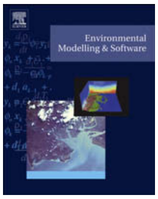
Environmental Modelling & Software 24 (2009) 517–529

Contents lists available at ScienceDirect

Environmental Modelling & Software

journal homepage: www.elsevier.com/locate/envsoft



More efficient PEST compatible model independent model calibration

Brian E. Skahill [a,*], Jeffrey S. Baggett [b], Susan Frankenstein [c], Charles W. Downer [a]

[a] Coastal and Hydraulics Laboratory, U.S. Army Engineer Research and Development Center, Hydrologic Systems Branch, Waterways Experiment Station, 3909 Halls Ferry Road, Vicksburg, MS 39180, USA
[b] Department of Mathematics, University of Wisconsin – La Crosse, La Crosse, WI 54601, USA
[c] Cold Regions Research and Engineering Laboratory, U.S. Army Engineer Research and Development Center, 72 Lyme Road, Hanover, NH 03755, USA

ARTICLE INFO

Article history:
Received 16 November 2007
Received in revised form
26 September 2008
Accepted 30 September 2008
Available online 20 November 2008

Keywords:
Calibration efficiency
Secant version of Levenberg–Marquardt
Multi-Level Single Linkage

ABSTRACT

This article describes some of the capabilities encapsulated within the Model Independent Calibration and Uncertainty Analysis Toolbox (MICUT), which was written to support the popular PEST model independent interface. We have implemented a secant version of the Levenberg–Marquardt (LM) method that requires far fewer model calls for local search than the PEST LM methodology. Efficiency studies on three distinct environmental model structures (HSPF, FASST, and GSSHA) show that we can find comparable local minima with 36–84% fewer model calls than a conventional model independent LM application. Using the secant LM method for local search, MICUT also supports global optimization through the use of a slightly modified version of a stochastic global search technique called Multi-Level Single Linkage [Rinnooy Kan, A.H.G., Timmer, G., 1987a. Stochastic global optimization methods, part I: clustering methods. Math. Program. 39, 27–56; Rinnooy Kan, A.H.G., Timmer, G., 1987b. Stochastic global optimization methods, part ii: multi level methods. Math. Program. 39, 57–78.]. Comparison studies with three environmental models suggest that the stochastic global optimization algorithm in MICUT is at least as, and sometimes more efficient and reliable than the global optimization algorithms available in PEST.

Published by Elsevier Ltd.

Software availability

Name of software: Model Independent Calibration and Uncertainty Analysis Toolbox (MICUT)
Developers: Brian E. Skahill and Jeffrey S. Baggett
Contact Address: Coastal and Hydraulics Laboratory, U.S. Army Engineer Research and Development Center, Hydrologic Systems Branch, Waterways Experiment Station, 3909 Halls Ferry Road, Vicksburg, MS 39180, USA or Department of Mathematics, University of Wisconsin – La Crosse, La Crosse, WI 54601, USA
Telephone: +1 601 634 3441 or +1 608 785 8393
fax: +1 601 634 4208 or +1 608 785 6602
E-mail: brian.e.skahill@usace.army.mil or baggett.jeff@uwlax.edu
Year first available: 2008
Hardware required: PC
Program language: C
Availability and cost: Available free of charge by email from first author

1. Introduction

Computer-based calibration of environmental models generally involves minimization of an “objective function” – a measure of model-to-measurement misfit. In simple cases this is comprised of a single objective; for example, in the watershed modeling context it could be specified as differences between measured and modeled stream flows at daily, hourly, or even smaller intervals. In more complex cases a multi-criterion objective function is constructed in which different measurement types, or the same measurement type processed in different ways, comprise separate components of a composite global objective function (in the watershed modeling context, see Madsen, 2000; Boyle et al., 2000; Cappelaere et al., 2003; Doherty and Johnston, 2003; Shrestha and Rode, 2008).

An important consideration in assessing the performance of a parameter estimation package is that of run time. Parameter estimation software, no matter what its algorithmic basis, must run the model to be calibrated many times in the course of minimizing the objective function that is used to characterize model-to-measurement misfit. Minimizing the number of model runs required during the calibration process is nearly always important, but particularly when the objective function landscape contains multiple local minima, model run times are high, or when multiple prediction specific calibrations must be conducted within the context of a single model deployment (Moore and Doherty, 2005).

[*] Corresponding author. Tel.: +1 601 634 3441; fax: +1 601 634 4208.
E-mail addresses: brian.e.skahill@usace.army.mil, beskahill@earthlink.net (B.E. Skahill).

1364-8152/$ – see front matter Published by Elsevier Ltd.
doi:10.1016/j.envsoft.2008.09.011

The Levenberg–Marquardt (LM) method of computer-based parameter estimation (Levenberg, 1944; Marquardt, 1963) was implemented to be a part of the Model Independent Calibration and Uncertainty Analysis Toolbox (MICUT) software, partially described herein. The LM method has features that make it attractive for model calibration. In calibration contexts where local optima are rare or nonexistent, the LM method is efficient in terms of its model run requirements. Also, is its ability to readily report estimates of parameter uncertainty, correlation, and (in)sensitivity as a by-product of its use both during and after the parameter estimation process. And moreover, the LM method is also easily adapted to include regularization devices to maintain numerical stability and robustness in the face of potential numerical problems, that adversely affect all parameter estimation methodologies, caused by parameter insensitivity and/or parameter correlation (Menke, 1984; DeGroote-Hedlin and Constable, 1990; Doherty and Skahill, 2006).

The model independent Levenberg–Marquardt (LM) method based parameter estimation software PEST (Doherty, 2004, 2007a,b), which quantifies model-to-measurement misfit in the weighted least squares sense, is now widely used to support environmental model calibration. In addition to its traditional groundwater model calibration application setting (Zyvoloskia et al., 2003; Tonkin and Doherty, 2005; Moore and Doherty, 2006; Gallagher and Doherty, 2007a), it is now employed to calibrate ecological models (Rose et al., 2007; Gaucherel et al., 2008), land surface models (Santanello Jr. et al., 2007) and models in other application areas including nonpoint source pollution (Baginska et al., 2003; Haydon and Deletic, 2007), surface hydrology (Doherty and Johnston, 2003; Gutiérrez-Magness and McCuen, 2005; Kunstmann et al., 2006; Skahill and Doherty, 2006; Doherty and Skahill, 2006; Gallagher and Doherty, 2007b; Goegebeur and Pauwels, 2007; Iskra and Droste, 2007; Kim et al., 2007; Maneta et al., 2007), and surface water quality (Rode et al., 2007).

The primary focus of this article is to show how it is possible to efficiently overcome a couple of drawbacks associated with LM-based Model Independent Parameter Estimation as implemented in PEST. One drawback associated with the LM method is that it requires the derivatives of the objective function with respect to the model parameters. Model independent LM implementations can become computationally costly when elements of the Jacobian matrix must be computed using finite differences based on model runs with incrementally varied parameter values. Certainly, using multiple processors can decrease the time required to construct the Jacobian matrix; however, it would be better, as demonstrated here, to not populate the entire Jacobian matrix unless really necessary.

Another drawback of the LM method is that it is a local search method. Thus, if there are different "regions of attraction" in parameter space, its solution will lead to just one of possibly many objective function minima, the particular one that is found is dependent upon the user-supplied set of initial parameter values. While stochastic global optimization (GO) can be employed as a remedy, one would like to utilize stochastic GO methods that are not only reliable in finding the global minimum, but also efficient in the sense that they minimize the return to previously visited local minima in parameter space. A modeler would possibly also like to receive some information on the locations of non-global minima, especially if these minima are little different in magnitude from the global minimum, but are widely separate from it in parameter space. Characterizing the structure of the objective function surface allows a modeler to qualitatively appraise the linearity and utility of his/her model, the uncertainty of parameters estimated though the parameter estimation process, and the information content of the data set that is currently available for the model calibration (Sorooshian and Arfi, 1982; Kuczera, 1990). However, such a characterization should not come at the possible expense of repeatedly returning to the same locations on the objective function surface.

To better enable the expectation that estimated parameter sets result in the best possible fit between model outputs and field measurements, Skahill and Doherty (2006) enhanced the PEST software (Doherty, 2004) LM local search method via the implementation of two stochastic global optimization methods, viz. Multistart and Trajectory Repulsion. Both of these methods can suffer from repeatedly locating the same local minima. Doherty (2003) also enabled an interface to the Shuffled Complex Evolution (SCE) general purpose global optimization method (Duan et al., 1992, 1993) and the Covariance Matrix Adaption Evolutionary Strategy (CMAES) general purpose optimization method (Hansen and Ostermeier, 2001; Hansen et al., 2003).

The objectives of this article are to report on the following:

1. Some of the capabilities of the independent parameter estimation software package MICUT that accommodates the PEST model independent and input control file protocol (Doherty, 2004),
2. Efficiency enhancements in the MICUT LM method implementation,
3. Implementation into MICUT of an alternative stochastic global optimization method, which uses the secant LM method for local search, that can be readily employed with only slight modification to an existing PEST input control file,
4. An efficiency comparison, using three distinct environmental model structures, of the newly implemented stochastic global optimization method in MICUT with Trajectory Repulsion, and the SCE and CMAES optimization methods that are currently available to users of the PEST Model Independent Parameter Estimation software (Doherty, 2004).

Table 10 lists and briefly defines acronyms used within this article.

2. Methodologies and enhancements

2.1. Levenberg–Marquardt method

As previously mentioned, the context for quantifying model-to-measurement misfit is nonlinear least squares minimization. Marquardt (1963) modified the Levenberg method (Levenberg, 1944), which is a blend of the gradient descent and Newton's methods of parameter estimation, to better accommodate Hessian information into the parameter upgrade vector. Marquardt's modification remedied the classic problem of "hemstitching", wherein successive parameter improvements result in oscillations across a long and narrow objective function valley, which is never actually penetrated (Doherty and Skahill, 2006). The Levenberg–Marquardt method is given by

$$\mathbf{p} - \mathbf{p}_0 = \left(\mathbf{X}^t\mathbf{Q}\mathbf{X} + \lambda\,\mathrm{diag}[\mathbf{X}^t\mathbf{Q}\mathbf{X}]\right)^{-1}\mathbf{X}^t\mathbf{Q}(\mathbf{h} - \mathbf{h}_0) \tag{1}$$

where $\mathbf{p}_0$, $\mathbf{X}$, $\mathbf{Q}$, λ, $\mathbf{h}$, and $\mathbf{h}_0$ represent current parameter values, the model Jacobian matrix, each row of which is comprised of the derivatives (i.e. sensitivities) of a particular model output (for which there is a corresponding field measurement) with respect to all elements of $\mathbf{p}$, a "weight matrix" wherein, ideally, each diagonal element is proportional to the inverse of the squared potential error associated with the corresponding processed measurement, the parameter which blends gradient descent (dominant when λ is large) and Newton's method (dominant when λ is small), and the n-dimensional vectors of observations and model outputs, respectively. The Hessian of the objective function has been approximated in the usual manner by assuming that the residuals can be estimated by linear functions or the residuals themselves are small. Where the model is nonlinear, $\mathbf{p}$ calculated through equation

(1) is not optimal (i.e. it does not minimize the objective function) unless $\mathbf{p}_0$ is close to optimal. Hence, after equation (1) is used to calculate an improved parameter set, a new set of sensitivities (i.e. $\mathbf{X}$) is calculated on the basis of the new parameter set, and the process is repeated until convergence to the objective function minimum is achieved. Skahill and Doherty (2006) and Doherty and Skahill (2006) both provide more lengthy summaries of the LM method.

2.2. *Secant version of the Levenberg–Marquardt method*

With a conventional model independent implementation of the LM method, the environmental model is a "black box" in that only outputs of the model are available and elements of the matrix **X** are often obtained by numerical differentiation. The LM method implemented in PEST (Doherty, 2004) requires anywhere between m and $2m$ forward model calls (dependent upon whether forward or central finite differences are employed) to populate the column space of the matrix **X** at each optimization iteration. It has been suggested that this is a general requirement for model independent derivate-based methods, such as LM, that employ perturbation sensitivities to populate the matrix **X** at each optimization iteration (Doherty, 2004; Tonkin and Doherty, 2005). To the contrary, there are well-established methods (Broyden, 1965) available that allow for better efficiency with respect to updating the matrix **X** at each optimization iteration.

2.2.1. *Broyden's rank one update*

Let **X** denote the current approximation to the Jacobian. Broyden (1965) developed a generalized secant method to update an approximation to the Jacobian, $\mathbf{X}_{new}$. Broyden's rank one update is given by

$$\mathbf{X}_{new} = \mathbf{X} + \frac{[\mathbf{y} - \mathbf{X}\mathbf{s}]\mathbf{s}^{\mathrm{T}}}{\mathbf{s}^{\mathrm{T}}\mathbf{s}} \tag{2}$$

where **y** and **s** represent the n-dimensional vector of differences in model outputs obtained from the current and previous parameter values, and the m-dimensional vector of the difference between the current and previous parameter values, respectively. Many update formulas are available (Fletcher, 1987); however, formula (2) has been shown to work well for nonlinear least squares problems.

Incorporating Broyden's rank one update into the LM implementation eliminates the requirement to conduct any additional forward model calls to populate the matrix $\mathbf{X}_{new}$ at each optimization iteration. To mitigate against the potential that $\mathbf{X}_{new}$ may eventually become a poor approximation to the true Jacobian after some optimization iterations, $\mathbf{X}_{new}$ can occasionally be fully updated in the usual manner using finite differences. Furthermore, this occasional full updating can also be supplemented through cyclic updating, using finite differences, at each optimization iteration, of anywhere between one and m individual columns of **X** (Madsen et al., 2004).

The independent LM implementation in MICUT accommodates the model independent PEST interface (Doherty, 2004) and includes the following additional abilities with respect to updating the matrix **X** (Skahill and Baggett, 2006) which in all cases is initially approximated by a full update using forward and/or central finite differences:

1. A full update, at each optimization iteration, using forward and/or central finite differences.
2. Use of the Broyden rank one update.
3. Use of the Broyden rank one update, with a recomputation, i.e. a full update of **X** whenever the ratio of the new and old objective function values is greater than a specified input value.
4. Use of the Broyden rank one update, with a recomputation, i.e. a full update of **X** whenever the ratio of the new and old objective function values is greater than a specified input value, and also cyclic updating, using finite differences, at each optimization iteration, of anywhere between one and m (a specified input) individual columns of **X**.

MICUT may be used as an alternative to PEST (Doherty, 2004, 2007a,b) for more efficient model independent LM-based parameter estimation. Only slight modifications to the PEST control file are required to utilize the Broyden update functionalities noted above. MICUT also provides linear based information on parameter uncertainty, correlation, and sensitivity. Doherty (2007a,b) reportedly did implement the Broyden rank one update, but evidently still computes a full update to the Jacobian matrix at each optimization iteration. This approach does not fully realize the potential efficiency gains of a secant version of the LM method and in some cases, as shown below, significantly increases the number of model runs required to find a local minimum.

2.3. *Stochastic global optimization*

As mentioned above, one of our objectives was to include into MICUT a more efficient and reliable stochastic global optimization algorithm than what is currently available in PEST. Efficient and reliable optimization methods, possibly constrained by a predetermined computational budget, that are capable of efficiently finding the locations of other good minima in addition to an estimate of the global minimum, are needed to identify environmental models. See, for example, Duan et al. (1993), Gupta et al. (2003), Shoemaker et al. (2007), Tolson and Shoemaker (2007), and references cited therein for applications of global optimization in the watershed modeling context. Stochastic global optimization algorithms estimate the global minimum of the objective function by initiating local searches from global, randomly sampled points. The local and global phases can be iterated and/or the local searches may be initiated at some or all or the globally sampled points. Stochastic global optimization algorithms are guaranteed to converge, with probability one, to the global minimum as the sample size approaches infinity. Stronger convergence properties are possible for some stochastic algorithms, as we mention below. Moreover, probabilistic-based stopping criteria can be developed for stochastic global optimization methods (Rinnooy Kan and Timmer, 1987a,b; Törn and Žilinskas, 1987); however, an a priori computational budget may preclude any concern regarding termination criteria.

2.3.1. *Multistart*

MICUT and PEST both incorporate a simple stochastic global optimization algorithm called Multistart, that involves both a global phase and a local phase. The Multistart method samples points from a uniform distribution over the feasible parameter space and starts a local search from each of the sample points. For both MICUT and PEST, the local search algorithm is the Levenberg–Marquardt method. Uniform random sampling ensures global reliability of the method and the estimate for the global minimum is the smallest local minimum found. Multistart is inefficient in that each local minimum, particularly those in large regions of attraction in the parameter space, is generally found multiple times. For both MICUT and PEST, Multistart naively stops after a user-specified number of iterations; however, more sophisticated stopping rules are possible. For instance, Boender (1984) developed optimal Bayesian stopping criteria for Multistart.

2.3.2. *Trajectory Repulsion*

In attempts to overcome the inefficiency of repeatedly locating the same local minima, Skahill and Doherty (2006) developed, and included into PEST, the Trajectory Repulsion method as a basis for extending LM-based parameter estimation to find multiple local optima. It is a stochastic global optimization algorithm designed to encourage maximal exploration of feasible parameter space. Trajectory Repulsion begins by evaluating the objective function on a single uniform random sample of points and discarding those points for which the objective function is above the median. A local search is begun from the point with the lowest objective function value. Subsequent local searches are initiated at points in the reduced sample set that are furthest from previous search trajectories, in attempts at avoiding repeatedly locating the same local minima. A variety of user-specified stopping criteria are implemented, and can be used to balance local and global exploration in parameter space. A complete specification of the Trajectory Repulsion method may be found in Skahill and Doherty (2006).

MICUT includes a slightly modified implementation of Trajectory Repulsion. To potentially increase efficiency, we have made a slight modification that terminates the current local search if the current trajectory is within a user-specified distance of any previous parameter trajectory, the presumption being that the current trajectory is headed toward an already found minimum.

Skahill and Doherty (2006) considered a single case study example, in the watershed modeling context, to evaluate the efficiency of Trajectory Repulsion relative to Multistart and SCE. Although its run-efficiency was shown to be at least as good as that of the SCE method for the single case study example considered, additional results indicated that it is nonetheless inefficient, much like Multistart, in that it repeatedly found the same local minima.

2.3.3. Clustering methods and Multi-Level Single Linkage

Multistart and Trajectory Repulsion are both inefficient in that they may return to previously visited local minima several times. Ideally, we would like to perform a single local search within the region of attraction of each local minimum. This would not only ensure that each local minimum is identified just once, but also that in fact we find all local minima. But we also want to employ a method that works well if one has a predetermined computational budget in that for a given effort it compares favorably with other methods. Clustering methods were designed to accommodate these requirements. They are variants of Multistart and the basic idea behind them is to group close points, sampled from the feasible parameter space and for which the specified groups presumably relate to actual regions of attraction in parameter space, and to apply a single local search procedure within each identified cluster. Either reduction; wherein sampled points associated with the highest objective function values are temporarily removed, or concentration; wherein the sampled points are transformed through application of one or a few iterations of a local search procedure, is employed to identify a reduced sample as part of the clustering process in order to provide some assurance that in fact the specified groups correspond to regions of attraction of actual local minima. Clustering methods are often iterative in that the global and local phases are repeated sequentially until a stopping criterion is satisfied.

With clustering methods, it is possible that one cluster intersects multiple regions of attraction; hence, the global minimum could be missed, or that one region of attraction contains more than one cluster, thus allowing for the same local minimum to possibly be identified more than once. Multi-Level Single Linkage (MLSL) is a clustering method that was developed to reduce the probability of not finding a local minimum or of finding a local minimum more than once (Rinnooy Kan and Timmer, 1987a,b).

MLSL mimics clustering by calculating a critical distance r_k at each iteration, k. This critical distance can be used to build clusters, but instead, in MLSL, the decision as to whether a local search is to be initiated from a given reduced sample point is simply based on whether there exists another reduced sample point within the distance r_k of the given point with a corresponding lower objective function value. The critical distance r_k is reduced at each iteration.

Under certain assumptions, MLSL has stronger convergence properties than simpler stochastic global optimization algorithms. First, if the algorithm continues forever, the number of local searches performed is finite. Second, if r_k tends to zero with increasing k, then every local minimum will be identified in finite time with probability one.

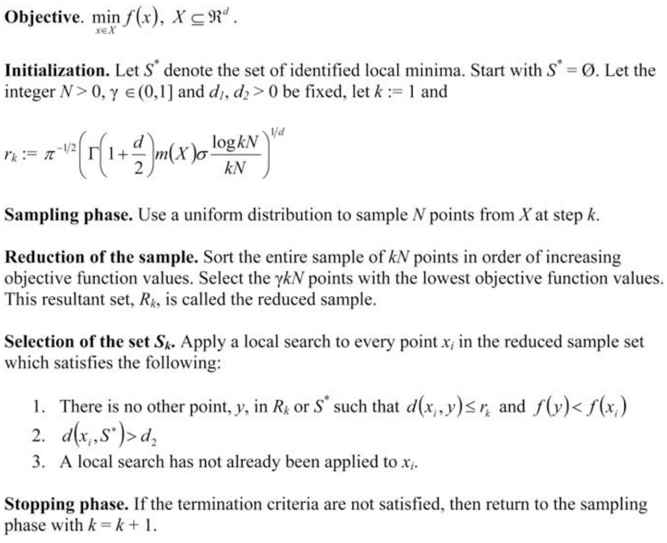
MLSL was implemented as part of the MICUT software and it uses the LM method for local search. The MLSL implementation follows that of Rinnooy Kan and Timmer (1987a,b) with a slight modification to sometimes avoid repeatedly finding the same local minima. Our implementation of the MLSL algorithm is summarized in Fig. 1. It can be utilized by simply appending the MLSL search parameters to the end of the control data section of a working PEST input control file. The following stopping criteria were included as part of our MLSL implementation:

1. The objective function has not been lowered over a specified number of local searches,
2. A specified maximum number of local searches have been performed,
3. The expected number of minima, in the Bayesian sense, exceeds the number of identified distinct local minima by less than 0.5 (Rinnooy Kan and Timmer, 1987a,b),
4. A specified maximum number of MLSL iterations have been performed,
5. The objective function has not been lowered over a specified number of MLSL iterations.

An instance of our MLSL implementation stops when any one of the above criteria are satisfied.

3. Examples

Efficiency gains that can be achieved from a properly implemented secant version of the LM method relative to conventional LM application, using the implementations in MICUT, are



Objective. $\min_{x \in X} f(x),\ X \subseteq \Re^d$.

Initialization. Let S^* denote the set of identified local minima. Start with $S^* = \emptyset$. Let the integer $N > 0$, $\gamma \in (0,1]$ and d_1, $d_2 > 0$ be fixed, let $k := 1$ and

$$r_k := \pi^{-1/2}\left(\Gamma\left(1+\frac{d}{2}\right)m(X)\sigma\frac{\log kN}{kN}\right)^{1/d}$$

Sampling phase. Use a uniform distribution to sample N points from X at step k.

Reduction of the sample. Sort the entire sample of kN points in order of increasing objective function values. Select the γkN points with the lowest objective function values. This resultant set, R_k, is called the reduced sample.

Selection of the set S_k. Apply a local search to every point x_i in the reduced sample set which satisfies the following:

1. There is no other point, y, in R_k or S^* such that $d(x_i, y) \le r_k$ and $f(y) < f(x_i)$
2. $d(x_i, S^*) > d_2$
3. A local search has not already been applied to x_i.

Stopping phase. If the termination criteria are not satisfied, then return to the sampling phase with $k = k + 1$.

Fig. 1. MICUT implementation of Multi-Level Single Linkage algorithm.

Table 1
HSPF parameters, their functions, and constraints imposed during the calibration process.

Parameter name	Parameter function	Bounds imposed during calibration process
IMP	Percent effective impervious area	11–19% (Alley and Veenhuis, 1983)
LZSN	Lower zone nominal storage	2–15 in (5–38 cm)
UZSN	Upper zone nominal storage	0.05–2 in (0.1–5 cm)
INFILT	Related to infiltration capacity of the soil	0.001–1.0 in/h (0.002–2.5 cm/h)
LZETP	Lower zone ET parameter – an index of the density of deep-rooted vegetation	0.1–0.9
INTFW	Interflow inflow parameter	1.0–10.0
IRC	Interflow recession parameter	0.30–0.85 day^{-1}
AGWRC	Groundwater recession parameter	0.833–0.999 day^{-1}

demonstrated in Section 4 by examining the reduction in the total number of model calls for single local searches. The efficiency gains from the independent LM implementation in MICUT are also compared against efficiencies associated with the model independent LM-based PEST software (Doherty, 2004, 2007a,b), using an eight parameter Hydrological Simulation Program – FORTRAN (HSPF) (Bicknell et al., 2001) hydrologic model, a ten parameter Fast All-season Soil Strength (FASST) state-of-the-ground model (Frankenstein and Koenig, 2004), and a sixteen parameter Gridded Surface Subsurface Hydrologic Analysis (GSSHA) (Downer and Ogden, 2003a,b) hydrologic model as case study examples. Using the three previously mentioned environmental models, we also compare the efficiencies, in terms of the number of model calls required to achieve a given objective function value, of the implementations of Trajectory Repulsion and MLSL in MICUT with that of SCE and CMAES as implemented in PEST (Doherty, 2007b).

3.1. HSPF model description

Use of the methodologies discussed in the preceding sections are now demonstrated by applying them to the calibration of an HSPF hydrologic model for the 6.6 square kilometer Wildcat Creek watershed located in Kitsap County, Washington. The model was developed to support a total maximum daily load study (ENVVEST Regulatory Working Group 2002). Its run time on a Pentium 4 computer with a 2 GHz processor was about 4 s.

Estimation of eight HSPF parameters was undertaken by matching observed and simulated daily flows over four non-contiguous time intervals spanning the period 1st Jan 2001–2nd Sep 2002, resulting in a total of 456 daily flow observations for use in the calibration process. (Data absences over this period were caused by a malfunctioning gage.) The shortcomings of such a short data set as a basis for reliable parameter estimation are well known (Yapo et al., 1996); unfortunately, however, no other data were available for calibration of this model. This is not of concern in the present instance as the purpose of this article is to demonstrate the capabilities of the methodologies discussed above in improving inversion run-efficiency for LM method based model independent calibration. The objective function was defined as the sum of weighted squared differences between modeled and observed log-transformed flows, with all weights assigned a value of 1.0. Thus **h** of equation (1) was comprised of the logs of daily flows, while the model represented by **X** in these equations calculated the model-generated counterparts to these logged flows. **Q** was the identity matrix.

Table 1 lists the names and functions of the HSPF parameters estimated through the calibration process. Also shown in this table are the bounds applied to these parameters; guidance in the setting of most of these bounds was obtained from USEPA (2000). Note that, in order to circumvent hypersensitivity of the AGWRC parameter as it approaches 1.0, it was transformed prior to estimation; the transformed parameter (named AGWRCTRANS in the present study) can vary between 5.0 and 999.0 as AGWRC varies between 0.833 and 0.999. See Skahill and Doherty (2006) for details. HSPF parameters other than those appearing in Table 1 were fixed at reasonable values.

3.2. FASST model description

The methodologies discussed in the preceding section were also demonstrated by applying them to the calibration of a FASST state-of-the-ground model for Yuma, Arizona. Ten FASST model parameters were estimated by matching observed and simulated surface soil moisture and surface soil temperature data for the period 15 March 1993–30 April 1993, resulting in a total of forty-seven surface soil moisture and 1122 surface soil temperature observations for use in the calibration process. The names and meanings of the FASST adjustable model parameters are listed in Table 2. As with the HSPF model application, in order to better accommodate scaling issues resulting from the use of different units for different parameters, and in an attempt to decrease the degree of nonlinearity of the parameter estimation problem, the logs of the adjustable parameters were estimated instead of their native values. Past experience has demonstrated that greater efficiency and stability of the parameter estimation process can often be achieved through this means (Skahill and Doherty, 2006). FASST input parameters other than those appearing in Table 2 were fixed at reasonable values. Weights were uniformly adjusted within the two observation groups constituting the objective function such that the

Table 2
Name and meaning of FASST adjustable model parameters.

Name	Meaning	Bounds imposed during calibration process
bddm	Bulk density of dry material (g/cm^3)	1.14–2.97
por	Porosity (0.0–1.0)	0.23–0.54000
lse	Longwave surface emissivity	0.80–0.99
qc	Quartz content (0.0–1.0)	0.03–0.54
tcbdm	Thermal conductivity of the bulk dry material (W/m K)	0.29–0.83
shdm	Specific heat of dry material (J/kg K)	803.9–850.6
shc	Saturated hydraulic conductivity (cm/s)	2.42E-07–8.61E-03
rwc	Residual water content (vol/vol) (0.0–1.0)	1.00000E-03–0.01000
vGBph	van Genuchten Bubbling pressure head (cm)	8.0645–141.254
vGe	van Genuchten exponent (n)	1.12500–4.808

Table 3
Name and meaning of GSSHA adjustable model parameters.

Name	Meaning	Bounds imposed during calibration process
ro_pine	Overland flow roughness coefficient – forest	0.075–0.45
ro_cottn	Overland flow roughness coefficient – cotton/soy fields	0.075–0.45
ro_pastr	Overland flow roughness coefficient – pasture	0.075–0.45
ro_gully	Overland flow roughness coefficient – gullied land	0.075–0.45
re_pine	Overland flow retention depth – forest	0.1–2.00 mm
re_cottn	Overland flow retention depth – cotton/soy fields	0.1–2.00 mm
re_pastr	Overland flow retention depth – pasture	0.1–2.00 mm
re_gully	Overland flow retention depth – gullied land	0.1–2.00 mm
hcnd_GSL	Soil saturated hydraulic conductivity – gullied land/silt loam	0.17–1.3 cm/h
hcnd_PCL	Soil saturated hydraulic conductivity – pasture/clay loam	0.025–0.41 cm/h
hcnd_CCL	Soil saturated hydraulic conductivity – cotton/clay loam	0.025–0.41 cm/h
hcd_PnCL	Soil saturated hydraulic conductivity – pine/clay loam	0.025–0.60 cm/h
hcd_PnSL	Soil saturated hydraulic conductivity – pine/silt loam	0.025–0.2 cm/h
hcnd_CSL	Soil saturated hydraulic conductivity – cotton/silt loam	0.07–1.5 cm/h
hcnd_PSL	Soil saturated hydraulic conductivity – pasture/silt loam	0.08–1.3 cm/h
ch_rough	Channel roughness coefficient	0.0275–0.0375

parameter estimation engine saw each of them to be of equal importance at the beginning of the parameter estimation process.

3.3. *GSSHA model description*

The methodologies discussed in the preceding sections were further demonstrated by applying them to the calibration of a GSSHA hydrologic model for the Goodwin Creek Experimental Watershed (GCEW) (Senarath et al., 2000; Downer and Ogden, 2003b). As with Senarath et al. (2000), spatially varied rainfall and runoff data measured at the outlet of Goodwin Creek for the period 22 May 1982–02 July 1982 were used to calibrate the GSSHA GCEW hydrologic continuous simulation model. Although available, no data from interior stream flow gauges were used to calibrate the model. The model was calibrated against hydrographs recorded at gauging station 1 only. The objective function was composed of a single observation group defined as the sum of weighted squared differences between the 233 modeled and observed transformed flows, with all weights assigned a value of 1.0. In order to reduce heteroscedascity, the Box–Cox transformation, $T(Q) = ((Q+1)^{\lambda} - 1)/\lambda$ with $\lambda = 0.3$ (Box and Jenkins, 1976; Misirli et al., 2003), was employed to transform the observed and modeled flows. Table 3 lists the names and meanings of the sixteen GSSHA adjustable model parameters. The logs of the adjustable parameters were estimated instead of their native values.

4. Efficiency of the secant version of the LM method

The three previously mentioned environmental model structures were employed to examine efficiencies associated with variations of our Secant LM (SLM) implementation relative to a conventional model independent LM application wherein the column space of the model sensitivity matrix is fully updated at each optimization iteration. For each of the three model structures, using MICUT, we performed thirty LM inversions and thirty SLM inversions for each variation considered, and in each case starting from the same initial points. Each individual trial; however, used a different initial guess. The mean number of model calls required to complete the thirty inversions, mean final objective function values, and the mean of their standard deviations are summarized in Tables 4–6 for the HSPF, FASST, and GSSHA models, respectively. Moreover, we used PEST (Doherty, 2004, 2007a,b), since it reportedly supports a variation of Secant LM, to repeat the same runs using the same input control files and initial points.

The three variations of our SLM implementation that we considered for the eight parameter HSPF hydrologic model achieved 79%, 69%, and 52% reductions in the mean number of total model calls relative to our conventional LM application, wherein a full update for the model sensitivity matrix was computed at each optimization iteration. All three SLM cases did not employ the option to fully update the model sensitivity matrix at any point during the inversion process, but did utilize cyclic updating of zero, one, or two columns of the model sensitivity matrix at each optimization iteration. The greatest reduction in the mean number of total model calls was achieved for the variation of SLM that did not employ full updating or cyclic updating of the model sensitivity matrix. While the total number of model calls is greatly reduced by employing our SLM method, there is an associated reduction in objective function improvement. For example, the variation of SLM that achieved the largest reduction in total model calls yielded a mean final objective function value seventeen percent greater than that achieved using LM. In many cases the model calibration cost savings will outweigh the possible slight degradation in model fit. For instance, Fig. 2 contains plots of simulated transformed flows for the seventh inversion runs of LM and SLM without full or cyclic updating. They achieved final objective function values of 25.1 and 34.2, respectively. These final objective function values are better and worse, respectively, than their associated mean values of 26.9 and 31.6 reported in Table 4. The simulated hydrographs displayed in Fig. 2, together with their observed counterparts, provide a basis for visually comparing calibration results associated with final objective function values commensurate with the computed

Table 4
Mean and standard deviation of final objective function value and mean total model calls using LM and variations of the secant version of LM. Results are based on thirty local searches with the eight parameter HSPF hydrologic model.

	Full update (LM)	Broyden update (SLM)		
		No full update & no cyclic updating	Cyclic updating with 1 column; no full update	Cyclic updating with 2 column; no full update
Mean of final objective function value	26.9	31.6	29.2	26.2
Standard deviation of final objective function value	5.7	7.0	6.0	5.0
Mean number of total model calls	367	62	94	142

Table 5
Mean and standard deviation of final objective function value and mean total model calls using LM and variations of the secant version of LM. Results are based on thirty local searches with the ten parameter FASST model.

	Full update (LM)	Broyden update (SLM)			
		No full update & no cyclic updating	With full update but no cyclic updating	Cyclic updating with 1 column; no full update	Cyclic updating with 5 column; no full update
Mean of final objective function value	735	1007	928	883	870
Standard deviation of final objective function value	700	725	714	709	710
Mean number of total model calls	307	48	144	79	179

Table 6
Mean and standard deviation of final objective function value and mean total model calls using LM and variations of the secant version of LM. Results are based on thirty local searches with the sixteen parameter GSSHA model.

	Full update (LM)	Broyden update (SLM)
		No full update & no cyclic updating
Mean of final objective function value	64.0	66.9
Standard deviation of final objective function value	4.9	7.0
Mean number of total model calls	426	60

means for the thirty trials for LM and SLM without full or cyclic updating. The hydrograph obtained using SLM was achieved with an 88% reduction in total model calls relative to its LM counterpart, also shown. Based on thirty trials, the SLM implementation in PEST (Doherty, 2004, 2007a,b), using the recommended input control file value for employment of Broyden updates, achieved a 32% reduction in total model calls relative to its conventional LM implementation.

Four variations of our implementation of SLM were considered with the ten parameter FASST state-of-the-ground model, and all four included cyclic updating of zero, one, or five columns of the model sensitivity matrix at each optimization iteration. The case that did not include cyclic updating evaluated both the option to and not to include a full update of the model sensitivity matrix at a pre-specified point during the inversion process. The single SLM case that did include full updating of the model sensitivity matrix activated this option when the ratio of the new and old objective function values was greater than one. A reduction of 83%, 71%, and 36% in the mean number of total model calls was achieved for the ten parameter FASST state-of-the-ground model when employing cyclic updating with zero, one, or five columns of the model sensitivity matrix at each optimization iteration. No cyclic updating together with activation of a full update as noted above resulted in a 47% reduction in total model calls. We expected to see the mean of the final objective function for the SLM case that did include a full update to be closer to the mean obtained from the thirty trials for the conventional LM case. Two possible reasons for the observed difference with our expectations could be objective function granularity and competing interests with respect to inversion stopping criteria and activation criteria for fully updating **X**. That is, full updating may have been activated too late in the inversion process. The SLM implementation in PEST (Doherty, 2004, 2007a,b), using the recommended input control file value for employment of Broyden updates, resulted in a 39% increase in total model calls relative to conventional LM.

Table 6 summarizes the two thirty trial numerical experiments that were conducted with the sixteen parameter GSSHA hydrologic model using our independent implementations of LM and SLM. The single SLM variation considered did not include a cyclic or full update of the model sensitivity matrix. The mean number of total model calls was reduced by 84% relative to the conventional LM application. Fig. 3 is a plot of the hydrographs associated with the sixth of the thirty trials for LM and SLM, which achieved final

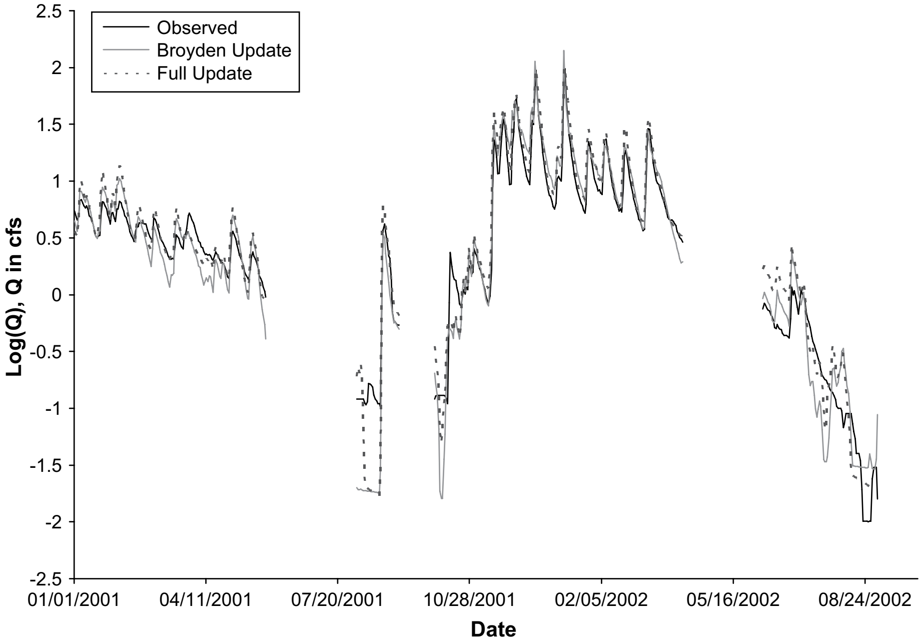


Fig. 2. Observed and simulated HSPF hydrographs for trial seven of thirty trials.

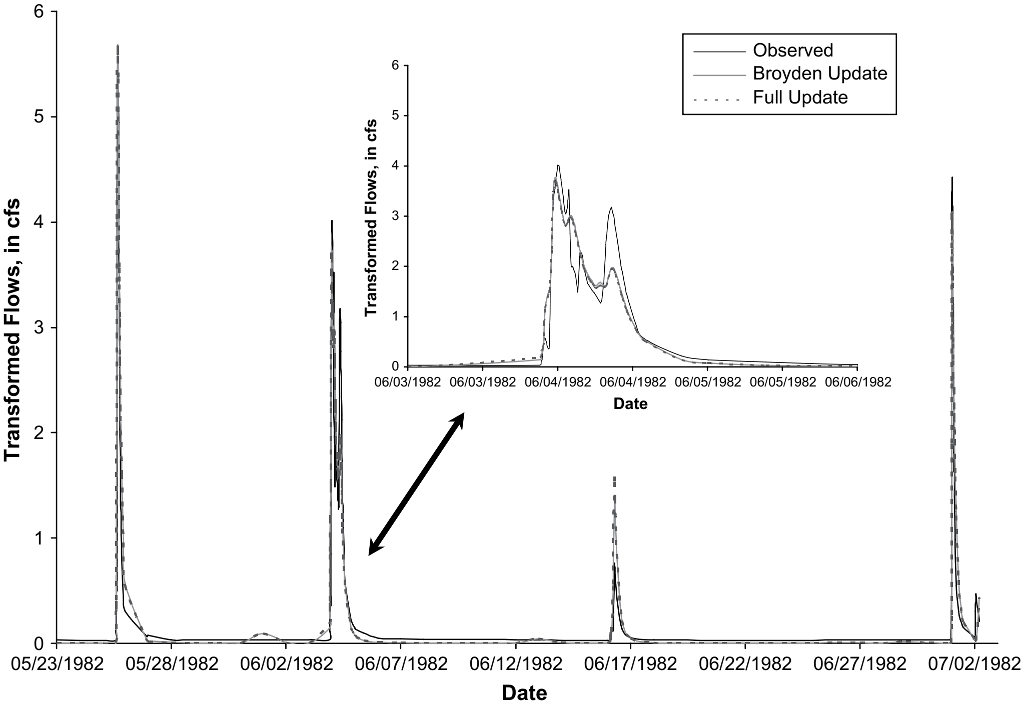


Fig. 3. Observed and simulated GSSHA hydrographs for trial six of thirty.

objective function values of 65.8 and 67.6, respectively. These final objective function values that were obtained for the sixth inversion runs for LM and SLM are comparable to the means obtained from all thirty trials. The SLM implementation in PEST (Doherty, 2004, 2007a,b), using the recommended input control file value for employment of Broyden updates, achieved a mean value of 608 for total model calls, which compares to a mean value of 421 when conventional LM was applied.

These case studies show that it is possible to find locally optimal parameter sets for model calibration at greatly reduced cost as compared to the standard LM method using full finite-difference updates of the Jacobian matrix. The Secant LM method, or SLM, achieved model run savings of anywhere from 36% to 84% with little or no difference in the value of the objective function at the local minimum. In Section 5 we examine global search strategies.

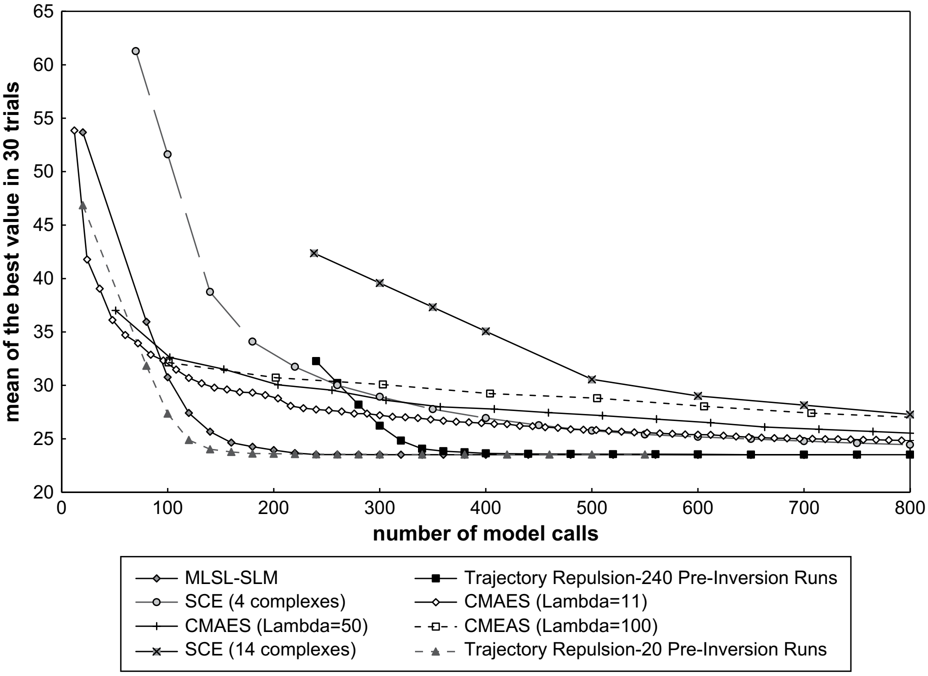


Fig. 4. Comparison of global optimization methods on the HSPF model. Note: nominal algorithm performance for CMAES and SCE.

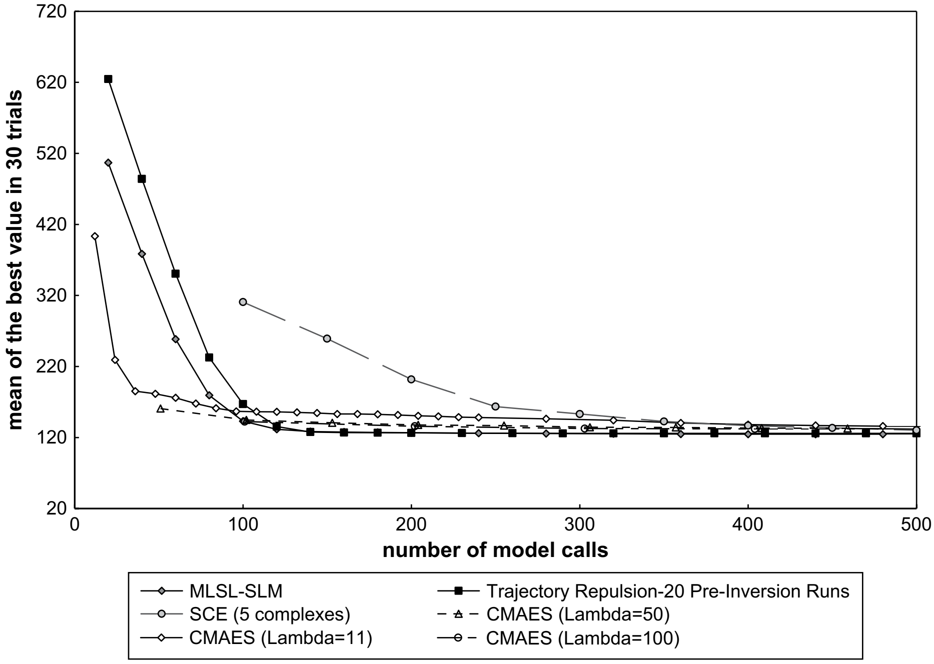


Fig. 5. Comparison of global optimization methods on the FASST model. Note: nominal algorithm performance for CMAES and SCE.

5. Efficiency of MLSL versus global optimization in PEST

To examine their relative efficiencies, thirty trials of MLSL, Trajectory Repulsion, CMAES, and SCE were conducted with each of the three distinct environmental model structures. MLSL and Trajectory Repulsion as implemented in MICUT were employed in the analysis. While PEST includes a Trajectory Repulsion implementation, the Trajectory Repulsion implementation from MICUT was utilized so that the local search implementation (SLM) would be identical to that employed with MLSL. CMAES and SCE were applied as interfaced to PEST (Doherty, 2004, 2007a,b). The processed data contained in Figs. 4–6 and Tables 7–9 were obtained by computing the average over thirty trials of the best objective function value obtained after a specified number of model calls. We wish to point out that we are very wary of comparing different software packages in this manner, for a package, and the methodology which it encapsulates, always performs best when operated by its designers. This is because program settings, particularly those pertaining to termination and convergence criteria, can have a huge effect on the performance of a method; a non-expert in the use of a particular package may not be aware of the optimal settings to use, especially in difficult cases. Hence, we do not pretend that the

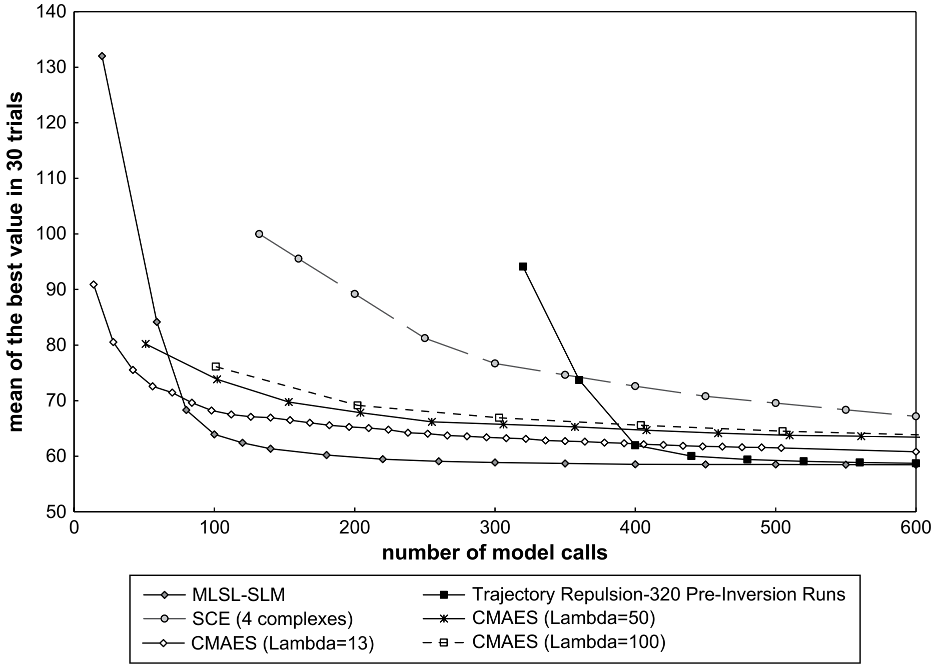


Fig. 6. Comparison of global optimization methods on the GSSHA model. Note: nominal algorithm performance for CMAES and SCE.

Table 7
Mean and length of one side of the 95% confidence interval for the best solution for the HSPF model for a specified number of model calls. Results based on thirty trials. Note: nominal algorithm performance for CMAES and SCE.

Algorithm	250 simulations		500 simulations	
	Mean	95% confidence interval	Mean	95% confidence interval
MLSL–SLM	23.533	0.033	23.510	0.005
Trajectory Repulsion – 240 pre-inversion runs	31.246	1.242	23.571	0.100
Trajectory Repulsion – 20 pre-inversion runs	23.561	0.040	23.532	0.034
SCE (4 complexes)	30.455	1.540	25.768	0.802
CMAES ($\lambda = 11$)	27.678	0.639	25.833	0.457
SCE (14 complexes)	41.829	3.274	30.571	1.185

Table 8
Mean and length of one side of the 95% confidence interval for the best solution for the FASST model for a specified number of model calls. Results based on thirty trials. Note: nominal algorithm performance for CMAES and SCE.

Algorithm	100 simulations		300 simulations	
	Mean	95% confidence interval	Mean	95% confidence interval
MLSL–SLM	143.077	11.046	125.494	1.075
Trajectory Repulsion – 20 pre-inversion runs	167.102	24.930	125.892	0.717
CMAES ($\lambda = 11$)	156.732	8.714	145.275	7.607
SCE (5 complexes)	310.571	31.220	153.054	7.957

results presented below provide a comprehensive basis for assessment of the comparative performance of SCE–PEST, CMAES–PEST and MLSL/MICUT. We hope, however, that they do provide a basis for at least a “ball park” comparison of the methods for the particular calibration cases considered herein.

For the eight parameter HSPF hydrologic model, and for the MLSL and Trajectory Repulsion global search methods, the local search procedure was our SLM implementation with two column cyclic updating specified at each optimization iteration. The option to fully update the model sensitivity matrix at a pre-specified point in the inversion process was deactivated. For the HSPF model, two sets of thirty trials were conducted using Trajectory Repulsion, one with 240 pre-inversion runs and one with twenty pre-inversion runs. Two sets of thirty trials were also performed using SCE, one with four complexes and one with fourteen complexes. For the MLSL application, the sample size, N, was specified to be twenty, the parameter for determining the reduced sample set, γ, was set at 0.1, and σ was set to four. Because CMAES is sensitive to the population size, λ (Hansen and Kern, 2004), three sets of thirty trials were performed using CMAES, one with a PEST computed default value of eleven for the population size, and then two additional trials with the population specified to be fifty and one hundred, respectively. In each case for CMAES, the number of parents was specified to be equal to one half the population size. For the MLSL and Trajectory Repulsion trials, all inversion runs stopped because three local searches had been performed with no objective function improvement; wherein the improvement fraction judged to be negligible was specified to be 0.0025. For the SCE runs with 4/14 complexes, the maximum number of trials allowed for each optimization run was specified to be 50,000/25,000, and the number of shuffling loops over which the objective function must improve by the specified percentage of 0.5, or else the optimization will be terminated, was set to five. For each of the thirty CMAES global optimization runs, the following stopping criteria were specified: (1a) Relative objective function improvement: 0.001, (1b) Number of iteration to which this applies: 40; (2a) Maximum relative parameter change: 0.001, (2b) Number of iterations to which this applies: 40; (3a) Relative generated objective function difference: 0.0025, (3b) Number of iterations to which this applies: 3; (4) Maximum number of iterations: 400. With CMAES, some of the global optimization trials were prematurely terminated, but only after the optimization process had progressed to a point wherein the known global min. objective function value (Skahill and Doherty, 2006) was in fact identified.

Fig. 4 and Table 7 summarize the results for the thirty trials associated with each of the four global optimization methods that were applied to the eight parameter HSPF model. CMAES, with the population size equal to eleven, performs the best with respect to efficiently reducing the objective function at the earliest stages of the optimization; however, it quickly plateaus. And after 1800 model calls, the mean of the best objective function value is 24.308, which is above the global objective function minimum of 23.5 (Skahill and Doherty, 2006). In contrast, both MLSL and Trajectory Repulsion with 20 pre-inversion runs have effectively identified the global minimum objective function value after 250 and 500 model calls, respectively. SCE with 14 complexes was inefficient; however, after approximately 3800 model calls the mean of the best objective function value had identified, with 95% confidence, the global minimum objective function value. Naturally, SCE with 4 complexes was more efficient but less effective at finding the global minimum objective function value (Duan et al., 1994).

Fig. 4 shows that Trajectory Repulsion with twenty pre-inversion runs appears to be the most efficient global search method among those tested, but it should be noted that this is not how Trajectory Repulsion would typically be applied. Trajectory Repulsion is not iterative, so its success is dependent on the initial sample size. A more typical application of Trajectory Repulsion would utilize twenty to fifty times the number of parameters as the

Table 9
Mean and length of one side of the 95% confidence interval for the best solution for the GSSHA model for a specified number of model calls. Results based on thirty trials. Note: nominal algorithm performance for CMAES and SCE.

Algorithm	350 simulations		600 simulations	
	Mean	95% confidence interval	Mean	95% confidence interval
MLSL–SLM	58.693	0.247	58.455	0.239
Trajectory Repulsion – 320 pre-inversion runs	78.819	2.711	58.732	0.165
CMAES ($\lambda = 13$)	62.720	0.984	60.827	0.817
SCE (5 complexes)	74.647	1.289	67.197	1.401

Table 10
List of acronyms.

	Acronym	Brief definition
AMALGAM	A Multi-Algorithm Genetically Adaptive Multiobjective	Optimization method
CMAES	Covariance Matrix Adaption Evolutionary Strategy	Optimization method
FASST	Fast All-season Soil Strength	State-of-the-ground model
GSSHA	Gridded Surface Subsurface Hydrologic Analysis	Watershed model
HSPF	Hydrological Simulation Program – FORTRAN	Watershed model
LM	Levenberg–Marquardt method	Derivative-based local search method
MICUT	Model Independent Calibration and Uncertainty Analysis Toolbox	Optimization and uncertainty analysis software
MLSL	Multi-Level Single Linkage	Stochastic global optimization method
PEST	Model Independent Parameter Estimation	Optimization and uncertainty analysis software
SCE	Shuffled Complex Evolution	Global optimization method
SLM	Secant Levenberg–Marquardt method	Derivative-based local search method

initial sample size. Such an example, with 240 pre-inversion runs are shown in Fig. 4, where it can be seen to not be as efficient as MLSL. If a small initial sample size is used with Trajectory Repulsion, it may fail to find the global minimum. To demonstrate this, MLSL and Trajectory Repulsion were both applied to the data fitting problem $\sum_{i=0}^{86}((\sin(m_1x_i)+\sin(m_2x_i))-(\sin(2/3)x_i+\sin(x_i)))^2$ where $x_i = 3.1+0.15i$, and $m_1, m_2 \in (0,2]$. MLSL was applied with N, γ, and σ equal to 10, 0.1, and 4, respectively, and Trajectory Repulsion was applied with ten pre-inversion runs. Based on thirty trials, similar efficiencies were exhibited for both MLSL and Trajectory Repulsion; however, Trajectory Repulsion failed to find the global minimum objective function value four times; whereas, MLSL was successful for all thirty trials. The iterative nature of the global phase for MLSL provides more insurance with respect to its reliability relative to Trajectory Repulsion.

For the ten parameter FASST model, both MLSL and Trajectory Repulsion used SLM for local search with one column cyclic updating specified at each optimization iteration. The option to fully update the model sensitivity matrix at a pre-specified point in the inversion process was deactivated. Input for application of MLSL and CMAES was the same as that previously described for the HSPF model. With MLSL and for all but one trial of global search with Trajectory Repulsion, all inversion runs stopped because three local searches had been performed with no objective function improvement; wherein the improvement fraction judged to be negligible was specified to be 0.0025. In one trial of Trajectory Repulsion, the global search was stopped because the maximum specified number, ten, of local searches had been performed. For SCE, the maximum number of function evaluations was set to 1500, and all thirty trials stopped due to this criterion. The same stopping criterion listed above for CMAES with HSPF were also applied with FASST, with the exception that the maximum number of iterations was set to 1500.

Fig. 5 and Table 8 summarize the results for the thirty trials associated with each of the four global optimization methods that were applied to the ten parameter FASST model. As with HSPF, CMAES performed the best with respect to reducing the objective function at the earliest stages of the optimization; however, its performance quickly plateaus. For example, with CMAES, at approximately 1800 model calls, with 95% confidence, the mean of the best value is approximately equal to that which was obtained using MLSL–SLM with only 300 model calls. SCE is inefficient relative to both MLSL and Trajectory Repulsion. If a predetermined computational budget of five hundred model calls were to be specified, then the data processed from the thirty trials suggest that MLSL–SLM would be the best choice relative to the other three methods.

The global search comparisons for the thirty trials of the sixteen parameter GSSHA model are similar and are summarized in Fig. 6 and Table 9. Again, CMAES is more efficient at reducing the objective function in the initial phases, but MLSL with SLM typically outperforms CMAES after only eighty model runs.

6. Conclusions

This article has described an independent implementation of the LM method included in MICUT, which accommodates the model independent PEST protocol (Doherty, 2004, 2007a,b) which is now widely used for environmental model calibration. Because environmental models are often very costly to run it is important to improve the efficiency of both local and global search methods used to calibrate these models. To improve the efficiency of the local searches, we implemented a secant version of LM (SLM) that typically requires far fewer model calls than the conventional LM method to achieve nearly the same objective function reduction. We also compared the efficiencies of our model independent LM and SLM implementations with related implementations in PEST (Doherty, 2004, 2007a,b). Our efficiency studies utilized three distinct environmental model structures: HSPF, FASST, and GSSHA. Based on our thirty trials with each of the three model structures, we found that we could find local minima using our SLM implementation with 36–84% fewer model runs than a conventional model independent LM application, and with only modest reductions in objective function improvement. In addition, we discovered that while PEST (Doherty, 2004, 2007a,b) reportedly does include the ability to utilize Broyden updates, that implementation does not realize the complete efficiency gains that are possible with a secant version of the LM method. For example, with the FASST and GSSHA model structures, the SLM implementation of PEST (Doherty, 2004, 2007a,b) required additional model calls. The results also suggest that implementing adaptive activation of cyclic updating could possibly achieve additional efficiency gains.

One of the strengths of the LM approach is that in cases of high parameter insensitivity and correlation the method can be readily modified by the inclusion of various regularization devices to maintain numerical stability and robustness; see for example Menke (1984), DeGroote-Hedlin and Constable (1990), and Tonkin and Doherty (2005). A further area to explore could be to examine the efficiency of SLM-based Tikhonov like regularization versus that of the hybrid method “SVD Assist” described in Tonkin and Doherty (2005), which was designed to support regularized inversion for highly parameterized environmental models.

In further consideration of the concern for efficient optimization, we also implemented a slight adaptation to MLSL, a stochastic global optimization method that utilizes our LM and SLM implementations for local search. We compared the performance of our implementations of MLSL and Trajectory Repulsion with those of CMAES and SCE, as interfaced to PEST (Doherty, 2004, 2007a,b). Based on numerical experiments involving thirty trials with each global optimization method, and for each of the three model structures, we recommend MLSL over Trajectory Repulsion for environmental model independent LM-based stochastic global optimization. Moreover, the results also suggest potential utilization of MLSL over SCE and CMAES, except for the case where only a very limited computational budget is available, in which case

CMAES might be preferable. The results further suggest exploring the development of additional model calibration methods that blend multiple optimization algorithms, perhaps in the spirit of the newly developed self-adaptive multi-method search algorithm AMALGAM (A Multi-Algorithm Genetically Adaptive Multi-objective) (Vrugt and Robinson, 2007). Clearly, it is almost certainly possible to tune SCE and CMAES to achieve better performance, but we are testing these algorithms as they are interfaced to PEST. The main parameter in CMAES is the population size, λ, and increasing it encourages global exploration at the expense of convergence speed. Our case studies bear this out. In SCE, there are other things that could be changed internally (Sorooshian et al., 2007), but as implemented through PEST the primary parameter is the number of complexes (Duan et al., 1994).

The methods that were described and evaluated herein, implemented in a manner to accommodate the popular PEST model independent and input control file protocol (Doherty, 2004, 2007a,b), fill in existing gaps (observed inefficiencies) associated with that software. Thus, the software described herein, available within MICUT, provides users of the noted popular PEST software with greater choice, and the potential for more efficient LM-based inverse model applications.

No single optimization algorithm can be expected to be the best for all problems. MICUT's LM-based optimization methods are a faster, compatible alternative to PEST's LM capabilities for calibration of environmental models. By adopting the popular PEST model independent interface, MICUT's LM-based optimization methods can easily be used by the modeling community for problems in which the derivatives are not too badly behaved and the objective function landscapes are not dominated by noise. There are many other algorithms that may be used in those circumstances. For users facing these complexities on a fixed computational budget, the Dynamically Dimensioned Search algorithm (Tolson and Shoemaker, 2007) can be utilized. Genetic algorithms or evolutionary strategies may be used, such as CMAES with a large population size (Hansen and Kern, 2004), or iteratively increasing population size (Auger and Hansen, 2005), or the recently developed AMALGAM (Vrugt and Robinson, 2007) which has been shown to be exceptionally competitive on a suite of test problems. Another algorithm which has recently been shown to have fast convergence in the early stages while being able to reliably locate the global minimum is a specially tuned version of SCE (Sorooshian et al., 2007). There are, of course, many optimization algorithms and we have only mentioned a few here. For a more complete review and comparison of some algorithms in the watershed modeling context see the recent work of Shoemaker et al. (2007).

Acknowledgments

Headquarters, US Army Corps of Engineers granted permission for publication of this paper. The authors gratefully acknowledge the constructive comments and suggestions of three anonymous reviewers which improved the quality of our paper. The second author gratefully acknowledges the support of ARO Grant W911NF-06-1-0306.

References

Alley, W.M., Veenhuis, J.E., 1983. Effective impervious area in urban runoff modeling. J. Hydraul. Eng. 109 (2), 313–319.

Auger, A., Hansen, N., 2005. A restart CMA evolution strategy with increasing population size. In: Proceedings of the IEEE Congress on Evolutionary Computation, CEC 2005, pp. 1769–1776.

Baginska, B., Milne-Home, W., Cornish, P.S., 2003. Modelling nutrient transport in Currency Creek, NSW with AnnAGNPS and PEST. Environ. Model. Softw. 18, 801–808.

Bicknell, B.R., Imhoff, J.C., Kittle, J.L., Jobes, T.H., Donigian, A.S., 2001. HSPF User's Manual. Aqua Terra Consultants, Mountain View, California.

Boender, C.G.E., 1984. The Generalized Multinomial Distribution: A Bayesian Analysis and Applications. Ph.D. Dissertation, Erasmus Universiteit Rotterdam.

Box, G.E.P., Jenkins, G.M., 1976. Time Series Analysis: Forecasting and Control. Holden-Day, San Francisco.

Boyle, D.P., Gupta, H.V., Sorooshian, S., 2000. Toward improved calibration of hydrologic models: combining the strengths of manual and automatic methods. Water Resour. Res. 36 (12), 3663–3674.

Broyden, C.G., 1965. A class of methods for solving nonlinear simultaneous equations. Math. Comput. 19, 577–593.

Cappelaere, B., Vieux, B.E., Peugeot, C., Maia, A., Séguis, L., 2003. Hydrologic process simulation of a semiarid, endoreic catchment in Sahelian West Niger. 2. Model calibration and uncertainty characterization. J. Hydrol. 279, 244–261.

DeGroote-Hedlin, C., Constable, S., 1990. Occam's inversion to generate smooth, two-dimensional models from magnetotelluric data. Geophysics 55 (12), 1613–1624.

Doherty, J., 2003. PEST Surface Water Utilities. Watermark Numerical Computing, Brisbane, Australia.

Doherty, J., Johnston, J.M., 2003. Methodologies for calibration and predictive analysis of a watershed model. J. Am. Water Resour. Assoc. 39 (2), 251–265.

Doherty, J., 2004. PEST: Model Independent Parameter Estimation. Fifth edition of user manual. Watermark Numerical Computing, Brisbane, Australia.

Doherty, J., 2007a. Addendum to the PEST Manual. Watermark Numerical Computing, Brisbane, Australia. January.

Doherty, J., 2007b. Addendum to the PEST Manual. Watermark Numerical Computing, Brisbane, Australia. August.

Doherty, J., Skahill, B.E., 2006. An advanced regularization methodology for use in watershed model calibration. J. Hydrol. 327, 564–577.

Downer, C.W., Ogden, F.L., 2003a. GSSHA User's Manual: Gridded Surface Subsurface Hydrologic Analysis, Version 1.43 for WMS 6.1. Technical Report, U.S. Army Corps of Eng., Eng. Res. and Dev. Cent., Vicksburg, Miss.

Downer, C.W., Ogden, F.L., 2003b. Prediction of runoff and soil moistures at the watershed scale: effects of model complexity and parameter assignment. Water Resour. Res. 39 (3), 1045.

Duan, Q.S., Sorooshian, S., Gupta, V.K., 1992. Effective and efficient global optimization for conceptual rainfall runoff models. Water Resour. Res. 28 (4), 1015–1031.

Duan, Q., Gupta, V.K., Sorooshian, S., 1993. A shuffled complex evolution approach for effective and efficient global minimization. J. Optim. Theory Appl. 76 (3), 501–521.

Duan, Q., Sorooshian, S., Gupta, V.K., 1994. Optimal use of the SCE-UA global optimization method for calibrating watershed models. J. Hydrol. (158), 265–284.

Fletcher, R., 1987. Practical Methods of Optimization, second ed. John Wiley, Chichester.

Frankenstein, S., Koenig, G., 2004. Fast All-season Soil STrength (FASST). U.S. Army Engineer Research and Development Center, Cold Regions Research and Engineering Laboratory, Hanover, New Hampshire, Special Report SR-04-1.

Gallagher, M., Doherty, J., 2007a. Predictive error analysis for a water resource management model. J. Hydrol. (334), 513–533.

Gallagher, M., Doherty, J., 2007b. Parameter estimation and uncertainty analysis for a watershed model. Environ. Model. Softw. 22, 1000–1020.

Gaucherel, C., Campillo, F., Misson, L., Guiot, J., Boreux, J.-J., 2008. Parameterization of a process-based tree-growth model: Comparison of optimization, MCMC and Particle Filtering algorithms. Environ. Model. Softw. 23, 1280–1288.

Goegebeur, M., Pauwels, V.R.N., 2007. Improvement of the PEST parameter estimation algorithm through extended Kalman filtering. J. Hydrol. (337), 436–451.

Gutiérrez-Magness, A.L., McCuen, R.H., 2005. Effect of flow proportions on HSPF model calibration accuracy. J. Hydrol. Eng. 10, 343–352.

Gupta, H.V., Sorooshian, S., Hogue, T.S., Boyle, D.P., 2003. Advances in automatic calibration of watershed models. In: Duan, Q., Gupta, H., Sorooshian, S., Rousseau, A., Turcotte (Eds.), Water Science and Application Series, vol. 6, pp. 197–211.

Hansen, N., Ostermeier, A., 2001. Completely derandomized self-adaptation in evolution strategies. Evol. Comput. 9, 159–195.

Hansen, N., Muller, S.D., Koumoutsakos, P., 2003. Reducing the time complexity of the derandomized evolution strategy with covariance matrix adaptation (CMA-ES). Evol. Comput. 9, 159–195.

Hansen, N., Kern, S., 2004. Evaluating the CMA evolution strategy on multimodal test functions. In: Eight International Conference on Parallel Problem Solving from Nature PPSN VIII, Proceedings. Springer, Berlin, pp. 282–291.

Haydon, S., Deletic, A., 2007. Sensitivity testing of a coupled *Escherichia coli* – hydrologic catchment model. J. Hydrol. (338), 161–173.

Iskra, I., Droste, R., 2007. Application of non-linear automatic optimization techniques for calibration of HSPF. Water Environ. Res. 79 (6), 647–659.

Kim, S.M., Benham, B.L., Brannan, K.M., Zeckoski, R.W., Doherty, J., 2007. Comparison of hydrologic calibration of HSPF using automatic and manual methods. Water Resour. Res. 43, W01402, doi:10.1029/2006WR004883.

Kuczera, G., 1990. Assessing hydrologic model nonlinearity using response surface plots. J. Hydrol. 118, 143–161.

Kunstmann, H., Krause, J., Mayr, S., 2006. Inverse distributed hydrological modelling of Alpine catchments. Hydrol. Earth Syst. Sci. 10 (3), 395–412.

Levenberg, K., 1944. A method for the solution of certain problems in least squares. Q. Appl. Math. 2, 164–168.

Madsen, H., 2000. Automatic calibration of a conceptual rainfall-runoff model using multiple objectives. J. Hydrol. 235, 276–288.

Madsen, K., Nielsen, H.B., Tingleff, O., 2004. Methods for Non-Linear Least Squares Problems. Informatics and Mathematical Modelling. Technical University of Denmark.

Maneta, M.P., Pasternack, G.B., Wallender, W.W., Jetten, V., Schnabel, S., 2007. Temporal instability of parameters in an event-based distributed hydrologic model applied to a small semiarid catchment. J. Hydrol. 341, 207–221.

Marquardt, D., 1963. An algorithm for least-squares estimation of non-linear parameters. SIAM J. Appl. Math. 11, 431–441.

Menke, W., 1984. Geophysical Data Analysis. Academic Press Inc., Orlando, Florida.

Misirli, F., Gupta, H.V., Sorooshian, S., Thiemann, M., 2003. Bayesian recursive estimation of parameter and output uncertainty for watershed models. In: Duan, Q., Gupta, H.V., Sorooshian, S. (Eds.), Calibration of 20 Watershed Models. American Geophysical Union, Washington, DC, pp. 113–124.

Moore, C., Doherty, J., 2005. The role of the calibration process in reducing model predictive error. Water Resour. Res. 41 (5), W05050.

Moore, C., Doherty, J., 2006. The cost of uniqueness in groundwater model calibration. Adv. Water Resour. 29 (4), 605–623.

Rinnooy Kan, A.H.G., Timmer, G., 1987a. Stochastic global optimization methods, part I: clustering methods. Math. Program. 39, 27–56.

Rinnooy Kan, A.H.G., Timmer, G., 1987b. Stochastic global optimization methods, part ii: multi level methods. Math. Program. 39, 57–78.

Rode, M., Suhr, U., Wriedt, G., 2007. Multi-objective calibration of a river water quality model – information content of calibration data. Ecol. Model. 204, 129–142.

Rose, K.A., Megrey, B.A., Werner, F.E., Ware, D.M., 2007. Calibration of the NEMURO nutrient–phytoplankton–zooplankton food web model to a coastal ecosystem: evaluation of an automated calibration approach. Ecol. Model. (202), 38–51.

Santanello Jr., J.A., Peters-Lidard, C.D., Garcia, M.E., Mocko, D.M., Tischler, M.A., Moran, M.S., Thoma, D.P., 2007. Using remotely-sensed estimates of soil moisture to infer soil texture and hydraulic properties across a semi-arid watershed. Remote Sens. Environ. (110), 79–97.

Senarath, S.U.S., Odgen, F.L., Downer, C.W., Sharif, H.O., 2000. On the calibration and verification of two-dimensional, distributed, Hortonian, continuous watershed models. Water Resour. Res. 36 (6), 1495–1510.

Shoemaker, C.A., Regis, R.G., Fleming, R.C., 2007. Watershed calibration using multistart local optimization and evolutionary optimization with radial basis function approximation. Hydrol. Sci. – J. 52 (3) Special issue: Hydroinformatics.

Shrestha, R.R., Rode, M., 2008. Multi-objective calibration and fuzzy preference selection of a distributed hydrological model. Environ. Model. Softw. 23, 1384–1395.

Skahill, B.E., Baggett, J.S., 2006. More efficient derivative-based watershed model calibration. Eos Trans. AGU 87 (52) Fall Meet. Suppl., Abstract 1521.

Skahill, B., Doherty, J., 2006. Efficient accommodation of local minima in watershed model calibration. J. Hydrol. (329), 122–139.

Sorooshian, S., Arfi, F., 1982. Response surface parameter sensitivity analysis methods for postcalibration studies. Water Resour. Res. 18 (5), 1531–1538.

Sorooshian, S., Behrangi, A., Khakbaz, B., Vrugt, J.A., Duan, Q., 2007. Global optimization with a limited budget of function evaluations. Eos Trans. AGU 88 (52) Fall Meet. Suppl., Abstract H14A-03.

Tolson, B.A., Shoemaker, C.A., 2007. Dynamically dimensioned search algorithm for computationally efficient watershed model calibration. Water Resour. Res. 43, W01413, doi:10.1029/2005WR004723.

Tonkin, M., Doherty, J., 2005. A hybrid regularised inversion methodology for highly parameterised models. Water Resour. Res. 41, W10412, doi:10.1029/2005WR003995.

Törn, A., Žilinskas, A., 1987. Global optimization. In: Lecture Notes in Computer Science, vol. 350. Springer-Verlag.

USEPA, 2000. BASINS Technical Note 6: Estimating Hydrology and Hydraulic Parameters for HSPF. EPA-823-R-00-012.

Vrugt, J.A., Robinson, B.A., 2007. Improved evolutionary optimization from genetically adaptive multimethod search. Proc. Natl. Acad. Sci. USA 104, 708–711, doi:10.1073/pnas.0610471104.

Yapo, P.O., Gupta, H.V., Sorooshian, S., 1996. Calibration of conceptual rainfall-runoff models: sensitivity to calibration data. J. Hydrol. 181, 23–48.

Zyvoloskia, G., Kwicklis, E., Eddebbarh, A.A., Arnold, B., Faunt, C., Robinson, B.A., 2003. The site-scale saturated zone flow model for Yucca Mountain: calibration of different conceptual models and their impact on flow paths. J. Contam. Hydrol. 62–63, 731–750.