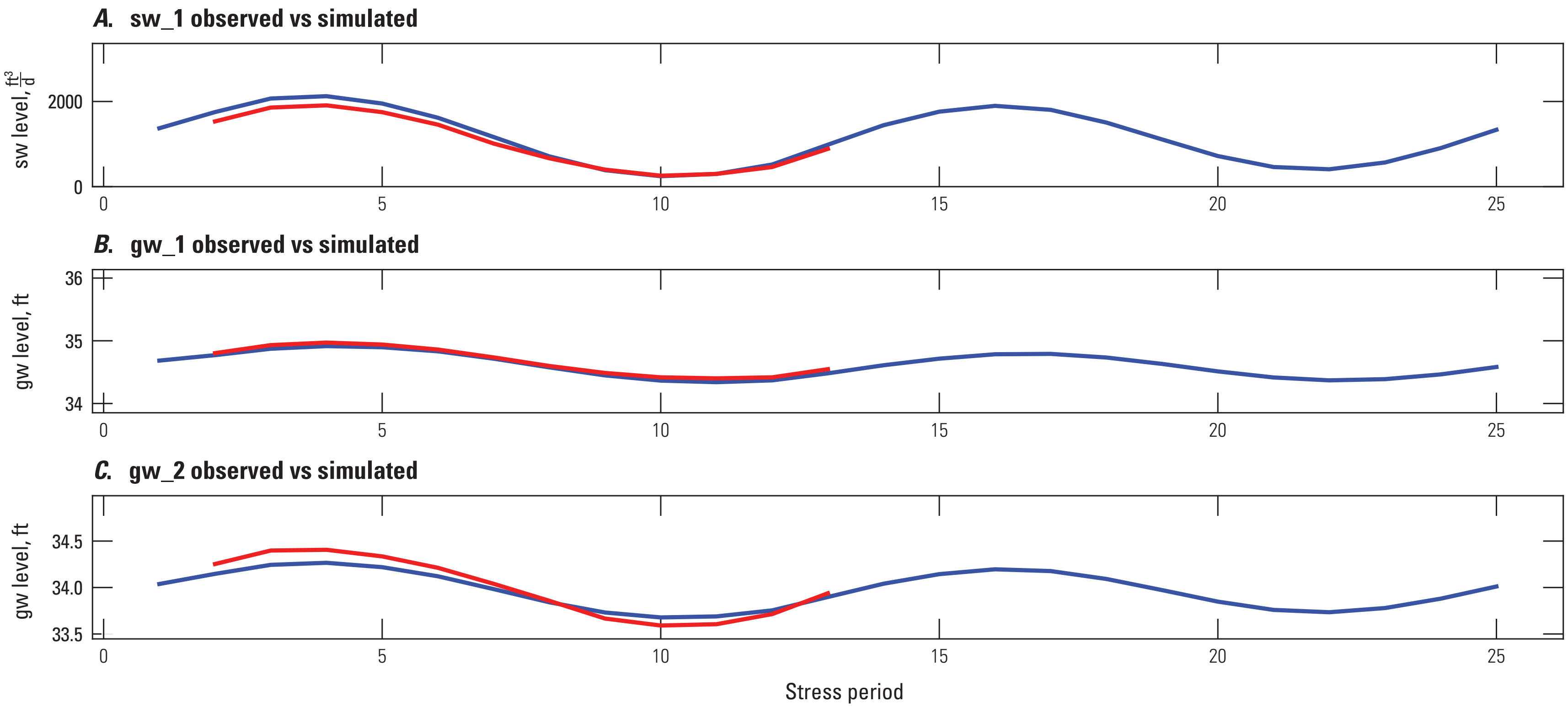
For the advanced PESTPP-GLM analysis, we used the regularized GLM solution process via the *glm_normal_form(prior)* option. Additionally, we used the automated subspace reparameterization process to further reduce the number of model evaluations through the *max_n_super* PEST++ option. We specified 200 posterior realizations with the *glm_num_reals* argument, resulting in 200 posterior realizations being drawn and evaluated once the four PESTPP-GLM iterations were complete. No additional files or modifications to the PEST interface were required to invoke Monte Carlo uncertainty analyses.

Results

The following PEST++ output files were evaluated:

- run record file (*.rec*),
- posterior FOSM forecast summary (*.pred.usum.csv*),
- posterior Monte Carlo parameter ensemble (*.post.paren.csv*), and
- posterior Monte Carlo observation ensemble (*.post.obsen.csv*).

The basic PESTPP-GLM analysis required 1,361 model runs to find a minimum error variance parameter set with a final objective function of 26. Figure 4 presents a summary of this analysis, including the results of the automated FOSM forecast uncertainty process. Figure 5 shows the final parameter set from the basic PESTPP-GLM analysis. As expected, the headwater forecast posterior distribution is similar to the prior



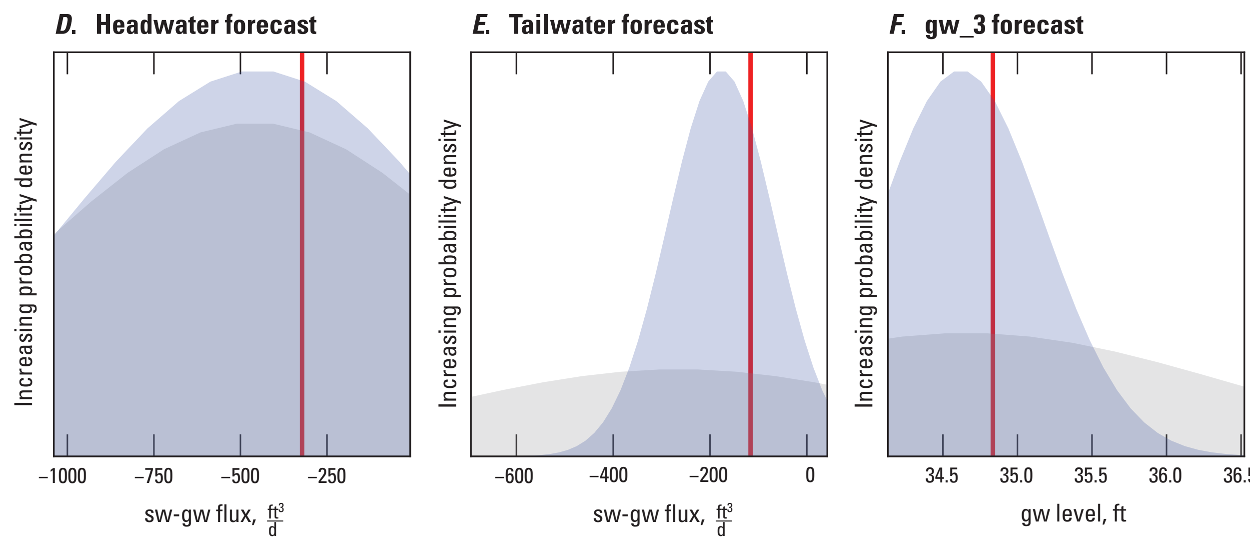


Figure 4. Summary of the basic PESTPP-GLM analysis using default values for the enhanced Freyberg model. *A*, *B*, and *C* show the observed (red) values versus simulated (blue) values. These results show a good level of agreement between the final minimum error variance parameter set and the observed states used for history matching. *D*, *E*, and *F* show how both the FOSM prior (gray gaussian curves) and posterior (blue gaussian curves) cover the true forecast values (red vertical bars). (sw, surface water; gw, groundwater, ft, feet; d, day; ft³, cubic feet)

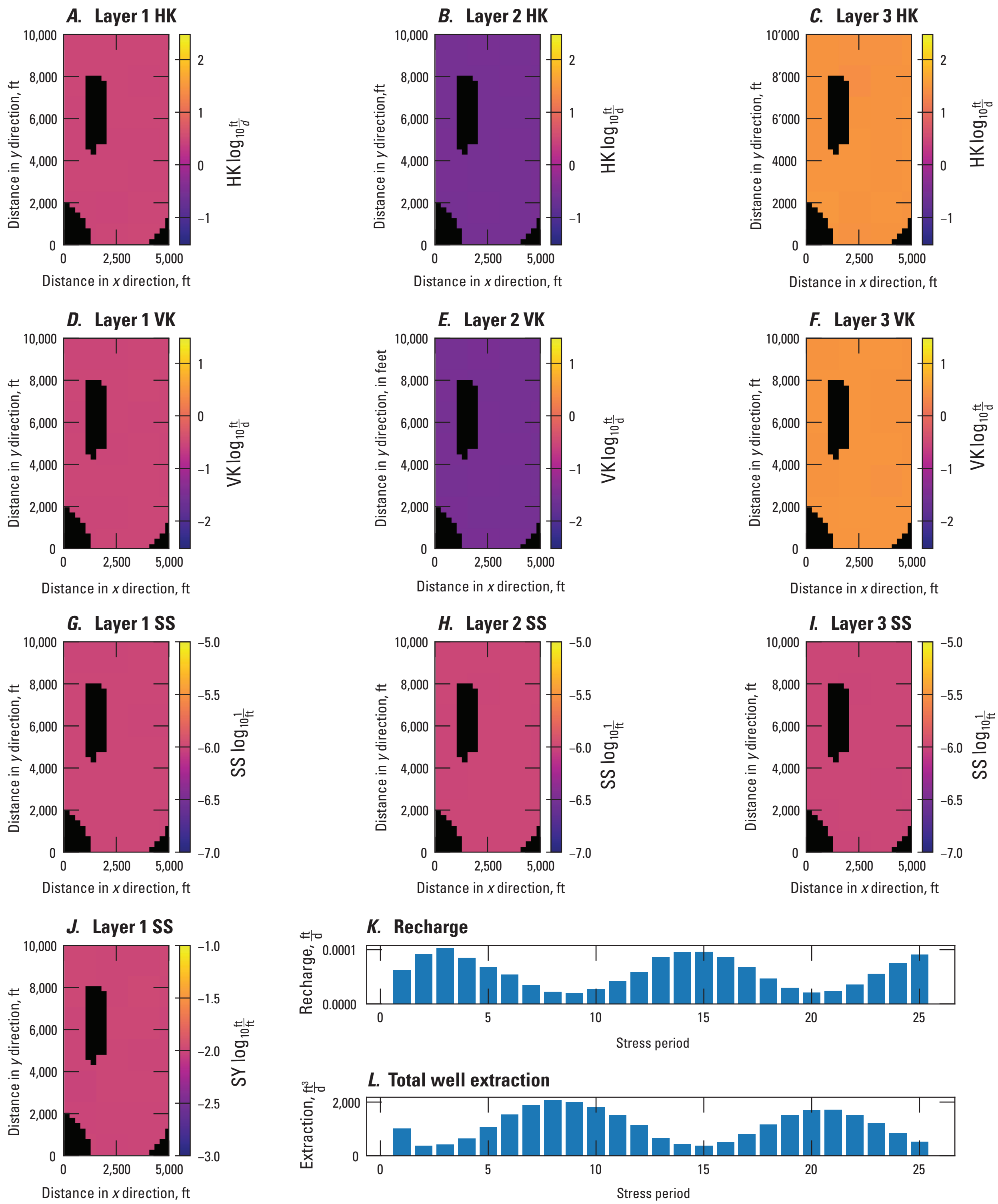


Figure 5. Final PESTPP-GLM optimal parameter values (objective function value of 26) from the basic PESTPP-GLM analysis using only default values. Generally, the spatial (*A–J*) and temporal (*K*, *L*) parameters show very little heterogeneity and are free of problematic extreme parameter values. For comparison, the true fields are shown in figure 2. (HK, horizontal hydraulic conductivity; VK vertical hydraulic conductivity; SY, specific yield; SS, specific storage; ft, feet; d, day; ft^3, cubic feet)

distribution, indicating a high null-space dependence for this forecast (that is, the observations did not contain appreciable information that could constrain parameters important for the forecast). On the other hand, the tailwater forecast posterior uncertainty is substantially narrower than the prior, indicating it has a low null-space dependence.

For the advanced PESTPP-GLM analysis, 482 model evaluations were required across four regularized super-parameter GLM iterations to find a minimum error variance parameter set with a final objective function value of 11. Following these three iterations, PESTPP-GLM proceeded to draw 200 realizations from the final-iteration FOSM-based posterior covariance matrix. These realizations were then evaluated in parallel, yielding 200 realized values for each of the observations used for history matching and for each of the three forecasts of interest. Because the PESTPP-GLM FOSM-based posterior realization selection relies on an assumed linear relation between parameters and observations not adjusted to improve matches to the historical observations, some of the resulting realizations may not acceptably reproduce the historical observations. As such, we filtered (also called "conditioned" or "rejected") unacceptable realizations by simply selecting the 50 realizations with the lowest-objective function (ranging from 35 to 153) for postprocessing (fig. 6). This type of filtering process is an important consideration for any posterior Monte Carlo analysis (FOSM-based or nonFOSM-based). By definition, posterior realizations must respect both the prior parameter distribution and the historical observations; therefore, excluding realizations that do not yield an acceptable fit to the historical observations adheres to the Bayesian framework used.

The filtered posterior realizations from PESTPP-GLM yielded an improved fit to the historical observations compared to the initial parameter values, as evidenced by both the observed-vs-simulated plots and the objective function histogram (fig. 6*A*, *B*, *C*, and *D*). Furthermore, FOSM-based and Monte Carlo-based posterior forecast uncertainty estimates cover the true forecast values (blue bell-shaped curves, histograms, and red line in figure 6*E*, *F*, and *G*, respectively), indicating the combined GLM parameter estimation and uncertainty estimation processes encoded in PESTPP-GLM yielded a successful uncertainty analysis for this demonstration problem (that is, the true value of each forecast is covered by the posterior distributions yielded by PESTPP-GLM). Recall that the conditions used for "truth" in the Freyberg example represented extreme 95-percent quantile conditions for the tailwater forecast.

Figure 7 shows the final minimum error variance (for example, final) parameter set from the PESTPP-GLM parameter estimation process, while figure 8 shows a single posterior parameter realization. Generally, the posterior parameter fields represent spatially expected patterns of heterogeneity with no discernable bias or extreme values.

PESTPP-IES Example

PESTPP-IES was originally documented in White (2018) and implements a form of the iterative ensemble smoother algorithm from Chen and Oliver (2013). Conceptually, at the iteration level, the algorithm in PESTPP-IES combines the efficiency of the GLM algorithm to minimize a least-squares objective function in high-dimensions with Monte Carlo, which is used to empirically estimate the Jacobian matrix and provide information regarding uncertainty.

As with the other tools in the PEST++ V5 suite, PESTPP-IES uses default internal values for all optional control file arguments and, in the absence of specifying any additional arguments, PESTPP-IES can be run on any existing set of PEST input files without modifications. When defaults are used, PESTPP-IES draws the prior parameter ensemble from a diagonal prior parameter covariance matrix constructed from parameter bound information, which assumes parameter bounds approximate the range of 95-percent confidence (this assumption can be adjusted via the PESTPP-IES *par_sigma_range* option). The ability to programmatically generate realizations and evaluate prior-based ensembles without user intervention facilitates an easy entry point for users who want to perform Monte Carlo analysis with an existing model and associated PEST interface (that is, control file, template file[s], instruction file[s]).

Theory: Ensemble Form of Regularized GLM

The parameter adjustment algorithm in PESTPP-IES is fundamentally the same as the PESTPP-GLM regularized GLM algorithm, except parameter and observation/simulated equivalent vectors are replaced with matrices that represent ensembles of parameter and observation/simulated equivalent values; that is, the PESTPP-IES equivalent of equation 3 shown above now becomes:

$$\Delta_{\Theta} = -\left(J^T \Sigma_{\varepsilon}^{-1} J + \lambda * (\Sigma_{\theta}^{-1})\right)^{-1} \Sigma_{\theta}^{-1} (\Theta_0 - \Theta) \\ + J^T \Sigma_{\varepsilon}^{-1} (D_{sim} - D_{obs}) \qquad (8)$$

Where Δ_{Θ} is a parameter change (for example, upgrade) ensemble, Θ is the initial (for example, prior) parameter ensemble, Θ is the current parameter ensemble, D_{obs} is an ensemble of observations values with additive realizations of measurement noise, and D_{sim} is a ensemble of simulated equivalents to observations (resulting from evaluation of the parameter ensemble Θ).

There are two notable differences between the PESTPP-GLM and PESTPP-IES algorithms. First, instead of operating on a single parameter set and single set of residuals, the parameter adjustment algorithm in PESTPP-IES uses an ensemble of residuals and adjusts an ensemble of parameters, meaning that PESTPP-IES yields ensembles of parameters

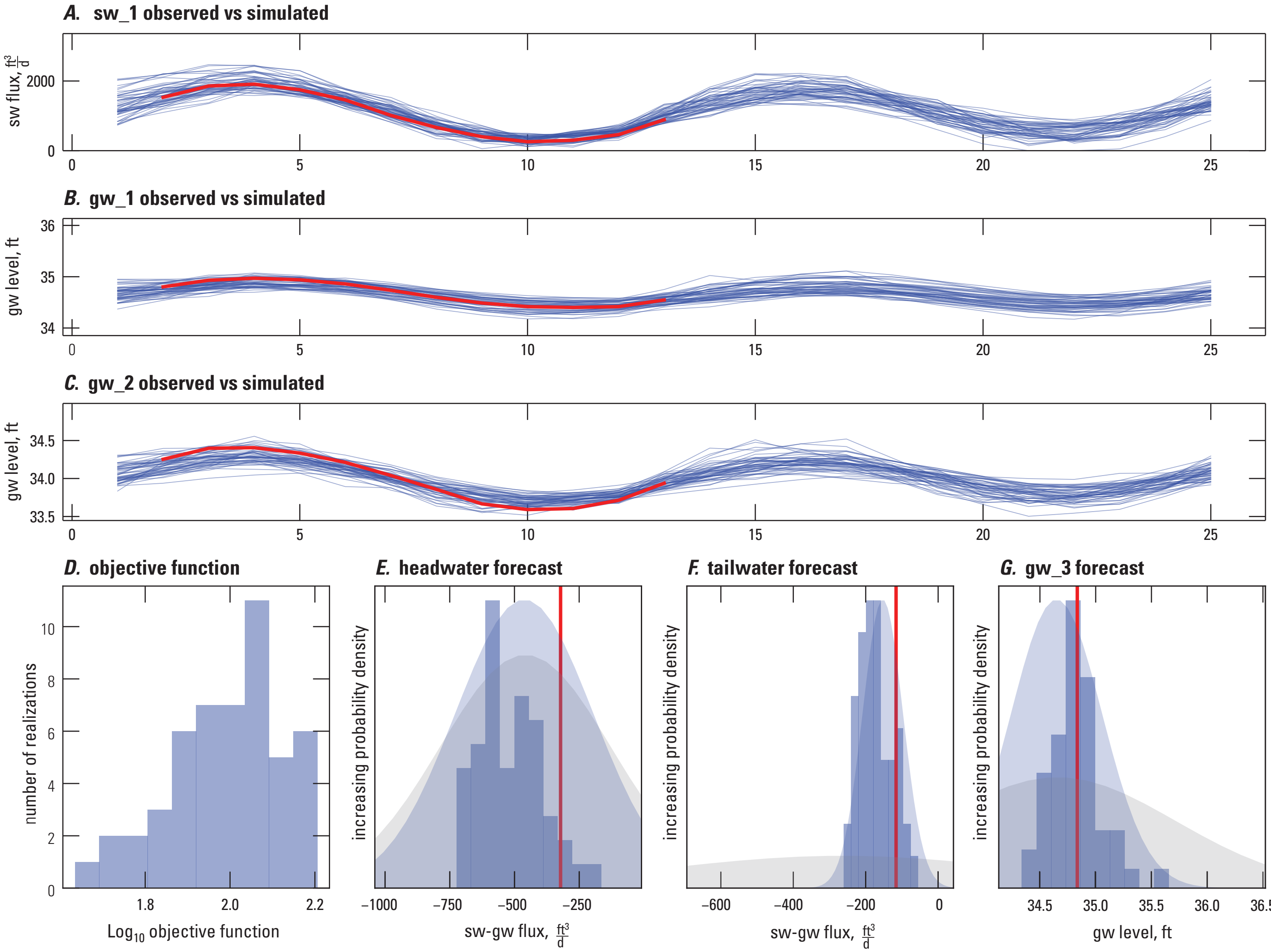


Figure 6. Summary of the advanced PESTPP-GLM analysis for the enhanced Freyberg model. *A*, *B*, and *C* show the observed (red) values versus simulated (blue) values for each of the posterior realizations. These results, taken with *D* (the objective function distribution for the filtered posterior ensemble) show a good level of agreement between the FOSM-based Monte Carlo realizations and the observed states used for history matching. *E*, *F*, and *G* show how both the FOSM prior (gray gaussian curves) and posterior (blue gaussian curves), as well as the Monte Carlo results (blue histograms), cover the true forecast values (red vertical bars). (sw, surface water; gw, groundwater, ft, feet; d, day; ft^3, cubic feet)

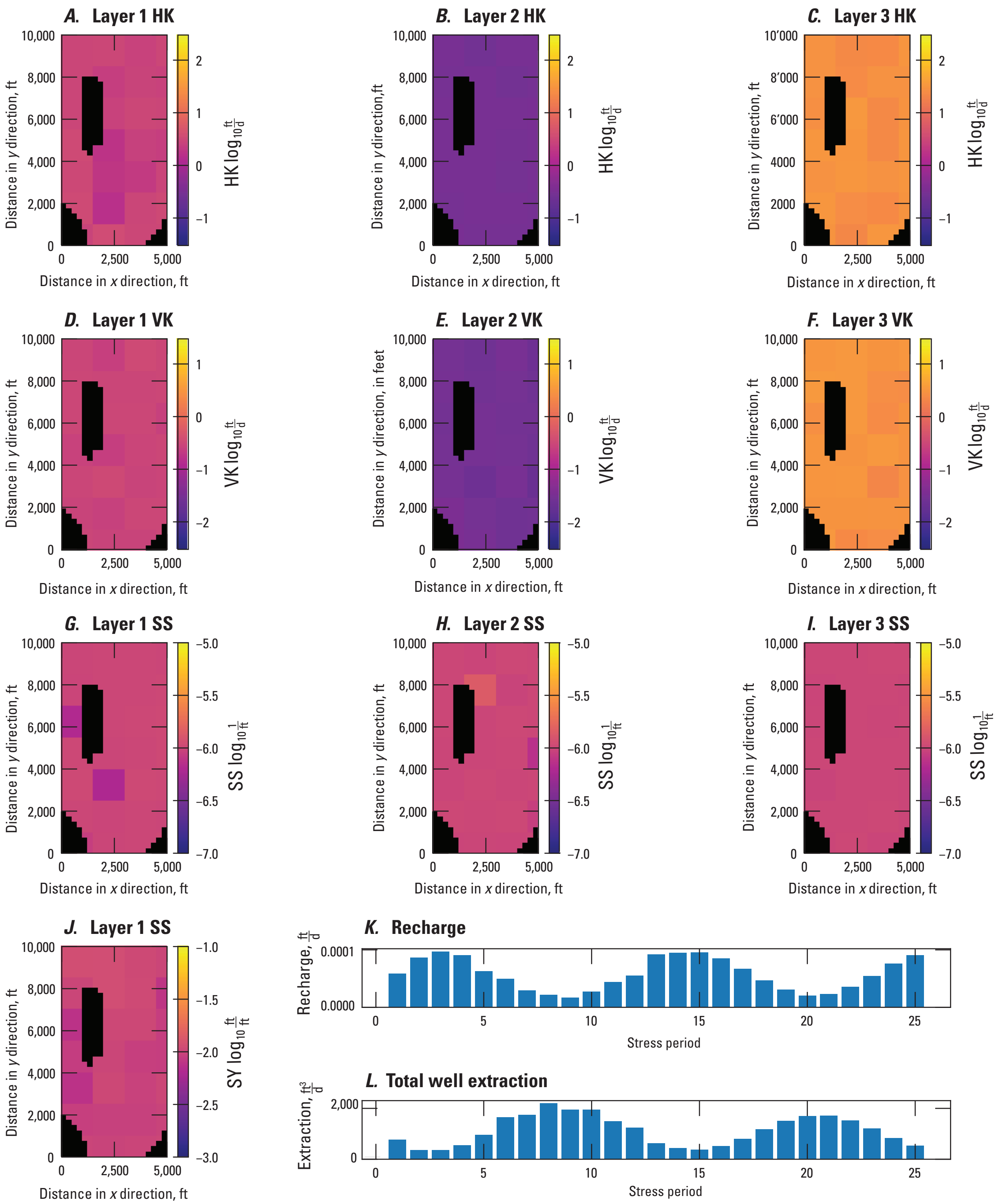


Figure 7. Final PESTPP-GLM optimal parameter values (objective function value of 11) from the advanced PESTPP-GLM analysis. Generally, the spatial (*A*–*J*) and temporal (*K*, *L*) parameters show very little heterogeneity and are free of problematic extreme parameter values. For comparison, the true fields are shown in figure 2. (HK, horizontal hydraulic conductivity; VK vertical hydraulic conductivity; SY, specific yield; SS, specific storage; ft, feet; d, day; ft^3, cubic feet)

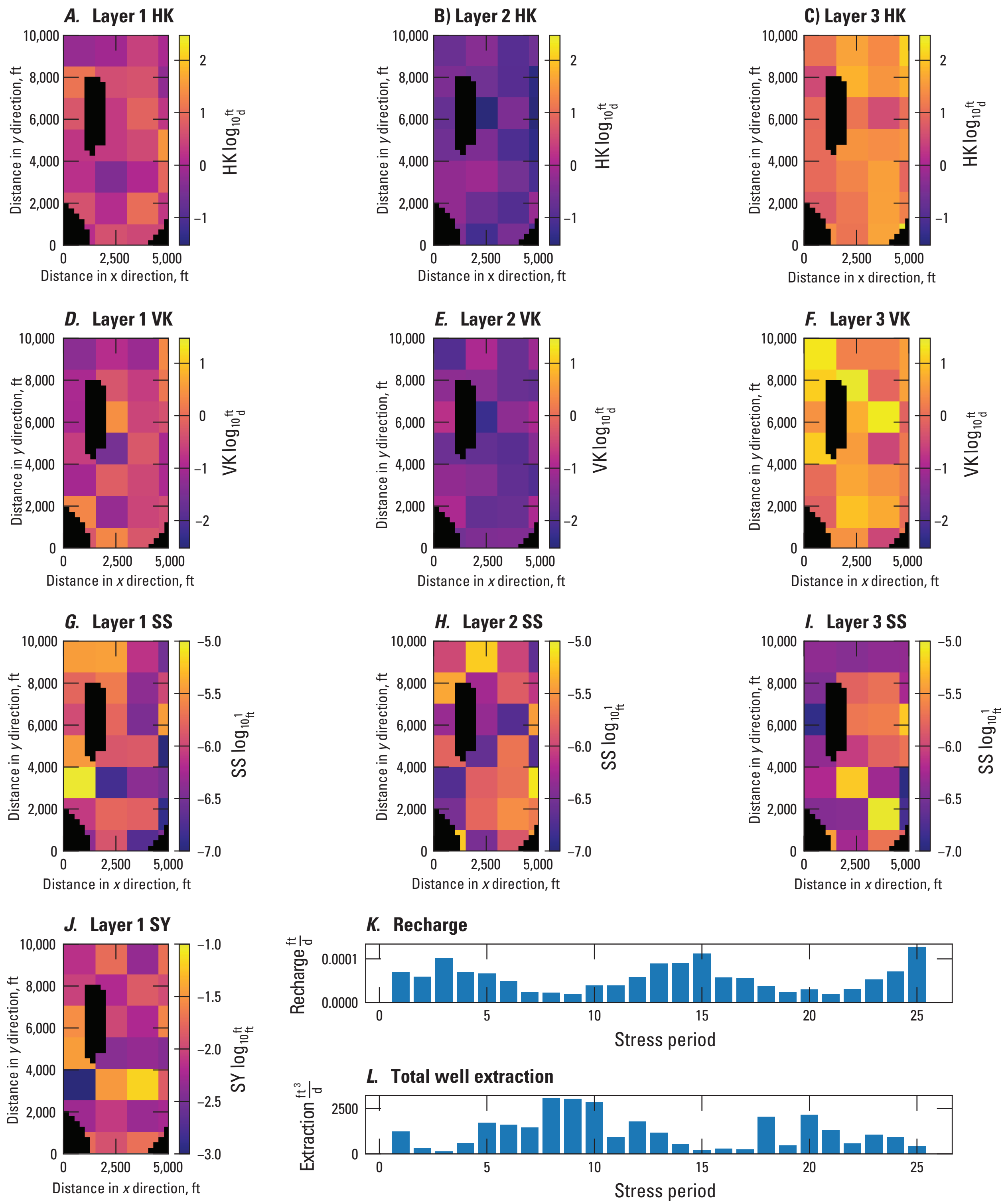


Figure 8. One of 50 realizations derived from the advanced PESTPP-GLM analysis posterior parameter ensemble (objective function value of 36). Compared to figure 7, substantially more spatial (*A*–*J*) and temporal (*K*, *L*) heterogeneity is shown and the blocky parameterization pattern can be seen. However, the patterns of heterogeneity are free of problematic extreme parameter values. For comparison, the true fields are shown in figure 2. (HK, horizontal hydraulic conductivity; VK vertical hydraulic conductivity; SY, specific yield; SS, specific storage; ft, feet; d, day; ft^3, cubic feet)

and simulated equivalents to observations after each iteration. Second, the Jacobian matrix used to construct candidate parameter upgrade matrices is approximated with results from a Monte Carlo-style ensemble evaluation (completed each iteration) via:

$$J \approx (D_{sim} - \overline{D}_{sim})\,(\Theta - \overline{\Theta})^{+} \tag{9}$$

Where $\overline{D}_{sim}$ is an ensemble where each observation vector is filled with the mean value of the corresponding observation vector in D_{sim}, $\overline{\Theta}$ is an ensemble where each parameter vector is filled with the mean value of the corresponding parameter vector in Θ, and $^{+}$ denotes a Moore-Penrose pseudoinverse (for example, Golub and Van Loan, 2012). Conceptually, the two terms of the righthand side of equation 9 are ensembles of deviations around the mean value for each simulated observation equivalent and parameter, respectively. In this way, we see that the form of Jacobian matrix used in PESTP-IES is formed from empirical cross-covariances between parameters and observations, as opposed to a causality (perturbation-based) Jacobian in PESTPP-GLM.

By using the ensemble form of regularized GLM, PESTPP-IES can operate in much higher dimensions than PESTPP-GLM because the number of model evaluations to fill the Jacobian matrix each iteration is controlled by the number of user-selected realizations in the parameter ensemble rather than requiring a minimum of one run per adjustable parameter. Furthermore, in the numerical implementation of equation 8 in PESTPP-IES, direct inversion of the normal matrix is avoided—only the quantity $(D_{sim} - \overline{D}_{sim})$ in equation 9 must be inverted.

Although PESTPP-IES does use an empirical Jacobian matrix for parameter upgrade calculations, this Jacobian matrix is not suitable for any form of first-order analyses such as data worth analyses or FOSM-based uncertainty estimation (Doherty 2015a, b; White and others, 2016) because this empirical Jacobian lacks the causality of the perturbation-based Jacobian used in PESTPP-GLM and PEST and PEST_HP. Users who are interested in first-order analyses are therefore encouraged to instead fill a full-rank Jacobian matrix with PESTPP-GLM once an optimal parameter set is selected.

For users who are interested in a single realization from a PESTPP-IES analysis, it is important to take care in selecting a representative realization. During initialization, PESTPP-IES will default to replacing the last stochastic realization in the prior parameter ensemble with the parameter values listed in the control file and will name this realization “base.” This base realization is then adjusted during the PESTPP-IES iterations (along with the other realizations in the parameter ensemble). The final values for the base realization might therefore be a suitable candidate to serve as a realization that represents the central tendency of the posterior parameter distribution. However, users are cautioned against picking any one realization from a PESTPP-IES analysis for predictive purposes without first evaluating how representative that single realization is.

Interested readers are referred to White (2018) and Chen and Oliver (2013) and references cited therein for more background and theory related to iterative ensemble smoothers.

Localization

The empirical Jacobian matrix used in PESTPP-IES greatly reduces the computational burden required to estimate the relation between parameters and simulated equivalents to observations in high dimensions. However, using an ensemble size that is much smaller than the number of parameters can result in spurious correlation (for example, Anderson, 2007); that is, because the relation between observations and parameters is estimated empirically, and because, in most cases, the number of realizations will be much less than the number of adjustable parameters, the estimated cross-covariance between each parameter and each observation is not without error, the magnitude of this error being a direct consequence of the small number of samples (that is, realizations). The end result is that nonphysical parameter-to-observation relations can emerge and lead to a phenomenon known as “ensemble collapse” where the posterior variance of the parameters is underestimated (for example, Chen and Oliver, 2017).

An obvious way to mitigate ensemble collapse is to use more realizations. However, this comes at a cost of increased computational burden. Another approach to mitigate ensemble collapse is through the use of localization (Anderson, 2007; Chen and Oliver, 2017). In concept, localization allows users to inject expert knowledge about how parameter and observations are and are not related (that is, magnitude of correlation between parameters and observations); for example, information cannot flow backwards in time so an observation from the past cannot be affected by a parameter that represents future forcing conditions. Other examples include disallowing groundwater level measurements effects on porosity parameters or not allowing observations and parameters to be related if they are separated by large distances.

Users can provide PESTPP-IES with a localizing matrix to enforce physically plausible parameter-to-observation relations. Several formats of this matrix are supported, including comma, tab, and space delimited, as well as binary formats. PESTPP-IES also has an option to automate this process by implementing a form of automatic adaptive localization based on the work of Luo and Bhakta (2020), which is briefly described here. With this functionality active, during each iteration, before commencing with parameter adjustments, the empirical correlation coefficient is calculated between each parameter and each observation. A “background” or “error” distribution for this correlation coefficient is also calculated by repeated circular shifting of the observation realization vector and recalculating a correlation coefficient. By definition, the circular shifted correlation coefficient is expected to be zero because there is no reason for the shifted observation vector to be related to the (unshifted) parameter realization vector. By comparing, in a statistical sense, the actual correlation coefficient and the associated error distribution, statistically significant correlations can be identified and retained. From a user perspective, the construction of the localization matrix in this case is automated without requiring user intervention as the elements in this matrix are the absolute values of the empirical correlation coefficients between a parameter and an observation. This localization matrix is then fed forward into the parameter adjustment calculations during a PESTPP-IES analysis.

Note that a localization matrix supplied by the user can be used in combination with the automatic adaptive localization process. In this setting, the automatic adaptive localization process works within the allowed parameter-to-observation relations defined by the localization matrix, whereby the automated process can only hold equal or decrease the value of elements in the localization matrix supplied by the user.

PESTPP-IES Example Application

Similar to the demonstration of PESTPP-GLM, we applied PESTPP-IES to the Freyberg example problem in two ways. First, a basic PESTPP-IES analysis was completed using the same block-zone parameterization scheme used for PESTPP-GLM with no further modifications of input options invoked. Then we applied PESTPP-IES using more advanced options using all 8,175 parameters.

Input Approach

Basic Analysis PEST++ options added to control file: None

Advanced Analysis PEST++ options added to control file:

- **option name:** *ies_parameter_ensemble,* **value:** *ies_prior_en.jcb*: use the parameter realizations stored in *ies_prior_en.jcb* as the initial realizations. This contains the first 50 realizations from the 300 realizations generated during the "truth" model setup process;
- **option name:** *ies_localizer,* **value:** *temporal_loc.jcb*: use the localization matrix stored in *temporal_loc.jcb*; and
- **option name:** *ies_autoadaloc,* **value:** *true*: use automatic adaptive localization in combination with the localization matrix *temporal_loc.jcb*.

For the advanced PESTPP-IES analysis, the full set of 8,175 adjustable parameters for the enhanced Freyberg demonstration were adjusted simultaneously. For this grid-scale parameterization, we used a prior parameter ensemble of 50 that was generated from the prior parameter distribution drawn as part of the "truth" model selection described in the "Example Problem Description" section. The objective function minimum, maximum, and mean for these 50 realizations were 125, 4,316, and 1,168, respectively.

To mitigate issues related to spurious correlations in the advanced PESTPP-IES analysis, we used temporal-based localization combined with automatic adaptive localization as follows: for each observation, recharge and well extraction parameters that occur after the observation's stress period or that occur more than 3 months before the observation were assigned a zero cross-covariance in the Jacobian matrix of equation 8. This operation prevents temporal-based recharge and well extraction parameters from being conditioned on observations that are not physically related. Within the allowable parameter-to-observation relations described by the localization matrix, the automatic adaptive localization process further reduced or removed relations that were identified as statistically spurious.

Results

The following PEST++ output files were evaluated:

- run record file (*.rec*),
- prior observation ensemble (*.0.obs.csv*),
- posterior observation ensemble (*.3.obs.csv*),
- posterior parameter ensemble (*.3.par.csv*), and
- objective function summary (*.phi.actual.csv*).

For both the basic and advanced PESTPP-IES analyses, 378 model evaluations across 4 iterations were required to yield a 50-realization posterior ensemble (no realizations resulted in a failed run, so all were retained). This is an important consideration in the use of PESTPP-IES: the number of model runs is independent of the number of adjustable parameters.

Collectively, the posterior ensemble from the basic and advanced PESTPP-IES analyses closely reproduces the observations, with final objective function values ranging from 10 to 67 in the basic analysis and 2.8 to 38 in the advanced analysis (figs. 9 and 10). Additionally, the posterior forecast probability distributions for both analyses cover the "true" value for all three forecasts with smaller posterior variances than what was produced from the PESTPP-GLM analyses – which again is notable given the extreme 95-percent quartile condition chosen for simulation. We also note the objective function across the PESTPP-IES posterior ensemble is commensurate with the advanced PESTPP-GLM final objective function value (11).

The posterior "base" parameter realization and a sample stochastic posterior parameter realization from both the basic and advanced analyses are shown in figures 11 and 12, and 13 and 14, respectively. These plots show physically plausible parameter values with an absence of apparent bias or extreme parameter values. Note that the total number of runs (378) is appreciably lower than the PESTPP-GLM example (Basic: 1,361, Advanced: 682) even though the number of parameters was much higher in the advanced PESTPP-IES analysis (8,175) than in the PESTPP-GLM analyses (330).

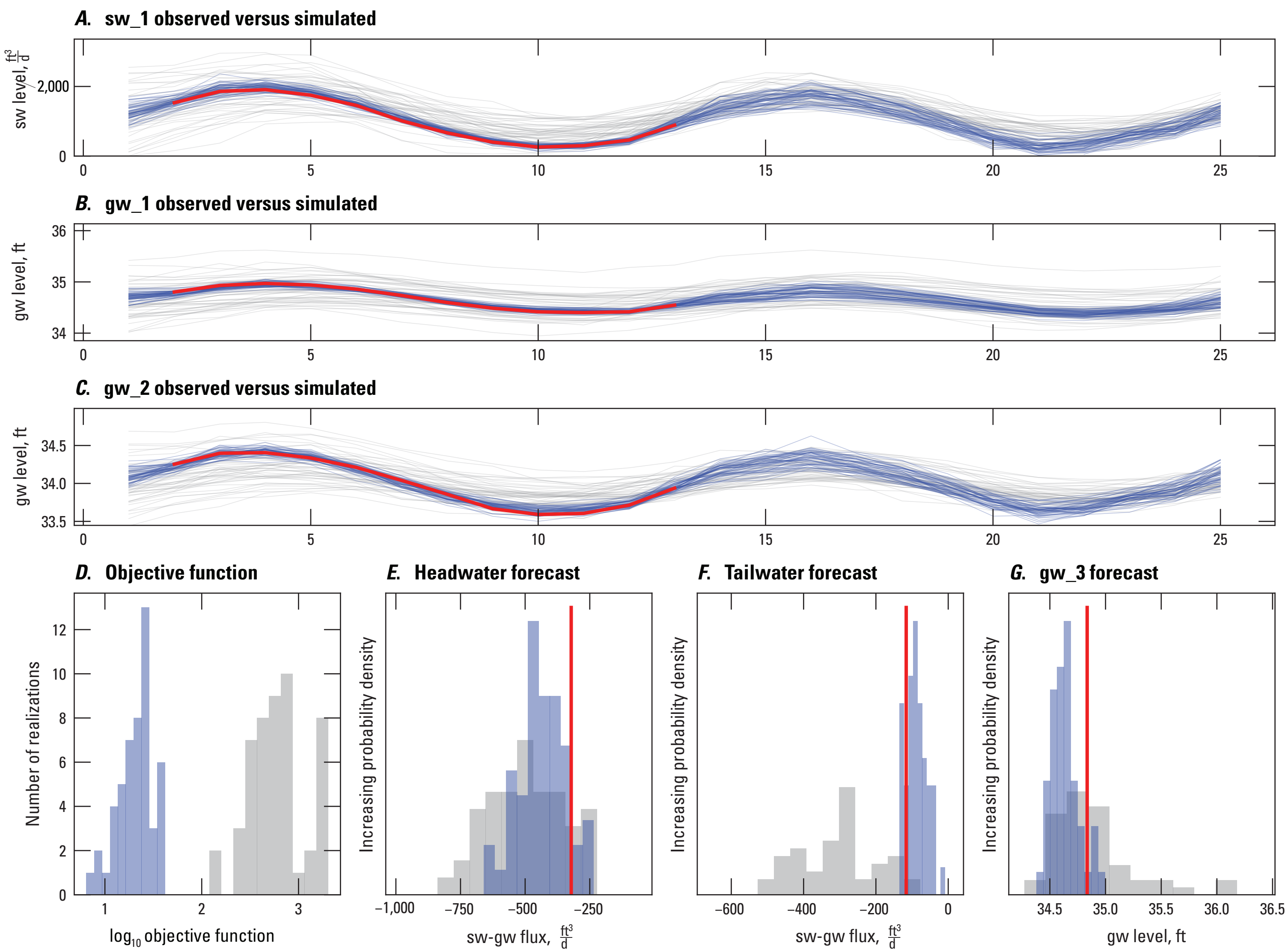


Figure 9. Summary of basic PESTPP-IES application to the enhanced Freyberg model using only default values. *A*, *B*, and *C* show the observed (red) values vs simulated equivalent values for each of the prior (grey traces) and posterior (blue traces) realizations. These results, taken with *D*, the objective function distribution for the posterior ensemble (blue), shows a good level of agreement between the posterior PESTPP-IES ensemble and the observed states used for history matching. Panels *E*, *F*, and *G* show how prior (grey) and posterior (blue) histograms cover the true forecast values (red vertical bars). (sw, surface water; gw, groundwater, ft, feet; d, day; ft^3, cubic feet)

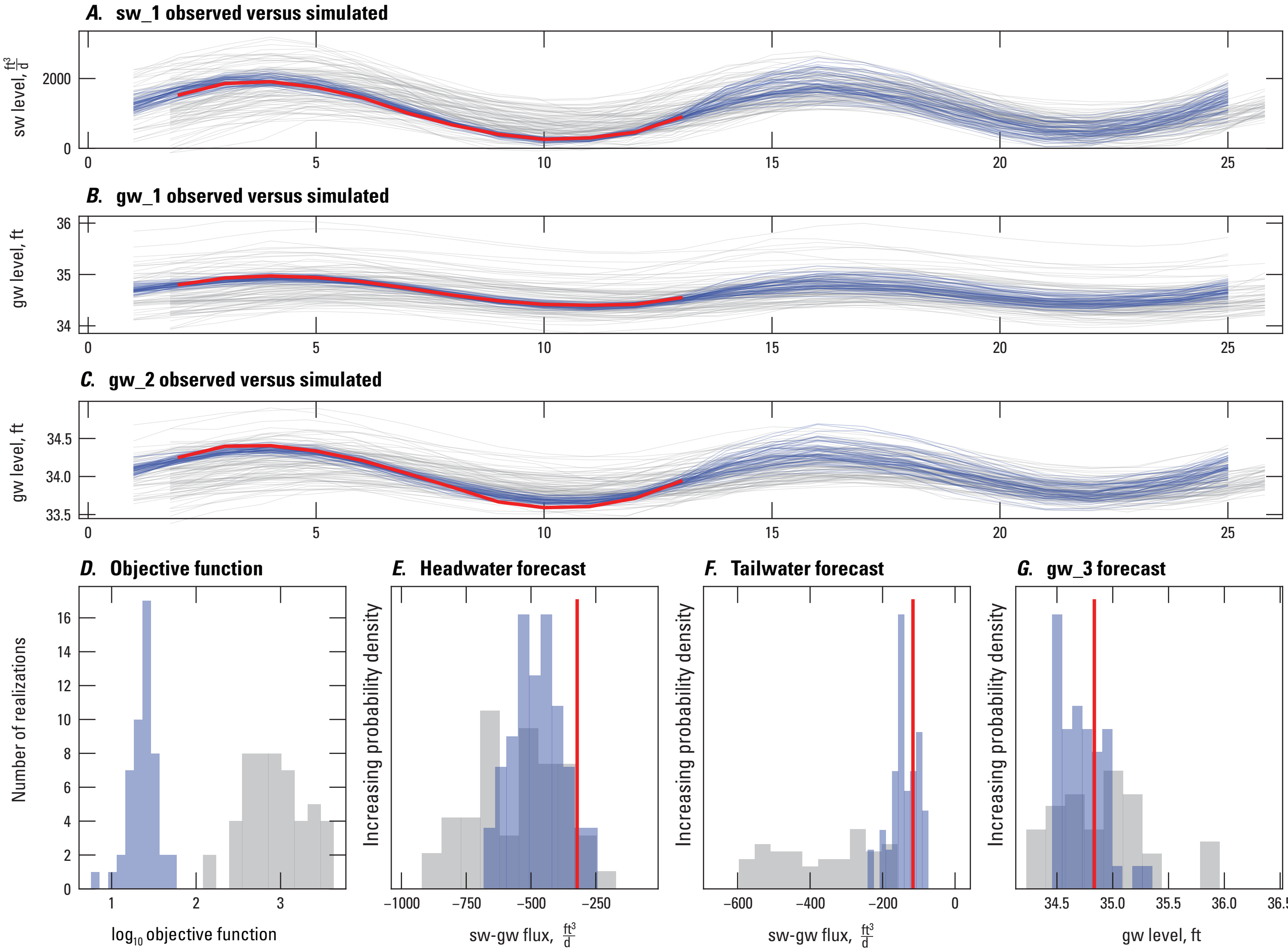


Figure 10. Summary of advanced PESTPP-IES application to the enhanced Freyberg model. *A*, *B*, and *C* show the observed (red) values vs simulated equivalent values for each of the prior (gray traces) and posterior (blue traces) realizations. These results, taken with *D*, the objective function distribution for the posterior ensemble (blue), shows a good level of agreement between the posterior PESTPP-IES ensemble and the observed states used for history matching. Panels *E*, *F*, and *G* show how prior (grey) and posterior (blue) histograms cover the true forecast values (red vertical bars). (sw, surface water; gw, groundwater, ft, feet; d, day; ft^3, cubic feet)

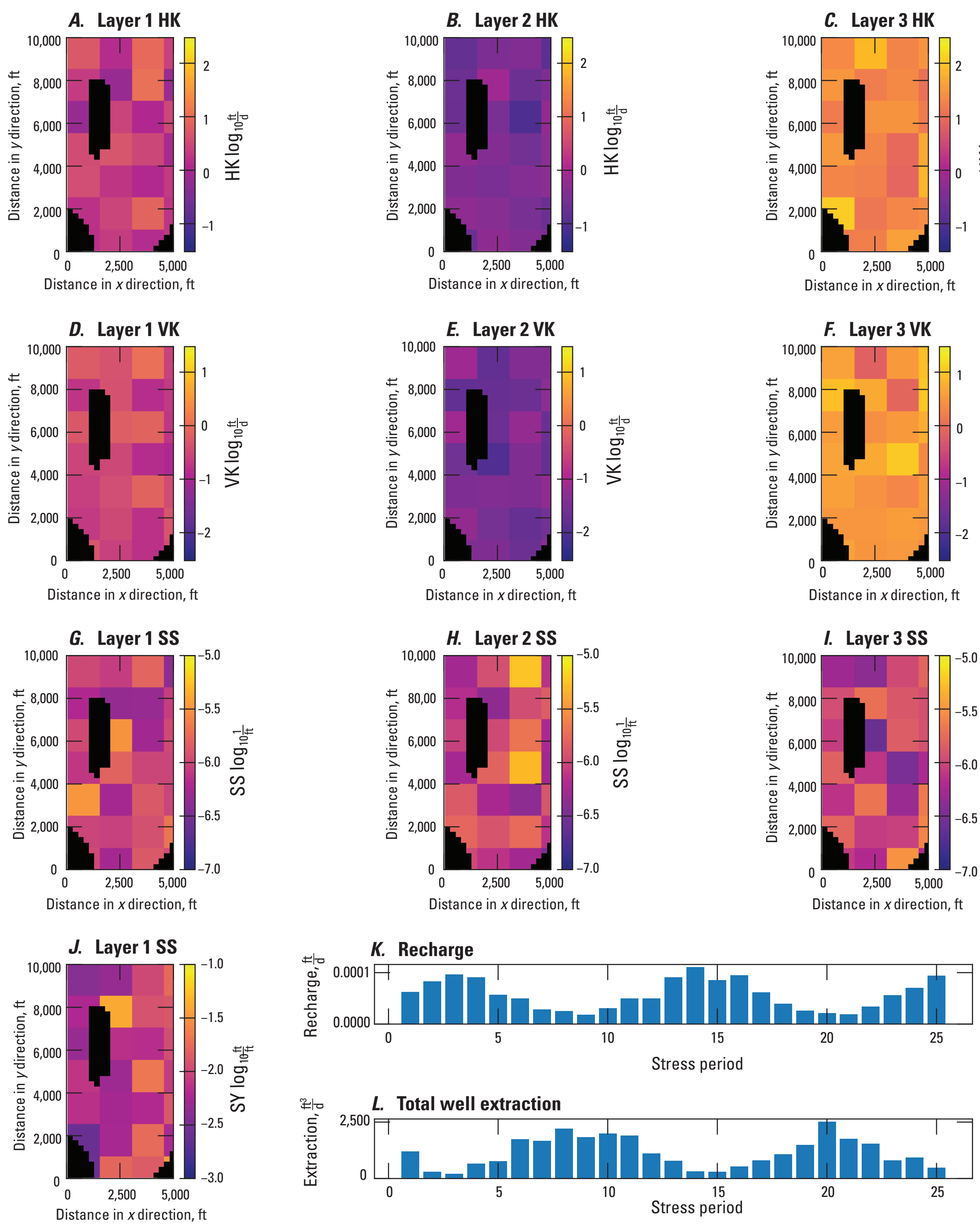


Figure 11. Base realization from the basic PESTPP-IES posterior parameter ensemble using only default values (objective function value of 11). Similar to the PESTPP-GLM analyses, the effect of the block-tying parameter reduction strategy can be seen in *A* through *J*. Spatial (*A*–*J*) and temporal (*K*, *L*) parameter relations display expected patterns and no discernable parameter extremes are evident. For comparison, the true fields are shown in figure 2. (HK, horizontal hydraulic conductivity; VK vertical hydraulic conductivity; SY, specific yield; SS, specific storage; ft, feet; d, day; ft^3, cubic feet)

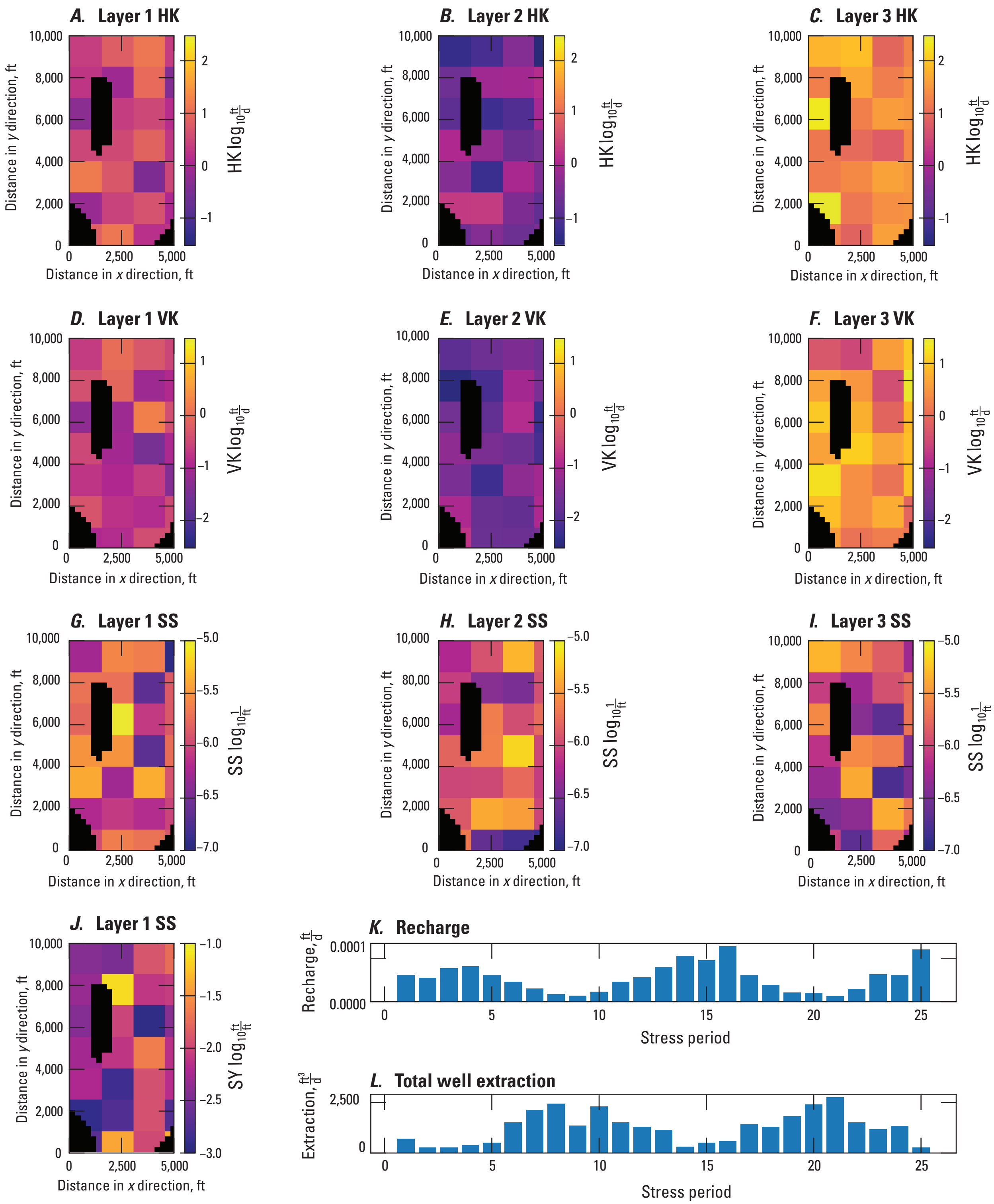


Figure 12. A single stochastic realization of the 50 realizations from the basic PESTPP-IES posterior parameter ensemble that used only default values (objective function value of 24). Similar to the PESTPP-GLM analyses, the effect of the block-tying parameter reduction strategy can be seen in *A* through *J*. Similar to figure 11, spatial (*A*–*J*) and temporal (*K*, *L*) parameter relations display expected patterns and no discernable extreme parameter values are evident. For comparison, the true fields are shown in figure 2. (HK, horizontal hydraulic conductivity; VK vertical hydraulic conductivity; SY, specific yield; SS, specific storage; ft, feet; d, day; ft^3, cubic feet)

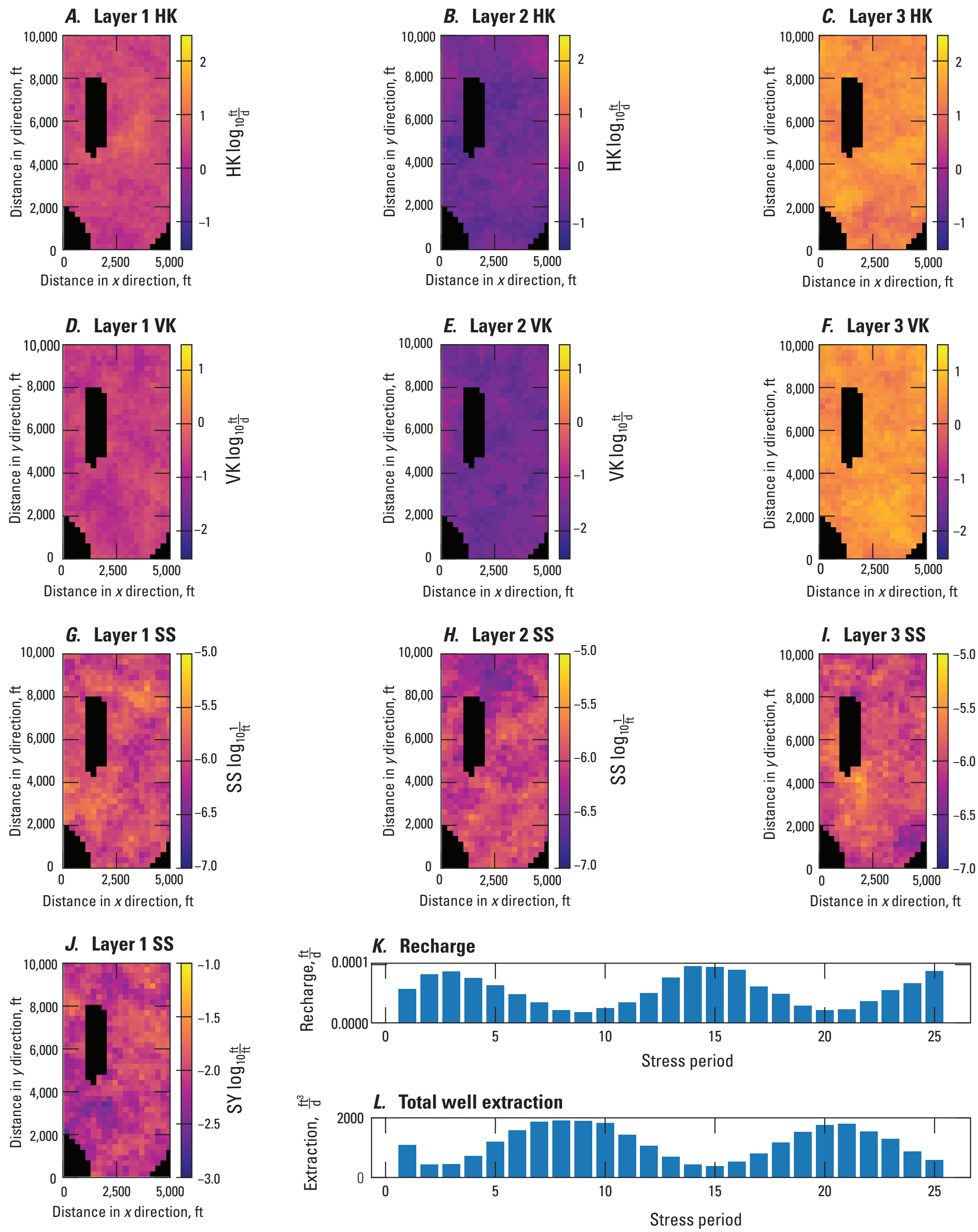


Figure 13. Base realization from the advanced PESTPP-IES posterior parameter ensemble (objective function value of 2.8). Compared to the block-tying parameter reduction strategy, the grid-scale parameterization can be seen on *A* through *J*. Spatial (*A*–*J*) and temporal (*K*, *L*) parameter relations display expected patterns and no discernable parameter extremes are evident. For comparison, the true fields are shown in figure 2. (HK, horizontal hydraulic conductivity; VK vertical hydraulic conductivity; SY, specific yield; SS, specific storage; ft, feet; d, day; ft^3, cubic feet)

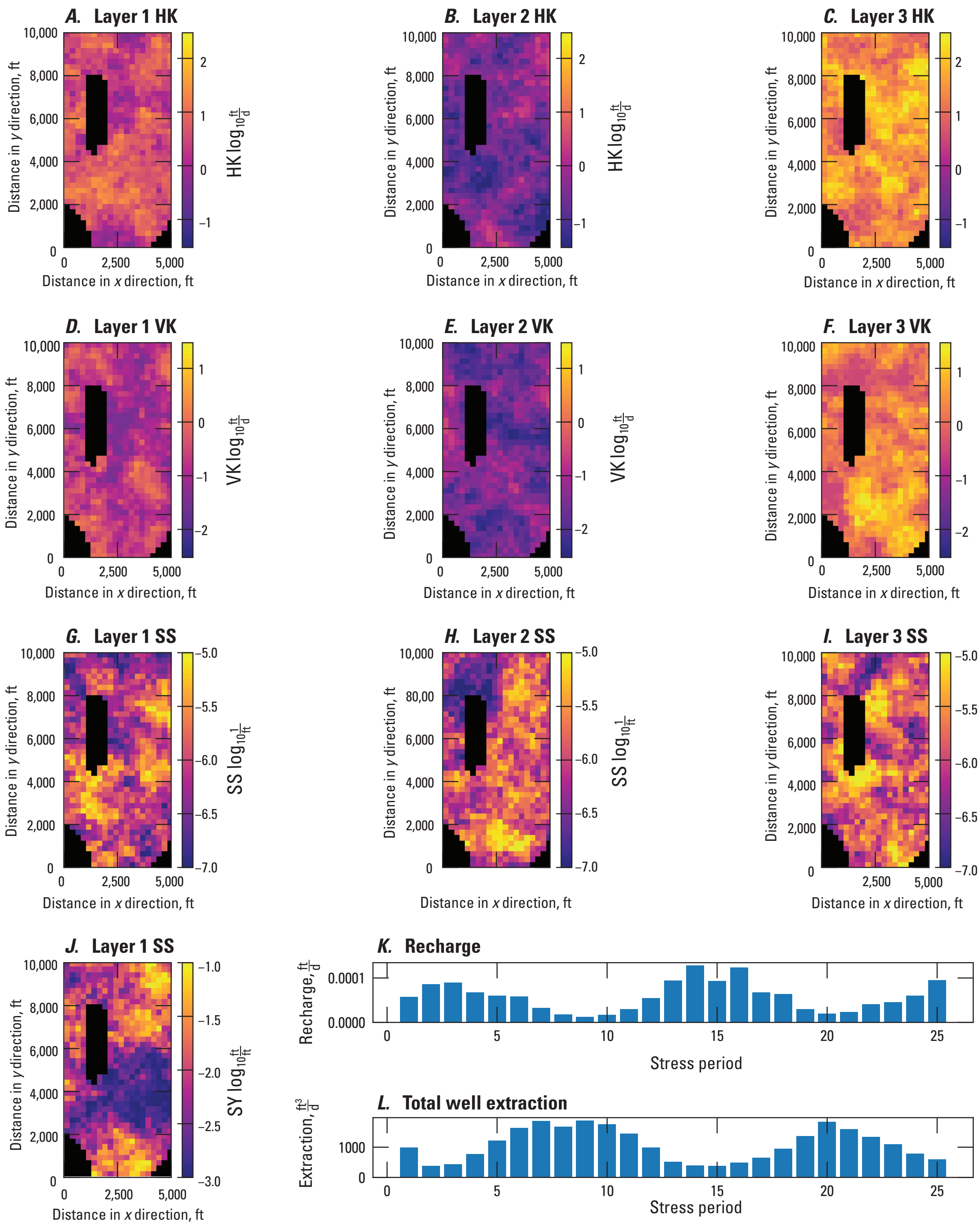


Figure 14. A single stochastic realization from the 50 realizations in advanced PESTPP-IES posterior parameter ensemble (objective function value of 18). Similar to figure 12, spatial (*A*–*J*) and temporal (*K*, *L*) parameter relations display expected patterns and no discernable extreme parameter values are evident. For comparison, the true fields are shown in figure 2. (HK, horizontal hydraulic conductivity; VK vertical hydraulic conductivity; SY, specific yield; SS, specific storage; ft, feet; d, day; ft^3, cubic feet)

PESTPP-OPT Example

PESTPP-OPT is a tool for management optimization under uncertainty (OUU). An example of a resource management optimization question is “what is the maximum pumpage that can be sustained while meeting a stream low-flow condition at given level of risk?,” where the pumpage is what is being optimized and the stream low-flow value is the “constraint” that limits the pumping. Originally documented in White and others (2018), PEST++ V5 implementation of PESTPP-OPT includes several enhancements to facilitate management optimization of high-dimensional decision variable spaces (for example, White and others, 2020) as well as support for multiple forms of chance constraints, which are discussed below.

Theory: Sequential Linear Programming

PESTP-OPT solves the sequential linear programming problem via the simplex method based on the algorithm of Dantzig and others (1955). Following Nocedal and Wright (2006), a minimizing linear programming problem can be written as:

$$\begin{aligned} \text{minimize} &: c^{\mathrm{T}}x. \\ \text{subject to} &: Ax \le b \\ & x \ge 0 \end{aligned} \qquad (10)$$

Where c^{T} is a transpose of the vector of objective function coefficients, x is a vector of decision variable values, b is a vector of required (for example, right-hand side) constraint values, and A is the so-called “response matrix” (Ahlfeld and Mulligan, 2000) that relates decision variable values to constraint values. Conceptually, the decision-variable solution computed by the simplex algorithm crawls along the edges of the sharp-sided high-dimensional polygon (simplex) that contains the feasible solution space until it finds the extrema of the objective function. The sides of the simplex are “sharp sided” because they represent linear constraints.

Interestingly, the response matrix in the linear programming problem is conceptually identical to the Jacobian matrix used in PESTPP-GLM in that it is first-order mapping of inputs (now decision variables) to outputs (now constraints). As such, the response matrix is filled in the same way as the Jacobian matrix in PESTPP-GLM: by evaluating the model once for each decision variable to calculate finite-difference partial first derivatives (eq. 2).

Ahlfeld and Mulligan (2000) and White and others (2018), and references cited therein, provide more background on mathematical optimization—and, specifically, sequential linear programming—in the context of environmental simulation.

Chance Constraints

PESTPP-OPT implements management OUU by use of chance constraints (Wagner and Gorelick, 1987). Conceptually, chance constraints recognize the uncertainty in constraints that are derived from model outputs. These constraints include uncertainty because the model’s ability to simulate a constraint is uncertain; that is, the constraints are uncertain in as much as the model inputs (represented by parameters) are uncertain and that the model outputs are sensitive to these parameters. In this way, there is a relation between the forecasts of interest during uncertainty analysis and the model-based constraints used in OUU.

PESTPP-OPT allows users to define uncertainty in model-based constraints in several ways. The simplest form of defining model-based constraint uncertainty is through the weights listed in the control file. If the *opt_std_weights(true)* option is invoked, the PESTPP-OPT interprets the weights for model-based constraints as standard deviation values. These standard deviations must be estimated externally from PESTPP-OPT using FOSM-based uncertainty analysis or empirically from an existing ensemble. The *opt_std_weights(true)* overrides all other chance constraint options. Note that these weights used for optimization are not related to the weights used to form the objective function in PESTPP-GLM and PESTPP-IES.

PESTPP-OPT also allows the user to utilize internal FOSM-based constraint uncertainty estimation directly. This form of chance constraint operation requires a Jacobian matrix relating (uncertain) parameters to model-based constraints. Equipped with this Jacobian matrix, PESTPP-OPT will calculate the uncertainty in model-based constraints using the same FOSM estimation process used to estimate forecast uncertainty in PESTPP-GLM (see equations 4, 5, and 6).

Both weight-based and FOSM-based chance constraints rely on an assumed Gaussian distribution. Combined with the user-specified risk value, the value of the model-based constraint that is used in the linear programming problem is shifted along the implied Gaussian distribution (see White and others [2018] for a schematic explanation of the concept of chance constraints and risk shifting). In this way, the resulting optimal solution respects the users’ stated risk stance while incorporating uncertainty in the model-based constraints yet yields a single, deterministic optimal management strategy—a modeling-analysis outcome that is more easily incorporated into decision making.

The third and most robust option for implementing chance constraints in PESTPP-OPT is through use of a “stack” (Bayer and others, 2008; Sreekanth and others, 2016). The term “stack” refers to an ensemble of parameter realizations and a corresponding ensemble of simulated model-based constraint values. By evaluating the stack at the current point in decision variable space, an empirical (distribution-free) probability density function can be used to risk-shift the model-based constraint values for use in the linear programming solver. In this way, stack-based chance constraints remove the assumption of a Gaussian distribution for each model-based constraint. Stack based chance constraints require an existing “stack” (for example, ensemble) of parameter values that represent parameter uncertainty. Furthermore, if users want to include posterior parameter uncertainty in a stack-based chance constraint analysis, a posterior ensemble of parameter values would need to be supplied. However, the stack can be derived using PESTPP-IES or the FOSM-based Monte Carlo analysis of PESTPP-GLM.

PESTPP-OPT Example Application

We applied PESTPP-OPT to the enhanced Freyberg model by constructing a hypothetical management optimization problem. In this problem, groundwater extraction rates are treated as decision variables and surface-water/groundwater exchange along the headwater and tailwater reaches are treated as constraints that require a minimum amount of groundwater to be discharged to surface water, thereby setting up a classic competition for resources or conjunctive use management optimization problem.

To form this hypothetical management optimization problem, extraction rates for each groundwater extraction well (6) were treated as a decision variable for each transient stress period (24), for a total of 144 decision variables. These decision variables were assigned a lower limit of 0.0 cubic feet per day (ft^3/d) and maximum of 500.0 ft^3/d for an individual well. Model-based constraints were specified such that the aggregated groundwater contribution to surface-water must be at least 250 ft^3/d for both the headwater and tailwater reaches for all 24 transient stress periods. Additionally, we specified a prior-information constraint that the total groundwater extraction rate for all wells must be at least 750 ft^3/d for each of the 24 transient stress periods. Combined with decision variable upper and lower bounds, these constraints form a management optimization problem that is bounded from above and below. By optimizing extraction rates for the full 24-month period, a 2-year-ahead management optimization problem is defined.

We completed three PESTPP-OPT analyses: (1) a risk-neutral analysis, where uncertainty in the model-based constraints is not considered; (2) a risk-averse analysis, where a 95-percent risk aversion stance is made using FOSM-based chance constraints; and (3) risk-averse analysis, where a 95-percent risk aversion stance is made using the posterior parameter ensemble from the advanced PESTPP-IES analysis as the parameter stack. Under the two risk-averse analyses, the optimal groundwater extraction strategy is allowed a 5-percent probability of reducing surface-water/groundwater exchange below the desired 250 ft^3/d threshold for both the headwater and tailwater reaches during any stress period, but lower groundwater extraction rates are expected to meet the required groundwater-to-surface-water discharge constraints.

The risk-neutral analysis does not require the user to define uncertainty in model-based constraints and the FOSM-based chance constraints, similar to the FOSM processes in PESTPP-GLM, relies entirely on quantities derived from inputs in the control file. However, stack-based chance constraints require specification of a parameter stack. In this demonstration, the stack input to PESTPP-OPT was derived from the previously calculated posterior parameter ensemble from the advanced PESTPP-IES example described previously. In this way, the workflow builds logically on previous steps and leverages linkages between history-matching and uncertainty analysis and between uncertainty analysis and decision support via OUU. As such, this example serves as a powerful and efficient PEST++ V5 workflow that can be applied by the user to a range of decision-making problems.

Input Approach

PEST++ options in control file:

- **option name:** *opt_dev_var_groups*, **value:** *welflux*: treat only the parameters in the parameter group welflux as decision variables;
- **option name:** *opt_direction,* **value:** *max*: maximize the objective function because well extraction rates are treated as positive parameters in the control file;
- **option name:** *opt_risk,* **value:** *0.95*: (risk averse analyses only) seek a 95-percent risk-averse solution; and
- **option name:** *opt_par_stack,* **value:** *par_stack.csv*: (stack-based analysis only) use the parameter realizations store in par_stack.csv as the parameter stack.

In addition to the PEST++ options, the following modifications to the control file were made to implement the PESTPP-OPT analyses:

- changed the parameter transformation of the well flux parameters to “none” and changed the upper and lower bounds parameters to 500.0 and 0.0 ft^3/d, respectively. This is necessary as the optimization solution in PESTPP-OPT does not accept log transformed decision variables, and we want to allow groundwater extraction at given stress periods and locations to be zero (as long as the minimum required supply constraint is satisfied);
- reassigned the observation values for the headwater and tailwater observations across all transient stress periods to be −250 ft^3/d and changed the group name of these observations to be “less_than_flux.” This results in those observations to be interpreted as less-than inequality constraints;
- added 24 prior information equations to the control file, one for each stress period, that are the sum of the six pumping well rates for that stress period. The summand for these equations was specified to be 750.0 ft^3/d and the group name for these equations was specified to be “greater_than_flux,” which results in these prior

information equations being interpreted as greater-than inequality constraints. This constraint ensures that a minimum of 750.0 ft^3/d is extracted from the groundwater system during each stress period; and

- for the analysis using FOSM-based chance constraints, all adjustable parameters except stress-period scale recharge parameters were fixed, yielding 25 parameters to represent model input uncertainty in the FOSM process. This limits the number of additional model evaluations devoted to the chance-constraint process while still accounting for the dominant source of uncertainty in the simulated surface-water/groundwater exchange, as seen in the PESTPP-SEN results.

Results

PEST++ output file evaluated:

- run record file (*.rec*),
- parameter (decision variable) value file (*.par*), and
- residual value (*.res*).

PESTPP-OPT required 147 model runs for the risk-neutral analysis; 172 model runs for the FOSM-based, risk-averse analysis; and 197 model runs for the stack-based, risk-averse analysis and reported an optimized total pumpage of 39,700 ft^3/d, 30,800 ft^3/d, and 26,400 ft^3/d respectively. The results of the PESTPP-OPT management optimization analyses are summarized in figure 15 and figure 16. As expected, a risk-averse management strategy results in less groundwater available for extraction and use; this represents the cost of the desire to hedge against uncertain, but unwanted, environmental outcomes. Furthermore, the FOSM-based PESTPP-OPT analysis yields more groundwater available for extraction compared to the stack-based PESTPP-OPT analysis. This is a result of the stack-based analysis including more parameters as uncertain (leading to higher uncertainty in the model-based constraints) and also the relaxation of the assumed Gaussian distribution of the model-based constraints.

Suggestions for Applying PEST++ V5

The examples discussed in the previous sections collectively provide one suggested workflow for moving progressively through a decision-support modeling problem using the PEST++ V5 tools. The PEST++ V5 software suite, however, can have many workflows, especially as it has increased sophistication and capability compared to previous versions. Moreover, the PEST++ V5 is also intended to complement the PEST software suite (Doherty, 2015a), which is also under active development; therefore, a user may want to access tools from both PEST++ V5 and PEST software suites. This section provides a distilled listing of additional suggestions for using PEST++ V5 as a stand-alone application as well as in concert with the PEST software suite.

Use of PEST++ Defaults

To facilitate robust results, tools in the PEST++ suite use default internal values for all optional control file arguments, which are automatically invoked in the absence of the user specifying a value for the input. As shown using the example problem, PEST++ V5 default values can provide reasonable solutions but can also often be improved upon with specification of PEST++ options. Therefore, extensive use of the default values is suggested when first implementing a PEST++ V5 tool as it is most likely to provide insights on the modeling problem—insights not confounded by potential suboptimal user-specification of PEST++ inputs. Once a result is obtained with defaults, modification to default values can be systematically used. The results from the default-value run can then be used to assess the effectiveness of the modification to a default value. Note that the value of each PEST++ option (default and user-supplied) is written to the run record file, which allows easy inspection of what values were used in any PEST++ run.

Interoperability with PEST, BeoPEST, and PEST_HP

A design goal of PEST++ Versions 1 and 3 was backward compatibility of input and output with the PEST software suite (www.pesthomepage.org). This compatibility allows tools from both software suites to be used and facilitates pre and postprocessing with a large set of existing utilities included in the PEST suite of tools. The tools in the PEST++ V5 suite can also be used interchangeably with the PEST (Doherty, 2015a, b), BeoPEST (Schreüder, 2010) and PEST_HP (Doherty, 2015a). However, to enhance access to new capabilities within PEST++ V5, the input and output have augmentations to the existing PEST formats; for example, the PEST software suite has a specific control file and template file format and model input length requirements that are relaxed in PEST++ V5. PEST++ V5 extends the allowable lengths for all quantities, including parameter and observations names, to 200 characters (compared to 12- and 20-character limits for parameter and observation names, respectively, in PEST). Likewise, PEST++ V5 template files do not have limits on the length of a line in template/input and instruction/output files (currently [2020] PEST uses a 2000-character limit).

PEST++ V5 also allows the user to use an optional more-human-readable control file format than previous versions. The optional format uses keyword-value pairs and supports external comma-separated-value files for the “parameter groups,” “parameter data,” “observation data,” and “prior information” sections of the control file. The standard and new control file formats are described in detail in appendix 1. Using the new format is not required; PEST++ V5 can read PEST input files without modification. However, if a user

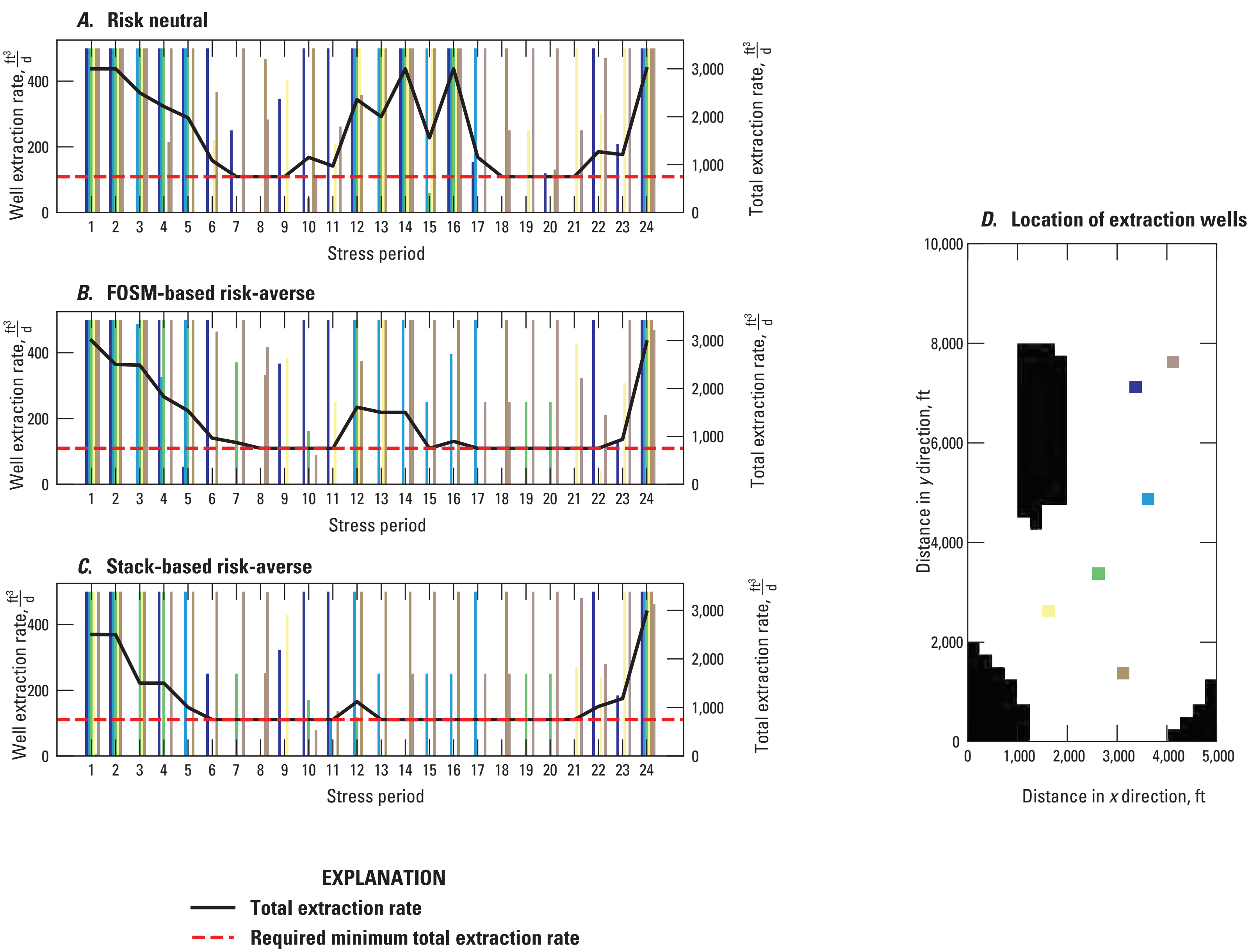


Figure 15. Summary of the PESTPP-OPT optimal decision variables for the *A*, risk-neutral, *B*, first-order, second-moment (FOSM) based risk-averse, and *C*, stack-based risk-adverse cases. The well extraction rate bar colors in *A*, *B*, and *C* correspond to the location markers in *D*. (ft, feet; d, day; ft³, cubic feet)

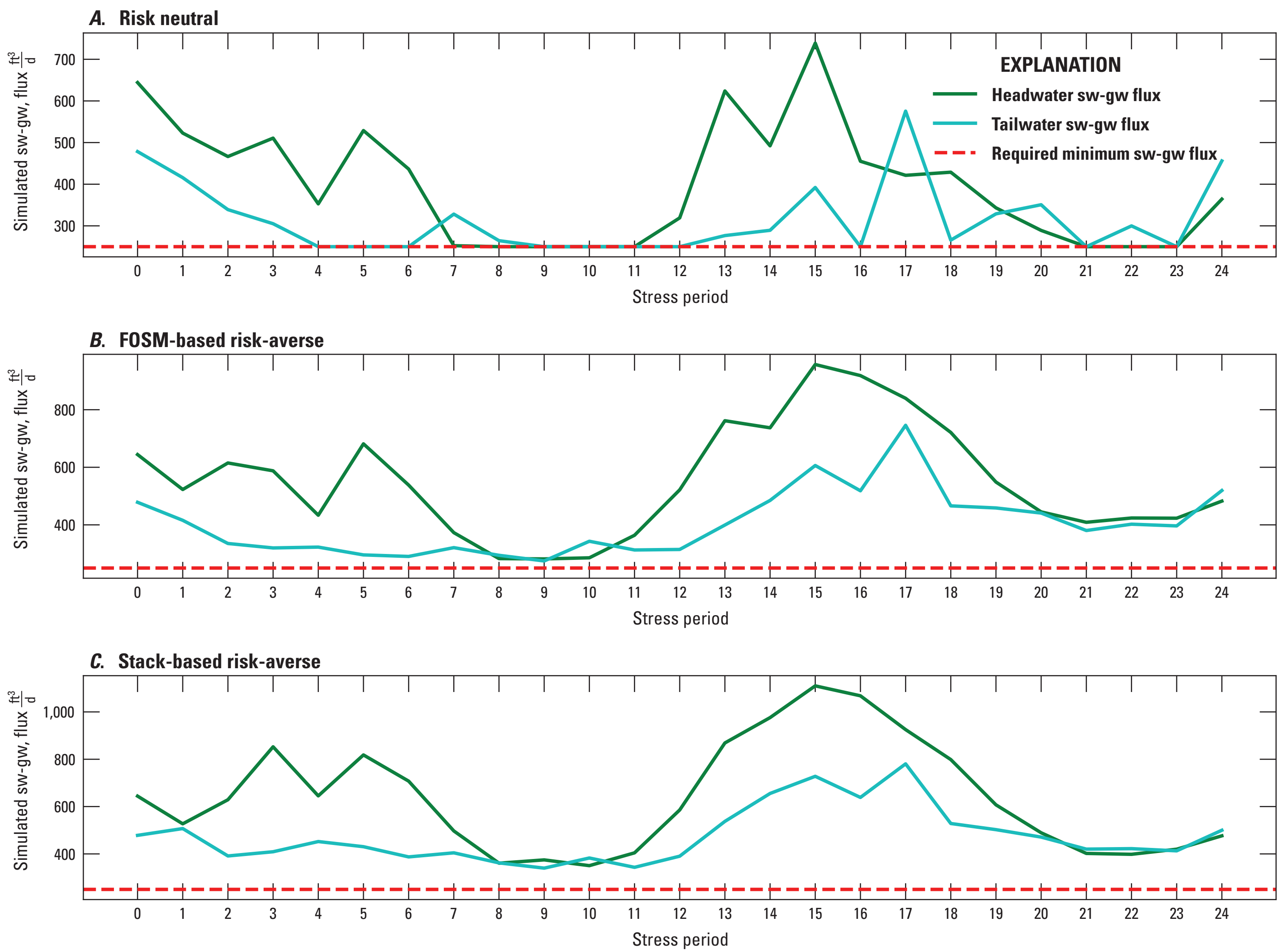


Figure 16. Summary of the PESTPP-OPT optimal constraint values for the *A*, risk-neutral, *B*, first-order, second-moment (FOSM) based risk-averse, and *C*, stack-based risk-averse case. For the risk-neutral case (*A*), the simulated surface-water/groundwater exchanges are used directly in the optimization solution process. For the FOSM-based chance constraint analysis (*B*), the simulated surface-water/groundwater exchanges are simulated slightly higher than the desired 250 ft^3/d threshold to account for the uncertainty in the model-based constraints and 95-percent risk-averse stance. For the stack-based analysis (*C*), the simulated surface-water/groundwater exchanges are simulated even higher than the FOSM-based analysis because the stack-based uncertainty in the model-based constraints is higher than the FOSM-based uncertainty. (sw, surface water; gw, groundwater; ft, feet; d, day; ft^3, cubic feet)

invokes the additional file format flexibility within PEST++, model inputs would need modification to conform with PEST protocols in order to run within the PEST software suite; the python module pyEMU (White and others, 2016) can translate between control file formats. This approach allows access to enhancements while maintaining the original design goal of input compatibility; once a user has setup a PEST-style interface around their model, they can use either software suite to implement sophisticated decisions-support analyses.

Workflows Using a Parallel Run Manager

All tools in the PEST++ V5 suite include both a serial and TCP/IP-based parallel run manager. As the name implies, the serial run manager runs each requested model evaluation in serial and in the directory where the PEST++ tool was invoked. Although the serial run manager is capable of fully executing any PEST++ tool to completion, in a highly parameterized context, typically the serial run manager is only invoked to check the construction of the chain of template file-model input-output extraction steps on a local machine (that is, setting NOPTMAX to 0 and running a PEST++ tool for a single model run). The PEST++ V5 tools are instantiated with the serial run manager by typing at the command prompt:

PESTPP-XXX <case name>.pst

where PESTPP-XXX refers to the PESTPP executable (PESTPP-SEN, PESTPP-GLM, PESTPP-IES, or PESTPP-OPT) and <case name> is the control file name (with or without the file extension).

The PEST++ suite TCP/IP-based parallel run manager is based on the PANTHER run manager (Welter and others, 2018), using the concepts of Schreüder (2010), which tolerates model run failures and includes multithreaded agents so that unneeded model runs can be preempted and runs allocated to lost agents can be efficiently rescheduled elsewhere. Master and agent mode of built-in PANTHER is instantiated in the same way as BeoPEST and PEST_HP:

- Master: PESTPP-XXX <case name>.pst /h:<TCP/IP port number>
- Agent: PESTPP-XXX <case name>.pst /h <master IP address>: <TCP/IP port number>

where PESTPP-XXX refers to the PESTPP executable (PESTPP-SEN, PESTPP-GLM, PESTPP-IES, or PESTPP-OPT), <case name> is the control file name (with or without the file extension), <TCP/IP port number> is the integer port number for communication, and <master IP address> is the IPv4 address or hostname of the machine where the master instance is running. The same TCP/IP port number must be specified on the Master and Agent command lines. Typically, agents are launched from separate run directories on multiple computers.

Limitations of Version 5

Most notable limitations of the PEST++ V5 software suite are the following:

- Use of a full observation noise covariance matrix is supported for the FOSM-based uncertainty calculations in PESTPP-GLM and in PESTPP-IES. However, a full observation noise covariance matrix is not supported in the PESTPP-GLM upgrade calculation process.
- Similar to an SVD-Assist run in PEST, PESTPP-GLM does not perform a final run with the best parameters. This requires the user to construct and execute a final run locally when using parallel run management. However, PEST++ does provide the best-fit parameter values, as well as the model output corresponding to observations and residuals associated with the best model run, so that a user simply needs to update the control file with the best parameters and rerun PEST++ with NOPTMAX set to 0. The PEST utility parrep.exe can facilitate this update, as can pyEMU.
- The Jacobian (*.jco*) file for PESTPP-GLM superparameter iterations is written in terms of the superparameters, whereas PESTPP-GLM writes the base parameter Jacobian to a *.jcb* file. Both of these files are compatible with the PEST *.jco* file specification.
- The localization solution of PESTPP-IES requires solving equation 8 for each adjustable parameter using the observations that meet localization criteria. As such, a localized solution for 500,000 or more parameters can take substantial time, depending on the number of observations. The *ies_num_threads* option will invoke a multithreaded solution process that can reduce the solve time.
- PESTPP-IES uses weights listed in the control file for two purposes. First, these weights are used during upgrade calculations and for formation of the objective function. This is how PESTPP-GLM, as well as PEST (and PEST_HP) use weights. But PESTPP-IES also uses weights for generating realizations of measurement noise through the covariance matrix implied by these weights, making use of the theoretical defini-tion of weights as the inverse of observation noise (Doherty, 2015a, b). These two uses can be noncommensurate if weights are adjusted subjectively to balance the objective function and steer the upgrade process towards the components of the objective function that are most important for decision support purposes (Doherty and Welter, 2010). In this case, the realizations of noise generated from the subjectively adjusted weights may not be representative of observation noise. In this case, users can generate a noise ensemble external to PESTPP-IES with the PEST utilities or pyEMU before subjectively adjusting the weights, or

users can employee the *ies_no_noise* option to ignore the contribution of measurement noise in the posterior parameter ensemble.

Although this program has been used by the U.S. Geological Survey (USGS), no warranty, expressed or implied, is made by the USGS or the U.S. Government as to the accuracy and functioning of the program and related program material nor shall the fact of distribution constitute any such warranty, and no responsibility is assumed by the USGS in connection therewith.

Summary

PEST++ Version 5 augments and extends capabilities of the previous PEST++ Version 3. Version 5 now represents an object-oriented suite of tools that implement a range of analyses designed to complement decision support using environmental models. These tools make extensive use of default values, work through a model-independent interface, and have seamless access to fault-tolerant parallel run management. The primary enhancements in Version 5 (V5) are as follows:

- The least-squares parameter estimation with integrated first-order, second-moment (FOSM) parameter and forecast uncertainty estimation (PESTPP-GLM) extends the former Version 3 PEST++ executable to allow a user access to FOSM-based posterior Monte Carlo sampling and parallel ensemble evaluation, where these automated uncertainty analyses are available for each iteration (base and super-parameter) and at completion of parameter adjustment iterations;

- An iterative, localized ensemble smoother (PESTPP-IES) combines the efficiency of the GLM algorithm to minimize a least-squares objective function in high-dimensions with Monte Carlo, which is used to empirically estimate the Jacobian matrix;

- A tool for management optimization under uncertainty (PESTPP-OPT) uses sequential linear programming to facilitate management optimization of high-dimensional decision variable spaces as well as to support multiple forms of chance constraints; and

- Additional capabilities added to V5 include other user-friendly options such as extensive use of default variables, optional easy-to-read keyword input format, and optional longer character limits for parameter and observation names.

Similar to Version 3, all PEST++ V5 code necessary to produce a statically linked PEST++ executable has been consolidated into the Microsoft Visual Studio 2019 solution for Windows environments, as well as CMake support for the Windows, Linux, and Apple environments.

References Cited

Ahlfeld, D.P., and Mulligan, A.E., 2000, Optimal management of flow in groundwater systems—An introduction to combining simulation models and optimization methods: Academic Press, 185 p.

Anderson, J.L., 2007, Exploring the need for localization in ensemble data assimilation using a hierarchical ensemble filter: Physica D. Nonlinear Phenomena, v. 230, no. 1–2, p. 99–111.

Anderson, M.P., Woessner, W.W., and Hunt, R.J., 2015, Applied groundwater modeling—Simulation of flow and advective transport 2nd ed: Academic Press, 564 p.

Bayer, P., Bürger, C.M., and Finkel, M., 2008, Computationally efficient stochastic optimization using multiple realizations: Advances in Water Resources, v. 31, no. 2, p. 399–417.

Chen, Y., and Oliver, D.S., 2013, Levenberg–Marquardt forms of the iterative ensemble smoother for efficient history matching and uncertainty quantification: Computational Geosciences, v. 17, no. 4, p. 689–703.

Chen, Y., and Oliver, D.S., 2017, Localization and regularization for iterative ensemble smoothers: Computational Geosciences, v. 21, no. 1, p. 13–30.

Dantzig, G.B., Orden, A., and Wolfe, P., 1955, The generalized simplex method for minimizing a linear form under linear inequality restraints: Pacific Journal of Mathematics, v. 5, no. 2, p. 183–195.

Doherty, J., 2003, Ground water model calibration using pilot points and regularization: Groundwater, v. 41, no. 2, p. 170–177.

Doherty, J.E., 2015a, PEST and its utility support software, theory: Brisbane, Australia, Watermark Numerical Computing, 353 p.

Doherty, J.E., 2015b, Calibration and uncertainty analysis for complex environmental models: Brisbane, Australia, Watermark Numerical Computing, 236 p.

Doherty, J.E., and Hunt, R.J., 2010, Approaches to highly parameterized inversion—A guide to using PEST for groundwater-model calibration: U.S. Geological Survey Scientific Investigations Report 2010–5169, 59 p., https://doi.org/10.3133/sir20105169.

Doherty, J.E., Fienen, M.N., and Hunt, R.J., 2010a, Approaches to highly parameterized inversion—Pilot-point theory, guidelines, and research directions: U.S. Geological Survey Scientific Investigations Report 2010–5168, 36 p.

Doherty, J., Hunt, R.J., and Tonkin, M.J., 2010b, Approaches to Highly Parameterized Inversion—A Guide to Using PEST for Model-Parameter and Predictive-Uncertainty Analysis: USGS Scientific Investigations Report 2010–5211, 71 p., https://doi.org/10.3133/sir20105211.

Doherty, J., and Moore, C., 2019, Decision support modeling—Data assimilation, uncertainty quantification, and strategic abstraction: Groundwater, v. 58, no. 3, p. 327–337.

Doherty, J., and Welter, D., 2010, A short exploration of structural noise: Water Resources Research, v. 46, no. 5, W05525.

Fienen, M.N., Doherty, J.E., Hunt, R.J., and Reeves, H.W., 2010, Using prediction uncertainty analysis to design hydrologic monitoring networks—Example applications from the Great Lakes water availability pilot project: U.S. Geological Survey Scientific Investigations Report 2010–5159, 44 p., https://doi.org/10.3133/sir20105159.

Freeze, R.A., Massmann, J., Smith, L., Sperling, T., and James, B., 1990, Hydrogeological decision analysis—1. A framework: Groundwater, v. 28, no. 5, p. 738–766.

Freyberg, D.L., 1988, An exercise in ground-water model calibration and prediction: Groundwater, v. 26, no. 3, p. 350–360.

Golub, G.H., and Van Loan, C.F., 2012, Matrix computations v. 3: Baltimore, MD, JHU press, 780 p.

Halko, N., Martinsson, P.G., and Tropp, J.A., 2011, Finding structure with randomness—Probabilistic algorithms for constructing approximate matrix decompositions: SIAM Review, v. 53, no. 2, p. 217–288.

Hanke, M., 1997, A regularizing Levenberg-Marquardt scheme, with applications to inverse groundwater filtration problems: Inverse Problems, v. 13, no. 1, p. 79–95.

Hunt, R.J., Feinstein, D.T., Pint, C.D., and Anderson, M.P., 2006, The importance of diverse data types to calibrate a watershed model of the Trout Lake Basin, northern Wisconsin: Journal of Hydrology (Amsterdam), v. 321, no. 1–4, p. 286–296.

Hunt, R.J., Doherty, J., and Tonkin, M.J., 2007, Are models too simple? Arguments for increased parameterization: Groundwater, v. 45, no. 3, p. 254–262.

Hunt, R.J., Fienen, M.N., and White, J.T., 2019, Revisiting “An exercise in groundwater model calibration and prediction” after 30 years—Insights and new directions: Groundwater, v. 58, no. 2, p. 168–182.

Langevin, C.D., Hughes, J.D., Banta, E.R., Niswonger, R.G., Panday, S., and Provost, A.M., 2017, Documentation for the MODFLOW 6 Groundwater Flow Model: U.S. Geological Survey Techniques and Methods, book 6, chap. A55, 197 p.

Luo, X., and Bhakta, T., 2020, Automatic and adaptive localization for ensemble-based history matching: Journal of Petroleum Science Engineering, v. 184, p. 106559.

Moore, C., and Doherty, J., 2005, Role of the calibration process in reducing model predictive error: Water Resources Research, v. 41, no. 5, p.1–14.

Moré, J.J., 1978, The Levenberg-Marquardt algorithm—Implementation and theory, *in* Watson, G.A., ed., Numerical analysis: Berlin, Heidelberg, Springer, p. 105–116.

Morris, M.D., 1991, Factorial sampling plans for preliminary computational experiments: Technometrics, v. 33, no. 2, p. 161–174.

Nocedal, J., and Wright, S., 2006, Numerical optimization: New York, NY, Springer Science & Business Media, 653 p.

Oliver, D.S., Reynolds, A.C., and Liu, N., 2008, Inverse theory for petroleum reservoir characterization and history matching: Cambridge, United Kingdom, Cambridge University Press, 380 p.

Saltelli, A., Ratto, M., Andres, T., Campolongo, F., Cariboni, J., Gatelli, D., Saisana, M., and Tarantola, S., 2008, Global sensitivity analysis—The primer: Hoboken, New Jersey, John Wiley & Sons, 219 p.

Schreüder, W., 2010. BeoPEST Programmer’s Documentation. Principia Mathematica. Document can be found via the following website address: http://www.prinmath.com.

Siade, A.J., Rathi, B., Prommer, H., Welter, D., and Doherty, J., 2019, Using heuristic multi-objective optimization for quantifying predictive uncertainty associated with groundwater flow and reactive transport models: Journal of Hydrology (Amsterdam), v. 577, p. 123999.

Shmueli, G., 2010, To explain or to predict?: Statistical Science, v. 25, no. 3, p. 289–310.

Sobol, I.M., 2001, Global sensitivity indices for nonlinear mathematical models and their Monte Carlo estimates: Mathematics and Computers in Simulation, v. 55, no. 1–3, p. 271–280.

Sreekanth, J., Moore, C., and Wolf, L., 2016, Pareto-based efficient stochastic simulation–optimization for robust and reliable groundwater management: Journal of Hydrology (Amsterdam), v. 533, p. 180–190.

Tarantola, A., 2005, Inverse problem theory and methods for model parameter estimation v. 89: Philadelphia, PA, SIAM Publishing, 354 p.

Tonkin, M.J., and Doherty, J., 2005, A hybrid regularized inversion methodology for highly parameterized environmental models: Water Resources Research, v. 41, no. 10, p. 1–16.

Wagner, B.J., and Gorelick, S.M., 1987, Optimal groundwater quality management under parameter uncertainty: Water Resources Research, v. 23, no. 7, p. 1162–1174.

Welter, D.E., Doherty, J.E., Hunt, R.J., Muffels, C.T., Tonkin, M.J., and Schreüder, W.A., 2012, Approaches in highly parameterized inversion—PEST++, a Parameter ESTimation code optimized for large environmental models: U.S. Geological Survey Techniques and Methods, book 7, section C5, 47 p.

Welter, D.E., White, J.T., Hunt, R.J., and Doherty, J.E., 2015, Approaches in highly parameterized inversion: PEST++ Version 3, a Parameter ESTimation and uncertainty analysis software suite optimized for large environmental models: U.S. Geological Survey Techniques and Methods, book 7, section C12, 54 p.

Welter, D.E., Doherty, J.E., and White, J.T., 2018, PANTHER parallel model run manager: Brisbane, Australia, Watermark Numerical Computing, 66 p.

White, J.T., 2018, A model-independent iterative ensemble smoother for efficient history-matching and uncertainty quantification in very high dimensions: Environmental Modelling & Software, v. 109, p. 191–201.

White, J.T., Fienen, M.N., and Doherty, J.E., 2016, A python framework for environmental model uncertainty analysis: Environmental Modelling & Software, v. 85, p. 217–228.

White, J.T., Fienen, M.N., Barlow, P.M., and Welter, D.E., 2018, A tool for efficient, model-independent management optimization under uncertainty: Environmental Modelling & Software, v. 100, p. 213–221.

White, J.T., Knowling, M.J., Fienen, M.N., Feinstein, D.T., McDonald, G.W., and Moore, C.R., 2020, A non-intrusive approach for efficient stochastic emulation and optimization of model-based nitrate-loading management decision support: Environmental Modelling & Software, v. 126, p. 104657.

Appendix 1. PEST++ Version 5 Input Instructions

For ease of reference, the required variables within the standard PEST control file are listed below, and of these, the variables used by PEST++ are shaded. PEST++ relies on the structure of the control file (Doherty, 2015) to read the necessary algorithmic parameters and reads only those algorithmic parameters that are needed; for example, there is no need to read the NOBS variable because each line in the "observation data" section of the control file specifies an observation; however, it is necessary to read the NPAR variable to know where specification of parameters ends and information on tied parameters begins. This list is followed by short explanation of each variable used by PEST++.

```
{PRIVATE} pcf
* control data
RSTFLE PESTMODE
NPAR NOBS NPARGP NPRIOR NOBSGP
NTPLFLE NINSFLE PRECIS DPOINT
RLAMBDA1 RLAMFAC PHIRATSUF PHIREDLAM NUMLAM
RELPARMAX FACPARMAX FACORIG
PHIREDSWH
NOPTMAX PHIREDSTP NPHISTP NPHINORED RELPARSTP NRELPAR
ICOV ICOR IEIG
* singular value decomposition
SVDMODE
MAXSING EIGTHRESH
EIGWRITE
* parameter groups
PARGP MEINCTYP DERINC DERINCLB FORCEN DERINCMUL DERMTHD
(one such line for each of NPARGP parameter groups)
* parameter data
PARNME PARTRANS PARCHGLIM PARVAL1 PARLBND PARUBND PARGPSCALE OFFSET DERCOM
(one such line for each of NPAR parameters)
PARNME PARTIED
(one such line for each tied parameter)
* observation groups
OBGNME
(one such line for each of NOBSGP observation group)
* observation data
OBSNME OBSVAL WEIGHT OBGNME
(one such line for each of NOBS observations)
* model command line
COMLINE
(one such line for each of NUMCOM command lines)
* model input/output
TEMPFLE INFLE
(one such line for each of NTPLFLE template files)
INSFLE OUTFLE
(one such line for each of NINSLFE instruction files)
* prior information
```

```
PILBL PIFAC * PARNME + PIFAC * log(PARNME) ... = PIVAL WEIGHT OBGNME
(one such line for each of NPRIOR articles of prior information)
* regularisation
PHIMLIM  PHIMACCEPT [FRACPHIM]
WFINIT  WFMIN  WFMAX
WFFAC  WFTOL IREGADJ

++# This line is a comment as are all lines that begin with “++#”
++# PEST++ input is parsed using key words that can be specified in any order
++# these are the PEST++ options used in the example problem
++additional_ins_delim(,)
++tie_by_group(true)
++n_iter_base(-1)
++n_iter_super(3)
++glm_num_reals(200)
++glm_normal_form(prior)
++parcov(glm_prior.cov)
++max_n_super(50)
++ies_par_en(ies_prior_en.jcb)
++ies_no_noise(True)
++ies_localizer(temporal_loc.jcb)
++ies_autoadaloc(true)
++opt_dec_var_groups(welflux)
++opt_direction(max)
++opt_par_stack(par_stack.csv)
++opt_risk(0.95)
```

Variables in "control data" section of control file.

Variable	Type	Values	Description
PESTMODE	Text	"Estimation," "prediction," "regularisation," "pareto"	PEST's mode of operation. Used by PESTPP-GLM.
NPAR	Integer	Greater than 0	Number of parameters.
NUMCOM	Integer	Optional; greater than zero	Number of command lines used to run model.
RELPARMAX	Real	Greater than 0	Parameter relative change limit. Used by PESTPP-GLM.
FACPARMAX	Real	Greater than 1	Parameter factor change limit. Used by PESTPP-GLM.
FACORIG	Real	Between 0 and 1	Minimum fraction of original parameter value in evaluating relative change. Used by PESTPP-GLM.
PHIREDSWH	Real	Between 0 and 1	Sets objective function change for introduction of central derivatives. Used by PESTPP-GLM.
NOPTMAX	Integer	−2, −1, 0, or any number greater than 0	Number of optimization iterations. Used by PESTPP-GLM, PESTPP-IES and PESTPP-OPT.
PHIREDSTP	Real	Greater than 0	Relative objective function reduction triggering termination. Used by PESTPP-GLM and PESTPP-IES.
NPHISTP	Integer	Greater than 0	Number of successive iterations over which PHIREDSTP applies. Used by PESTPP-GLM and PESTPP-IES.
NPHINORED	Integer	Greater than 0	Number of iterations since last drop in objective function to trigger termination. Used by PESTPP-GLM and PESTPP-IES.
RELPARSTP	Real	Greater than 0	Maximum relative parameter change triggering termination. Used by PESTPP-GLM.
NRELPAR	Integer	Greater than 0	Number of successive iterations over which RELPARSTP applies. Used by PESTPP-GLM.

Variables in optional "singular value decomposition" section of control file.

Variable	Type	Values	Description
MAXSING	Integer	Greater than 0	Number of singular values at which truncation occurs. Used by PESTPP-GLM and PESTPP-IES.
EIGTHRESH	Real	0 or greater, but less than 1	Eigenvalue ratio threshold for truncation. Used by PESTPP-GLM and PESTPP-IES.

Variables required for each parameter group in "parameter groups" section of control file.

Variable	Type	Values	Description
PARGP	Text	12 characters or less	Parameter group name. Used by PESTPP-GLM and PESTPP-OPT.
INCTYP	Text	"Relative," "absolute," "rel_to_max"	Method by which parameter increments are calculated. Used by PESTPP-GLM and PESTPP-OPT.
DERINC	Real	Greater than 0	Absolute or relative parameter increment. Used by PESTPP-GLM and PESTPP-OPT.
DERINCLB	Real	0 or greater	Absolute lower bound of relative parameter increment. Used by PESTPP-GLM and PESTPP-OPT.
FORCEN	Text	"Switch," "always_2," "always_3," "switch_5," "always_5"	Determines whether central derivatives calculation is undertaken and whether three points or four points are employed in central derivatives calculation. Used by PESTPP-GLM.
DERINCMUL	Real	Greater than 0	Derivative increment multiplier when undertaking central derivatives calculation. Used by PESTPP-GLM.
DERMTHD	Text	"Parabolic," "outside_pts," "best_fit," "minvar," "maxprec"	Method of central derivatives calculation. PEST++ Version 3 only supports "parabolic." Used by PESTPP-GLM.

Variables required for each parameter in "parameter data" section of control file.

Variable	Type	Values	Description
PARNME	Text	12 characters or less	Parameter name. Used by PEST++ V5 suite.
PARTRANS	Text	"Log," "none," "fixed," "tied"	Parameter transformation. Used by PEST++ V5 suite.
PARCHGLIM	Text	"Relative," "factor," or "absolute(n)"	Type of parameter change limit. Used by PESTPP-GLM.
PARVAL1	Real	Any real number	Initial parameter value. Used by PEST++ V5 suite.
PARLBND	Real	Less than or equal to PARVAL1	Parameter lower bound. Used by PEST++ V5 suite.
PARUBND	Real	Greater than or equal to PARVAL1	Parameter upper bound. Used by PEST++ V5 suite.
PARGP	Text	12 characters or less	Parameter group name. Used by PEST++ V5 suite.
SCALE	Real	Any number other than 0	Multiplication factor for parameter. Used by PEST++ V5 suite.
OFFSET	Real	Any number	Number to add to parameter. Used by PEST++ V5 suite.
PARTIED	Text	12 characters or less	The name of the parameter to which another parameter is tied. Used by PEST++ V5 suite.

Variables required for each observation group in "observation groups" section of control file.

Variable	Type	Values	Description
OBGNME	Text	12 characters or less	Observation group name.

Variables required for each observation in “observation data” section of control file.

Variable	Type	Values	Description
OBSNME	Text	20 characters or less	Observation name. Used by PEST++ V5 suite.
OBSVAL	Real	Any number	Measured value of observation. Used by PEST++ V5 suite.
WEIGHT	Real	0 or greater	Observation weight. Used by PEST++ V5 suite.
OBGNME	Text	12 characters or less	Observation group to which observation assigned. Used by PEST++ V5 suite.

Variables in “model command line” section of control file.

Variable	Type	Values	Description
COMLINE	Text	System command	Command to run model. Used by PEST++ Version 5 suite.

Variables in “model input/output” section of control file.

Variable	Type	Values	Description
TEMPFLE	Text	A filename	Template file. Used by PEST++ V5 suite.
INFLE	Text	A filename	Model input file. Used by PEST++ V5 suite.
INSFLE	Text	A filename	Instruction file. Used by PEST++ V5 suite.
OUTFLE	Text	A filename	Model output file. Used by PEST++ V5 suite.

Variables in “prior information” section of control file.

Variable	Type	Values	Description
PILBL	Text	20 characters or less	Name of prior information equation. Used by PESTPP-GLM and PESTPP-OPT.
PIFAC	Text	Real number other than 0	Parameter value factor. Used by PESTPP-GLM and PESTPP-OPT.
PARNME	Text	12 characters or less	Parameter name. Used by PESTPP-GLM and PESTPP-OPT.
PIVAL	Real	Any number	Observed value of prior information. Used by PESTPP-GLM and PESTPP-OPT.
WEIGHT	Real	0 or greater	Prior information weight. Used by PESTPP-GLM and PESTPP-OPT.
OBGNME	Text	12 characters or less	Observation group name. Used by PESTPP-GLM and PESTPP-OPT.

Variables in optional "regularization" section of control file.

Variable	Type	Values	Description
PHIMLIM	Real	Greater than 0	Target measurement objective function. Used by PESTPP-GLM.
PHIMACCEPT	Real	Greater than PHIMLIM	Acceptable measurement objective function. Used by PESTPP-GLM.
FRACPHIM	Real	Optional; 0 or greater but less than 1	Set target measurement objective function at this fraction of current measurement objective function. Used by PESTPP-GLM.
WFINIT	Real	Greater than 0	Initial regularization weight factor. Used by PESTPP-GLM.
WFMIN	Real	Greater than 0	Minimum regularization weight factor. Used by PESTPP-GLM.
WFMAX	Real	Greater than WFMAX	Maximum regularization weight factor. Used by PESTPP-GLM.
WFFAC	Real	Greater than 1	Regularization weight factor adjustment factor. Used by PESTPP-GLM.
WFTOL	Real	Greater than 0	Convergence criterion for regularization weight factor. Used by PESTPP-GLM.
IREGADJ	Integer	0, 1, 2, 3, 4, or 5	Instructs PEST to perform interregularization group weight factor adjustment or to compute new relative weights for regularization observations and prior information equations. Used by PESTPP-GLM.

PEST++ Additions to the Standard Control File

Information in the control file specific to PEST++ is marked by lines starting with "++." Although the examples provided in this report place all PEST++ input in a single section at the end of the control file, this is not a requirement. This information does not need to be contiguous and can reside anywhere in the file. Lines starting with "++#" are considered comments and are ignored by PEST and PEST++.

Unlike the rest of the control file, PEST++ uses keywords rather than location to specify variables. Lines are parsed using the space, tab, and parenthesis characters as separators. Parentheses are used to more clearly delineate the values assigned to the variable (for example, ++N_ITER_BASE(1) specifies N_ITER_BASE=1).

PANTHER run manager options

Variable	Type	Role
MAX_RUN_FAIL(3)	Integer	Number of tolerated run failures for a given run before it is marked as a failure.
YAMR_POLL_INTERVAL(1)	Integer	Number of seconds between attempts of a PANTHER agent to connect to the master.
OVERDUE_RESCHED_FAC(1.15)	Real	A run with a run time that is greater than *overdue_resched_fac* times average run time is rescheduled on a different agent if agents are available and no more than three other agents are already making the same run.
OVERDUE_GIVEUP_FAC(100)	Real	A run with a run time that is greater than *overdue_giveup_fac* times average run time is marked as a run failure and terminated.
OVERDUE_GIVEUP_MINUTES(1E+30)	Real	A run with a run time that is greater than *overdue_giveup_minutes* is marked as a run failure and terminated.
PANTHER_AGENT_RESTART_ON_ERROR(FALSE)	Boolean	Flag to automatically restart a PANTHER agent if it exits on an error.
PANTHER_AGENT_NO_PING_TIMEOUT_SECONDS(300)	Integer	If a PANTHER agent cannot communicate with the master at least once every *panther_agent_no_ping_timeout_seconds*, it will exit.

PEST++ options used by all tools

Variable	Type	Role
TIE_BY_GROUP(FALSE)	Boolean	Flag to tie all adjustable parameters by parameter group name
ADDITIONAL_INS_DELIMITERS	Text	Characters to treat as whitespace when parsing model output files with instruction files.
DEBUG_PARSE_ONLY(FALSE)	Boolean	Flag to only parse the control file and exit
CHECK_TPL_INS(TRUE)	Boolean	Flag to check template and instruction file contents against the quantities listed in the control file.
ENFORCE_TIED_BOUNDS(FALSE)	Boolean	Flag to limit adjustable parameters based on the bounds of any parameters that are tied to it.
FILL_TPL_ZEROS	Boolean	Flag to fill template file parameter marker entries with leading zeros instead of leading whitespace.

PESTPP-GLM options

Variable	Type	Role
MAX_N_SUPER()	Integer	The maximum number of superparameters to use when conducting SVD-assisted inversion. The default is the number of adjustable parameters, in which case the number of superparameters is effectively set by *super_eigthresh()*.
SUPER_EIGTHRESH(1.0E-8)	Real	The ratio to maximum singular value of $\mathbf{J}^t\mathbf{QJ}$ at which truncation takes place to form superparameters. Note, however, that if the number of superparameters calculated in this way exceeds *max_n_super()* then the value of the latter variable takes precedence.
N_ITER_BASE(1)	Integer	Where superparameters are estimated in some iterations and base parameters are estimated in other iterations, this variable sets the number of sequential base parameter iterations to undertake before commencing an iteration in which superparameters are adjusted. If *n_iter_base()* is set to −1, this instructs PESTPP-GLM to emulate PEST behavior; a base parameter Jacobian matrix is calculated; then superparameters are estimated as soon as they are defined based on this matrix. Super parameters are estimated in all succeeding iterations.
N_ITER_SUPER(4)	Integer	Where superparameters are estimated in some iterations and base parameters are estimated in other iterations, this variable sets the number of sequential superparameter iterations to undertake before commencing an iteration in which a base parameter Jacobian matrix is recalculated and base parameters are adjusted.
JAC_SCALE(TRUE)	Boolean	Scale parameters by their sensitivities when calculating parameter upgrades. This can increase numerical precision; however, it may incur a numerical cost.
SVD_PACK(REDSVD)	Text	This informs PESTPP-GLM of the package that it must employ to undertake singular value decomposition of the $\mathbf{J}^t\mathbf{QJ}$ matrix (appropriately modified to include the Marquardt lambda and regularization). Options are “eigen” and “redsvd.”
LAMBDAS(0.1,1,10,100,1000)	Real numbers	Values for the Marquardt lambda used in calculation of parameter upgrades. Note that this base list is augmented with values bracketing the previous iteration’s best lambda. However, if a single value is specified, only that lambda (and no other lambda) is used in all iterations.
LAMBDA_SCALE_FAC(.75,1.0,1.1)	Real numbers	These values are used to scale each parameter upgrade vector calculated using different values of lambda. This results in a line search along each upgrade vector direction. The number of tested parameter upgrades (and hence model runs) is equal to the number of lambdas times the number of scaling factors. Set *lambda_scale_fac()* to 1.0 to disable an upgrade direction line search.
BASE_JACOBIAN()	Text	Provide the name of a JCO file. The Jacobian matrix contained in this file will be used for the first iteration of the inversion process.
HOTSTART_RESFILE()	Text	Specify the name of a residuals file from a previous PESTPP-GLM run. PESTPP-GLM will assume that these are model outputs corresponding to initial parameter values. It will use these instead of undertaking the initial model run.
UNCERTAINTY(TRUE)	Boolean	Flag to activate or deactivate first-order, second-moment (FOSM)-based parameter and (optionally) forecast uncertainty estimation.
PARCOV()	Text	Provide the name of a JCO, JCB, UNC or COV file from which the prior parameter covariance matrix used in FOSM analysis is read.
PAR_SIGMA_RANGE(4.0)	Real	The difference between a parameter’s upper and lower bounds expressed as standard deviations.

PESTPP-GLM options—Continued

Variable	Type	Role
FORECASTS(..,..,..)	Text	Provide the names of one or more observations featured in the "observation data" section of the control file; these are treated as predictions in FOSM predictive uncertainty analysis. If not provided and *uncertainty(true)*, then all zero-weighted observations are treated as forecasts.
GLM_NUM_REALS(100)	Integer	Number of parameter realizations to draw from the posterior parameter distribution (using final, estimated parameter values as the parameter mean vector, and the FOSM-based posterior covariance matrix). After generation of the realizations, the model is run once for each realization. The resulting observation ensemble is saved in a CSV file named *case.obs.csv*; the parameter ensemble is saved in a CSV file named *case.par.csv.*
SAVE_BINARY(FALSE)	Boolean	A flag to save parameter and observation ensembles in binary format. If this is set to *true*, parameter and observation ensembles are saved in files named *case.par.jcb* and *case.obs.jcb*.
TIE_BY_GROUP(FALSE)	Boolean	Flag to tie all adjustable parameters by group designation; however, all user-supplied parameter tied-parent relationships are preserved. The effective number of adjustable parameters thus becomes the number of parameter groups (which contain at least one adjustable parameter) plus the number of parameters that are listed as having others tied to it.
ITERATION_SUMMARY(TRUE)	Boolean	This flag activates or deactivates the writing of CSV files summarizing parameters (*case.ipar*), objective functions (*case.iobj*), sensitivities (*case.isen*), trial parameter upgrades (*case.upg.csv*), and parameter-to-run-id mapping (*case.rid*).
DER_FORGIVE(TRUE)	Boolean	If set to "true," then if model run failure occurs when calculating finite-difference derivatives with respect to a certain parameter, that parameter is frozen at its current value for the remainder of the iteration. If set to "false," PESTPP-GLM terminates execution with an appropriate message if a model run fails during calculation of a derivative.
ENFORCE_TIED_BOUNDS(FALSE)	Boolean	Flag to enforce parameter bounds on any tied parameters.
GLM_ACCEPT_MC_PHI(FALSE)	Boolean	Flag to accept FOSM-based realization phi each base iteration if the phi is lower than the lambda-testing phi. Default is false.
RAND_SEED(358183147)	Unsigned integer	Seed for the random number generator. Used for FOSM-based Monte Carlo.
GLM_REBASE_SUPER	Boolean	A flag to run the superparameter truncated values once at the start of the first superparameter iteration to provide a more accurate base run for calculating sensitivity numerators. Only applies if *n_iter_base* = -1 and *base_jacobian* is supplied. Default is "false," which indicates use either the *hotstart_resfile* residuals or use the base run previously completed.
GLM_ITER_MC(FALSE)	Boolean	Flag to undertake FOSM-based posterior Monte Carlo during each iteration of PESTPP-GLM. Default is "false," which will result in Monte Carlo only after iterations are done (depending on the *glm_num_reals* and *uncertainty* flags).

PESTPP-SEN options

Control variable	Type	Role
GSA_METHOD(MORRIS)	Text	Methods are "morris" and "sobol."
GSA_MORRIS_R(4)	Integer	Sample size. The number of times that an elementary effect is computed for each parameter; that is, the number of sequences of *m* model runs undertaken by PESTPP-SEN, where *m* is the number of adjustable parameters featured in the control file.
GSA_MORRIS_P(4)	Integer	The number of levels employed to grid the interval [0,1] associated with each transformed parameter. The number of intervals into which [0, 1] is therefore subdivided is $p-1$.
GSA_MORRIS_DELTA()	Real	The default value for *morris_delta()* is $p/2[(p-1)]$. The value supplied for this variable must be a multiple of $1/2[(p-1)]$. No check is made to ensure this is the case so users must take care if specifying this argument.
GSA_MORRIS_OBS_SEN(TRUE)	Boolean	If supplied as "false", PESTPP-SEN computes parameter sensitivities for the objective function only. If supplied as "true," PESTPP-SEN computes parameter sensitivities for the objective function, as well as for each model output corresponding to observations featured in the "observation data" section of the control file.
TIE_BY_GROUP(FALSE)	Boolean	Flag to tie all adjustable parameters together within each parameter group. Initial parameter ratios are maintained as parameters are adjusted. Parameters that are designated as already tied, or that have parameters tied to them, are not affected.
ENFORCE_TIED_BOUNDS(FALSE)	Boolean	Flag to enforce parameter bounds on any tied parameters.
GSA_SOBOL_SAMPLES(4)	Integer	The number of samples to use in computing variances. The number of model runs is actually twice this number because of the need to employ two series of parameter samples. See Saltelli and others (2008) for details.
GSA_SOBOL_PAR_DIST(NORM)	Text	Specifies whether parameter samples should be drawn from a uniform or normal distribution. Values are "unif" or "norm" respectively. In the latter case, samples are centered on parameter values provided in the control file, while the standard deviation is a quarter of the difference between a parameter's upper and lower bounds. Log-uniform and log-normal distributions are employed for parameters which are denoted as log-transformed in the control file.

PESTPP-OPT options

Variable	Type	Role
OPT_DEC_VAR_GROUPS()	Text	Comma-delimited string identifying which parameter groups are to be treated as decision variables. If not supplied, all adjustable parameters are treated as decision variables.
OPT_EXTERNAL_DEC_VAR_GROUPS()	Text	Comma-delimited string identifying which parameter groups are to be treated as external decision variables, that is decision variables that do not affect model outputs and that therefore do not require a finite-difference run of the model to fill the pertinent column of the response matrix.
OPT_CONSTRAINT_GROUPS()	Text	Comma- delimited string identifying which observation and prior information groups are to be treated as constraints. Group names for "less than" constraints must start with "l_" or "less_"; group names for "greater than" constraints must start with "g_" or "greater_." If this control variable is omitted, all observation and prior information groups that meet these naming conventions are treated as constraints.
OPT_OBJ_FUNC()	Text	String identifying the prior information equation or two-column ASCII file that contains coefficients used in formulation of the objective. If this control variable is not supplied, then each decision variable is given a coefficient of 1.0 in formulation of the objective function.
OPT_DIRECTION(MIN)	Text	Either "min" or "max." "min" specifies that the objective function be minimized, while "max" specifies that it be maximized.
OPT_RISK(0.5)	Real	A number between 0.0 and 1.0. A value of 0.5 signifies risk neutrality. A value of 0.95 seeks a 95-percent risk averse application of optimization constraints, while a value of 0.05 seeks a 5-percent risk tolerant application of optimization constraints.
OPT_RECALC_CHANCE_EVERY(1)	Integer	Number of iterations of the SLP process over which chance constraints are reused. If set to 1, a calibration Jacobian matrix is calculated during every iteration of the SLP constrained optimization process if FOSM-based chance constraints are used or the stack is reevaluated if stack-based chance constraints are being used.
PARCOV()	Text	Provide the name of a JCO, JCB, UNC or COV file from which the prior covariance matrix used in FOSM analysis is read.
PAR_SIGMA_RANGE(4.0)	Real	The difference between a parameter's upper and lower bounds expressed as standard deviations.
OPT_ITER_TOLL(0.001)	Real	Solution closure criterion applied to objective function and decision variables.
BASE_JACOBIAN()	Text	Provide the name of a Jacobian matrix file (with extension .jco or .jcb). Sensitivities read from this file are used for the first iteration of the constrained optimization process.
HOTSTART_RESFILE()	Text	The name of a residuals file produced by PESTPP-GLM or PESTPP-OPT. PESTPP-OPT assumes that model output values contained in this file correspond to the values of parameters (including decision variables) listed in the control file. Hence, it does not carry out the initial model run.
OPT_COIN_LOG(1)	Integer	Level of verbosity of solution information recorded by optimization library functions.

PESTPP-OPT options—Continued

Variable	Type	Role
OPT_STD_WEIGHTS(FALSE)	Boolean	Flag that identifies constraint weights as standard deviations. If set to true, PESTPP-OPT skips FOSM-based constraint uncertainty calculation and uses observation weights directly as standard deviations in the calculation of risk. These standard deviations can be calculated externally via PREDUNC or pyEMU or can be derived empirically from an ensemble. Setting this flag to true will override all other chance constraint flags and options.
OPT_SKIP_FINAL(FALSE)	Boolean	Flag to skip the final model run.
TIE_BY_GROUP(FALSE)	Boolean	Flag to tie all adjustable parameters together within each parameter group. Initial parameter ratios are maintained as parameters are adjusted. Parameters that are designated as already tied, or that have parameters tied to them, are not affected.
ENFORCE_TIED_BOUNDS(FALSE)	Boolean	Flag to enforce parameter bounds on any tied parameters.
OPT_STACK_SIZE(0)	Integer	Number of realizations to use in the stack. If positive, stack-based chance constraints are used. If *opt_par_stack* is not supplied, *opt_stack_size* realizations are drawn from the Prior. If *opt_par_stack* is supplied and the stack in that file is larger than *opt_stack_size*, the stack is truncated to *opt_stack_size*.
OPT_PAR_STACK()	Text	File containing a parameter stack. The file extension is used to determining CSV for binary (JCB) format. The stack in this file must constrain all adjustable parameters.
OPT_OBS_STACK()	Text	File containing an observation stack. The file extension is used to determining CSV for binary (JCB) format. Supplying this file will forego evaluating the stack for the first iteration and possibly subsequent iterations depending on the value if *opt_recalc_chance_every.*

PESTPP-IES options

Variable	Type	Role
IES_NUM_REALS(50)	Integer	The number of realizations to draw in order to form parameter and observation ensembles.
PARCOV()	Text	The name of a file containing the prior parameter covariance matrix. This can be a parameter uncertainty file (extension *.unc*), a covariance matrix file (extension *.cov*) or a binary JCO or JCB file (extension *.jco* or *.jcb*).
PAR_SIGMA_RANGE(4.0)	Real	The difference between a parameter's upper and lower bounds expressed as standard deviations.
IES_PARAMETER_ENSEMBLE()	Text	The name of a CSV or JCO/JCB file (recognized by its extension) containing user-supplied parameter realizations comprising the initial (prior) parameter ensemble. If this keyword is omitted, PESTPP-IES generates the initial parameter ensemble itself.
IES_OBSERVATION_ENSEMBLE()	Text	The name of a CSV or JCO/JCB file (recognized by its extension) containing user-supplied observation realizations comprising the observation ensemble. If this keyword is omitted, PESTPP-IES generates the observation ensemble itself.
IES_ADD_BASE(TRUE)	Boolean	If set to "true," instructs PESTPP-IES to include a realization in the initial parameter ensemble comprised of parameter values read from the "parameter data" section of the control file. The corresponding observation ensemble is comprised of measurements read from the "observation data" section of the control file.
IES_RESTART_OBSERVATION_ENSEMBLE()	Text	The name of a CSV or JCO/JCB file (recognized by its extension) containing model outputs calculated using a parameter ensemble. If it reads this file, PESTPP-IES does not calculate these itself, proceeding to upgrade calculations instead.
IES_RESTART_PARAMETER_ENSEMBLE()	Text	The name of a CSV or JCO/JCB file (recognized by its extension) containing a parameter ensemble that corresponds to the *ies_restart_observation_ensemble()*. This option requires that the *ies_restart_observation_ensemble()* control variable also be supplied.
IES_ENFORCE_BOUNDS(TRUE)	Boolean	If set to "true" PESTPP-IES will not transgress bounds supplied in the control file when generating or accepting parameter realizations and (or) when adjusting these realizations.
IES_INITIAL_LAMBDA()	Real	The initial Marquardt lambda. The default value is <math xmlns="http://www.w3.org/1998/Math/MathML" xmlns:tps="http://www.typefi.com/ContentXML"
IES_LAMBDA_MULTS(0.1,1.0,10.0)	Comma-separated reals	Factors by which to multiply the best lambda from the previous iteration to yield values for testing parameter upgrades during the current iteration.
LAMBDA_SCALE_FAC(0.75,1.0,1.1)	Comma-separated reals	Line search factors along parameter upgrade directions computed using different Marquardt lambdas.
IES_SUBSET_SIZE(5)	Integer	Number of realizations used in testing and evaluation of different Marquardt lambdas.

PESTPP-IES options—Continued

Variable	Type	Role
IES_USE_APPROX(TRUE)	Boolean	Use complex or simple formula provided by Chen and Oliver (2013) for calculation of parameter upgrades. The more complex formula includes a function which constrains parameter realizations to respect prior means and probabilities.
IES_REG_FACTOR(0.0)	Real	Regularization objective function as a fraction of measurement objective function when constraining parameter realizations to respect initial values.
IES_BAD_PHI(1.0E300)	Real	If the objective function calculated as an outcome of a model run is greater than this value, the model run is deemed to have failed.
IES_BAD_PHI_SIGMA(1.0E300)	Real	If the objective function calculated for a given realization is greater than the current mean objective function of the ensemble plus the objective function standard deviation of the ensemble times *ies_bad_phi_sigma()*, that realization is treated as failed.
IES_USE_PRIOR_SCALING(FALSE)	Boolean	Use a scaling factor based on the prior parameter distribution when evaluating parameter-to-model-output covariance used in calculation of the randomized Jacobian matrix.
IES_USE_EMPIRICAL_PRIOR(FALSE)	Boolean	Use an empirical, diagonal parameter covariance matrix for certain calculations. This matrix is contained in a file whose name is provided with the *ies_parameter_ensemble()* keyword.
SAVE_LAMBDA_ENSEMBLES(FALSE)	Boolean	Save a set of CSV or JCB files that record parameter realizations used when testing different Marquardt lambdas.
IES_VERBOSE_LEVEL(1)	0, 1 or 2	The level of diagnostic output provided by PESTPP-IES. If set to 2, all intermediate matrices are saved to ASCII files. This can require a considerable amount of storage.
IES_ACCEPT_PHI_FAC(1.05)	Real > 1.0	The factor applied to the previous best mean objective function to determine if the current mean objective function is acceptable.
IES_LAMBDA_DEC_FAC(0.75)	Real < 1.0	The factor by which to decrease the value of the Marquardt lambda during the next IES iteration if the current iteration of the ensemble smoother process was successful in lowering the mean objective function.
IES_LAMBDA_INC_FAC(10.0)	Real > 1.0	The factor by which to increase the current value of the Marquardt lambda for further lambda testing if the current lambda testing cycle was unsuccessful.
IES_SUBSET_HOW(RANDOM)	“first,” “last,” “random,” “phi_based”	How to select the subset of realizations for objective function evaluation during upgrade testing. Default is “random.”

PESTPP-IES options—Continued

Variable	Type	Role
IES_NUM_THREADS(0)	Integer > 1	The number of threads to use during the localized upgrade solution process, the automatic adaptive localization process and the generation of the initial parameter ensemble (if the prior parameter covariance matrix is none diagonal). If the localizer contains many (greater than 10,000) rows, then multithreading can substantially speed up the upgrade calculation process. *ies_num_threads()* should not be greater than the number of physical cores on the host machine. Note also that if large numbers of parameters are in each group and the prior parameter covariance matrix is nondiagonal, users must take care not to saturate memory by using too many threads (this will manifest as a seg fault on Linux/mac and an out or memory error on windows).
IES_LOCALIZER()	Text	The name of a matrix to use for localization. The extension of the file is used to determine the type: *.mat* is an ASCII matrix file, *.jcb or .jco* signifies use of (enhanced) Jacobian matrix format (a binary format), and *.csv* signifies a comma-delimited file. Note that adjustable parameters not listed in localization matrix columns are implicitly treated as “fixed” while nonzero weighted observations not listed in rows of this matrix are implicitly treated as zero-weighted.
IES_GROUP_DRAWS(TRUE)	Boolean	A flag to draw from the (multivariate) Gaussian prior by parameter/observation groups.
IES_SAVE_BINARY(FALSE)	Boolean	A flag to save parameter and observation ensembles in binary (in other words, JCB) format instead of CSV format.
IES_CSV_BY_REALS(TRUE)	Boolean	A flag to save parameter and observation ensemble CSV files by realization instead of by variable name. If true, each row of the CSV file is a realization. If false, each column of the CSV file is a realization.
IES_AUTOADALOC(FALSE)	Boolean	Flag to activate automatic adaptive localization.
IES_AUTOADALOC_SIGMA_DIST(1.0)	Real	Real number representing the factor by which a correlation coefficient must exceed the standard deviation of background correlation coefficients to be considered significant. Default is 1.0.
TIE_BY_GROUP(FALSE)	Boolean	Flag to tie all adjustable parameters together within each parameter group. Initial parameter ratios are maintained as parameters are adjusted. Parameters that are designated as already tied, or that have parameters tied to them, are not affected.
IES_ENFORCE_CHGLIM(FALSE)	Boolean	Flag to enforce parameter bounds transgression and parameter change limits (via FACPARMAX and RELPARMAX) in a way similar to PEST and PESTPP-GLM (by scaling the entire realization). Default is “false.”
IES_CENTER_ON(BASE)	Text	A realization name that should be used for the ensemble center in calculating the approximate Jacobian matrix. The realization name must be in both the parameter and observation ensembles. If not passed, the mean vector is used as the center.

PESTPP-IES options—Continued

Variable	Type	Role
ENFORCE_TIED_BOUNDS(FALSE)	Boolean	Flag to enforce parameter bounds on any tied parameters.
IES_NO_NOISE(FALSE)	Boolean	Flag to not generate and use realizations of measurement noise. Default is “false” (that is, to use measurement noise).
IES_DROP_CONFLICTS(FALSE)	Boolean	Flag to remove nonzero weighted observations that are in a prior-data conflict state from the upgrade calculations. Default is “false.”
IES_PDC_SIGMA_DISTANCE()	Real	The number of standard deviations from the mean used in checking for prior-data conflict.
IES_SAVE_RESCOV(TRUE)	Boolean	Flag to save the iteration-level residual covariance matrix. If *ies_save_binary* is “true,” then a binary format file is written, otherwise an ASCII format (.cov) file is written. The file name is case.N.res.cov/.jcb.
OBSCOV()	Text	The name of a file containing the observation noise covariance matrix. This can be a parameter uncertainty file (extension *.unc*), a covariance matrix file (extension *.cov*) or a binary JCO or JCB file (extension *.jco* or *.jcb*).
RAND_SEED(358183147)	Unsigned integer	Seed for the random number generator.

Example of the new optional control file format:

```
# this is comment line - these lines can be used throughout
pcf
# control data keyword is a new section that combines all the algorithmic
# sections of the control file including control data, singular value
# decomposition and PEST++ "++" options
* control data keyword
noptmax 3
maxsing 10
additional_ins_delim,
# options used by PESTPP-SEN
tie_by_group true
# options used by PESTPP-GLM
n_iter_base -1
n_iter_super 3
glm_num_reals 200
glm_normal_form prior
parcov glm_prior.cov
max_n_super 50
# options used by PESTPP-IES
ies_par_en prior.jcb
ies_num_reals 50
ies_no_noise True
ies_localizer temporal_loc.jcb
ies_autoadaloc true
# options used by PESTPP-OPT
opt_dec_var_groups welflux
opt_direction max
opt_par_stack par_stack.csv
opt_risk 0.95
* parameter data external
# these external files must have the required entries from the * parameter
# data section of the standard control file but can also have additional
#columns
Hk_pars.csv
Recharge_pars.csv
Other_pars.csv
* observation data
Head_obs.csv
Flux_obs.csv
Additional_important_obs.csv
* model command line
mf6
** model input external
```

```
# this section contains the template files and corresponding model
# input files
Most_tpl_files.csv
A_few_more_tpl_files.csv
* model output external
# this section contains the instruction files and corresponding model
# output files
Ins_files.csv
```

References Cited

Chen, Y., and Oliver, D.S., 2013, Levenberg–Marquardt forms of the iterative ensemble smoother for efficient history matching and uncertainty quantification: Computational Geosciences, v. 17, no. 4, p. 689–703.

Doherty, J.E., 2015, PEST and its utility support software, theory: Brisbane, Australia, Watermark Numerical Computing, 353 p.

Saltelli, A., Ratto, M., Andres, T., Campolongo, F., Cariboni, J., Gatelli, D., Saisana, M., and Tarantola, S., 2008, Global sensitivity analysis—The primer: Hoboken, New Jersey, John Wiley & Sons, 219 p.

For additional information contact:

Director, USGS Upper Midwest Water Science Center
U.S. Geological Survey
8505 Research Way
Middleton, WI 53562

608–828–9901

For additional information, visit: https://www.usgs.gov/centers/umid-water

White and others—**Approaches to Highly Parameterized Inversion: PEST++ Version 5**—Techniques and Methods 7–C26

ISSN 2328-7055 (online)
https://doi.org/10.3133/tm7C26