Journal of Hydrology (2006) **329**, 122–139

available at www.sciencedirect.com

ScienceDirect

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

Efficient accommodation of local minima in watershed model calibration

Brian E. Skahill [a,*], **John Doherty** [b]

[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 Civil Engineering, University of Queensland, St. Lucia, Australia 4068, Watermark Numerical Computing, Brisbane, Australia*

Received 8 February 2005; received in revised form 23 January 2006; accepted 2 February 2006

KEYWORDS
Calibration;
Local minima;
Parameter estimation;
Objective function;
Watershed modeling

Summary The Gauss–Marquardt–Levenberg (GML) method of computer-based parameter estimation, in common with other gradient-based approaches, suffers from the drawback that it may become trapped in local objective function minima, and thus report ''optimized'' parameter values that are not, in fact, optimized at all. This can seriously degrade its utility in the calibration of watershed models where local optima abound. Nevertheless, the method also has advantages, chief among these being its model-run efficiency, and its ability to report useful information on parameter sensitivities and covariances as a by-product of its use. It is also easily adapted to maintain this efficiency in the face of potential numerical problems (that adversely affect all parameter estimation methodologies) caused by parameter insensitivity and/or parameter correlation.

The present paper presents two algorithmic enhancements to the GML method that retain its strengths, but which overcome its weaknesses in the face of local optima. Using the first of these methods an ''intelligent search'' for better parameter sets is conducted in parameter subspaces of decreasing dimensionality when progress of the parameter estimation process is slowed either by numerical instability incurred through problem ill-posedness, or when a local objective function minimum is encountered. The second methodology minimizes the chance of successive GML parameter estimation runs finding the same objective function minimum by starting successive runs at points that are maximally removed from previous parameter trajectories. As well as enhancing the ability of a GML-based method to find the global objective function minimum, the latter technique can also be used to find the locations of many non-global optima (should they exist) in parameter space. This can provide a useful means of inquiring into the well-posedness of a parameter estimation problem, and for detecting the presence of bimodal parameter and predictive probability distributions.

* Corresponding author. Tel.: +1 601 634 3441; fax: +1 601 634 4208.
E-mail address: brian.e.skahill@erdc.usace.army.mil (B.E. Skahill).

0022-1694/$ - see front matter
doi:10.1016/j.jhydrol.2006.02.005

The new methodologies are demonstrated by calibrating a Hydrological Simulation Program-FORTRAN (HSPF) model against a time series of daily flows. Comparison with the SCE-UA method in this calibration context demonstrates a high level of comparative model run efficiency for the new method.

Introduction

Computer-based calibration of surface water quantity and quality models generally involves minimization of an ''objective function'' — a measure of model-to-measurement misfit. In simple cases this is comprised of differences between measured and modeled flows at, for example, daily, hourly or even smaller intervals. In many cases, observed and modeled flows are transformed (for example through a Box-Cox transformation) before fitting, and/or residuals are fitted to an ARMA model prior to formulation of an objective function, in order to reduce heteroscedascity and temporal correlation (Box and Tiao, 1973; Box and Jenkins, 1976; Kuczera, 1983; Bates and Campbell, 2001). 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 (Madsen, 2000; Boyle et al., 2000; Doherty and Johnston, 2003).

A unique solution to the inverse problem of model calibration can only be guaranteed if the information content of a calibration dataset is sufficient to allow values to be assigned to all parameters for which estimation is sought through the calibration process. Often this is ensured by adherence to the so-called ''principle of parsimony'' in design of the inverse problem, whereby the number of parameters for which estimated values are sought is kept to a minimum while at the same time retaining enough parameters to allow a satisfactory fit between model outputs and field data to be achieved (Hill, 1998). It is often recommended that, prior to model calibration, a sensitivity analysis be conducted to identify those parameters that are estimable and those that are not; the latter are then fixed at realistic values while the ''identifiable'' parameters are estimated. Unfortunately however, especially where models are highly nonlinear, it is the parameter estimation process itself that is the final arbiter of parameter identifiability, for it is not always possible to select an appropriate subset of parameters for estimation ahead of actually undertaking the parameter estimation process. If too few parameters are selected for estimation, the calibration objective function will not be lowered to the extent that it possibly could be if other parameters were involved in the inversion process. However, in some cases the involvement of these extra parameters may lead to nonuniqueness in their estimation and, depending on the parameter estimation package employed, possibly poor performance of that package due to consequential numerical instability. Furthermore, even if the parameter estimation process is successful in minimizing the objective function under these circumstances, the final parameter set will lie within a long valley that defines the loci of objective function minima in parameter space. Should such a valley (rather than a bowl containing a unique minimum) exist, the parameter estimation package should notify the user of this, and of the fact that parameter estimates forthcoming from the calibration process are nonunique.

Whether or not an inverse problem is poorly posed, and whether or not the objective function minimum is elongate or round, it is rarely possible to avoid the fact that when calibrating watershed models the objective function will often contain local minima in addition to its global minimum; see Duan et al. (1992) for a full discussion of this topic. This presents challenges to the design of automatic calibration software, for a modeler who uses such software has the right to expect that estimated parameter sets result in the best possible fit between model outputs and field measurements (with due account taken of parameter believability). Ideally, however, a modeler would also like to receive some information from a calibration package 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. Indeed, information on the structure of the objective function surface can be of great assistance in allowing 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 dataset that is currently available for its calibration (Sorooshian and Arfi, 1982; Kuczera, 1990).

A further 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 whose task it is to calibrate many times in the course of minimizing the objective function that is used to characterize model-to-measurement misfit. Where model run times are high, model run efficiency of the calibration process becomes of paramount concern. It is inevitable that the challenges posed by parameter nonuniqueness and local objective function minima will lead to the necessity to carry out more model runs than that required for solution of an inverse problem characterized by a convex objective function surface with a single minimum. However, if the cost of meeting these challenges is too high, a parameter estimation package may simply be unusable in many modeling contexts, despite what may be a high degree of numerical robustness.

Choice of parameter estimation package

Much has been written concerning the suitability of various parameter estimation strategies for calibration of watershed models; see for example Thyer et al. (1999),

Madsen et al. (2002), Gupta et al. (2003) and papers cited therein. In light of the above discussion, desirable features of a parameter estimation package include the following.

1. It must perform well in numerical contexts where parameter nonuniqueness prevails. Moreover, it must inform the user that the values estimated for at least some parameters in such contexts are very uncertain.
2. It must find the global minimum of the objective function, notwithstanding the existence of local minima. However, it should provide some information to the user on the existence and/or locations of other minima. In particular, if a local minimum exists within parameter space that is not much higher than the global minimum, but that yields more realistic parameter estimates, the user must be made aware of this, for the higher minimum may be preferable.
3. To be generally useable, it must perform these tasks using as small a number of model runs as possible.

In accommodating the local optima problem, optimization strategies based on global search algorithms have an obvious appeal. The shuffled complex evolution (SCE-UA) method developed by Duan et al. (1992) has gained a justifiably good reputation as an efficient global optimization method and, as a result of this, has found widespread use in the calibration of watershed and other types of models. Its capabilities have been extended by combining its global search engine with an adaptive Markov Chain Monte-Carlo (MCMC) algorithm (Vrugt et al., 2003); this extension of its capabilities allows exploration of parameter space in the vicinity of the global minimum. However, in spite of the numerical efficiencies of SCE-based algorithms when compared with other global search methodologies, the number of model runs required for their implementation can still make them impractical for use with many commonly-used models because of the run-times required by these models; furthermore, run-time penalties tend to increase dramatically as the number of parameters requiring estimation increases. Another consideration is that MCMC methods can experience difficulties in contexts of high parameter correlation. It has also been the experience of the authors that these methods can encounter difficulties in accommodating multi-modal probability distributions arising out of the existence of two or more objective function minima of about the same magnitude but widely separated in parameter space.

Gradient-based methods such as the Gauss Marquardt Levenberg (GML) method have been criticized for poor performance in the face of local optima (Gupta et al., 2003). Use of such methods can lead to the determination of a parameter set that corresponds to a local, rather than global, objective function minimum, leaving the user with no idea of whether another location exists within parameter space for which the objective function is lower. However, certain features of the GML method make it difficult to reject outright as a serious contender for use in watershed model calibration. These features include the following.

1. In calibration contexts where local optima are rare or nonexistent, the GML method is parsimonious in terms of its model run requirements.
2. Estimates of parameter uncertainty, correlation and (in)sensitivity are readily available as a by-product of its use both during and after the parameter estimation process.
3. 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), Tonkin and Doherty (2005), and many other references particularly from the geophysical literature. The model-independent parameter estimation package PEST (Doherty, 2005) provides both Tikhonov and truncated singular value decomposition techniques as regularization alternatives.
4. Various enhancements can be made to the GML method that allow it to carry out linear or nonlinear post-calibration predictive uncertainty analysis with model run efficiencies that far exceed those of MCMC methods (Vecchia and Cooley, 1987).

It follows that if a methodology can be found that retains the advantages of the GML method, while eradicating its propensity to be trapped in local optima, such a method would deserve serious consideration for use in watershed model calibration.

Brief overview of the GML method

Though used extensively in the calibration of nonlinear models, the theoretical underpinnings of the GML method have their roots in linear parameter estimation theory. That theory, and its extension to nonlinear parameter estimation, will now be briefly described.

Let the matrix $\mathbf{X}$ represent the action of a linear model. Let the vector $\mathbf{p}$ represent its m parameters, the vector $\mathbf{h}$ represent the n observations (or ''processed observations'' as discussed above) comprising the calibration dataset, and the n-dimensional vector ε represent the noise associated with those (processed) observations. The relationship between these quantities is embodied in the equation:

$$\mathbf{Xp} = \mathbf{h} + \varepsilon \tag{1}$$

Let an objective function Φ defined as:-

$$\Phi = (h - \mathbf{Xp})^t \mathbf{Q} (\mathbf{h} - \mathbf{Xp}) \tag{2}$$

serve as a measure of model-to-measurement misfit, where $\mathbf{Q}$ is proportional to the inverse of $C(\varepsilon)$, the covariance matrix of measurement noise. It is normally the purpose of measurement transformations discussed above to ensure that this is a diagonal matrix.

It can be shown that minimization of Φ is achieved for $\mathbf{p}$ calculated as

$$\mathbf{p} = (\mathbf{X}^t \mathbf{Q} \mathbf{X})^{-1} \mathbf{X}^t \mathbf{Q} \mathbf{h} \tag{3}$$

and that the uncertainty associated with these estimates of $\mathbf{p}$ can be characterized by the covariance matrix $C(\mathbf{p})$ calculated as:-

$$C(\mathbf{p}) = \sigma_r^2 (\mathbf{X}^t \mathbf{Q} \mathbf{X})^{-1} \tag{4}$$

where the ''reference variance'' σ_r^2 is calculated as

$$\sigma_r^2 = \Phi_{\min} / (n - m) \tag{5}$$

Eq. (4) can be used for the computation of parameter confidence intervals; where the sensitivities of a prediction to model parameters are known, a slight extension of this formula then allows calculation of predictive confidence intervals.

When the $\mathbf{X}^t\mathbf{Q}\mathbf{X}$ matrix of Eq. (3) cannot be inverted (this being a direct result of parameter nonuniqueness), a number of methods are available for nevertheless obtaining a solution to the inverse problem. One of the better known of these methods is truncated singular value decomposition (TSVD). This method allows estimates of orthogonal linear combinations of parameters spanning the ''estimable subspace'' of parameter space to be made, this subspace being of smaller dimensions than the number of parameters featured in the parameter estimation problem. The dimensionality of this subspace can be directly chosen by the user; alternatively the user can allow the parameter estimation package to choose its dimensionality itself on the basis that the post-truncation ratio of highest to lowest eigenvalue of the $\mathbf{X}^t\mathbf{Q}\mathbf{X}$ matrix (a measure of its condition number) be no greater than a specified amount (normally between 10^6 and 10^7). Parameter combinations comprising the inestimable ''calibration null space'' are left unaltered from their initial values. If parameters are scaled according to their innate variability, this leads to maximum likelihood estimates for both estimable and inestimable parameter combinations. See Moore and Doherty (2005) for details.

Where a model is nonlinear, implementation of Eq. (3) becomes an iterative process which commences from a user-supplied set of initial parameter estimates. Furthermore, the Jacobian matrix $\mathbf{J}$ (this being the sensitivity of every model output for which there is a corresponding element of $\mathbf{h}$ with respect to every adjustable parameter) replaces $\mathbf{X}$. The pre-inversion $\mathbf{J}^t\mathbf{Q}\mathbf{J}$ term of Eq. (3) is then supplemented by the addition of a ''Marquardt lambda'' to its diagonal elements. This increases efficiency in nonlinear contexts, and acts as a rudimentary numerical stabilization device where encroaching nonuniqueness raises the condition number of this matrix and thereby threatens its ability to be inverted. The nonlinear parameter estimation process then becomes one of successive linearization and calculation of a parameter upgrade vector using Eq. (3); the upgraded parameter set then marks the spot at which the next linearization takes place. The process is terminated when the objective function is minimally lowered over two or more successive iterations, and/or when parameter values undergo minimal change between iterations.

One pronounced advantage of the GML method is that it can generally complete a parameter estimation process with an extremely high level of model run efficiency, even if elements of the Jacobian matrix must be computed using finite differences based on model runs with incrementally varied parameter values. Indicators of parameter nonuniqueness resulting from ill-posedness of the parameter estimation problem are available through the calculated condition number of the $\mathbf{X}^t\mathbf{Q}\mathbf{X}$ matrix and from the $\mathbf{C}(\mathbf{p})$ matrix calculated through inversion of this matrix. Either of these matrices can also indicate which parameters are insensitive and/or correlated and are therefore responsible for the ill-posedness of a particular inverse problem. Such information is vital if the inverse problem is to be re-formulated in such a way as to make it numerically tractable.

As mentioned above, the chief disadvantage of the GML method is its propensity to find local minima rather than the global minimum. Thus, if there are different ''regions of attraction'' in parameter space, iterative solution of Eq. (3) will lead to just one of possibly many objective function minima, the particular one that is found on any particular implementation of the method being dependent on the user-supplied set of parameters from which the iterative solution process was started. Malperformance of the method is further exacerbated by the fact that local minima can occur as ''pits'' in the objective function surface, this possibly forestalling the method's ability to locate the objective function minimum that may dominate a particular region of parameter space, whether or not this is the global objective function minimum.

We now present two enhancements to the GML scheme which promote its ability to operate in parameter estimation contexts in which all of these phenomena occur, viz. parameter nonuniqueness leading to elongate objective function minima, the existence of different regions of attraction in parameter space surrounding separate objective function minima, and local ''pits'' within the objective function surface defining these regions of attraction. The ''trajectory repulsion'' scheme described below provides some assistance in overcoming the second of these problems. ''Temporary parameter immobilization'', also described below, is a useful tool for accommodating the first and third of these phenomena. (Existing regularization schemes such as TSVD can also easily accommodate the first of these problems.) Collectively, these enhancements allow successful and efficient use of the GML method in surface water modeling contexts where less robust implementations of the method have encountered severe difficulties in the past.

Parameter estimation package

In the current study GML-based parameter estimation was implemented using the PEST parameter estimation package (Doherty, 2005). PEST is a ''model-independent'' parameter estimator, its model-independence being based on the fact that it communicates with a model through the model's own input and output files. Hence, use of PEST with a particular model can be carried out without the requirement that the model be recast as a subroutine or undergo any other changes. In fact, the ''model'' calibrated by PEST can even be a series of programs encapsulated in a batch or script file.

Most models are not programmed to calculate derivatives of their outputs with respect to their parameters for filling of the Jacobian matrix required for implementation of the GML method. Hence, PEST calculates these derivatives itself using finite differences. During each iteration of the GML method, PEST varies each parameter incrementally from its currently estimated value and re-runs the model. The ratio of model output differences to parameter differences approximates the derivative. For greater accuracy in derivatives calculation, parameter values can be both incremented and decremented. Using this ''central difference'' scheme, derivatives can be approximated by taking differences between incremented and decremented

model outputs and parameter values, or by including current parameter values and corresponding model outputs in a three-point quadratic interpolation scheme. In normal PEST operation, derivatives are first calculated using forward differences only; an automatic change to a more run-expensive central difference scheme is made when progress of the parameter estimation process appears to slow later in a PEST run.

Implementation of a finite-difference derivatives scheme requires that a user select an appropriate parameter increment. If an increment is too large, the finite difference ratio through which derivatives are calculated is a poor approximation to the derivative. If the increment is too small, numerical precision is lost as numbers which are similar in magnitude are subtracted from each other. In many parameter estimation contexts an increment of one percent of current parameter values can be employed for forward-difference-based derivatives computation, while twice this can be used for central-difference-based derivatives computation.

The trajectory repulsion scheme

The robust performance of the SCE-UA method, as well as that of most other global search methods, is based on two principles. These are as follows.

1. The injection of a certain degree of randomness into the parameter estimation process allows it to go in directions that may eventually prove fruitful, even if the attractiveness of a new direction may be shielded by the promise of local, more immediate, rewards.
2. The benefits of randomness are partly offset by the cost of making mistakes. Hence, by incorporating into a global optimization process an ability to learn from mistakes, the likelihood of incurring large run-time penalties through repeatedly making the same (or a similar) mistake is minimized.

Based on these principles, an enhancement of the GML method was developed in order to increase the capacity of this method to work well in contexts where local minima occur. The software package that encapsulates this enhancement takes the form of a ''PEST driver'', in which GML parameter estimation is still conducted by PEST (and can thus employ one or more of PEST's inbuilt devices for numerical stabilization of difficult problems and, like PEST, is model-independent), but in which successive PEST runs are undertaken under intelligent control. The package is presently named ''PD_MS2'' (for ''*P*EST *D*river — *M*ultiple *S*tarting Points *2*'').

PD_MS2 commences execution by running the model that it must calibrate N times. PD_MS2 employs random parameter values for these runs; these are sampled from a uniform or log-uniform distribution defined between user-supplied upper and lower parameter bounds. Experience indicates that between 20 and 50 times the number of parameters requiring estimation is a suitable value for N.

PD_MS2 next ranks the outcomes of the N random runs in order of increasing objective function value. It then disregards all runs for which the objective function is above the median. Next, it initiates a PEST run, with initial values for this run being equal to the random parameter sample for which the objective function was lowest. PD_MS2 monitors this run, recording optimized parameter values, as well as parameter values calculated by PEST during every iteration of the nonlinear GML method which it implements. Normally, between five and fifteen such iterations are required to reach an objective function minimum. Each such iteration requires that at least as many model runs be undertaken as there are parameters requiring estimation (in order to fill the Jacobian matrix), plus a few more (for testing the effects of different Marquardt lambdas on the parameter upgrade process); see Doherty (2005) for details.

After completion of the first PEST run, another PEST run is initiated. For this run, it is desired that the chances of finding the same objective function minimum as that which was encountered on the first PEST run be minimized. Hence, from among the $N/2$ retained pre-calibration samples of parameter space, a starting point is chosen that is maximally distant from any point on the parameter trajectory taken by the initial PEST run. Selection of such a starting point is based on the rationale that the closer is a point in parameter space to the previous parameter trajectory, the more likely it is to lie in the ''catchment area'' of the previously-encountered objective function minimum. (Note that ''distance'' as measured in parameter space is a Euclidean metric, normalized in each direction of parameter space by the range of the pertinent parameter.)

After the next PEST run is complete, another parameter set is selected from the $N/2$ potential starting points. The parameter set selected is that which is maximally distant from all previous points on all previous trajectories. The process is then repeated.

A number of criteria can be used to terminate the PD_MS2 global optimization process. Where model run efficiency is an issue, PD_MS2 can be instructed to cease execution if the objective function has not been lowered over the last M_1 PEST runs. Alternatively, PD_MS2 can be asked to undertake M_2 PEST runs regardless of the outcomes of these runs. If M_2 is moderate to large, this enables PD_MS2 to find the locations of many local optima in parameter space (should these exist), thus providing the user with powerful insights into the structure of the objective function surface.

It is worth noting that, as well as providing insights into the ''broad scale'' structure of the objective function response surface, PD_MS2 (through its use of PEST) provides insights into the structure of this surface in the vicinity of the global objective function minimum as well. As has already been mentioned, the GML method can provide parameter sensitivities and can calculate a linear approximation to the parameter covariance matrix, as well as statistics derived from this matrix including correlation coefficients and eigenvectors/eigenvalues of the covariance matrix.

Temporary parameter immobilization

''Temporary parameter immobilization'' (or ''automatic user intervention'' as it is referred to in the PEST manual, but will be referred to as ''TPI'' herein) can be used as both a regularization device and as a device for conducting ordered attempts to break out of local pits in parameter

space. PEST only implements this scheme if the objective function improvement attained during a particular iteration of the GML process is less than a user-supplied threshold (normally 10%). In implementing this scheme PEST selects the most insensitive parameter, and temporarily removes it from the optimization process. With the dimensionality of estimable parameter space thus reduced (and with the most troublesome parameter being temporarily removed from the parameter estimation process), the parameter upgrade vector (which now has no component in the subspace of parameter space occupied by the temporarily frozen parameter) is re-calculated using Eq. (3). A model run is then conducted on the basis of the trial parameter set thus calculated in order to compute the objective function associated with this parameter set. Unless the objective function has fallen by a significant amount, the next most troublesome parameter is temporarily frozen (in addition to the first), and the parameter upgrade calculation procedure is repeated. After a number of parameters have been successively frozen in this manner (with already frozen parameters maintained in their frozen state), the process is abandoned, and then re-commenced using a different value of the Marquardt lambda. For a parameter estimation problem involving m parameters, up to half of these parameters may be progressively frozen for up to three Marquardt lambdas, this requiring $3m/2$ model runs for that iteration for the testing of parameter upgrade vectors in addition to the (depending on whether forward differences or central differences are employed) m or $2m$ model runs required for filling of the Jacobian matrix. (Note however that the process is immediately abandoned if a suitable objective function improvement is obtained.) Thus, implementation of the TPI process may lead to the requirement that between twice and three times (at the very most) the number of model runs be carried out compared to normal GML operations. However, experience has demonstrated that on most occasions in which the TPI method is employed about fifty percent extra model runs need to be carried out, and that this is generally a small price to pay for the benefits that it brings in terms of increased numerical stability in situations of parameter nonuniqueness, and for a dramatic reduction in the risk of becoming trapped in local objective function pits.

The decreased probability of ensnarement in local optima that attends use of the TPI scheme has its roots in a number of properties of this scheme. One obvious reason for a heightened probability of success in finding its way out of small regions of attraction of limited extent in parameter space is the sheer number of parameter upgrades that are attempted by this scheme, together with the fact that the directions pertaining to these upgrade attempts tend to be maximally different with respect to each other. This maximality of difference is a result of two factors. The first is the fact that the upgrade direction tends to be dominated by insensitive parameters where all parameters are involved in the computation of this direction; this is a direct result of the fact that, because of their insensitivity, the GML parameter estimation algorithm calculates that these parameters require larger movement than other parameters to affect the objective function. As dimensions of parameter space are progressively closed to the parameter upgrade vector through the temporary immobilisation of insensitive parameters, and new upgrade directions are accordingly computed in spaces of lower dimensions, these new directions will tend to be orthogonal to the original upgrade vector which was dominated by the now-omitted dimensions. The penchant for orthogonality is further increased as a result of the fact that the entire dimensionality reduction process is repeated for widely different Marquardt lambda values. As documented in works such as Bard (1974), computed upgrade directions can vary between that of steepest descent down the objective function surface when the Marquardt lambda is high, to a direction that can be almost orthogonal to this when the Marquardt lambda is low.

Another important factor behind the success of the TPI scheme is that it lowers the chances of upgraded parameters finding local optima in the first place. Unless objective function improvement during a particular iteration is acceptably large without the help of the TPI scheme (which often occurs in the early stages of the parameter estimation process), use of the TPI scheme requires that model runs be carried out specifically to test the ability of different upgrade vectors (often with very different directions as discussed above) to lower the objective function. The upgrade vector that results in the largest objective function decline is that which is selected as the basis for the next linearization of the inverse problem. Of all the upgrade vectors tested, this is the one least likely to lead to a local objective function minimum, for the encroachment of global or local optimality (for which derivatives of the objective function with respect to all model parameters is zero) is normally marked by smaller and smaller declines in the objective function per iteration as the GML method ensures that a parameter set is found from which all directions lead uphill. In fact, the more nonlinear is the problem, the less likely it is that a parameter upgrade vector resulting in a large objective function decline will lead directly to the bottom of an objective function minimum (due to the fact that the equations upon which this upgrade vector are calculated are based on an assumption whose inapplicability grows with increasing parameter movement, and/or increasing changes in model outputs on account of this movement).

An additional factor that contributes to the success of the TPI scheme in both avoidance of local minima of small lateral extent, and in extricating itself from such minima, is PEST's use of finite differences for parameter derivatives calculation. As was mentioned above, parameter increments of one percent are often employed for forward difference derivatives calculation and two percent for central difference derivatives calculation (these being settings that work well in many calibration contexts in which PEST is used with HSPF and other models). These increments are large enough for PEST to ''see'' outside of a small pit in which it may be currently trapped. Alternatively, if current parameter values lie just outside of a small pit, these increments are large enough for the effect of the pit to exert a smaller influence on calculated derivatives than would be the case if derivatives were exact. Thus, the use of finite-difference-based parameter derivatives provides a kind of filtering mechanism through which finer details of the objective function surface are prevented from concealing the broader features of that surface.

So, through a combination of the fact that many upgrade vectors are tested, that a parameter upgrade selection

procedure is adopted that minimizes the chances of being trapped in a local minimum in the first place, and maximizes the chances of escaping from that minimum if ensnarement does indeed occur, and because parameter upgrades possess some immunity to the effects of pits because their calculation is based on finite-difference derivatives rather than point derivatives, use of the TPI method in calibration of surface water models has consistently resulted in good PEST performance in estimating parameters for those models.

(Note that selection of a TPI activation threshold of 10% improvement in the objective function is somewhat arbitrary. However experience has demonstrated that this normally results in efficient implementation of the method. If the threshold is set too high, TPI-based parameter upgrade re-computation will be undertaken on most GML optimisation iterations, irrespective of proximity, or otherwise, to an objective function minimum. This can result in wasted model runs if rapid objective function improvement is taking place without the need for TPI upgrade repetitions. On the other hand, if the improvement threshold is set too low, then needless ''struggling'' of the GML method in the face of difficulties incurred through problem ill-posedness or proximity to a local minimum, resulting in only small improvements in the objective function in successive iterations, can be avoided.)

An example

Description

Use of the methodologies discussed in the preceding section are now demonstrated by applying them to the calibration of an HSPF (Bicknell et al., 2001) 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 8 HSPF parameters was undertaken by matching observed and simulated daily flows over four non-contiguous time intervals spanning the period 1st Jan 2001 to 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 dataset 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 paper is to demonstrate the capabilities of the methodologies discussed above in accommodating local minima. 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 Eq. (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 Doherty and Johnston (2003) for details. HSPF parameters other than those appearing in Table 1 were fixed at reasonable values.

Finding objective function minima

For the present calibration problem, the objective function has a value of 23.5 at its global minimum. Fig. 1 shows the fit between modeled and observed daily flows at this minimum.

In an attempt to locate as many local minima as possible, PD_MS2 was asked to run PEST 100 times from a succession of starting values that were maximally distant from all previous parameter trajectories, as discussed above. These starting values were selected from 300 random parameter samples for which objective functions were calculated prior to the undertaking of any PEST runs. For this initial PD_MS2 run PEST's TPI functionality was not activated. Instead, in order to guarantee numerical stability in the face of a problem that (as will be demonstrated shortly) is not well-posed, TSVD was employed as a regularization device; a maximum to minimum eigenvalue ratio of 10^7 was set as the truncation criterion, this resulting in a PEST-selected solution subspace of dimensionality between five and eight for most iterations of the parameter estimation process. In order to remove the possibility of PEST misinterpreting locally slow progress of the parameter estimation process as convergence of that

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}

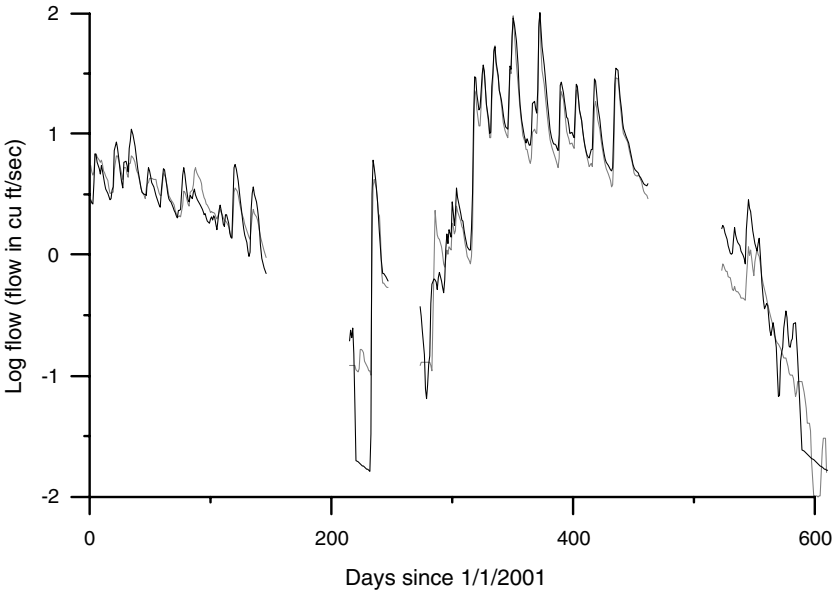

Figure 1 Observed (dark) and modeled (light) daily flows corresponding to global objective function minimum.

process to a global or local minimum, PEST convergence criteria were set unusually tight (this incurring a higher cost in terms of model runs than would be the case in normal PEST usage). Specifically, PEST did not terminate a calibration exercise unless it failed to lower the objective function by more than a relative amount of .0005 over four iterations.

Fig. 2 depicts the outcomes of this exercise. Each of the eight graphs appearing in this figure pertains to one of the eight estimated parameters. For each graph the objective function is plotted on the *x* axis, while an optimized parameter value is plotted on the *y* axis. For all but AGWRC, the *y* axis exactly spans the allowed range of the pertinent parameter. In each graph, each point represents the outcome of one PD_MS2-supervised PEST run; corresponding points from different graphs (representing corresponding values for different parameters) can be matched vertically through their common objective function. It is readily apparent from this figure that many of the outcomes of successive optimization runs are grouped into ''parameter clumps'' of nearly constant objective function value; these clumps define regions of attraction in parameter space. ''Tight'' clumps indicate a well-defined region of attraction; vertical spreading of clumps indicates difficulties in parameter identification through parameter correlation and/or insensitivity. For those minima situated at the bottom of broad objective function valleys defining different regions of attraction in parameter space, local minima are often in close proximity. Other local minima appear to exist in isolation from these more populous clumps.

The PD_MS2 run was repeated with TPI replacing TSVD as a device for numerical stabilization of the inverse problem and, as discussed above, as a device for escaping local pits in the objective function surface. The results are provided in Fig. 3. It is apparent that use of TPI significantly reduces the chances of the parameter estimation process terminating in an isolated pit in the objective function surface. However, this benefit comes at a cost. The 100 PEST runs whose outcomes are depicted in Fig. 3 required about 52000 model runs for completion; those whose outcomes are depicted in Fig. 2 required half this number. (Note that the number of model runs in each of these cases is larger than what they would be in normal modeling practice because of the tight convergence criteria employed by PEST as discussed above.)

Another PD_MS2 run involving 100 PEST runs (and with TPI implemented) was undertaken with the INTFW parameter fixed at a value of 3.4. Fig. 4 shows the results.

Impressions gained from an inspection of Figs. 2—4 are as follows.

1. Where 8 parameters are estimated, the global objective function minimum of 23.5 occurs in an elongate valley rather than a bowl. For some parameters (particularly INTFW, LZSN and UZSN) the parameter value at the global minimum cannot be defined. It is thus apparent that these parameters can be estimated only with a considerable degree of uncertainty on the basis of the dataset available for model calibration.
2. A minimum exists at an objective function value of 24.0, this being slightly above the global objective function minimum of 23.5. For some parameters (e.g. LZSN and AGWRC), values corresponding to this minimum are not too different from those corresponding to the global minimum; however for other parameters (for example IMP), values pertaining to this ''perched'' objective function valley are significantly different from those corresponding to the global objective function minimum. If, due to the fact that their values are so close, either of these two objective functions can be considered to calibrate the model, then some parameters (and model predictions that depend on them) will have bimodal probability distributions. To the extent that a prediction depends on IMP (impervious area), it may be calculated to have a significantly different value depending on which parameter set is selected as the set of ''calibrated parameters''.
3. Strong regions or attraction appear to exist around objective function minima of 26, 34 and 40.

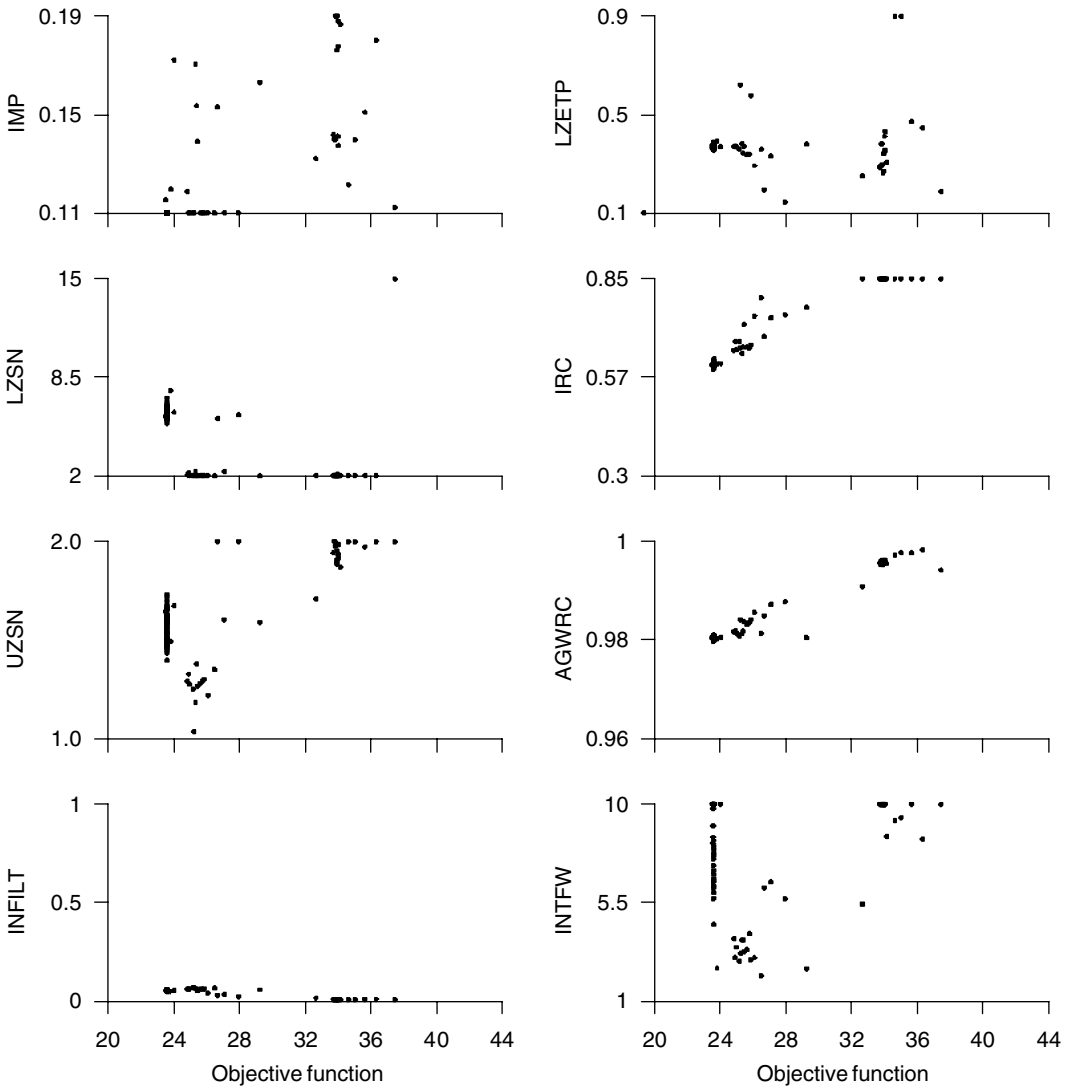

Figure 2 End-points in parameter space of 100 PEST runs undertaken under the control of PD_MS2. Parameters comprising an optimized set are linked vertically between graphs by objective function. PEST's TPI functionality was not operative.

4. Removal of the INTFW parameter from the parameter estimation process allows better estimates to be made of all other parameters. However, the discrete objective function minima at 23.5 and 24, with very different IMP values corresponding to each of these minima, still remain.

Ninety-five percent confidence intervals calculated by PEST for parameters corresponding to the global objective function minimum are provided in Table 2. It should be carefully noted that calculation of these confidence limits is based on a model linearity assumption that is grossly violated. Nevertheless, they are powerful indicators of the relative uncertainty of the different parameters for which estimation is attempted through the calibration process, and hence of any elongation of the response surface in the vicinity of the global objective function minimum. The limited ability of the calibration process to allow inference of the UZSN and INTFW parameters based on the current calibration dataset is readily apparent from this table. Further analysis of PEST outputs (including an inspection of composite parameter sensitivities and eigencomponents of the parameter covariance matrix) reveals that insensitivity of these parameters, rather than correlation with one or a number of other parameters, is the principal reason for difficulty of their estimation, though some correlation between UZSN and INTFW is indicated (correlation coefficient of 0.54); this, no doubt, is responsible for the decreased elongation of the UZSN clump in Fig. 4 when compared to that of Fig. 3 when INTFW is removed from the parameter estimation process.

To gain further understanding of the structure of the complex response surface associated with the current calibration problem, a ''traverse in parameter space'' was undertaken along a path comprised of four linear segments joining parameter sets situated within the various regions of attraction depicted in Fig. 3. The parameter sets comprising the beginnings/ends of these segments are marked as crosses in this figure; the path commences at the global objective function minimum and moves to progressively

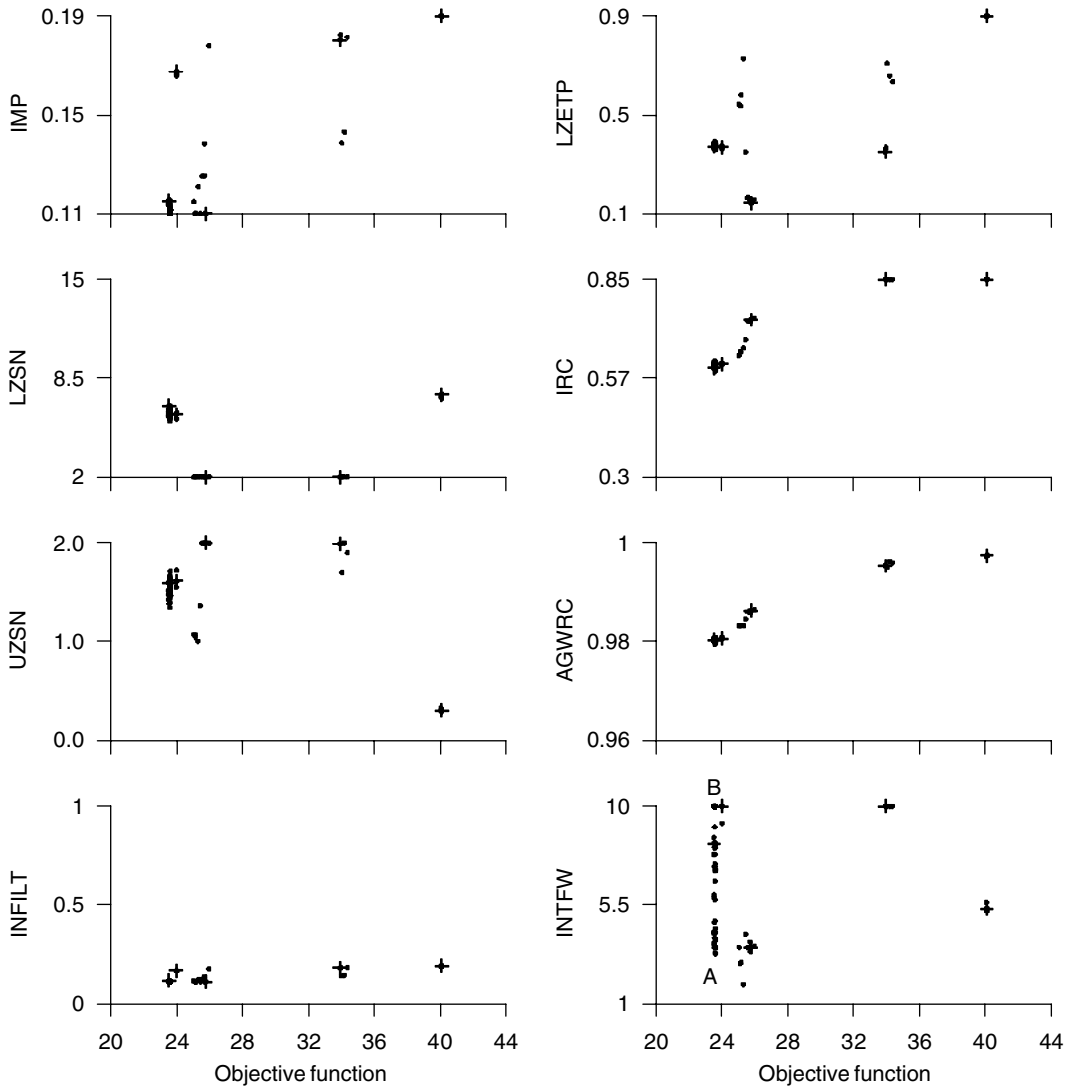

Figure 3 End-points in parameter space of 100 PEST runs undertaken under the control of PD_MS2. Parameters comprising an optimized set are linked vertically between graphs by objective function. PEST's TPI functionality was operative. Crosses mark the breakpoints of the objective function traverse depicted in Fig. 5.

higher minima. Thirty model runs were undertaken along each linear segment in equal increments along the segment. The profile is depicted in Fig. 5. In this figure, distance along the traverse (plotted on the horizontal axis) is measured from its start; the ''distance'' between two parameter sets is calculated as the square root of the sum of squared differences of parameters comprising each set, with each such difference normalized through division by the allowed parameter range.

Fig. 6 is another profile in parameter space comprised of 9 segments; however, in this case the objective function pertains to the 7-parameter inversion problem depicted in Fig. 4. The points which form the individual segment beginning and end points are shown as crosses in the first of the graphs comprising this figure. The irregular nature of the objective function surface in fine detail is obvious from this figure, as is the fact that, notwithstanding activation of PEST's TPI functionality, the trapping of some points in local optima was not completely avoided.

The nature of the objective function surface is explored in more detail in Fig. 7. In producing this figure UZSN and LZSN were varied about the objective function minimum while all other parameters were held at the values corresponding to this minimum. UZSN was varied between 1.2 and 1.7 in. in increments of 0.005 in., while LZSN was varied between 5 in. and 8 in. in increments of 0.025 in. For the sake of clarity, however, the values of these variables are normalized with respect to these ranges in Fig. 7a. The ''striated'' nature of the objective function surface (for which evidence is also available in Fig. 6) is readily apparent in this figure. The surface is shown contoured in Fig. 7b, its irregular geometry being also readily apparent from this figure.

A profile was plotted of the shape of the objective function surface along a transect joining points marked ''A'' and ''B'' in the last frame of Fig. 3. This profile is not reproduced in this paper as it is flat, demonstrating the fact that this linear feature is in fact a long valley in parameter space marking the elongate minimum of the objective function.

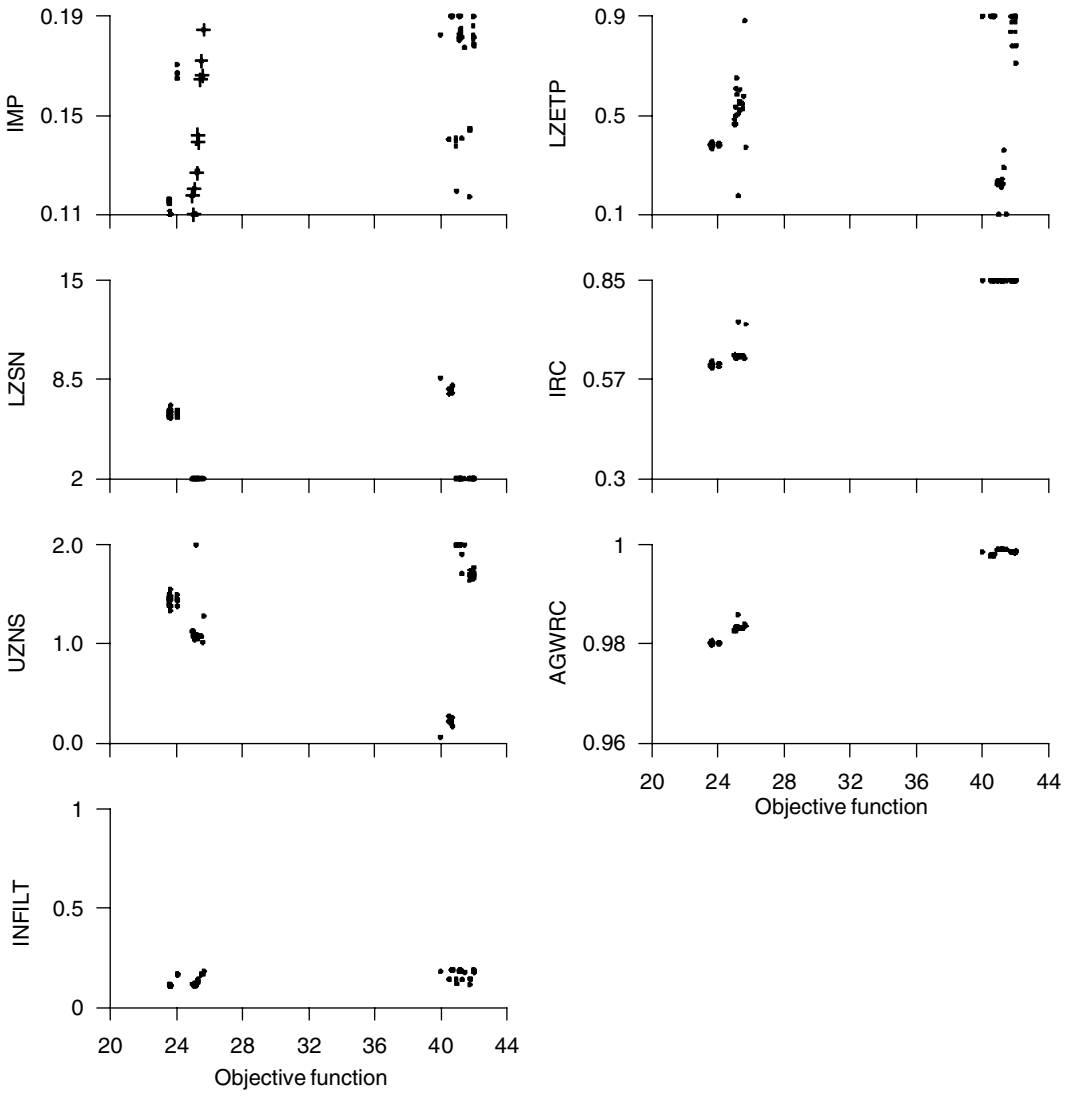

Figure 4 End-points in parameter space of 100 PEST runs undertaken under the control of PD_MS2 with INTFW fixed at 3.4. Parameters comprising an optimized set are linked vertically between graphs by objective function. PEST's TPI functionality was operative. Crosses (only shown for first frame of this figure) mark the breakpoints of the objective function traverse depicted in Fig. 6.

Table 2 Parameter values corresponding to global optimum; also shown are linear parameter confidence limits calculated as a by-product of the GML parameter estimation process

Parameter	Estimated value	Lower 95% confidence limit	Upper 95% confidence limit
IMP	0.116	8.21E − 02	0.164
LZSN	6.58	4.32	10.0
UZSN	1.58	0.978	2.54
INFILT	5.33E − 02	3.760E − 02	7.59E − 02
LZETP	0.378	0.318	0.448
INTFW	3.4	1.151E − 02	8687
IRC	0.605	0.530	0.691
AGWRC	0.98	0.977	0.983

Efficiency

In the previous subsection, PD_MS2 was deployed in a model-run-intensive exercise whose aim was to learn as much about the structure of the objective function surface as possible. During this exploration, use of the trajectory repulsion scheme lowered the chances of finding the same minimum twice (more on this follows), and therefore raised the chances of finding as many different local optima as possible on this surface within the 100 PEST runs allocated to this task. In normal PD_MS2 usage, however, a modeler will wish to find the global objective function minimum as quickly as possible. More than this, a user will wish not only to have found the global objective function minimum, but also to be certain that he/she has found the global objective function minimum. In achieving both of these aims, he/she will wish to waste as few model runs as possible.

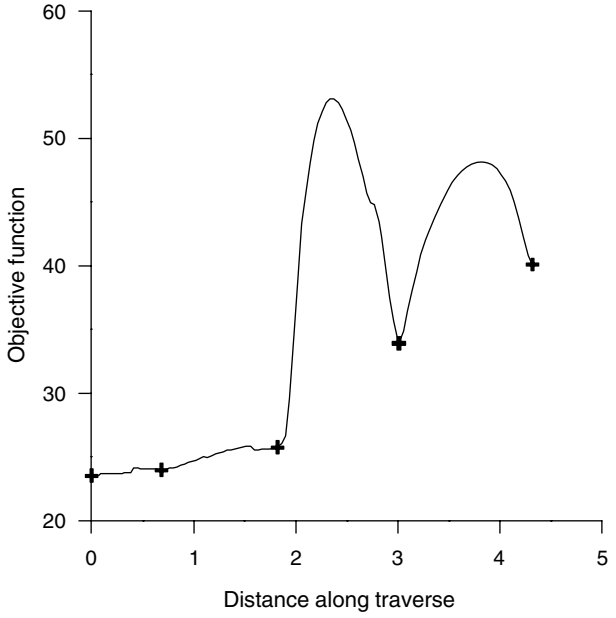

Figure 5 Objective function along traverse between parameter sets marked with crosses in Fig. 3. These same parameter sets are also shown as crosses in the transect.

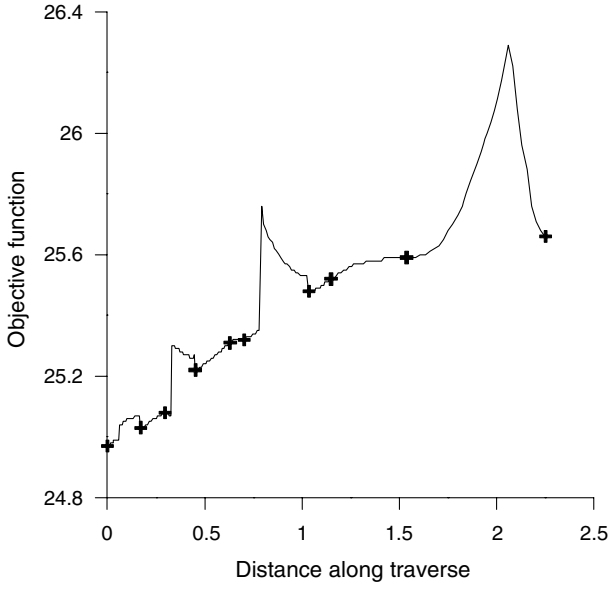

Figure 6 Objective function along traverse between parameter sets marked with crosses in the first frame of Fig. 4. These same parameter sets are also shown as crosses in the transect.

To test this aspect of PD_MS2's performance, a series of PD_MS2 runs was carried out in which pre-inversion random sampling of parameter space was undertaken using different seeds for the random number generator upon which selection of these samples is based. For each such PD_MS2 run, 240 pre-inversion model runs were carried out prior to commencement of the sequence of PEST runs supervised by PD_MS2. PD_MS2 was instructed to cease execution upon failure of three successive PEST runs to lower the objective function by a relative amount of .0025. Stable PEST operation in the face of potential numerical instability (particularly where 8 parameters were estimated) was ensured through activation of its TSVD functionality (with maximum to minimum eigenvalue ratio set at 10^7) rather than its TPI functionality. While, as is documented above, this increased the risk of encountering a local optimum (and thus had the potential to force PD_MS2 to undertake more PEST runs than it would otherwise have needed to find the global minimum of the objective function), it did make each PEST run considerably faster. As will be demonstrated below, this strategy appeared to work well in the present case. Note also that PEST was instructed to cease each of its parameter estimation runs if it failed to lower the objective function by a relative amount of 0.001 over 3 successive iterations.

Thirty such PD_MS2 runs were undertaken for this efficiency test. On no occasion was more than 1241 model runs required to find the global objective function minimum (mostly far fewer than this), and on no occasion did PD_MS2 carry out more than 2151 model runs before ceasing execution. The average number of model runs required to find the global objective function minimum was 493 while the average number of model runs required for completion of PD_MS2 execution was 1332. See Table 3 for a summary of the results of this numerical experiment.

This efficiency study was repeated with the INTFW parameter held fixed at 3.4. Furthermore, in each PD_MS2 run, only 120 pre-inversion model runs were carried out. In this case, the global objective function minimum was always found within 989 model runs (mostly far fewer than this), and on no occasion did PD_MS2 carry out more than 1400 model runs before ceasing execution. The average number of model runs required to find the objective function minimum was 331 and the average number of model runs required for completion of PD_MS2 execution was 1023. See Table 4 for a summary of the results of this numerical experiment.

In order to test the dependence of PD_MS2 performance on the magnitude of the finite difference derivative increment, the 8 parameter experiment was repeated with the increment varied downwards from 1%, to 0.1% and 0.01% of current parameter values. The results are shown in Table 5. Use of the same random number seeds for respective PD_MS2 runs in this numerical experiment as those employed in the previous experiment allowed PD_MS2 runs to be compared on a run by run basis. In some cases greater precision in derivatives calculation achieved through use of an increment of 0.1% allowed faster convergence of the GML algorithm. However in many cases, faster PEST execution did not lead to faster PD_MS2 execution as more PEST runs were required to find the global objective function minimum, and for assurance that the global minimum had indeed been found. Where an even smaller increment was used, this situation was exacerbated. (It should be noted that use of an increment as small as 0.01% would not be possible at all in conjunction with a model that exhibited any numerical noise caused, for example, by the necessity to calculate system states using an iterative solver; hence use of an increment as low as 0.01% is not generally recommended.)

Comparison with SCE_UA

For purposes of comparison, the SCE-UA method was employed to calibrate the same model. We wish to point out that we are very wary of comparing different software

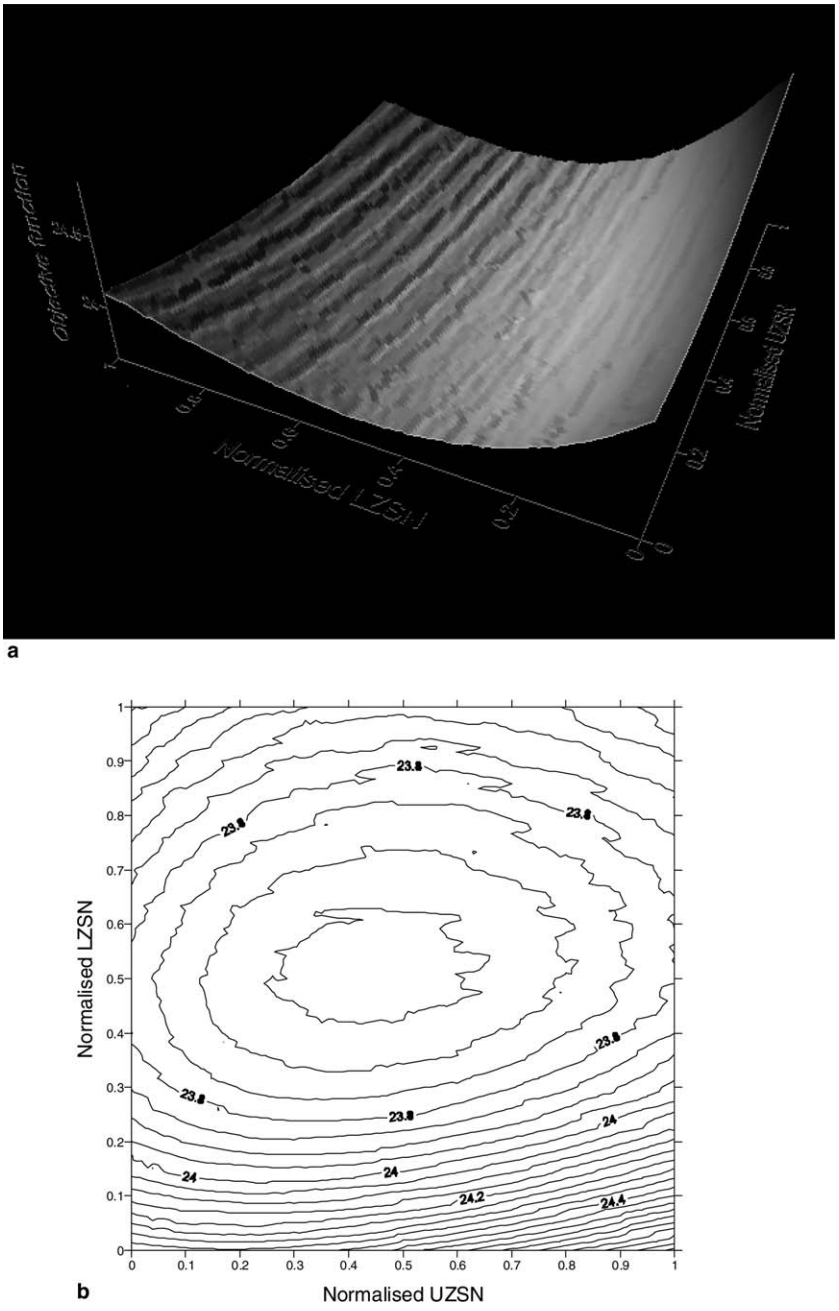

Figure 7 (a) Objective function surface computing by varying UZSN and LZSN incrementally about their optimized values and sampling at small, regular intervals. (b) Objective function contours corresponding to (a).

Table 3 Performance of PD_MS2 in estimating 8 HSPF parameters

	Minimum	Maximum	Average
Runs to find global objective function minimum	208	1214	493
Runs required for completion of PD_MS2 execution	626	2151	1332

For comparison, SCE_UA required about 4500 model runs for calibration of the same model.

Table 4 Performance of PD_MS2 in estimating 7 HSPF parameters

	Minimum	Maximum	Average
Runs to find global objective function minimum	198	989	331
Runs required for completion of PD_MS2 execution	517	1400	1023

For comparison, SCE_UA required about 2600 model runs for calibration of the same model.

Table 5 Performance of PD_MS2 in estimating 8 HSPF parameters for different settings of finite difference derivatives increment

	Fractional increment	Minimum	Maximum	Average
Runs to find global objective function minimum	10^{-2}	208	1214	493
	10^{-3}	200	1624	520
	10^{-4}	203	2342	701
Runs required for completion of PD_MS2 execution	10^{-2}	626	2151	1332
	10^{-3}	589	2456	1286
	10^{-4}	603	3432	1843

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 results presented below provide a comprehensive basis for assessment of the comparative performance of SCE-UA and PD_MS2/PEST. We hope, however, that they do provide a basis for at least a ''ball park'' comparison of the two methods for this particular calibration case.

After some experimentation with SCE-UA settings (the most important being the number of complexes to employ — see Duan et al. (1992), for further details) it was found that SCE-UA could be guaranteed to find the global objective function minimum of the eight-parameter minimization problem in about 4500 model runs. Our impression was that the narrowness of the objective function valley in which the global optimum was situated did not make SCE-UA's task an easy one, for it was able to achieve an objective function value of 24.0 in about 3000 model runs, and required the further 1500 model runs to penetrate to the depths of the elongate objective function surface surrounding the global minimum. Furthermore, unless more than 10 complexes were employed, SCE-UA failed to find an objective function below 24.0 at all; therefore, unless a modeler undertook a number of experimental SCE-UA runs with varying numbers of complexes, it would be possible for him/her to remain unaware of the fact that the global objective function was less than this.

Fig. 8 shows the best parameters estimated by SCE-UA at the end of each shuffling loop, together with the objective function to which they pertain, the format of this plot being the same as that of Figs. 2—4; to allow easy comparison with these figures, only objective function values below 44.0 are depicted in Fig. 8. They thus record the late history of the parameter estimation process as undertaken by SCE-UA. The structural details of the objective function surface that are apparent in Figs. 2—4, are not apparent in Fig. 8.

The optimization process was repeated with INTFW held fixed at its optimal value of 3.4. SCE-UA required about 2600 model runs to achieve global optimization.

Comparison with random starting point selection

A further numerical experiment was undertaken whose purpose was to compare the performance of PD_MS2 with that of a scheme in which repeated PEST runs are undertaken from random starting points in parameter space, with no assistance being provided in selection of these points by pre-inversion ranking or by avoidance of proximity to previous trajectories. One hundred such PEST runs were undertaken under the control of a PEST driver named PD_MS1. In this experiment, INTFW was fixed at its optimal value, so that 7 parameters were estimated. It was found that of the 100 PEST runs undertaken by PD_MS1, 54 of these runs found the global objective function minimum.

With a failure rate of 46 per 100, the chances of failing to find the global optimum in two independent PEST runs is 21.2%; for 3 PEST runs it is 9.7%. Outcomes of the 30 PD_MS2 runs discussed above must be interpreted in the light of these statistics. Examination of the records of these runs revealed that the global objective function minimum was found within 2 PEST runs for all but 3 of the 30 PD_MS2 runs; it was found within 3 PEST runs for all but 2 of them. It does thus appear that use of PD_MS2 results in increased global optimization efficiency over that of PD_MS1 in this case.

The numerical experiment was repeated with INTFW adjustable. In this case, it was established through the use of PD_MS1 (200 PEST runs were undertaken in this experiment) that the probability of failure in finding the global objective function minimum on any one PEST run for which starting parameter values are randomly selected is 56%. Thus failure probabilities for 2, 3 and 4 successive PEST runs from random starting points are 31.4%, 17.6% and 9.8% respectively. Examination of the records of the 30 eight-parameter PD_MS2 runs discussed above revealed that PD_MS2 was able to find the global objective function minimum within 2 PEST runs in 23 out of these 30 PD_MS2 runs. Three PEST runs or less were required in 25 cases, while 4 PEST runs or less were required in 28 cases. The latter two figures approach PD_MS1 probabilities.

Further verification of the trajectory repulsion scheme

Work documented earlier in this section demonstrated that use of the TPI scheme considerably heightened, but did not remove, the propensity for ensnarement of the GML method in a local objective function minima. The example of the preceding section demonstrated that, in the present case at least, the propensity for PD_MS2 to find the global objective function within one to three PEST runs (as an outcome of pre-calibration sampling and trajectory repulsion) mitigated the need for this scheme to some extent (in the present case at least), and that use of TSVD as a device for numerical stabilization (with its consequential doubling of model run efficiency) resulted in good PD_MS2 performance. It should be noted that this may not always be the case, as objective function surfaces far more pitted than that illustrated in Fig. 7 may be encountered in the course of calibrating surface water models. In such cases, use of the TPI scheme may be fundamental to good PEST performance.

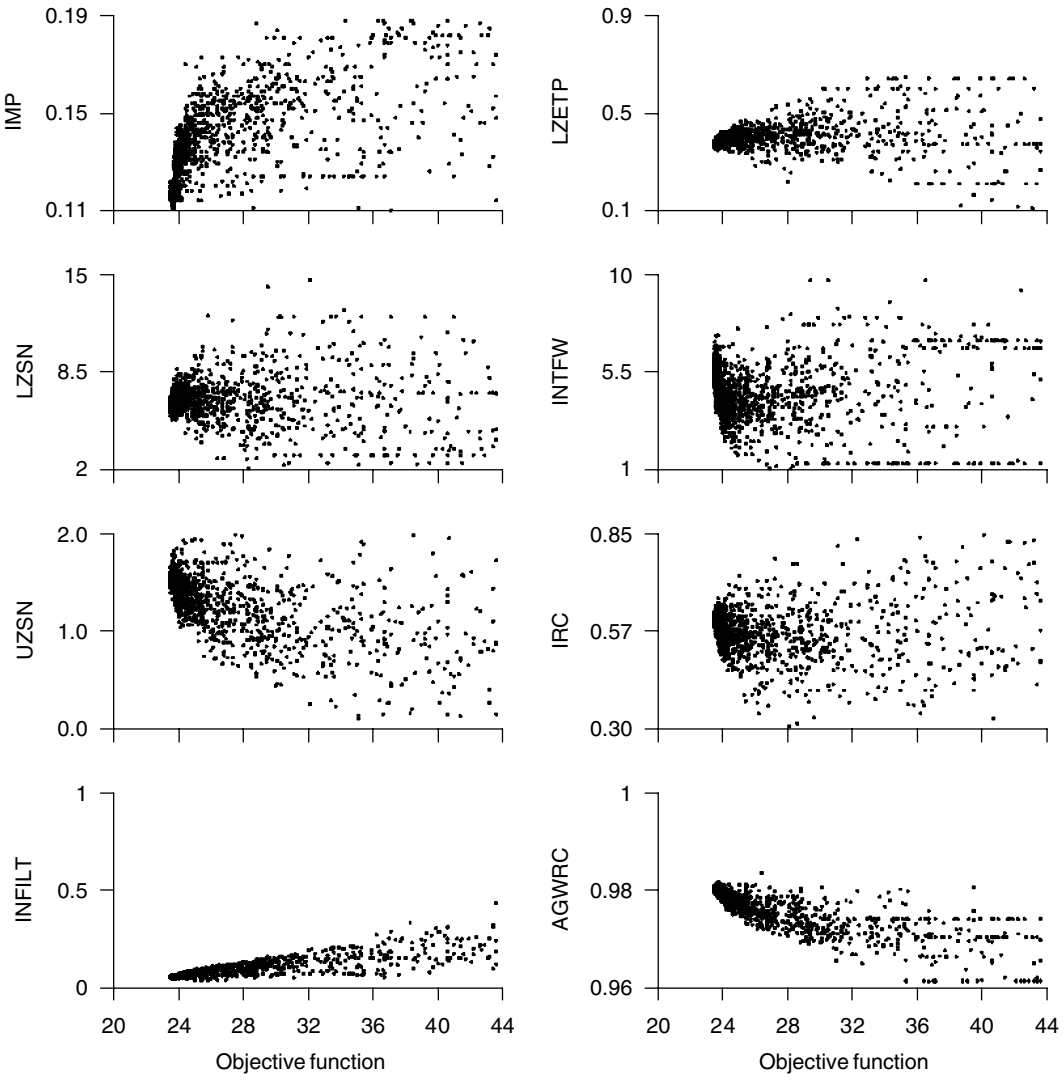

Figure 8 Best parameters and objective functions obtained by SCE-UA at the end of each shuffling loop of the 8-parameter calibration problem. (Note that only objective functions below 44.0 are represented in this figure; the objective function at the commencement of the parameter estimation process was over 200.)

A numerical experiment was undertaken in which the TPI algorithm was again disabled so that the efficacy of the trajectory repulsion scheme alone could be tested. One hundred PEST runs were undertaken, using TSVD (with the same truncation criteria as for other TSVD implementations documented herein) as a stabilization device, from randomly-selected starting points in parameter space (using the PD_MS1 driver). The full set of 8 HSPF parameters were estimated in each case. A histogram of objective function values achieved from these runs is depicted in the lower part of Fig. 9.

The upper part of Fig. 9 shows objective function values resulting from 100 PEST runs undertaken under the control of PD_MS2; these are the same results as those depicted in Fig. 2. A comparison of these two histograms reveals the following.

1. For both methods, the number of times that PEST found the global objective function minimum was about the same. It must not be construed from this, however, that PD_MS1 is as efficient in finding the global minimum of the objective function as PD_MS2 is, for it must be remembered that, after having found the minimum objective function once, it is PD_MS2's task to find other objective function minima from that point onwards. Efficiency rests on finding the global minimum as quickly as possible. Testing this efficiency requires that multiple PD_MS2 runs with different random number seeds and appropriate termination criteria be carried out as described in the preceding section.
2. A number of PEST runs undertaken under the control of PD_MS1 resulted in relatively high objective function minima being found. This was not the case for PEST runs conducted under the supervision of PD_MS2. This is probably an outcome of PD_MS2's rejection of parameter sets with high objective function values as starting parameter candidates for a PEST run.
3. Within the range of medium to low objective functions, excluding the optimum objective function, PEST runs undertaken under the control of PD_MS2 appear to have

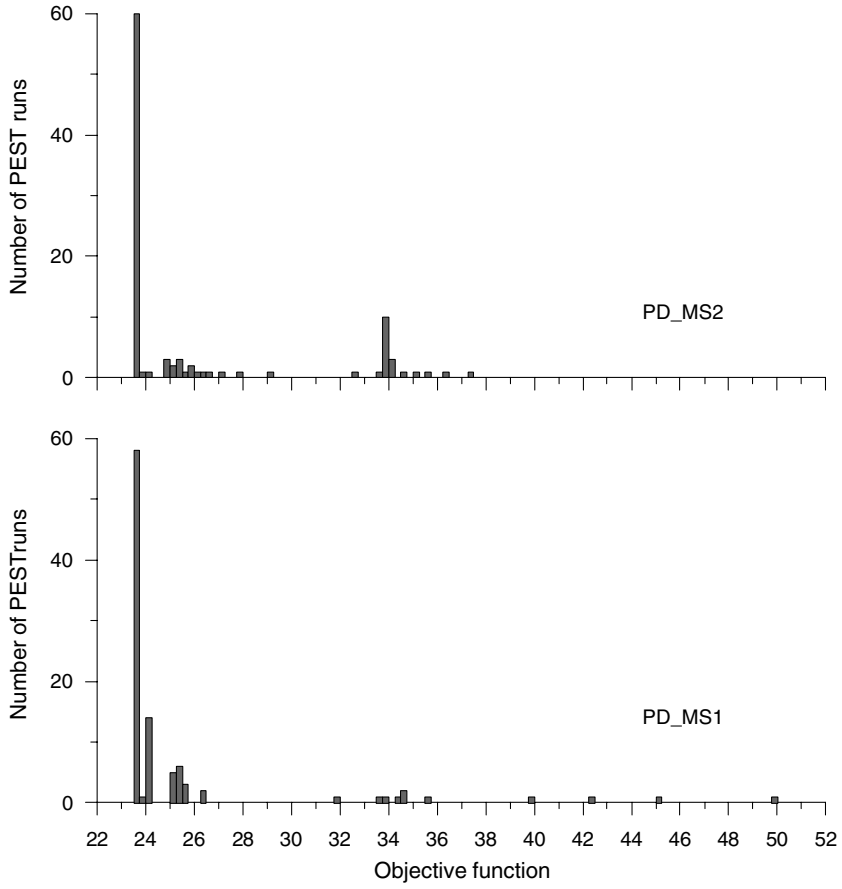

Figure 9 Objective function occurrences for 100 PEST runs based on random starting locations (lower figure), and starting locations selected during a single, long, PD_MS2 run (upper figure).

encountered a broad range of objective functions. On the other hand, PD_MS1-calculated objective functions appear to be concentrated in fewer values. The greater spread of minimized objective functions for PD_MS2 is probably an outcome of PD_MS2's use of the trajectory repulsion algorithm in selecting parameter starting points that are maximally distant from all points on previous parameter trajectories.

Discussion and conclusions

The results of the above numerical experiments demonstrate that a GML-based method can perform well in finding the global minimum of a complex objective function surface with a run-efficiency that is at least as good as that of the SCE-UA method if that scheme is enhanced to perform better in such inversion contexts. For the PD_MS2 package discussed herein, these enhancements consisted of:

1. the inclusion, within the PEST inversion engine, of a truncated singular value decomposition algorithm to prevent deterioration in numerical performance where one or more parameters became inestimable;
2. the inclusion within PEST of the TPI algorithm which not only provides a means to circumvent numerical instability in situations where the $\mathbf{X}^t\mathbf{Q}\mathbf{X}$ matrix of Eq. (3) is difficult to invert, but also provides the basis for an intelligent search strategy through which ensnarement in local objective function pits is avoided;
3. the undertaking of a number of pre-inversion model runs prior to selecting initial parameter sets that enhance the chances of a GML-based inversion engine finding its way into the region of attraction of parameter space dominated by the lowest objective function in that space; and
4. a trajectory repulsion scheme that lowers the chances of finding the same objective function minimum twice.

As well as providing an efficient means of finding the global minimum of the objective function, use of the PD_MS2 package brings with it further advantages. One of these advantages is that PD_MS2 can be configured to undertake a long run in which its task is to find as many different objective function minima as possible. In doing this it effectively probes parameter space for information concerning the structure of the objective function surface. Where the number of parameters being estimated is high, and the dimensionality of parameter space is thus also high, exploration of parameter space, by whatever means, becomes a numerically intensive process. The ability of PD_MS2 to find the locations of different objective function minima within this space provides, at relatively low numerical cost, valuable information on those aspects of the objective function surface that are of most relevance to a particular model

calibration problem. Of particular importance is the existence of local objective function minima with similar objective function values to that of the global minimum, but for which at least one parameter is significantly different between these two minima (as occurs in the example documented above). This phenomenon may have a profound effect on predictive probability distributions; its existence could be easily missed using other parameter estimation methods.

Another advantage that accompanies the use of a GML-based method is the fact that, as a by-product of its use, important information on parameter uncertainty, correlation and sensitivity is available at no extra computational cost from the beginning of the parameter estimation process. On the basis of this information, a modeler is able to determine at an early stage of this process whether the inverse problem that he/she is attempting to solve is in fact capable of unique solution. As the parameter estimation process progresses, information on the uncertainties associated with parameter estimates becomes available. Though local, and based on a linearity assumption that is violated by most models, such information can still be valuable in allowing a modeler to distinguish between those parameters which are well-estimated through the inversion process and those which are not. Should the process need to be repeated, such information provides the basis for making decisions pertaining to which parameters should be fixed, rather than estimated, in subsequent calibration operations.

The GML method has suffered problems in the past when applied to the calibration of watershed models as a result of its propensity to become trapped in local objective function minima and because of its susceptibility to numerical instability where inverse problems are poorly posed. However, in other modeling contexts it is widely used because of its high run-efficiency. Furthermore, as has already been discussed, numerical instability problems are easily overcome through the inclusion of various regularization schemes. It is worth noting that not only can use of such schemes circumvent the deleterious affects on the inversion process of numerical instability; they can also provide a mechanism for assimilation of valuable ''outside knowledge'' into that process, with the result that parameter estimates, thus informed by a modeler's expertise, are more suitable for use in the making of important predictions by that model than would otherwise be the case. (See Doherty and Skahill (2006) for an example of a sophisticated regularization scheme applied to simultaneous, composite calibration of a number of watershed models.) For these reasons the GML method, and variants of it, form the basis of many different data interpretation and image processing methodologies in many different industries where the estimation of hundreds, or even thousands, of parameters must be undertaken with maximum efficiency; see, for example, Haber et al. (2000) and references cited therein. Even where only a few parameters require estimation, model run efficiency is a consideration of overriding importance where model run times are large. Its importance is highlighted by the fact that the parameter estimation process must often be repeated many times in the course of finalizing the calibration of a model prior to using that model for environmental management.

It is hoped that the local optima problem facing use of the GML method in the surface water modeling context has been at least partially addressed by the advent of PD_MS2. Using this software global optimization can be implemented with a run-efficiency that is easily comparable with that of purpose-built global optimizers. Furthermore, through use of this package, the penchant of the GML method to find local minima can actually be turned to advantage, for this characteristic of the method can be used as a basis for exploration of the objective function surface while incurring a relatively small cost in terms of model runs. This, in turn, can lead to a better understanding of the parameter estimation problem as presently posed, and may possibly lead to re-formulation of that problem so that it can be more easily, or uniquely, solved.

The purpose of the present paper is not to discredit the use of global search methods in the calibration of watershed models. Rather, it is to demonstrate that robust parameter estimation packages based on the GML method should not be excluded from consideration for the calibration of these models, especially when enhanced by techniques such as those encapsulated in PD_MS2 and PEST that allow this method to perform well in calibration contexts where local objective function minima are a common occurrence. The fact that linear and/or nonlinear predictive uncertainty analysis can be undertaken following a GML-based calibration run with zero (in the case of linear analysis) or fairly modest (in the case of nonlinear analysis) model run requirements further adds to the attractiveness of GML-based methods; see Vecchia and Cooley (1987), Doherty and Johnston (2003) and Doherty (2005) for details.

Software

PEST, PD_MS1, PD_MS2 and supporting software are available free of charge from the following site:- http://chl.erdc.usace.army.mil/pest

Acknowledgement

The Puget Sound Naval Shipyard and Intermediate Maintenance Facility (PSNS & IMF) Environmental Investment (ENVVEST) Project supported this study. The authors would like to thank the PSNS & IMF ENVVEST Project participants for their contributions to this effort. Headquarters, US Army Corps of Engineers granted permission for publication of this paper.

References

Alley, W.M., Veenhuis, J.E., 1983. Effective impervious area in urban runoff modeling. Journal of Hydraulic Engineering 109 (2), 313–319.

Bates, B.C., Campbell, E.P., 2001. A Markov chain Monte Carlo scheme for parameter estimation and inference in conceptual rainfall-runoff modeling. Water Resources Research 37 (4), 937–947.

Bard, Jonathon, 1974. Nonlinear Parameter Estimation. Academic Press, NY, p. 341.

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.

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

Box, G.E.P., Tiao, G.C., 1973. Bayesian Inference in Statistical Analysis. Addison-Wesley, Reading, MA.

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

DeGroote-Hedlin, C., Constable, S., 1990. Occam's inversion to generate smooth, two-dimensional models from magnetotelluric data. Geophysics 55 (12), 1624–1631.

Doherty, J., 2005. PEST: Model Independent Parameter Estimation. fifth edition of user manual. Watermark Numerical Computing, Brisbane, Australia.

Doherty, J., Johnston, J.M., 2003. Methodologies for calibration and predictive analysis of a watershed model. Journal of American Water Resources Association 39 (2), 251–265.

Doherty, J., Skahill, B.E., 2006. An advanced regularization methodology for use in Watershed Model Calibration. Journal of Hydrology.

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

ENVVEST Regulatory Working Group, 2002. FINAL Fecal Coliform Total Maximum Daily Load Study Plan for Sinclair and Dyes Inlets Quality Assurance Project Plan. October 4, 2002. http://www.ecy.wa.gov/programs/wq/tmdl/watershed/sinclair-dyes_inlet/fc_tmdl_studyplan_final_draft_print.pdf.

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

Haber, E., Ascher, U.M., Oldenburg, D., 2000. On optimization techniques for solving nonlinear inverse problems. Inverse Problems 16, 1263–1280.

Hill, M.C., 1998. Methods and Guidelines for Effective Model Calibration. U.S. Geological Survey Water-Resources Investigations Report 98-4005.

Kuczera, G., 1983. Improved parameter inference in catchment models. 1. Evaluating parameter uncertainty. Water Resources Research 19 (5), 1151–1172.

Kuczera, G., 1990. Assessing hydrologic model nonlinearity using response surface plots. Journal of Hydrology 118, 143–161.

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

Madsen, H., Wilson, G., Ammentorp, H.C., 2002. Comparison of different automated strategies for calibration of rainfall-runoff models. Journal of Hydrology 261, 48–59.

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

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

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

Thyer, M., Kuczera, G., Bates, B.C., 1999. Probabilistic optimization for conceptual rainfall-runoff models: a comparison of the shuffled complex evolution and simulated annealing algorithms. Water Resources Research 35 (3), 767–773.

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

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

Vecchia, A.V., Cooley, R.L., 1987. Simultaneous confidence and prediction intervals for nonlinear regression models with application to a groundwater flow model. Water Resources Research 23 (7), 1237–1250.

Vrugt, J.A., Gupta, H.V., Bouten, W., Sorooshian, S., 2003. A shuffled complex evolution metrolopolis algorithm for optimization and uncertainty assessment of hydrologic model parameters. Water Resources Research 39 (8), 1201–1215.

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