An agent-based approach to global uncertainty and sensitivity analysis

Dylan R. Harp[1] and Velimir V. Vesselinov[1]

A novel approach to global uncertainty and sensitivity analyses of modeling results utilizing concepts from agent-based modeling is presented. A plausible model parameter space is discretized and sampled by a particle swarm where the particle locations represent unique model parameter sets. Particle locations are optimized based on a model performance metric using a standard particle swarm optimization (PSO) algorithm. Locations producing a performance metric below a specified threshold are collected. In subsequent visits to the location, a modified value of the performance metric, proportionally increased above the acceptable threshold (i.e. convexities in the response surface become concavities), is provided to the PSO algorithm. As a result, the methodology promotes thorough exploration of a plausible parameter space, and discourages reinvestigation of previously explored regions, which effectively alters the strategy of the PSO algorithm from optimization to global uncertainty and sensitivity analysis. The viability of the approach is demonstrated on 2D Griewank and Rosenbrock functions. The practical application of the approach is demonstrated on a 3D synthetic contaminant transport case study to evaluate the global uncertainty of the horizontal source location and transverse and longitudinal dispersivities.

1. Introduction

Inverse approaches are routinely used to identify appropriate values of model parameters that provide simulations with the highest degree of consistency with existing observations. These approaches can be considered to provide answers to the question "What do the observations and model tell us about the parameters?". An often neglected question is "What do the observations and model have the ability to tell us about the parameters?". The answer to the second question is required to properly evaluate the significance and uncertainty of the answers to the first question. Approaches that answer the second question explore the effect of changes in parameter values on a performance metric and are considered model-based uncertainty analysis (UA) approaches.

UA is often based on sensitivity analysis techniques. Local sensitivity analyses evaluate the sensitivities surrounding the current solution by calculating derivatives of model simulations with respect to model parameters [*Vecchia and Cooley*, 1987; *Cooley*, 1993] or adjoint solutions of the governing equation [*Neuman*, 1980; *Sykes et al.*, 1985; *Li and Yeh*, 1998]. Local sensitivity analysis approaches are computationally efficient, requiring relatively few model calls operating under the assumption that parameter probability distributions are normally distributed. These techniques are commonly utilized in gradient-based optimization strategies for parameter estimation. The information provided by these techniques in a UA is limited to a region surrounding the current parameter values, to models with a continuous parameter space, and by the assumption of normally distributed parameter uncertainty.

Null-Space Monte-Carlo (NSMC) combines concepts from error variance analysis theory and Monte Carlo (MC) sampling to perform UA on highly parameterized models [*Tonkin and Doherty*, 2009]. The null-space is defined from local sensitivities of a calibrated model. The null-space is defined for a given set of best model parameter estimates as a subspace of the parameter space comprised of parameter combinations that have negligible impact on the performance metric. An MC sampling is utilized to produce parameter realizations by modifying parameter values within the calibration null-space. If, in the process of MC sampling, a parameter realization produces an uncalibrated model, parameters in the calibration solution space are re-estimated to re-calibrate the model. This produces a local UA capable of reducing the computational burden imposed by a large numbers of parameters.

Most global sensitivity analysis approaches are based on evaluating the relative contribution of individual and combinations of parameters to the variance of a performance metric [*Sobol*, 2001; *van Werkhoven et al.*, 2008; *Wagener et al.*, 2009]. These approaches provide scalar indices of global sensitivity. This information indicates parameters of interest and correlated parameter estimates. The information from such analyses does not provide specific information about sensitivities at any specific point in the parameter space, and is therefore of limited use in a UA.

Evaluation of global uncertainty of a model is typically based on global sampling approaches. *Vrugt et al.* [2008] introduced a Markov chain Monte Carlo (MCMC) approach entitled Differential Evolution Adaptive Metropolis (DREAM). This approach provides estimates of posterior density functions of parameters in a formal Bayesian framework. An informal Bayesian approach to global UA is the Generalized Likelihood Uncertainty Analysis (GLUE) developed by *Beven and Binley* [1992]. This approach performs an MC analysis using a statistically informal likelihood function to rank model performance. Recently, *Harp and Vesselinov* [2011] developed a sampling approach for global UA of stochastic models of flow medium heterogeneity introducing the concept of an acceptance probability of a stochastic parameter set. Sampling approaches have the ability to provide detailed information directly addressing the UA. The drawback to such approaches is that the number of model calls is often too large for many practical applications involving process-based models [*Keating et al.*, 2010].

The approach presented here intends to provide an alternative to existing UA approaches that will be useful for complex problems for which a local UA is known to be inappropriate, but for which a rigorous sampling based approach is infeasible due to computational constraints. We will refer to this approach as Agent-Based Analysis of

[1]Earth and Environment Science Division, Los Alamos National Laboratory, Los Alamos, NM, USA.

0094-8276/11/$5.00

Global Uncertainty and Sensitivity (ABAGUS). Concepts from agent-based modeling have been utilized extensively in optimization algorithms, such as particle swarm optimization (PSO) [*Kennedy and Eberhart*, 1995; *Clerc*, 2006; *Vesselinov and Harp*, 2011] and ant colony optimization [*Dorigo and Stützle*, 2004]. However, to our knowledge, their direct application to global UA has not been explored. The ABAGUS computational framework is based on integrating concepts of agent-based social simulation with the Standard PSO 2006 (SPSO2006) algorithm [*Paricle Swarm Central*, 2006], effectively altering the strategy of SPSO2006 from optimization to global UA. SPSO2006 is chosen here as it implements a parsimonious and efficient version of particle swarm optimization.

The strategy of ABAGUS is to efficiently explore a discretized parameter space by storing information about locations producing simulations consistent with observations within specified uncertainty bounds. The algorithm alters the response surface at the already sampled locations by increasing the associated performance metric (e.g. objective function, fitness function). As a result, if points within a local area of attraction are already visited by the algorithm, the region appears as a region of concavity (repulsion), as opposed to a region of convexity (attraction), discouraging future exploration. Similarities can easily be drawn between ABAGUS and the Sugarscape agent-based social simulator [*Epstein and Axtell*, 1996], designed to model the survival of a population on a regenerative resource; however, in ABAGUS, the resource is not regenerative, encouraging global exploration of the parameter space.

Since the ABAGUS algorithm is based on SPSO2006, a brief discussion of this algorithm is presented in section 2. The ABAGUS algorithm is discussed in section 3. Section 4 demonstrates the performance of ABAGUS on 2D Griewank and Rosenbrock functions. Section 5 presents a synthetic five parameter contaminant transport problem that is utilized to demonstrate the use of ABAGUS on a practical application.

2. Standard PSO 2006 algorithm

SPSO2006 modifies a population of solutions called particles defined by their position and velocity in a D-dimensional parameter space. The position and velocity of the ith particle can be represented as $\vec{P_i} = [p_{i,1}, p_{i,2}, \ldots, p_{i,D}]$ and $\vec{V_i} = [v_{i,1}, v_{i,2} \ldots, v_{i,D}]$, respectively. An empirical formula for determining the swarm size S has been suggested as $S = 10+\sqrt{D}$ [*Paricle Swarm Central*, 2006]. Particles retain a record of the best location they have visited so far denoted as $\vec{B_i} = [b_{i,1}, b_{i,2}, \ldots, b_{i,D}]$. Particles are also informed of the best location that K other randomly chosen particles have visited, denoted as $\vec{G_i} = [g_{i,1}, g_{i,2}, \ldots, g_{i,D}]$. A standard value for K is 3 [*Paricle Swarm Central*, 2006]. These networks of informers are reinitialized after iterations with no improvement in the global best location of the swarm. The velocity of the ith particle in the jth dimension is updated from swarm iteration step k to $k+1$ as

$$\begin{aligned} v_{i,j}(k+1) &= wv_{i,j}(k) + c_1 r_1 (b_{i,j} - p_{i,j}(k)) + \\ &\quad c_2 r_2 (g_{i,j} - p_{i,j}(k)), \quad k = \{1, \ldots, D\}, \end{aligned} \tag{1}$$

where w is a constant referred to as the inertia weight, c_1 and c_2 are constants referred to as acceleration coefficients, r_1 and r_2 are independent uniform random numbers in $[0, 1]$. The swarm iteration steps are also referred to as time steps because they represent the progress of swarm development in the parameter space. The parameter w controls the level of influence of the particles previous displacement on its current displacement, c_1 and c_2 scale the random influence of the particles memory and its network of informers, respectively. Values of $w = 0.72$ and $c_1 = c_2 = 1.2$ have been demonstrated to perform well an many problems [*Clerc*, 2006]. A limitation on the magnitude of the velocity V_{max} is commonly employed. The particle position at each iteration is updated as

$$p_{i,j}(k+1) = p_{i,j}(k) + v_{i,j}(k+1), \quad k = \{1, \ldots, D\}. \tag{2}$$

Additional details on SPSO2006 are available in *Clerc* [2006] and *Cooren et al.* [2009]. The source code is available for download at *Paricle Swarm Central* [2006].

3. ABAGUS algorithm

Concepts from agent-based modeling have found significant utility in global optimization. The following discusses the first, to our knowledge, utilization of agent-based modeling to perform global UA.

ABAGUS works on a discretized parameter space based on user-provided parameter-specific resolution (each parameter can be assigned a distinct resolution). The resolution of the analysis can therefore be controlled by the user depending on computational constraints and/or desired level of detail. ABAGUS runs can also be nested, using the samples from previous coarser runs as starting points for finer resolution runs. The discretization of the parameter space does not hinder the UA as the strategy is to identify regions of the parameter space producing indistinguishably consistent simulations with observations.

ABAGUS collects locations with a performance metric Φ below a defined threshold ϵ, and inverts the value of the performance metric as

$$\Phi_{inv} = 2\epsilon - \Phi, \quad \Phi < \epsilon, \tag{3}$$

where Φ_{inv} is the value of the inverted performance metric. Φ_{inv} is provided to particles on subsequent visits in place of recalculating Φ. The value of ϵ can be defined based on theoretical (e.g. confidence levels under certain assumptions; cf. [*Vecchia and Cooley*, 1987; *Cooley*, 1993]) or problem-specific considerations. The potentially large number of locations that must be collected are managed by a KD-tree, allowing the collected locations to be efficiently searched in a binary fashion in a K-dimensional space [*Tsiombikas*, 2009]. A nearest neighbor search of the KD-tree is utilized to identify if a location has been collected previously [*Tsiombikas*, 2009]. As a result, revisiting collected locations has a relatively insignificant cost to the algorithm, particularly in cases involving long model execution times.

Equation 3 effectively adds the discrepancy between ϵ and Φ to ϵ and assigns the value to Φ_{inv}. The larger the discrepancy, the less attractive the position appears to future visits. As a result, convexities in the response surface become concavities.

As the ABAGUS algorithm progressively identifies and collects acceptable locations in the parameter space, the coefficients w, c_1, and c_2 are dynamically modified to maintain an appropriate balance between exploration and intensification. An exploration rate metric ρ quantifies the level of exploration at each iteration of the ABAGUS run as

$$\rho = N_e / N_r \tag{4}$$

where N_e is the number of new locations visited and N_r is the number of revisits to previously collected positions. One iteration of ABAGUS involves updating and evaluating the population of solutions (particles). The following rules are used to maintain a reasonable value for ρ:

$$\begin{aligned} \text{if } \rho < \rho_0 : w &= w * (1 + a), \\ c_1 &= c_1 * (1 + a), \\ c_2 &= c_2 * (1 + a). \\ \text{if } \rho > \rho_0 : w &= w * (1 - d), \\ c_1 &= c_1 * (1 - d), \\ c_2 &= c_2 * (1 - d). \end{aligned}$$

where ρ_0 is set by the user to a value deemed to be reasonable and a and d are constants greater than zero. In the cases investigated here, values of $\rho_0 = 1$ and $a = d = 10^{-5}$ were found to be effective. More complex strategies for controlling ρ by modifying w and c are easily conceptualized, and will take time and effort to evaluate on varied response surfaces.

ABAGUS is allowed to run to a maximum number of model evaluations or until the swarm runs out of energy. The initial energy of the swarm is specified by the user, where larger values of initial energy will allow more exploration of the parameter space. Each particle move decrements the swarm energy by one. Each identification of an acceptable location increments the swarm's energy. Incrementing the swarm energy by 10% of the initial energy is used for the cases investigated here. For an initial investigation of the parameter space, an initial energy of 10,000 is reasonable for the test cases presented here.

4. Test functions

The performance of ABAGUS is demonstrated on 2D Griewank and Rosenbrock test functions, defined as

$$z = \frac{x^2 + y^2}{4000} - cos\left(\frac{x}{\sqrt{2}}\right) * cos\left(\frac{y}{\sqrt{3}}\right) + 1 \qquad (5)$$

and

$$z = (1 - x)^2 + 100 * (y - x^2)^2, \qquad (6)$$

respectively. The Griewank and Rosenbrock functions are benchmark problems presenting challenging response surfaces for optimization strategies. The Griewank function contains numerous local minima with a single global minimum of zero at (0,0). The Rosenbrock function contains a large smooth valley with an ill-defined global minimum of zero at (1,1).

Parameter bounds for x and y are both $[-100, 100]$ and the parameter space is discretized to a 0.1 resolution for both functions, resulting in 4×10^6 possible locations. The value of ϵ is set to 0.1 for the Griewank run and 20 for the Rosenbrock run. The initial swarm energy is set to 10,000 and the number of function evaluations is limited to $2{\times}10^6$. Initial values for w, c_1 and c_2 are set according to the constant values commonly utilized by SPSO2006 *Paricle Swarm Central* [2006] (w=0.72; $c = c_1 = c_2$=1.2, where c_1 and c_2 will be referred to collectively as c hereafter). In order to evaluate the performance of the ABAGUS algorithm, one particle is initialized to the global minimum ((0,0) for the Griewank function, (1,1) for the Rosenbrock function). This eliminates the initial search from random locations prior to the identification of an area of attraction, which, for the ABAGUS algorithm, is identical to SPSO2006 *Paricle Swarm Central* [2006]. In fact, the utilization of ABAGUS in this manner (i.e. beginning the ABAGUS run from a known optimal location obtained by a prior optimization) is an attractive approach.

Figures 1 (a) and (b) present maps of the response surfaces for the parameter space considered in the ABAGUS runs for the Griewank and Rosenbrock functions, respectively ($[-100, 100]$). Figures 1 (c) and (d) present 3D plots of the response surfaces near the global minimum for each case. The results of the ABAGUS runs are presented in Figures 1 (e) and (f) as maps of the response surfaces at identified locations.

The Griewank run collected 1788 locations with $\Phi < \epsilon = 0.1$ from approximately $2.00{\times}10^6$ function evaluations with approximately $2.08{\times}10^6$ revisits to collected locations. The Rosenbrock run collected 325 locations with $\Phi < \epsilon = 20$ from 109,060 function evaluations. The Griewank run took 15 seconds with approximately $1.3{\times}10^5$ function evaluations per second and the Rosenbrock run took 1 second with approximately $1.1{\times}10^5$ function evaluations per second on a 2.8GHz processor.

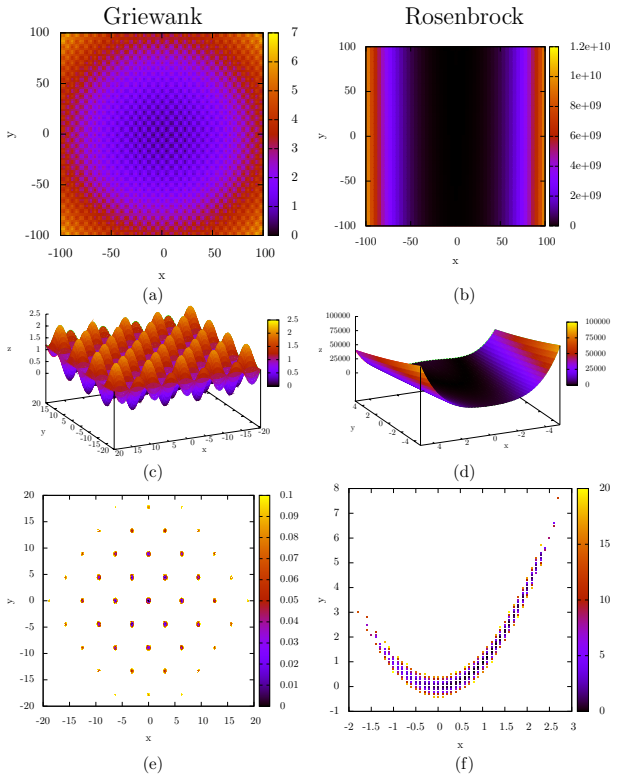

Figure 1. Griewank and Rosenbrock test function analyses. Maps of the response surface for the full parameter space considered in the search [-100,100] are presented in (a) and (b). Subplots (c) and (d) present 3D surfaces of the objective function near the region of the parameter space with values below the cutoff. Subplots (e) and (f) present the results of the ABAGUS runs identifying the solutions below predefined cutoffs equal to 0.1 and 20, respectively. A global minimum of 0 exists at (0,0) for the Griewank function and (1,1) for the Rosenbrock function.

5. Contaminant transport case study

The ABAGUS approach is demonstrated on a synthetic contaminant transport problem to explore the model-based uncertainty of distributed contaminant concentrations in an analytical contaminant transport model [*Vesselinov and Harp*, 2010] considering uncertainty in the plume source location (x_s, y_s) and dispersivities (a_x, a_y, a_z). Flow is in the x-direction. True concentrations are collected from a simulation of the model given true parameter values (x_s = 759 m, y_s = 1448 m, a_x = 70 m, a_y = 15 m, and a_z = 0.3 m) and are listed in Table 1. Information regarding the parameters (e.g. value, min, max, and resolution) is presented in Table 2. The collected concentrations have been rounded to provide values similar in resolution to field-collected measurements. A concentration map of the "truth" at $t = 49$ a is presented in Figure 2.

The performance metric for the contaminant transport case study is a sum-of-the-squared-residuals (SSR) expressed as

$$\Phi(\theta) = \sum_{i=1}^{N} (\hat{c}_i(\theta) - c_i)^2, \tag{7}$$

where Φ is the performance metric, θ is a vector containing the parameter values, $\hat{c}(\theta)$ is a vector of simulated concentrations resulting from θ, c is a vector containing the observed concentrations, and N is the number of observations. Due to the rounding of the collected concentrations, a value of Φ=0.14 is obtained from the true parameter values.

It is assumed that we are interested in collecting parameter sets producing values of Φ below 100 (ϵ = 100, refer to equation 3). The true parameters are provided to define the location of one of the initial particles. The initial energy is set to 10,000 and the maximum number of model calls is 200,000. As in the test functions, initial values for w and c are set according to the constant values commonly utilized by SPSO2006 *Paricle Swarm Central* [2006] (w=0.72;

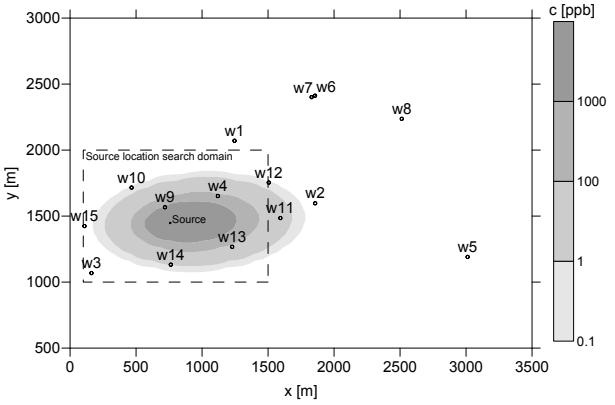

Figure 2. "True" contaminant concentration map at 49 years. Circles represent monitoring well locations. A dashed line rectangle indicates the parameter bounds for x_s and y_s. The "true" contaminant source is indicated.

Table 1. Well coordinates, screen top (z_{top}) and bottom (z_{bot}) depths below the water table, and year and value of observed contaminant concentrations

Well	x [m]	y [m]	z_{top} [m]	z_{bot} [m]	t [a]	c [ppb]
w1	1246	2071	5.57	12.55	49	0
w2	1857	1597	36.73	55.14	49	0
w3	162	1068	0	15.04	49	0
w4	1120	1652	13.15	20.41	44	35
					49	61
w5	3012	1191	26.73	33.71	49	0
w6	1856	2412	69.01	83.98	49	0
w7	1830	2401	11.15	18.19	49	0
w8	2513	2237	4.86	11.87	49	0
w9	719	1567	3.66	10.09	49	876
w10	466	1716	3.32	9.63	49	0.1
			23.2	26.24	49	0
w11	1593	1485	4.94	7.99	49	5.2
			32.46	35.48	49	0.1
w12	1505	1754	3.59	6.64	49	0.1
			32.51	38.61	49	0
w13	1228	1267	3	6	50	51
			36	42	50	0.25
w14	763	1132	3	6	51	0.8
			30	33	51	0
w15	109	1424	3	6	51	0
			30	33	51	0

Table 2. Parameter values and resolution for the contaminant transport case study. Log-transformed dispersivities are presented in parenthesis as these are the values provided to ABAGUS for the case study.

	$x_s[m]$	$y_s[m]$	$a_x[m]$ ($\log(a_x)$)	$a_y[m]$ ($\log(a_y)$)	$a_z[m]$ ($\log(a_z)$)
value	759	1448	70 (1.845)	15 (1.176)	0.3 (-0.523)
min	100	1000	25 (1.398)	8 (0.903)	0.1 (-1.0)
max	1500	2000	200 (2.301)	40 (1.602)	2 (0.301)
resolution	0.1	0.1	1.0 (0.005)	0.1 (0.002)	0.0025 (0.002)

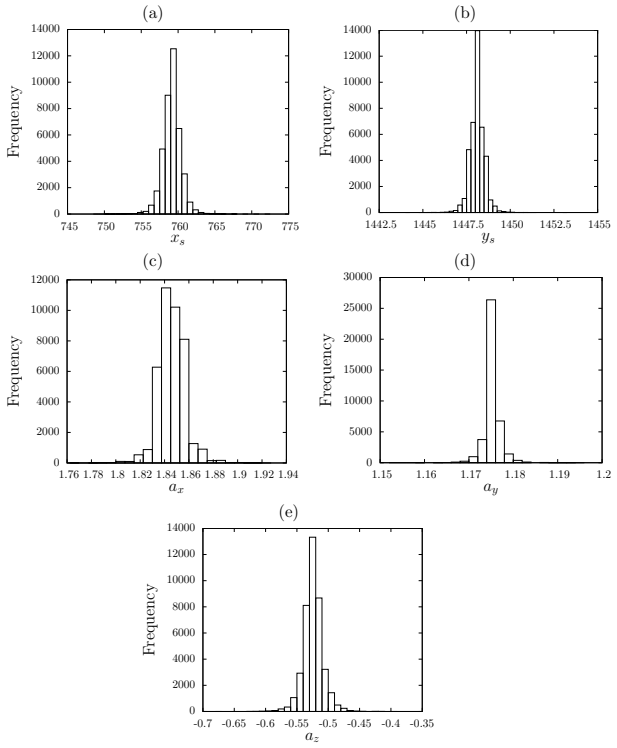

Figure 3. Histograms of parameter values obtained from ABAGUS evaluation.

c=1.2). The results for the ABAGUS run on the contaminant transport problem are presented as histograms of the parameter values in Figure 3, where it is apparent that the histograms are centered on the "true" values for all parameters (Table 2). The algorithm also allows predictive uncertainty analysis for model predictions at uncalibrated locations in the model domain or future predictions of calibration points.

The ABAGUS run collected 40,291 parameter sets producing $\Phi < \epsilon = 100$ from 2×10^5 model evaluations and 3.3×10^5 revisits to collected locations. The total number of plausible locations in the discretized parameter space is 4×10^{15}. The ABAGUS run took approximately 23 minutes on a 2.8GHz processor, with approximately 117 model evaluations per second. The resolution of x_s and y_s are 10 cm. This level of detail is not likely significant in a practical application, but is used here for demonstration purposes. A coarser level of detail in x_s and y_s would require fewer model calls.

Inspection of the results summarized by Figure 3 provide information answering the question discussed in the introduction: "What do the observations and model have the ability to tell us about the parameters?". The summary provided by the histograms in Figure 3 indicate the frequency of discrete parameter values producing equally consistent simulations to the observations considering a value of $\epsilon = 100$, indicating model parameter uncertainty and sensitivity.

6. Conclusions

The utilization of concepts from agent-based modeling coupled with the efficiency of KD-tree data storage provide a novel approach to perform a global UA. The efficiency of the approach can be tailored to the computational constraints of a problem by specifying the resolution of the search. ABAGUS does not produce formal posterior distributions of parameter probabilities consistent with Bayes' rule, instead focusing on identifying regions of the parameter space producing simulations acceptably consistent with observations. The performance of ABAGUS is evaluated on two test functions with known response surfaces, demonstrating the viability of the approach. The use of ABAGUS on a practical application is evaluated on a 5-parameter synthetic contaminant transport case study, demonstrating the approaches ability to identify regions of the response surface producing simulations acceptably consistent with observations surrounding the "true" parameter values. ABAGUS provides a discretized global UA approach filling the gap between local UA approaches and rigorous sampling-based global UA approaches. ABAGUS will be an attractive alternative for complex problems where it is recognized that a local UA is inappropriate, but for which a rigorous sampling-based global UA is infeasible due to computational constraints. The ABAGUS algorithm has been realized within the MADS toolbox [*Vesselinov and Harp*, 2010].

Acknowledgments. This work was supported by various projects within the Environmental Programs directorate of the Los Alamos National Laboratory.

References

Beven, K. J., and A. M. Binley (1992), The future of distributed models: Model calibration and uncertainty prediction, *Hydrological Processes*, *6*, 279–298.

Clerc, M. (2006), *Particle Swarm Optimization*, ISTE, London.

Cooley, R. L. (1993), Exact scheffé-type confidence intervals for output from groundwater flow models 1. Use of hydrogeologic information, *Water Resources Research*, *29*(1), 17–33.

Cooren, Y., M. Clerc, and P. Siarry (2009), Performance evaluation of TRIBES, an adaptive particle swarm optimization algorithm, *Swarm Intelligence*, *3*, 149–178, doi:10.1007/s11721-009-0026-8.

Dorigo, M., and T. Stützle (2004), *Ant Colony Optimization*, MIT Press, Cambridge, MA.

Epstein, J. M., and R. Axtell (1996), *Growing Artificial Societies: Social Science from the Bottom Up*, MIT Press, Cambridge, MA.

Harp, D. R., and V. V. Vesselinov (2011), Analysis of hydrogeological structure uncertainty by estimation of hydrogeological acceptance probability of geostatistical models, *Advances in Water Resources*, in review.

Keating, E. H., J. Doherty, J. A. Vrugt, and Q. Kang (2010), Optimization and uncertainty assessment of strongly nonlinear groundwater models with high parameter dimensionality, *Water Resources Research*, *46*, W10517, doi: 10.1029/2009WR008584.

Kennedy, J., and R. Eberhart (1995), Particle swarm optimization, in *Proceedings of the IEEE International Conference on Neural Networks*, pp. 1942–2948, IEEE Press, Piscataway.

Li, B., and T.-C. Yeh (1998), Sensitivity and moment analyses of head in variably saturated regimes, *Advances in Water Resources*, *21*, 477–485.

Neuman, S. P. (1980), A statistical approach to the inverse problem of aquifer hydrology 3. Improved solution method and added perspective, *Water Resources Research*, *16*(2), 331–346.

Paricle Swarm Central (2006), http://www.particleswarm.info/Standard_PSO_2006.c.

Sobol, I. (2001), Global sensitivity indices for nonlinear mathematical models and their monte carlo estimates, *Mathematics and Computers in Simulation*, *55*, 271–280.

Sykes, J., J. Wilson, and R. Andrews (1985), Sensitivity analysis for steady state groundwater flow using adjoint operators, *Water Resources Research*, *21*(3), 359–371.

Tonkin, M., and J. Doherty (2009), Calibration-constrained Monte Carlo analysis of highly parameterized models using subspace techniques, *Water Resources Research*, *45*, W00B10, doi:10.1029/2007WR006678.

Tsiombikas, J. (2009), kdtree: A simple C library for working with KD-Trees, http://code.google.com/p/kdtree/.

van Werkhoven, K., T. Wagener, P. Reed, and Y. Tang (2008), Characterization of watershed model behavior across a hydroclimatic gradient, *Water Resources Research*, *44*, W01429, doi: 10.1029/2007WR006271.

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

Vesselinov, V. V., and D. R. Harp (2010), MADS, modeling analysis and decision support toolkit in C, [web page] http://www.ees.lanl.gov/staff/monty/codes/mads.html, [Accessed on 15 Nov. 2010.].

Vesselinov, V. V., and D. R. Harp (2011), Contaminant source identification using adaptive hybrid optimization of inverse groundwater transport model, *Water Resources Research*, in review.

Vrugt, J. A., C. J. ter Braak, M. P. Clark, J. M. Hyman, and B. A. Robinson (2008), Treatment of input uncertainty in hydrologic modeling: Doing hydrology backward with Markov chain Monte Carlo simulation, *Water Resources Research*, *44*, W00B09, doi:10.1029/2007WR006720.

Wagener, T., K. van Werkhoven, P. Reed, and Y. Tang (2009), Multiobjective sensitivity analysis to understand the information content in streamflow observations for distributed watershed modeling, *Water Resources Research*, *45*, W02501, doi: 10.1029/2008WR007347.