International Environmental Modelling and Software Society (iEMSs)
2010 International Congress on Environmental Modelling and Software
Modelling for Environment's Sake, Fifth Biennial Meeting, Ottawa, Canada
David A. Swayne, Wanhong Yang, A. A. Voinov, A. Rizzoli, T. Filatova (Eds.)
http://www.iemss.org/iemss2010/index.php?n=Main.Proceedings

Calibrating Environmental Models using ParaMESH

Daniel Princz[1], L. Shawn Matott[2]
[1]*Undergraduate Research Assistant, Dept. of Civil and Environmental Engineering, University of Waterloo, dprincz@uwaterloo.ca*
[2]*Assistant Professor, Dept. of Civil and Environmental Engineering, University of Waterloo, lsmatott@uwaterloo.ca*

Abstract: ParaMESH (Parameter management for Models of Environmental Systems and Hydrology) is a graphical interface for advanced parameter management. Originally developed to be compatible with selected hydrologic and climatologic models, recent enhancements support any environmental model or evaluation tool that uses text-based input-output files. ParaMESH provides an intuitive graphical environment in which to configure arbitrary models. Additional features include parameter verification and execution management. To demonstrate ParaMESH as a flexible tool-independent graphical interface, a new extension was developed which links ParaMESH with OSTRICH (Optimization Software Tool for Research In Computational Heuristics), a model-independent optimization toolkit. The linkage provides a convenient user interface for configuring and executing a model calibration exercise. Using ParaMESH's graphical interface, users easily navigate through the various aspects of model calibration, including: selecting a calibration algorithm, configuring and defining calibration parameters, and importing observation data and identifying simulated equivalents. Once configured, ParaMESH runs the OSTRICH-based model calibration and monitors and displays calibration progress. When calibration is complete, ParaMESH extracts and displays several post-calibration statistics, including: parameter confidence intervals and measures of correlation and sensitivity.

Keywords*:* model calibration; parameter estimation; graphical user interface; model-independent software

1. INTRODUCTION

Physics-based environmental simulation models are being increasingly relied upon to support decision making and policy analysis at local, national and international scales. Such models are often *calibrated* in order to adapt them to a particular site or problem – a process by which uncertain model input parameters are adjusted in order to achieve an acceptable correspondence between model outputs (i.e. simulated equivalents) and corresponding observations and response data (Hill and Tiedeman [2007]). In addition to trial-and-error calibration, many tools for automatic calibration have been developed. Several of these calibration tools are *model-independent* in that they can be adapted to work with a variety of environmental models – examples include OSTRICH (Optimization Software Tool for Research In Computational Heuristics, Matott [2005]), UCODE (Poeter and Hill [1999]), PEST (Doherty [2004]), and UNCSIM (Reichert [2005]).

A common feature of currently available model-independent calibration tools is that using such tools requires reading through a (typically expansive) user-manual and following the instructions in the manual to create a set of text-based input files. After creating the necessary input files, the model-independent calibration tool can then be invoked. Once the calibration tool finishes the automatic calibration exercise, users can then examine one or more output files to evaluate calibration results and any other statistical and diagnostic information that may be tabulated by the calibration tool. Interpretation of calibration output generally requires consultation with the user-manual.

The flexibility of model-independent calibration tools is clearly an advantage over tools that are embedded within (and therefore limited to) specific models. However, from a novice user's perspective, these model-independent tools have several disadvantages. For example, expecting users to read through an extensive manual is unrealistic and users have come to expect much more friendly alternatives that incorporate usage information directly into the program and provide a convenient mechanism for users to learn about a given program feature in a "just-in-time" or "as-needed" type of format. The result is a user experience that is much more tailored to the needs of a given user and application.

Another disadvantage of currently available model-independent calibration tools is their requirement that users manually create text-based input configuration files. One limitation of this approach is that much of the file preparation is tedious data entry (e.g. entering hundreds, if not thousands, of observational data points). Furthermore, users are prone to creating input files that contain crucial typographical errors (e.g. a single misspelled keyword can result in unpredictable tool behaviour). There are also disadvantages to presenting calibration results to the user via test-based output files. For example, relative to picking through a text-based output file, graphical analysis of calibration results (e.g. plots of parameters with confidence bounds, and plots of residuals and model fit) can often provide a much faster and more reliable means of analysis (Tufte [2001]). Such graphical analyses can make an important result obvious (e.g. the presence of an outlier or a poorly constrained parameter estimate), whereas such result may be obscured if it is buried within the myriad of numbers that would be found in a lengthy text-based output file.

1.1 Research Objectives

Recognizing the advantages and limitations of existing model-independent calibration tools, the primary research objective was to link a representative calibration tool (i.e. OSTRICH, described below in Section 2.2) with a newly developed graphical interface tool (i.e. ParaMESH). As described below (in Section 2.1), ParaMESH is a model- and tool-independent code that was designed to provide a convenient graphical interface replacement for the usual text-based input-output approach of a given model or modelling tool. Linking ParaMESH with OSTRICH was therefore expected to yield an automatic calibration tool that is more user-friendly than the standard text-file approach described previously. To demonstrate the linked ParaMESH-OSTRICH tool, it was applied to a previously published case-study (Matott and Rabideau [2008]) involving an 8-parameter calibration of a subsurface reactive transport model.

1.2 Related Work

ParaMESH provides a unique graphical user interface that can be flexibly tailored to interoperate with arbitrary environmental models as well as arbitrary model evaluation tools (i.e. model calibration, and sensitivity and uncertainty analysis). A similar tool has been recently developed as part of the SADA (Spatial Analysis and Decision Assistance) software tool (Stewart and Purucker [2006]). Whereas SADA emphasizes the use of statistical and geostatistical models of observational data sets, ParaMESH is focused on the development and evaluation of physics-based models.

The ParaMESH tool is also complementary to several ongoing efforts encouraging interoperability among models and model evaluation tools. Examples include FRAMES (Framework for Risk Analysis in Multimedia Environmental Systems) (Babendreier and Castleton [2005]), JUPITER (Joint Universal Parameter IdenTification and Evaluation of Reliability) (Banta et al. [2008]), and the industry-standard PEST protocols (Doherty [2004, [2007] and Skahill et al. [2009]). ParaMESH can be tailored to these interoperability standards, thereby providing a convenient graphical front-end for the associated tools and models.

2. METHODS

The methods section is organized as follows: Section 2.1 describes the features and capabilities of ParaMESH, Section 2.2 describes the features and capabilities of OSTRICH, Section 2.3 describes the software that was developed for linking ParaMESH and OSTRICH, and Section 2.4 describes the calibration case study that was used to evaluate the new ParaMESH-OSTRICH linkage.

2.1 Features and Capabilities of ParaMESH

ParaMESH was developed at the University of Waterloo (Ontario, Canada) with support from the National Hydrology Research Centre of Environment Canada (www.ec.gc.ca/inre-nwri). ParaMESH provides users with a convenient interface to the parameters of console-oriented models and associated text-based input-output files. The graphical interface of ParaMESH is designed specifically with users in mind and provides a controlled and user-friendly environment via a series of forms (Princz [2008]). As forms are filled in, ParaMESH validates user inputs and prevents typographic error. For example, ParaMESH provides basic parameter verification to ensure the parameter values entered for a given model are validated (i.e. are actual numbers) and are formatted to the highest level of significant digits. ParaMESH also provides additional parameter verification by checking user-assigned values against parameter limits (e.g. established maximum and minimum values) and by checking that model-specific parameter relationships are maintained (e.g. ensuring that the denitrification rate is slower than the aerobic degradation rate). ParaMESH is extremely flexible and the look-and-feel of the graphical environment can be easily adjusted by the user to match both aesthetic tastes and the specific needs of a given model. Preferred settings can then be saved and applied to any number of working directories to provide a user-centric experience.

In addition to helping validate user-supplied parameter values, ParaMESH also serves as a parameter management tool in that it maintains a history of previously used parameter values and provides the capability to restore the values of a given parameter. Execution management is also provided, allowing users to call a user-defined modelling scheme. Example modelling schemes include: directly invoking a model executable; calling a batch file that contains the model executable and pre- and post-processors; and running the model via calls to a dynamically linked library (DLL). As detailed in Section 2.3, recent ParaMESH enhancements provide for the development of DLLs that support parameter calibration and other model evaluation tasks. Once written, these DLLs allow a given external model evaluation tool to take advantage of the many graphical interface and parameter management features of ParaMESH. For this study, the new ParaMESH enhancements were demonstrated by creating a DLL that links ParaMESH with the OSTRICH calibration tool.

2.2 Features and Capabilities of OSTRICH

OSTRICH (Matott [2005]) is a multi-algorithm and model-independent calibration and optimization tool. Previous applications of OSTRICH have included: optimization of pump-and-treat systems for containing plumes of subsurface contaminants (Matott et al. [2006b]); designing multi-layer sorptive barrier systems (Matott et al. [2006a]); and calibrating alternative models of subsurface nitrate transport (Matott and Rabideau [2008]). Core features of OSTRICH utilized for this research are: an extensive library of calibration algorithms; built-in support for the weighted-sum-of-squared residuals (WSSR) calibration objective function; and a suite of post-calibration diagnostics and statistics that assist with evaluating calibration results and parameter uncertainty.

2.3 Linking ParaMESH and OSTRICH

Recent enhancements to ParaMESH allow the software to support any environment (i.e. any model or model evaluation tool) that uses text-based input-output files. Furthermore, ParaMESH extensions can now be developed to link the functionality and features of ParaMESH with other external model-independent tools. The cornerstone of these extensions is the specification of a DLL for interfacing with tools for automatic calibration, simulation-optimization, sensitivity analysis, uncertainty analysis, and other model run-time tools (Princz [2009]). Features of this DLL specification are described in the following sub-sections.

2.3.1 DLL Specification for ParaMESH Extensions

The first step for developing a ParaMESH extension is to create a generic DLL using the 'Class Library' application type within the desired software development environment (e.g. Microsoft Visual Studio). Within this DLL structure, the 'ParaMESH' root namespace must be included and an 'Optimize' class must be defined. To help potential developers with the creation of a properly formatted ParaMESH extension, templates are under development that will provide a skeletal structure and definitions of these required components.

2.3.2 Required and Optional DLL Functions

All functions required by the DLL specification must be implemented as member functions of the 'Optimize' class. These member functions are formed into two groups: 'definition' functions and 'run-time' functions. Definition functions allow the external evaluation tool to inform ParaMESH about the level of support provided by the specific tool. For example, the `OnlyNumeric()` function defines the types of variables supported by the external tool and should return **true** if only numeric values are supported and **false** otherwise. Similar definition functions inform ParaMESH whether parameters are to be adjusted individually (e.g. as would be done in one-at-a-time sensitivity analysis) or simultaneously (e.g. as would be done during automatic calibration). The ParaMESH graphical interface and parameter management features will be adjusted to accommodate the level of support offered by the external tool. For example, ParaMESH will not allow users to configure non-numeric parameters if only numerical values can be calibrated.

The required run-time functions for the ParaMESH extension DLL are comprised of functions for setting the initial parameter values of the model (i.e. prior to invoking the external tool) and a function (i.e. `RunModel()`) that invokes the desired model evaluation tool (e.g. runs the desired calibration program). Return arguments for the `RunModel()` function are arrays containing any results (e.g. calibrated parameter values) and diagnostics (e.g. error messages) that the external tool would like to pass on to ParaMESH. Optional run-time functions can also be specified that allow ParaMESH to request context-sensitive help information.

Importantly, ParaMESH DLL extensions are not permitted to make call-back requests directly to ParaMESH. Instead, ParaMESH makes calls to the specified DLL functions and the external tool communicates with ParaMESH solely via the return arguments of these functions. This 'feed-forward' design allows extension developers to focus solely on implementing the DLL specification and ensures that the external tool and the ParaMESH code are maximally decoupled.

2.3.3 Implementation of the linked ParaMESH-OSTRICH Software

Following the DLL specifications outlined above, a ParaMESH extension was developed to link the ParaMESH and OSTRICH software. For brevity, this linked software will be referred to as ParaMESH-OSTRICH. In addition to the required definition and run-time functions, the optional context-sensitive help functions were also implemented. The definition functions were implemented to indicate OSTRICH support for only numerical parameters along with support for calibration of both individual and multiple parameters. Arrays returned by the `RunModel()` run-time function included: calibrated parameter values, 95% linear confidence intervals for each parameter, measures of composite scaled parameter sensitivity, and measures of parameter correlation. Finally, the modelling scheme invoked by the `RunModel()` function was configured to run both OSTRICH and a previously developed monitoring program (see Figure 4 in Section 3.2).

2.4 Case Study for Evaluating ParaMESH-OSTRICH

A previously-published model calibration exercise was used to test and evaluate the linked ParaMESH-OSTRICH code. As described by Matott and Rabideau [2008], the calibration problem was motivated by data collection and modelling activities at the Lizzie Research Station in North Carolina (Kraemer et al. [2003], Spruill et al. [2005], Tesoriero et al. [2005]). An active hog farm operates on the site and soil fertilization using swine waste has led to significant nitrate contamination in the underlying aquifer. Groundwater samples collected at the site suggest that various biodegradation processes, including denitrification, have resulted in the development of distinct redox zones. These findings motivated one-dimensional subsurface reactive transport modelling of the Lizzie site. The model employs multiple-Monod biokinetics for aerobic degradation and denitrification.

Calibration parameters for the model are summarized in Table 1 and these were calibrated using a hypothetical time-series of groundwater concentrations (see Matott and Rabideau [2008] for details). The hypothetical observations included a variety of chemical species synthetically sampled at various groundwater depths. For this study, the calibration performance of the dynamically-dimensioned search (DDS) algorithm was investigated using the new linked ParaMESH-OSTRICH software. DDS is a promising and efficient global optimizer that requires no algorithm parameter tuning.

Table 1: Calibration Parameters for the Subsurface Reactive Transport Modeling Case Study

Parameter [units]	Range[a]	Description
`VEL` [mm/d]	0.86 to 26	vertical seepage velocity
`DIS` [m]	0.05 to 10	longitudinal dispersivity
`AER_MAX`[b] [mg/L per d]	0.028 to 140	maximum aerobic degradation rate
`AER_TEA`[b] [mg/L]	0.0025 to 500	½ saturation constant for aerobic degradation electron acceptor (O_2)
`AER_SUB`[b] [mg/L]	0.0067 to 1300	½ saturation constant for aerobic degradation substrate (CH_2O)
`DEN_MAX`[c] [mg/L per d]	5.36×10^{-4} to 27	maximum denitrification rate
`DEN_TEA`[c] [mg/L]	0.0025 to 16000	½ saturation constant for denitrification electron acceptor (NO_3^-)
`DEN_SUB`[c] [mg/L]	0.013 to 2600	½ saturation constant for denitrification substrate (CH_2O)

[a] – lower and upper bounds for the calibration exercise.

[b] – Multiple-Monod kinetics for aerobic degradation

$$\frac{d[\mathrm{O_2}]}{dt} = (\mathrm{AER\_MAX})\left(\frac{[\mathrm{O_2}]}{(\mathrm{AER\_TEA})+[\mathrm{O_2}]}\right)\left(\frac{[\mathrm{CH_2O}]}{(\mathrm{AER\_SUB})+[\mathrm{CH_2O}]}\right)$$

[c] – Multiple-Monod kinetics for denitrification

$$\frac{d[\mathrm{NO_3^-}]}{dt} = (\mathrm{DEN\_MAX})\left(\frac{[\mathrm{NO_3^-}]}{(\mathrm{DEN\_TEA})+[\mathrm{NO_3^-}]}\right)\left(\frac{[\mathrm{CH_2O}]}{(\mathrm{DEN\_SUB})+[\mathrm{CH_2O}]}\right)$$

3. RESULTS AND DISCUSSION

The new ParaMESH-OSTRICH tool was applied to the calibration case-study described previously in Section 2.4. Since DDS is a stochastic algorithm, with different trials potentially yielding different results, multiple (i.e. 5) DDS trials were completed. The process of configuring and running the trials using the ParaMESH-OSTRICH software allowed for a subjective assessment of the advantages and limitations of the tool.

3.1. Configuring and Running the Calibration Exercise using ParaMESH

Screenshots from configuring the OSTRICH calibration exercise using the ParaMESH interface are provided in Figures 1-3. Figure 1 illustrates the configuration of calibration parameters, which involves two steps: parameter selection and parameter configuration. During the parameter selection (Figure 1a), users are presented with a representative model input file and highlight those portions of the input file that correspond to the desired calibration parameters. During parameter configuration (Figure 1b) users fill out a convenient form for each selected parameter – form entries include: initial parameter value, lower and upper bounds, and any applicable transformations (e.g. log10 or ln).

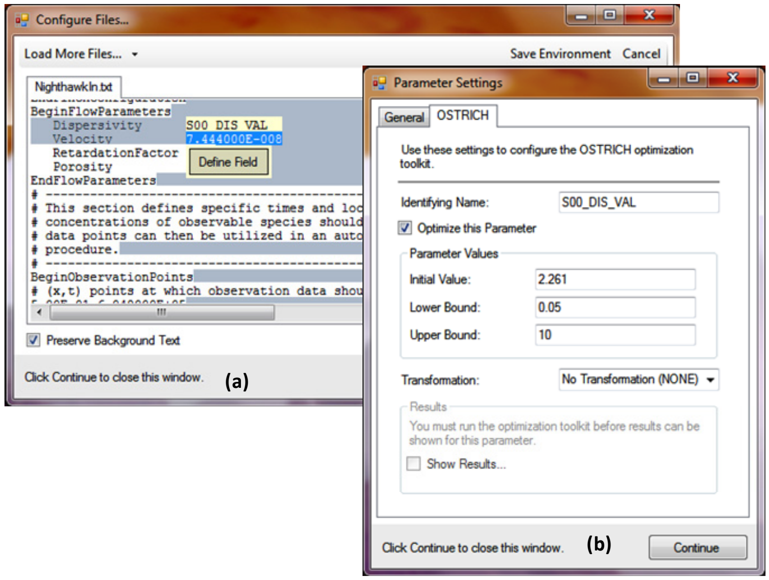

Figure 1: Defining Calibration Parameters Using ParaMESH
(a) Parameter Selection, (b) Parameter Configuration.

Figure 2 illustrates the ParaMESH form for selecting and configuring an OSTRICH calibration algorithm. The form is dynamic in that it will contain only entries that are relevant to the "Optimization Algorithm" drop-down selection. For example, Figure 2a illustrates the configuration form for the DDS algorithm while Figure 2b illustrates the form that is created when the Powell algorithm is selected. After setting up the various aspects of the calibration, users can run the calibration within ParaMESH by clicking on the "Run Model" button – this will cause ParaMESH to launch OSTRICH along with an interface that allows users to graphically monitor the status of the calibration (see Section 3.2).

ParaMESH also provides a graphical interface for importing available observation data into the calibration exercise. The interface allows users to specify weights and simulated equivalent information for individual observations as well as groups of observations. The ability to configure groups of observations and corresponding simulated equivalents allows for rapid completion of this portion of the calibration setup. In contrast, setting up observations and simulated equivalents is usually extremely tedious when using the usual text-file approach. The look-and-feel of the interface for importing observations is similar to the interface for configuring parameters (see Figure 1).

As discussed above and illustrated in Figures 1-2, all configuration steps for a given calibration exercise can be completed within ParaMESH via filling out graphical forms. When all of these forms are filled out, ParaMESH will transparently create a syntactically correct OSTRICH input file. In contrast, configuring the selected calibration exercise using the standard text-file approach would have required typing in 190 lines of text. An error in any of these lines could cause the program to crash or behave in an unexpected manner. For example, OSTRICH defaults to the Gauss-Marquardt-Levenberg algorithm if an alternative algorithm type is not specified correctly in the input file.

Figure 3 illustrates the context-sensitive help that is embedded within ParaMESH and which was tailored (via a ParaMESH extension) to support configuration of the OSTRICH tool. Within a given ParaMESH form, users simply click on a given form field to retrieve helpful information and relevant guidance via an interactive, user-dismissible pop-up form. In this way, ParaMESH supports on-demand learning about various aspects of model calibration, in general, and OSTRICH, in particular.

3.2. DDS Calibration Performance

As described previously, several DDS trials were completed and this resulted in a range of calibration results. Each trial was limited to a computational budget of 300 transport model evaluations. Table 2 summarizes the median DDS performance across all trials, measured in terms of root mean squared error (RMSE). Also included in Table 2 are results reported previously for the Gauss-Marquardt-Levenberg (GML), particle swarm optimization (PSO), and hybrid PSO-GML algorithms. Relative to the previously utilized PSO algorithm, the DDS algorithm performed reasonably well and yielded comparable (albeit somewhat larger) RMSE values using a significantly reduced computational budget.

While a given calibration exercise is running, the ParaMESH-OSTRICH tool provides a convenient a graphical interface for monitoring calibration progress. As shown in Figure 4, the calibration monitor provides views of overall calibration progress (top window section), the progress of the current algorithm step or iteration (middle section), and the progress of the current model evaluation (bottom section).

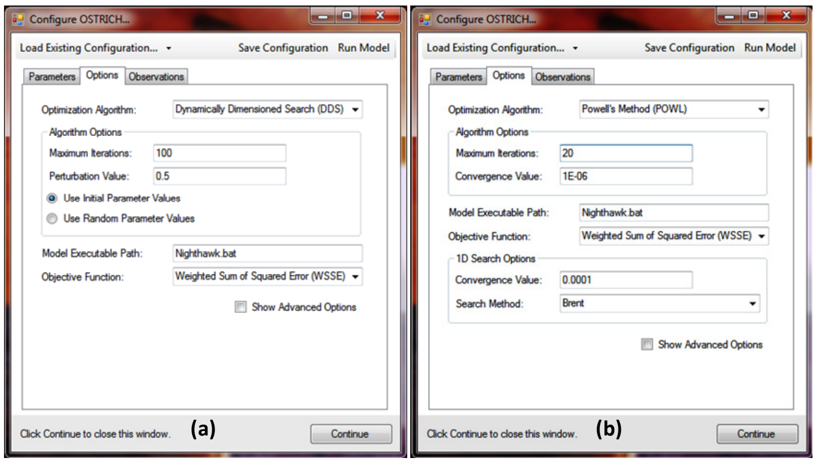

Figure 2: Configuring Calibration Algorithms Using ParaMESH Forms

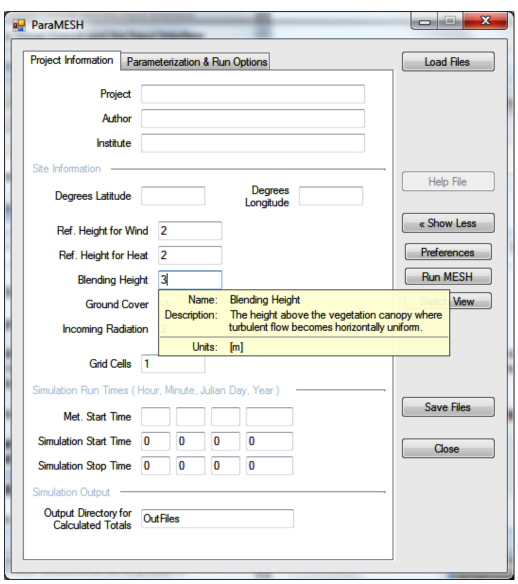

Figure 3: Example of Context-Sensitive Help

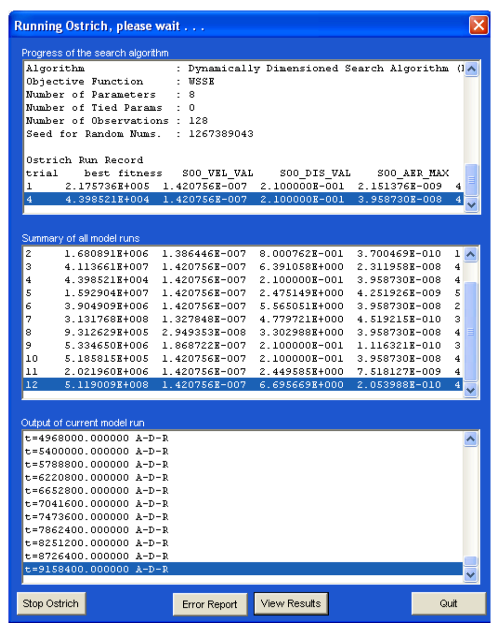

Figure 4: The OSTRICH Progress Monitor

Table 2: Comparison of the Performance of Selected Calibration Algorithms

Algorithm[a]	RMSE	Model Evaluations
GML[b]	5.86	1701
PSO	8.16	1528
PSO-GML	5.77	1601

DDS	11.5	300
[a] – GML, PSO, and PSO-GML results from Matott and Rabideau [2008], [b] – 9 multi-starts		

3.3. Estimated Parameter Values and Calibration Diagnostics

Figures 5 and 6 summarize the estimated parameter values and calibration diagnostics (i.e. linear confidence intervals and measures of parameter sensitivity) for a selected DDS trial. The user may choose from several different views, including a one-at-a-time format (Figure 5a), a tabular format (Figure 5b), and a graphical format that includes sensitivity bars as well as normalized box-and-whisker confidence intervals (Figure 6). These alternative views allow for rapid analysis of the calibration results. Inspection of Figure 6, for example, makes it immediately clear that the `AER_MAX` parameter (i.e. the maximum aerobic degradation rate) dominates in terms of sensitivity while the large (essentially unbounded) confidence bands of the `DEN_TEA` and `DEN_SUB` parameters (i.e. the ½ saturation constants for denitrification) indicates that these parameters are poorly constrained.

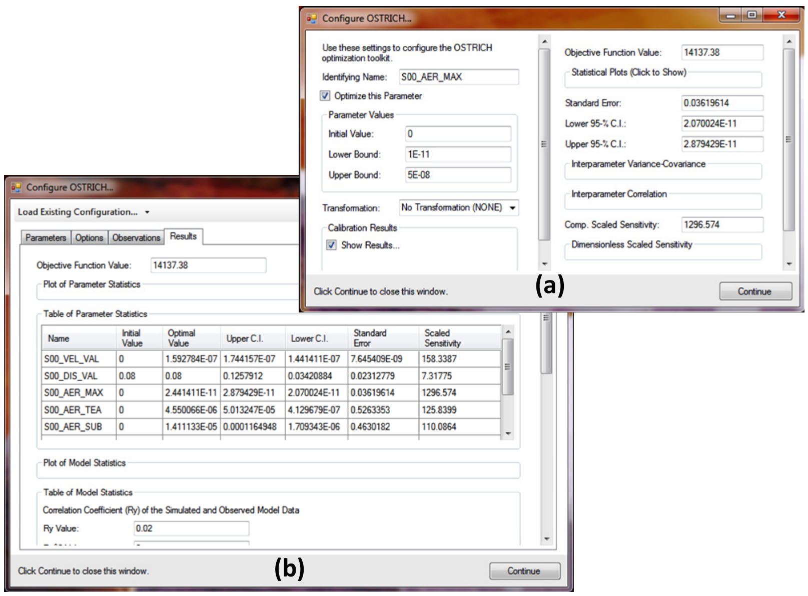

Figure 5: ParaMESH Interface for Viewing OSTRICH Output
(a) Individual parameter results, (b) Tabulation of results for all parameters

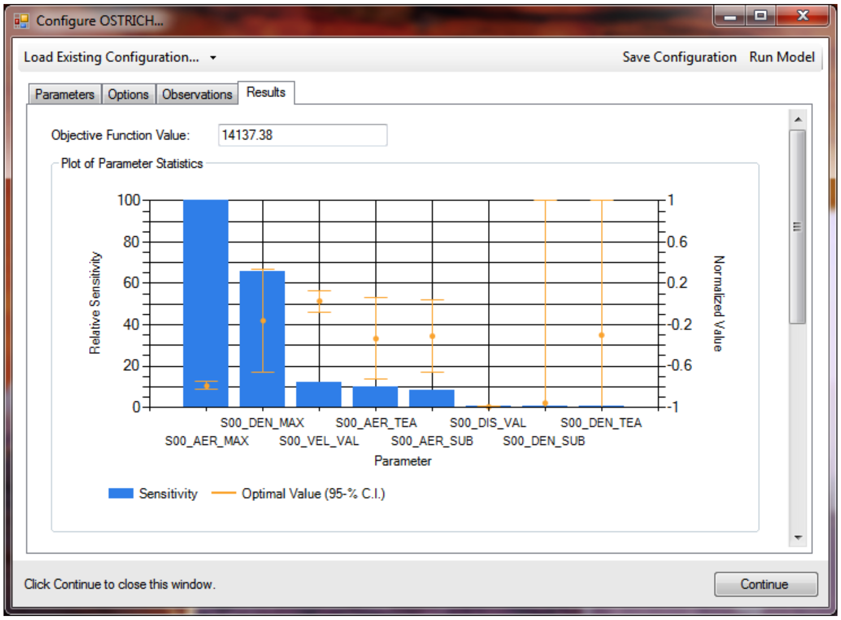

Figure 6: Plots of Parameter Sensitivities and Confidence Intervals

4. CONCLUSIONS AND RECOMMENDATIONS

ParaMESH is a new software tool that can be utilized to develop graphical user interfaces for various model evaluation tasks, such as parameter estimation and sensitivity and uncertainty analysis. To demonstrate the ParaMESH tool, it was linked with the OSTRICH multi-algorithm and model-independent calibration code. The linked ParaMESH-OSTRICH software was applied to a case-study involving subsurface transport and transformation of nitrate.

Results highlight some of the advantages of using ParaMESH to configure and run a calibration exercise. For example, ParaMESH provides convenient form-based mechanisms for configuring calibration parameters, importing observation data and matching this data with simulated equivalents, and selecting and configuring a calibration algorithm. These mechanisms are more user friendly and less error prone than the alternative approach, which requires the user to create a text-based input file using a syntax described in the lengthy OSTRICH user-manual. The linked ParaMESH-OSTRICH code also contains context-sensitive help instructions, allowing users to learn about both the calibration process and the OSTRICH code in a convenient "on-demand" manner.

Additional features of the ParaMESH tool include a graphical interface for monitoring calibration progress, and tabular and graphical representations of calibration results and related parameter diagnostics. These features allow users to rapidly analyze the results of a calibration.

Government agencies and academics are increasingly interested in using leading-edge model evaluation technologies, but would generally prefer an easy-to-use interface. This study demonstrates that ParaMESH is a promising tool for adding graphical convenience to existing non-graphical model evaluation tools, and several additional calibration tools (i.e. PEST, UCODE, and UNSIM), can easily take advantage of the ParaMESH software.

REFERENCES

Babendreier, J.E., and K.J. Castleton, Investigating uncertainty and sensitivity in integrated, multimedia environmental models: tools for FRAMES-3MRA, *Environmental Modelling & Software*, *20*(8), 1043-1055, 2005.

Banta, E.R., M.C. Hill, E. Poeter, J.E. Doherty, and J. Babendreier, Building model analysis applications with the Joint Universal Parameter IdenTification and Evaluation of Reliability (JUPITER) API, *Computers & Geosciences*, *34*(4), 310-319, 2008.

Doherty, J., PEST: Model-independent parameter estimation, User Manual, 5th edition, Watermark Numerical Computing, 336 pp., Brisbane, Australia, 2004.

Doherty, J., Addendum to the PEST Manual, Watermark Numerical Computing, pp., Brisbane (Australia), 2007.

Hill, M.C., and C.R. Tiedeman, *Effective Groundwater Model Calibration: With Analysis of Data, Sensitivities, Predictions, and Uncertainty*, Wiley and Sons, 480 pp.,Hoboken, NJ, 2007.

Kraemer, S.R., T.B. Spruill, A.J. Tesoriero, H.E. Mew, and K.M. Farrell, Modeling to evolve understanding of the shallow ground water flow system beneath the Lizzie Research Station, North Carolina In: *Guidebook of the Carolina Geological Society Annual Field Trip*, 35-46, Durham, NC, 2003.

Matott, L.S., Ostrich: an Optimization Software Tool, Documentation and User's Guide, Version 1.6, University at Buffalo, Department of Civil, Structural, and Environmental Engineering, 127 pp., Buffalo, NY, 2005.

Matott, L.S., S.L. Bartelt-Hunt, A.J. Rabideau, and K.R. Fowler, Application of Heuristic Optimization Techniques and Algorithm Tuning to Multi-layered Sorptive Barrier Design, *Environmental Science & Technology*, *40*(20), 6354-6360, 2006a.

Matott, L.S., and A.J. Rabideau, Calibration of subsurface batch and reactive-transport models involving complex biogeochemical processes, *Advances in Water Resources*, *31*(2), 269-286, 2008.

Matott, L.S., A.J. Rabideau, and J.R. Craig, Pump-and-treat optimization using analytic element method flow models, *Advances in Water Resources*, *29*(5), 760–775, 2006b.

Poeter, E.P., and M.C. Hill, UCODE, a computer code for universal inverse modeling, *Computers & Geosciences*, *25*(4), 457-462, 1999.

Princz, D., Comparing routing schemes using the semi-distributed model MESH and simulating streamflow in the Baker Creek watershed, University of Waterloo, 19 pp., Waterloo, Ontario, Canada, 2008.

Princz, D., Evaluating two optimization schemes and assessing the automated calibration process of ParaMESH on the Baker Creek watershed using MESH, University of Waterloo, 22 pp., Waterloo, Ontario, Canada, 2009.

Reichert, P., UNCSIM - A computer programme for statistical inference and sensitivity, identifiability, and uncertainty analysis, In: *Proceedings of the 2005 European Simulation and Modelling Conference (ESM 2005)*, 51-55, Porto, Portugal, October 24-26, 2005.

Skahill, B.E., J.S. Baggett, S. Frankenstein, and C.W. Downer, More efficient PEST compatible model independent model calibration, *Environmental Modelling & Software*, *24*(4), 517-529, 2009.

Spruill, T.B., H.E. Tesoriero, H.E. Mew, K.M. Farrell, S.L. Harden, A.B. Colosimo, and S.R. Kraemer, Geochemistry and Characteristics of Nitrogen Transport at a Confined Animal Feeding Operation in a Coastal Plain Agricultural Watershed, and Implications for Nutrient Loading in the Neuse River Basin, North Carolina, 1999-2002, USGS, 57 pp., 2005.

Stewart, R.N., and S.T. Purucker, SADA: A Freeware Decision Support Tool Integrating GIS, Sample Design, Spatial Modeling, and Risk Assessment, paper presented at the Third Biennial Meeting of the International Environmental Modelling and Software Society, Burlington, Vermont, July 9-13, 2006.

Tesoriero, A.J., T.B. Spruill, H.E. Mew, K.M. Farrell, and S.L. Harden, Nitrogen transport and transformations in a coastal plain watershed: Influence of geomorphology on flow paths and residence times *Water Resources Research*, *41*(2), W02008, doi:02010.01029/02003WR002953, 2005.

Tufte, E.R., *The Visual Display of Quantitative Information*, Graphics Press, 197 pp.,Cheshire, Connecticut, 2001.