A generalized global optimization application and its use in determining the hydraulic properties of unsaturated sand

Marc Lebeau & Jean-Marie Konrad
Department of Civil and Water Engineering – Laval University
Quebec, QC, Canada

GEOQuébec 2015
Challenges from North to South
Des défis du Nord au Sud

ABSTRACT
Global optimization consists in finding the best solution to a problem that may contain multiple sub-optimal solutions. Although it dates back to the origins of geometry, it is now extensively applied in various fields of engineering. This paper focuses on the deterministic algorithms that have been implemented in a generalized global optimization application. The application integrates the algorithms into a comprehensive modeling environment that uses the JUPITER API to communicate with any process model with batch processing capabilities. After a brief description of its structure and algorithms, the capabilities of the application are illustrated by finding the hydraulic property functions of unsaturated silica sand through inverse modeling of a multistep outflow experiment with both suction and cumulative outflow measurements. The canonical differential evolution algorithm is shown to be particularly well suited to this optimization problem in which the objective function is topographically complex. The estimated hydraulic property functions are also shown to be in excellent agreement with those independently determined using steady-state methods.

RÉSUMÉ
L'optimisation globale consiste à trouver la meilleure solution à un problème pouvant contenir une multitude de solutions non optimales. Bien qu'elle remonte aux origines de la géométrie, l'optimisation est maintenant utilisée dans différents domaines de l'ingénierie. Cet article porte sur les algorithmes déterministes qui ont été mis en œuvre dans une application d'optimisation globale généralisée. L'application intègre les algorithmes dans un environnement de modélisation complet qui utilise l'IPA JUPITER afin de communiquer avec tout modèle pouvant exécuter des traitements par lots. Après une brève description de sa structure et de ses algorithmes, les capacités de l'application sont illustrées en trouvant les propriétés hydrauliques d'un sable de silice non saturé par modélisation inverse d'un essai de laboratoire en régime transitoire avec mesures de la succion et du volume d'eau extrait. La version canonique de l'algorithme à évolution différentielle s'avère particulièrement bien adaptée à ce problème d'optimisation dont la fonction objective est topographiquement complexe. Les propriétés hydrauliques ainsi estimées s'avèrent être en excellent accord avec ceux déterminés indépendamment en utilisant des méthodes de mesure en régime permanent.

1 INTRODUCTION

Inverse modelling is one of the most important and well-studied subjects in science and engineering. In contrast to conventional modelling, it starts with the results and then determines the model parameters. To do so, the inverse problem is formulated as an optimization problem in which an objective function describes the differences between the observed and estimated model responses. In many cases, the objective function contains multiple local minima in which local optimization algorithms tend to get stuck. In an attempt to circumvent this difficulty, the global optimization algorithms search the entire parameter space in a more or less intelligent manner. The literature provides a number of different global optimization algorithms that can be divided in two main classifications: deterministic and stochastic algorithms. In the deterministic algorithms, the model parameters are fixed and unique whereas the parameters of stochastic algorithms are treated as random variables with joint posterior probability distributions. This paper focuses on the deterministic algorithms that have been implemented in a Generalized Global Optimization Application, called GGOA. All of these algorithms are based on evolutionary strategies in which the underlying idea is that alternative, or candidate, solutions play the role of individuals in a population that evolves within the environment defined by the objective function. These include the genetic algorithm, the canonical differential evolution algorithm and its self-adaptive variants.

The objectives of the paper are to (1) describe the structure and algorithms of the application, (2) present a versatile and precise experimental apparatus for conducting outflow experiments, (3) illustrate the capabilities of the application in estimating the hydraulic properties of uniformly-graded porous media from a multistep outflow experiment, and (4) evaluate the ability of Richards' equation to describe flow in the multistep outflow experiment.

2 DESCRIPTION OF THE APPLICATION

The purpose of the application is to find the model parameters that provide the best solution to a given linear or nonlinear problem. In order to accomplish this task, the best parameters are estimated by minimizing an objective function that expresses the difference between observed and computed model response. In a common approach,

the objective function is defined as the sum of weighted least-squares

$$\Phi\left(\vec{\beta}\right)=\sum_{j=1}^{m}\sum_{i=1}^{n_{\vartheta,j}}w_{i,j}\left[\vartheta_{i,j}-\hat{\vartheta}_{i,j}\left(\vec{\beta}\right)\right]^2 \qquad [1]$$

where $\vec{\beta}=\left(\beta_1,\beta_2,\beta_3,...,\beta_{k=n}\right)$ is the parameter vector, n is the number of parameters, $w_{i,j}$ is the weighting factor for the *i*th observation of *j*th type, $\mathcal{J}_{i,j}$ is the *i*th observation of *j*th type, $\hat{\mathcal{J}}_{i,j}$ is the *i*th estimation of *j*th type, m is the number of observation types, $n_{\mathcal{J},j}$ is the number of observations of *j*th type. The differences between observed and estimated (or simulated) values, $\Delta\vartheta=\vartheta_{i,j}-\hat{\vartheta}_{i,j}\left(\vec{\beta}\right)$, are called residuals. As formulated, the objective function is most appropriate when the residuals are uncorrelated and heteroscedastic (do not have the same variance). In order to provide independent residuals, the weighting factors should ideally be set equal to the inverse of the error variance-covariance matrix (Draper and Smith 1998). In the case of uncorrelated residuals, the weight matrix is diagonal and each non-zero element of the weight matrix equals one over the variance of measurement error, $w_{i,j}=1/s_{i,j}^2$.

In highly nonlinear problems, the objective function may be topographically complex, and contain multiple local minima. This makes it far more difficult to find the global minimum. To circumvent this difficulty, GGOA adopts a post-hybridization strategy in which a global optimization algorithm searches the entire parameter space, and a local optimization algorithm follows the path of least resistance towards the best solution. In so doing, the application inherits the advantages of both search methods. Once it has identified the best parameters, GGOA performs an extensive error analysis that provides statistical information about the residuals, and the ability to discriminate among model alternatives.

2.1 Application structure

Figure 1 shows the different steps of the generalized global optimization application, and its interaction with the process model. The application essentially consists of three main steps: (1) Initialize the problem by reading the application controls, model execution command lines, parameter bounds and observations. (2) Optimize the objective function. In this step, the optimization algorithm proposes new parameters that are transferred to the process model. The model is executed, the computed response is extracted, and the objective function is calculated. These steps are repeated until the end of the optimization process. (3) Compute relevant statistics. The application is written in Fortran, and uses JUPITER API modules (Banta et al. 2006) to input and exchange data, and interact with any process model with batch processing capabilities.

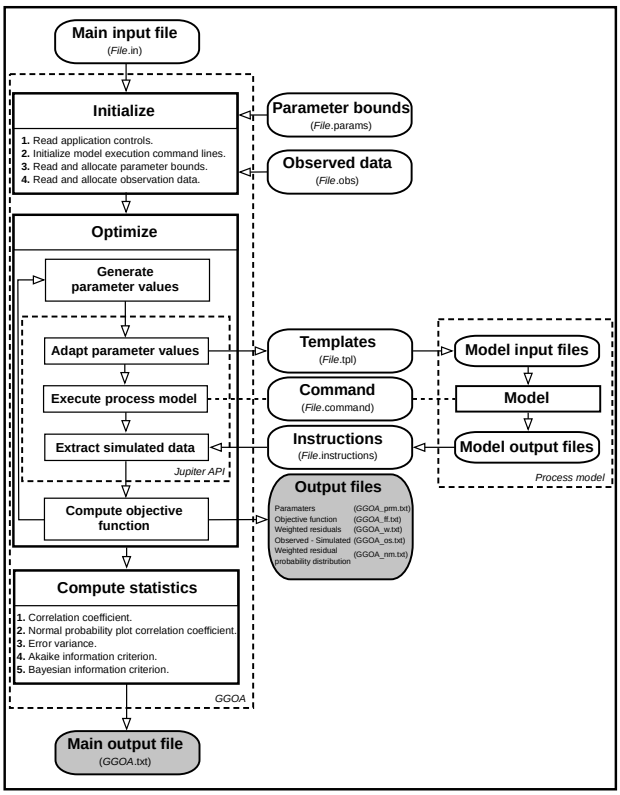

Figure 1. Flowchart showing the structure of the application, and its interaction with the process model.

2.2 Global optimization algorithms

The literature provides numerous different evolutionary algorithms in which candidate solutions play the role of individuals in a population that evolves within the environment defined by the objective (or fitness) function. Based on the fitness of the various candidate solutions, some are selected to seed the next generation by applying a number of mechanisms inspired by biological evolution (crossover, mutation, and selection). These algorithms may generally be differentiated by their representation of the candidate solutions. For instance, the genetic algorithm uses strings to represent the candidate solutions whereas real-valued vectors are used in the differential evolution algorithm. Both of these types of algorithms are implemented into GGOA.

2.2.1 Genetic algorithm, GA

GAs were introduced by John H. Holland in the latter portion of the twentieth century, and have since then been applied to a number of inverse modelling problems in geotechnical engineering (McCombie and Wilkinson 2002; Levasseur et al. 2008). GAs generally differ in their modes of selection, crossover and/or mutation. The current version of GGOA implements the most recent form of the PIKAIA algorithm (Charbonneau 2002). In this and the other global optimization algorithms, the parameters are defined in a floating-point interval, such that $b_k \in\left[0,1\right]$. This implies that the parameters must be

transformed prior to model execution, and data exchange. In this particular algorithm, each parameter vector (or individual) is encoded as a chromosome-like structure using 1-digit base 10 integers. In the case of a *n*-dimensional parameter vector with n_d significant digits (or genes), the encoding process produces a one-dimensional integer array (or chromosome) of length $n' n_d$. In order to avoid any bias, the algorithm starts by initializing each of the one-dimensional arrays with a random number generator. The fitness of each individual is then calculated and used to rank the individuals from best to worst. By convention, the rank $R \in [1, n_p]$ where n_p is the number of individuals in the population, is set equal to 1 for the fittest individual. The following sections describe the most important components of the PIKAIA algorithm.

Parent selection technique

Although there are many different types of selection techniques, PIKAIA uses what is known as *roulette wheel selection*. In this technique, each individual is thought of as a pocket on a roulette wheel, where the size of the pocket is proportional to its probability of selection. Given that $F_l = (fdif + 1)(n_p + 1) - 2R_l\, fdif$ is the relative fitness of the *l*th individual with relative fitness differential *fdif*, its probability of being selected is $p_l = F_l / \sum_{l=1}^{n_p} F_l$. Parents are drawn independently by generating a random number in the $[0,1]$ interval. In this manner, individuals with lesser fitness may be selected as parents. This may be beneficial as less fit individuals may include some genes that could prove useful following the crossover process. In general, however, individuals with greater fitness are more likely to be selected as parents. Therefore, genes that result in greater fitness will become more abundant in the next generation.

Crossover operators

Once individuals have been selected as parents, they must take part in the act of reproduction in order to produce offspring. The most common way to generate these offspring is through an operation called crossover. Although there are many kinds of crossover, PIKAIA incorporates both one-point crossover and two-point crossover. These crossover operators act on the pair of parent-chromosomes to produce a pair of offspring-chromosomes. The one-point crossover operator begins by randomly selecting a cutting point along the chromosomes by generating an integer in the $[1, n \times n_d]$ interval. Once cut, the chromosomal fragments located right of the cutting point are exchanged from one parent to the other, and concatenated to the fragments left of the cutting point. This complex operation is best understood through an example. Let us therefore consider two prototypical parents in a two-dimensional parameter space: $(b_1, b_2)_1$ = (0.256359,0.489238), and $(b_1, b_2)_2$ = (0.134787,0.731686). Encoding these parents with six significant digits produces the following parent-chromosomes: 256359489238 and 134787731686. If the randomly selected crossover point is equal to three, the parent-chromosomes read as follows 256 | 359489238 and 134 | 787731686. The sections of chromosome from each parent are then used to form the following offspring-chromosomes: 256787731686 and 134359489238. In the case of two-point crossover, two cutting points are randomly selected along the chromosomes. The fragments located in between these points are then exchanged in a manner otherwise identical to that described for one-point crossover.

As can be seen from the example for one-point crossover, this type of operator has a positional bias whereby the probability of a gene being swapped is dependent on the position of that gene within the chromosome. Given that positional bias generally decreases as the number of crossover points increases, PIKAIA adds a probabilistic test to the crossover operation through which either one-point or two-point crossover is chosen with equal probability. It is to be noted that crossover does not always occur but is rather a consequence of a probabilistic test, which depends on a user defined crossover rate. To ensure more diversity, and to avoid premature convergence, the crossover rate is generally set equal to 0.85. In the event that crossover does not occur, the parent-chromosomes are copied directly as offspring-chromosomes.

Mutation operator

The offspring-chromosomes are now subjected to mutation in order to preserve diversity from one generation to the next. In general, mutation operators involve a probability that an arbitrary gene will be changed from its original state. PIKAIA incorporates two mutation operators known as uniform mutation and creep mutation. In these mutation operators, the replacement of each of the genes of a chromosome is a consequence of a probabilistic test that depends on a user defined mutation rate, which will be discussed below. In the event of mutation, the uniform mutation operator substitutes the gene by a random integer in the $[0,9]$ interval. In the case of creep mutation, the gene is either subjected to a unit increment or decrement and this, with equal probability. If the gene to be incremented is a 9, it becomes 0 and the gene located to its left is incremented by unity or, then again, it is also set equal to 0 if the gene happens to be a 9. This process is repeated as many times as needed or until the left boundary of the substring is reached, in which case the genes are reverted to their initial values. Given that uniform mutation results in greater exploration of the parameter space, it is either used separately or in conjunction with creep mutation.

Although the mutation rate may be set constant, experience has shown that the selection of an optimal mutation rate is extremely problem-dependant. In this context, it may be preferable to use a dynamically adjusted mutation rate that offers the advantage of self-adapting the balance between the need to explore the parameter space at early stages, and to refine well adapted popula-

tions at the end of the evolution process. In order to adjust the mutation rate, PIKAIA monitors the degree of clustering of the population at the end of each generational iteration. The degree of clustering is either evaluated in terms of the normalized fitness difference between the best and median individuals

$$\Delta_{\Phi} = \frac{\Phi\left(\vec{\beta}\right)_{r=1} - \Phi\left(\vec{\beta}\right)_{r=n_p/2}}{\Phi\left(\vec{\beta}\right)_{r=1} + \Phi\left(\vec{\beta}\right)_{r=n_p/2}} \quad [2]$$

or, the metric distance between the best and median individuals

$$\mathrm{D}_d = \frac{1}{n}\left[\sum_{k=1}^{n}\left(b_{k,r=1} - b_{k,r=n_p/2}\right)^2\right]^{1/2} \quad [3]$$

As long as the degree of clustering is superior to 0.05 and inferior to 0.25, the mutation rate remains equal to its initial value, which is usually set equal to 0.005. Whenever the degree of clustering falls below the lower limit (or above the upper limit), the mutation rate is increased (or decreased) by a factor of 1.5, and may not exceed user defined maximum and minimum limits. Although fairly simple, mutation is essential in preventing premature convergence to a local optimum.

Survivor selection technique

Survivor selection, often called replacement, determines which offspring will be allowed in the next generation. As opposed to parent selection, which is typically stochastic, survivor selection is most often deterministic. For instance, the generational survivor selection technique consists in selecting only the offspring. In such a case, individuals have a fixed lifetime equal to a single generation. Steady-state survivor selection differs from the generational technique in that the offspring are inserted into the population as soon as they are generated. A deletion strategy then defines which individual of the population will be replaced by the offspring. PIKAIA incorporates two different deletion strategies: *delete-worst* and *delete-random*. Under the steady-state-delete-worst technique, the least fit member of the population is deleted and replaced with the offspring. In the steady-state-delete-random technique, the offspring replaces a randomly selected individual. Hence, the fittest individual is only guaranteed to survive under the steady-state-delete-worst technique. Yet, a strategy known as elitism may be used with the other techniques to ensure that the fittest individual is passed on to the next generation. It is to be noted that only the full generational survivor selection technique lends itself to a parallel implementation as the fitness of the offspring are computed in a single step.

2.2.2 Differential evolution algorithm, DE

Storn and Price (1995) introduced DE as a simple direct search approach for minimizing nonlinear and non-differentiable functions. It has since then proven itself in a number of competitions (Price 1996) and geotechnical problems (Zhao et al. 2015). In contrast to GAs, which are inspired by genetic evolution, DE was entirely derived from geometrical considerations. As with other evolutionary algorithms, DE generates new parameter vectors (or individuals) by perturbing existing individuals. In order to do so, every *l*th individual of the *m*th generation is represented by a *n*-dimensional parameter vector, $\vec{\beta}_{l,m} = \left(\beta_1, \beta_2, \beta_3, ..., \beta_{k=n}\right)_{l,m}$. The first parameter values are chosen randomly so that the n_p vectors of the population cover the entire parameter space. For each target vector, $\vec{\beta}_{l,m}$, the difference of one or two pairs of randomly selected parameter vectors is used as the source of random variation (or mutation) for a non-mutated parameter vector. As described in the following sections, selection of the non-mutated parameter vector depends on the type of mutation operator. Once this task is completed, the mutated vector, $\vec{\mathrm{B}}_{l,m}$, is combined with the target vector to form a trial vector, $\vec{\beta}^{*}_{l,m+1}$. This operation, referred to as crossover, is followed by a fitness biased survivor selection technique whereby the fittest of the trial and target parameter vectors is passed on to the next generation. The following sections describe the various mutation and crossover operators implemented in GGOA.

Mutation operator

The mutated vector can be created with a number of different mutation operators. Of all mutation operators, the most useful are (Das and Suganthan 2011):

$$\begin{aligned}
&\text{rand/1:} && \vec{\mathrm{B}}_{l,m} = \vec{\beta}_{r_1,m} + F\left(\vec{\beta}_{r_2,m} - \vec{\beta}_{r_3,m}\right)\\
&\text{best/1:} && \vec{\mathrm{B}}_{l,m} = \vec{\beta}_{best,m} + F\left(\vec{\beta}_{r_1,m} - \vec{\beta}_{r_2,m}\right)\\
&\text{current to best/1:} && \vec{\mathrm{B}}_{l,m} = \vec{\beta}_{l,m} + F\left[\left(\vec{\beta}_{r_1,m} - \vec{\beta}_{r_2,m}\right) + \left(\vec{\beta}_{best,m} - \vec{\beta}_{l,m}\right)\right]\\
&\text{best/2:} && \vec{\mathrm{B}}_{l,m} = \vec{\beta}_{best,m} + F\left[\left(\vec{\beta}_{r_1,m} - \vec{\beta}_{r_2,m}\right) + \left(\vec{\beta}_{r_3,m} - \vec{\beta}_{r_4,m}\right)\right]\\
&\text{rand/2:} && \vec{\mathrm{B}}_{l,m} = \vec{\beta}_{r_1,m} + F\left[\left(\vec{\beta}_{r_2,m} - \vec{\beta}_{r_3,m}\right) + \left(\vec{\beta}_{r_4,m} - \vec{\beta}_{r_5,m}\right)\right]
\end{aligned} \quad [4]$$

where indices r_1, r_2, r_3, r_4, and r_5 are random and mutually exclusive integers in the $\left[1, n_p\right]$ interval, which must be different from the target vector index. F is a scaling factor in the $\left[0,2\right]$ interval that controls the amplification of the differential variation, and $\vec{\beta}_{best,m}$ is the fittest (or best) parameter vector of the *m*th generation.

The performance of the various mutation operators depends on the selection of an appropriate scaling factor. As highlighted by Storn and Price (1997), scaling factors smaller than 0.4 and larger than 1.0 are only occasionally effective when using the rand/1 mutation operator. Although an appropriate scaling factor may be determined by trial-and-error, it is often preferable to use a dynamically adjusted scaling factor that self-adapts to the need for local and global search capabilities. Such strategies

are included in a number of self-adaptive variants of DE (Das and Suganthan 2011). For instance, the jDE algorithm self-adapts the scaling factor of the rand/1 mutation operator (Brest et al. 2006). The strategy consists in using either the scaling factor of the *l*th parameter vector for the *m*th generation or a randomly generated scaling factor that lies in the $\left[0.1,1\right]$ range. The replacement of the scaling factor is a consequence of a probabilistic test with a ten percent replacement probability. In the case of the SaDE algorithm, the scaling factor of the *l*th parameter vector is randomly selected within a normal distribution, $N\left[0.5,0.09\right]$, and constrained to the $\left(0,2\right]$ range (Qin and Suganthan 2005). This strategy is either applied to the rand/1 or current-to-best/1 mutation operators. In fact, SaDE probabilistically selects the mutation operator of a given parameter vector. The replacement probability depends on the success rate of each of the mutation operators for the parameter vector, and is updated at every fiftieth generation.

Crossover operator

To complement the mutation operator, DE also uses a crossover operator that builds trial vectors with the target and mutated parameter vectors. In the uniform (or binomial) crossover operation, used in GGOA, the replacement of each of the parameters of the target vector is the consequence of a probabilistic test that depends on a user-defined crossover constant. In the event of crossover, the operator substitutes the parameter of the target vector with that of the mutated vector. In order to ensure that the trial vector obtains at least one parameter of the mutated vector, crossover is also forced to occur at a randomly generated index. As with other control parameters, the crossover constant is problem-dependent and should be determined by trial-and-error. Alternatively, one may use self-adaptive variants of DE. The adaptive strategy of the jDE algorithm consists in using either the crossover constant of the *l*th parameter vector for the *m*th generation or a randomly generated constant in the $\left[0,1\right]$ range. The replacement of the crossover constant is a consequence of a probabilistic test with a ten percent replacement probability. In the case of the SaDE algorithm, the crossover constant is randomly selected within a normal distribution, $N\left[m,0.01\right]$, at every fifth generation. The mean, m, of the normal distribution is initially set equal to 0.5, and is updated every twenty-fifth generation using the crossover constants that resulted in successful trial vectors.

2.3 Local optimization algorithm

Nelder and Mead (1965) introduced the downhill simplex method for minimizing functions without derivatives. In essence, the algorithm defines a simplex that consists of $n+1$ vertices, where n is the number of parameters. It then evaluates the objective function at each vertex, and replaces the worst vertex by moving this vertex through the opposite face of the simplex. The method also expands or contracts the simplex as needed. GGOA uses the best solution of the global optimization algorithm as one of the vertices, and randomly choses the other vertices within the range defined by the parameter bounds. It must be noted that the algorithm is programmed in accordance with that found in Numerical Recipes (Press et al. 1996).

2.4 Statistical indicators

A number of indicators are used to evaluate the ability of the model to realistically represent the simulated system. The first of these indicators is the calculated error variance (Cooley and Naff 1990):

$$s^2 = \Phi\left(\vec{\beta}\right)\Big/\left(n_\vartheta - n\right) \qquad [5]$$

and its ninety-five percent confidence interval (Ott and Longnecker 2010):

$$\left[\frac{\left(n_J - n\right)s^2}{c^2_{n_J - n,0.025}}, \frac{\left(n_J - n\right)s^2}{c^2_{n_J - n,0.975}}\right] \qquad [6]$$

where $n_J = \sum_{j=1}^{m} n_{J,j}$ is the total number of observations, and $c^2_{n_J - n,0.025}$, $c^2_{n_J - n,0.975}$ are the upper and lower tail values of the chi-square distribution. As pointed out by Hill and Tiedeman (2007), the calculated error variance is sometimes criticized for not sufficiently representing the drawbacks associated with increasing the number of estimated parameters. The Akaike information criterion, AIC, and Bayesian information criterion, BIC, were specifically developed to address this criticism. These statistics consist in adding the maximum likelihood objective function, $\Phi'\left(\vec{\beta}\right)$, to various terms that become large when the number of estimated parameters increases. The AIC and BIC statistics are herein calculated as (Carrera and Neuman 1986):

$$\mathrm{AIC} = \Phi'\left(\vec{\beta}\right) + 2n = \left[n_\vartheta \ln\left(\Phi\left(\vec{\beta}\right)\Big/n_\vartheta\right)\right] + 2n \qquad [7]$$

and

$$\mathrm{BIC} = \Phi'\left(\vec{\beta}\right) + \ln\left(n_\vartheta\right) = \left[n_\vartheta \ln\left(\Phi\left(\vec{\beta}\right)\Big/n_\vartheta\right)\right] + \ln\left(n_\vartheta\right) \qquad [8]$$

It must be emphasized that the AIC and BIC statistics support the principle of parsimony in that, everything being equal, the model with the smallest number of parameters is the most acceptable (Russo 1988).

3 SAMPLE CASE STUDY

The use of rigorous models that describe flow in porous media is often hindered by a lack of knowledge of the unsaturated hydraulic properties. While many laboratory and field methods are now available for determining these highly nonlinear properties, most of the methods require that experiments reach steady-state or equilibrium conditions. This can make measurements time consuming and

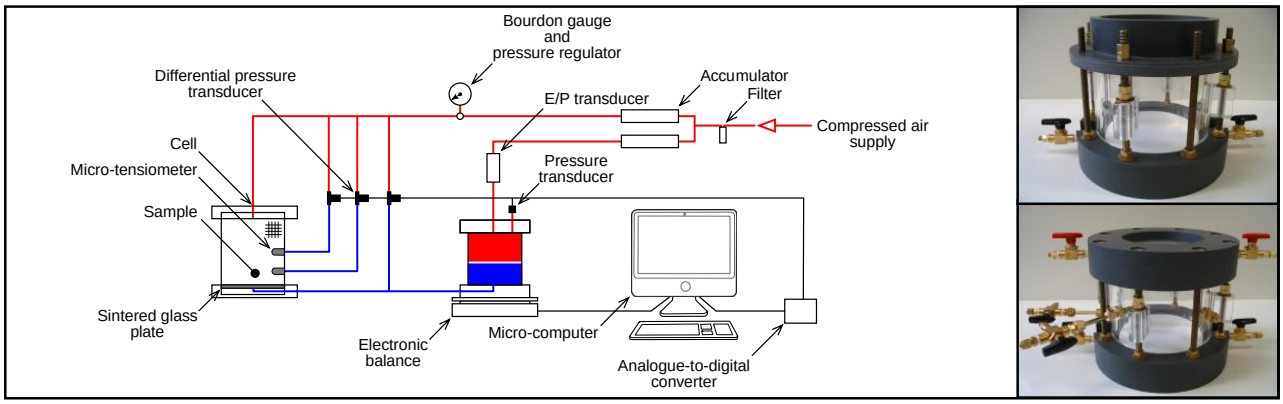

Figure 2. Diagram and photographs of the outflow experiment apparatus.

expensive. Rather than measuring the properties directly, the inverse or parameter estimation method estimates the unsaturated hydraulic properties from transient experiments that allow for considerable time saving, and greater flexibility in the choice of boundary conditions. These experiments are carried out under controlled conditions with measurements of various flow variables. A process model that simulates the transient flow regime of the experiment is then iteratively solved to find the best parameters of the functional relations that describe the unsaturated hydraulic properties.

Whisler and Watson (1968) were among the first to suggest using the inverse method to estimate the parameters of functional relations describing the hydraulic properties of unsaturated porous media. More than a decade later, Zachmann et al. (1981) applied the inverse method to a numerical solution of Richards' equation for artificially generated transient drainage experiments. A number of studies have since then focused on the inverse modeling of outflow experiments. Additional background information can be found in Hopmans et al. (2002).

3.1 Materials and methods

3.1.1 Soil sample

Outflow experiments were conducted with standard 20/30 silica sand manufactured by Unimin Corporation. Features of the sand particles include chemical purity, low organic content, high particle sphericity, and a narrow particle-size distribution with predominant particle-sizes ranging from 600 to 850 μm. The relative density of the silica sand, $D_R = 2.65$, is found to be similar to that of pure quartz.

To establish an initial condition of complete saturation, the sample was prepared by wet pluviation with de-aired and de-mineralized water. The sample was then compacted by placing the cell on a vibratory table for eight (8) minutes. This process resulted in a completely water saturated sample with a porosity of 0.35.

3.1.2 Multistep outflow experiment

Overview

The experimental apparatus was based on the axis translation technique. In contrast to most studies, however, the pneumatic pressure was kept constant while applying a series of step decreases of pore-water pressure at the lower boundary of the pressurized cell.

After preparing the sample by wet pluviation, the suction was increased to a value that slightly exceeded the air-entry value of the porous media. This resulted in a continuous gaseous phase, which reduced the problems associated with the occurrence of non-uniform flow at the start of the multistep experiment (Hopmans et al. 1992). The sample was subsequently subjected to five (5) consecutive computer-controlled steps of increasing suction. The applied values of suction were 0.91, 1.12, 1.31, 1.51, and 1.71 kPa. Cumulative outflow and suction within the cell were sampled simultaneously at 10 s intervals. As is generally recommended, approximately 100 validated space-time flow variables of each measurement type were selected exponentially.

Details

Figure 2 shows a diagram of the experimental apparatus with gas and water flow controls. The core of the apparatus consisted of a large cell with two (2) micro-tensiometers inserted at $H/2$ and $H/4$ from the bottom of the soil sample. Though the quality of data from both micro-tensiometers was satisfactory, suction within the cell was taken from that measured at $H/4$. The internal dimensions of the cell, 11.64 cm in height and 12.54 cm in diameter, were set so as to comply with the representative elementary volume (REV) of a wide range of cohesionless materials. In addition to the acrylic cylinder, the cell consisted of two (2) acrylic receptacles that were assembled into polyvinyl chloride bases. The lower receptacle was fitted with a low impedance sintered glass plate. During sample preparation, the upper receptacle and base were substituted by a polyvinyl chloride extension collar. As can be seen from the photographs of Figure 2, the micro-tensiometers were laterally offset to minimize flow disturbance. Apart from the precisely machined brass tubes, the micro-tensiometers were made of commercially available components. To obtain rapid response times, the micro-tensiometers were connected to temperature compensated differential pressure transducers by short pieces of tubing. The pressure transducers were calibrated with liquid manometers. The observed standard deviation of suction measurement error was equal to 0.076 kPa.

A precision pressure regulator was used to set a constant pneumatic pressure while suction at the lower boundary was controlled with a miniature E/P transducer and a proportional-integral-derivative controller. Suction measurement at the lower boundary was carried out with a limited range differential pressure transducer, which allowed for a precise dynamic control of suction.

To obtain a fully computer-automated system, the bottom boundary of the cell was connected to a pressurized vessel, which was mounted on an electronic balance. A layer of silicon fluid was used to inhibit air dissolution into the de-aired and de-mineralized water of the vessel. The measurements provided by the electronic balance were converted into cumulative outflow values and corrected for vessel and tubing volume changes as well as air density variations. As the experiment was conducted in a temperature controlled laboratory environment, air density was taken to be a function of the measured pressure within the vessel. The observed standard deviation of outflow measurement error was equal to 5.548×10^{-6} m³/m².

3.1.3 Process model

The movement of water in the unsaturated sand is assumed to follow the mixed-form of Richards' equation (Celia et al. 1990), which can be written as follows

$$\frac{\partial q}{\partial t} = \frac{\partial}{\partial z}\left[k\left(\frac{\partial h_m}{\partial z}+1\right)\right] \qquad [9]$$

where q is the volumetric water content, t is the time, z is the spatial coordinate, k is the hydraulic conductivity, and h_m is the matric head. The lower boundary condition consists in a stepwise decrease of matric head, which is computed from the imposed values of suction. The hydraulic properties, on the other hand, are described with the following functional relations (Kosugi 1996):

$$q = \left(q_s - q_r\right)\left[\frac{1}{2}\mathrm{erfc}\left(\frac{\ln\left(h_m/h_{m,median}\right)}{\sqrt{2}s}\right)\right] + q_r \qquad [10]$$

and

$$k = k_s k_r = k_s \mathrm{Q}^L\left[\frac{1}{2}\mathrm{erfc}\left(\frac{\ln\left(h_m/h_{m,median}\right)}{\sqrt{2}s}\right)+\frac{s}{\sqrt{2}}\right]^2 \qquad [11]$$

where q_s is the saturated volumetric water content, q_r is the residual volumetric water content, $\mathrm{erfc}(\ )$ is the complementary error function, $h_{m,median}$ is the matric head that corresponds to the median pore radius, s is the standard deviation of the log-transformed pore-radius distribution, k_s is the saturated hydraulic conductivity, k_r is the relative hydraulic conductivity, $\mathrm{Q} = \left(q - q_r\right)/\left(q_s - q_r\right)$ is the normalized volumetric water content, and L is a lumped parameter that accounts for both pore tortuosity and connectivity. The computed values of matric head at the position of the micro-tensiometer are internally converted into suction for inverse modelling purposes. It must also be noted that the equations are solved with a script-driven partial differential equation solver called FlexPDE (PDE Solutions 2011).

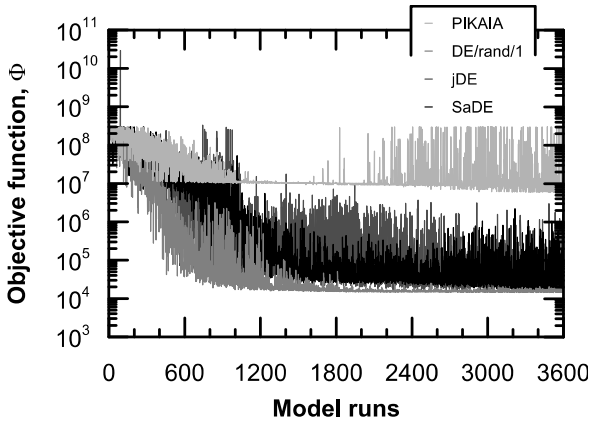

Figure 3. Convergence graph for the multistep outflow problem.

3.1.4 Inverse modelling

The inverse or parameter estimation process begins by selecting the parameter vector and its associated bounds. As is often the case, the saturated volumetric water content is set equal to the independently determined value of porosity. The parameter vector therefore contains parameters q_r, $h_{m,median}$, s, k_s, and L. Each global optimization algorithm is then used to find the best solution over 50 generations using default algorithm settings, and a population of 100 individuals.

3.2 Results

3.2.1 Algorithm performance

Figure 3 shows the evolution of the objective function with the number of model runs. In this particular case, PIKAIA converges towards a local minimum whereas the canonical DE outperforms its self-adaptive variants by reaching the minimum in less model runs. The parameters of this best solution are q_r = 0.034 $h_{m,median}$ = -0.11 m, s = 0.18, k_s = 6.68×10^{-4} m/s, and L = 0.25.

3.2.2 Model performance

Figure 4 shows the applied lower boundary condition with the resulting suction and cumulative outflow. As expected, the experimental apparatus provides outstanding control of the lower boundary condition. A close inspection of the observed values of suction also reveals that the static equilibrium condition, illustrated by the dotted line in Figure 4, is reached in the first three steps of the experiment. Yet, as often occurs in coarse-textured porous media, static equilibrium conditions are not reached at larger values of suction. The simulated values

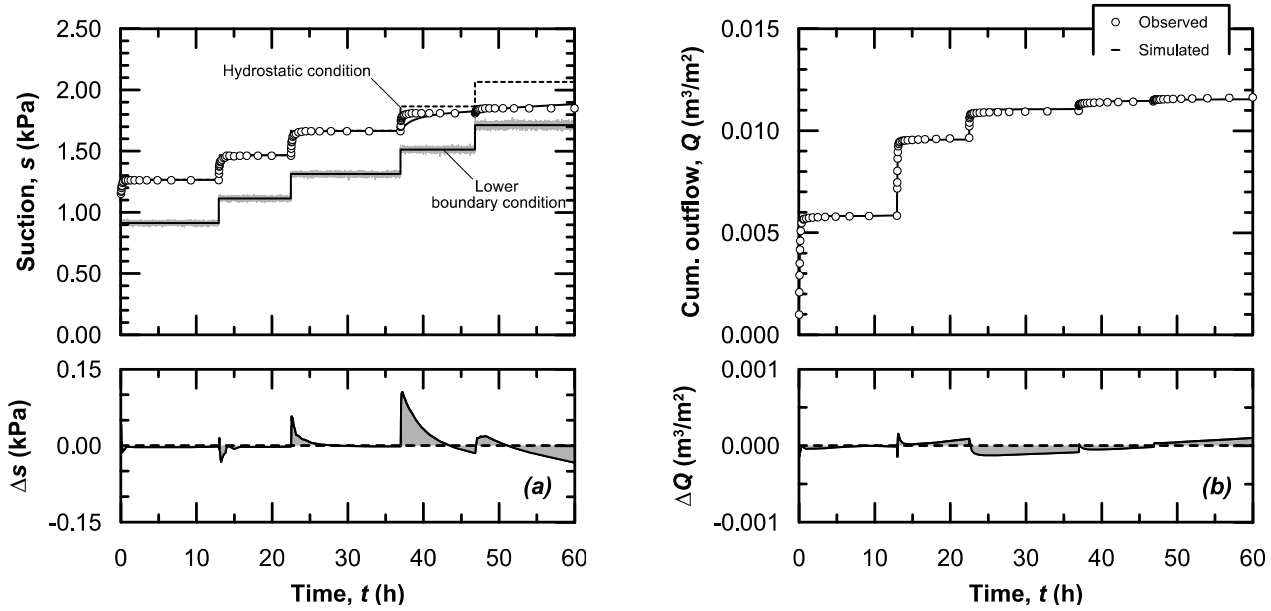

Figure 4. Results of the multistep outflow analysis. (a) Suction. (b) Cumulative outflow.

are nonetheless in very close agreement with the observed data. The residual plots reveal that most of the model errors occur at the onset of the stepwise increases in suction. The coefficient of determination, r^2, provides a measure of how closely these residuals lie along a straight line. For the data at hand, the coefficient of determination is equal to 0.99. This suggests that the numerical model quite accurately reproduces the general dynamics of the outflow experiment.

Figure 5 provides a comparison of the estimated hydraulic properties with those independently determined by Schroth et al. (1996). Although determined with very different experimental methods, the estimated water retention function shows good agreement with the functional representation of the independent data. It must be noted that the independently determined water retention data was obtained through the achievement of a sequence of static hydraulic equilibrium, and is therefore treated as the integral of the soil-water distribution over the height of the cell (Schroth et al. 1996). For this reason, large differences are observed between the independently determined data and its functional relation. The estimated relative hydraulic conductivity is also in good agreement with the independently determined data, which was measured with the steady-state flux control method described by Klute and Dirksen (1986). It must be emphasized that the estimated value of saturated hydraulic conductivity has no physical meaning since the multistep outflow experiment was started under unsaturated conditions. These results clearly indicate that the parameter estimation method with outflow experiments is an attractive and rapid alternative to conventional methods, which allows for the simultaneous determination of both the water retention and hydraulic conductivity functions.

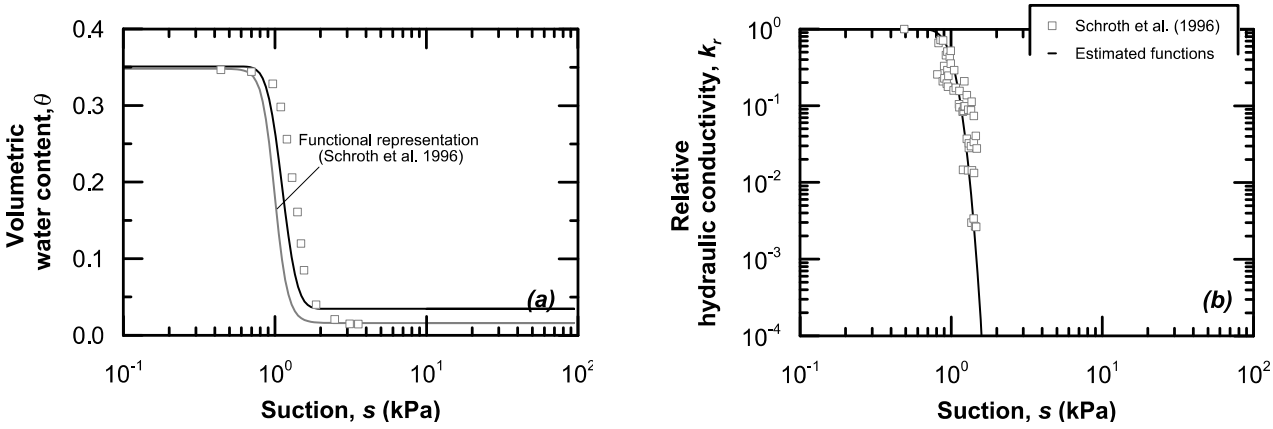

Figure 5. Hydraulic properties of the 20/30 silica sand. (a) Water retention function. (b) Hydraulic conductivity function.

3.3 Discussion

GGOA performs an extensive error analysis that provides statistical information about the residuals, and numerous indicators of the ability of the model to represent the simulated system. The first of these measures is the calculated error variance, which is an indicator of the overall magnitude of the weighted residuals and is expected to be equal to one when the regression is consistent with data accuracy. In this particular case, the calculated error variance is equal to 77.19, which implies that the residuals are much larger than the error of the measurement devices. As the ninety-five percent confidence interval of the calculated error variance is significantly larger than one, which is common, it may be concluded that the model fit is inferior to that expected based on the analysis of error used to determine the weighting. Similar conclusions may also be drawn from the AIC and BIC information criteria.

Given that the residuals are not randomly distributed, the calculated error variance suggests that model error is significant. Vrugt et al. (2003) point out that the majority of process models fail this type of adequacy test, and that this does not necessarily mean that the estimated parameter values are meaningless. In fact, the results of this study clearly demonstrate that these types of models possess excellent predictive capabilities. Yet, the Akaike and Bayesian information criteria may be used to compare alternative models, such as those obtained when using functional relations that account for film flow (Lebeau and Konrad 2010), or then again, when accounting for dynamic nonequilibrium processes (Diamantopoulos and Durner 2012).

4 CONCLUSION

Inverse modelling plays an important role in parameter estimation and model development. This paper presented the structure and algorithms of a generalized global optimization application that possesses the ability to communicate with any process model with batch processing capabilities. The capabilities of the application were illustrated through a sample inverse modelling problem in which the hydraulic properties of unsaturated silica sand were estimated using the results of a multistep outflow experiment. Although the estimated hydraulic properties were in agreement with those independently determined using steady-state methods, the extensive error analysis of the application revealed that the process model error was significant.

ACKNOWLEDGEMENTS

The authors gratefully acknowledge the financial support of NSERC-Hydro-Québec Industrial Research Chair for life cycle optimisation of embankment dams. The authors also extend their appreciation to their industrial partners: Hydro-Québec, SNC-Lavalin, WSP, Qualitas, AECOM, Klohn Crippen Berger, Golder Associates, and ConeTec. Finally, the authors thank Nicolas Venkovic for his help in programming and testing the application.

REFERENCES

Banta, E.R., Poeter, E.P., Doherty, J.E., and Hill, M.C. 2006. JUPITER: Joint Universal Parameter IdenTification and Evaluation of Reliability - An Application Programming Interface (API) for Model Analysis. *In* Techniques and Methods 6-E1. U.S. Geological Survey. p. 268.

Brest, J., Greiner, S., Boškovic, B., Mernik, M., and Žumer, V. 2006. Self-adapting control parameters in differential evolution: A comparative study on numerical benchmark problems. IEEE Transactions on Evolutionary Computation **10**(6): 646-657.

Carrera, J., and Neuman, S.P. 1986. Estimation of aquifer parameters under transient and steady state conditions: 1. Maximum likelihood method incorporating prior information. Water Resources Research **22**(2): 199-210.

Celia, M.A., Bouloutas, E.T., and Zarba, R.L. 1990. A general mass-conservative numerical solution for the unsaturated flow equation. Water Resources Research **26**(7): 1483-1496.

Charbonneau, P. 2002. Release notes for PIKAIA 1.2. High Altitude Observatory, National Center for Atmospheric Research, Boulder, Colorado. p. 49.

Cooley, R.L., and Naff, R., L. 1990. Regression modeling of ground-water flow. U.S. Geological Survey. pp. 232.

Das, S., and Suganthan, P.N. 2011. Differential evolution: A survey of the state-of-the-art. IEEE Transactions on Evolutionary Computation **15**(1): 4-31.

Diamantopoulos, E., and Durner, W. 2012. Dynamic nonequilibrium of water flow in porous media: A review. Vadose Zone Journal **11**(3).

Draper, N.R., and Smith, H. 1998. Applied regression analysis. 3rd ed. John Wiley & Sons, Inc. pp. 736.

Hill, M.C., and Tiedeman, C.R. 2007. Effective groundwater model calibration with analysis of data, sensitivites, predictions, and uncertainty. John Wiley & Sons, Inc. pp. 455.

Hopmans, J.W., Šimůnek, J., Romano, N., and Durner, W. 2002. Inverse methods. *In* Methods of soil analysis. Part 4. Physical methods. *Edited by* J.H. Dane and G.C. Topp. Soil Science Society of America. pp. 963-1008.

Hopmans, J.W., Vogel, T., and Koblik, P.D. 1992. X-ray tomography of soil water distribution in one-step outflow experiments. Soil Science Society of America Journal **56**(2): 355-362.

Klute, A., and Dirksen, C. 1986. Hydraulic conductivity and diffusivity: Laboratory methods. *In* Methods of Soil Analysis: Part 1-Physical and Mineralogical Methods. *Edited by* A. Klute. Soil Science Society of America, American Society of Agronomy, Madison, WI. pp. 687-734.

Kosugi, K. 1996. Lognormal distribution model for unsaturated soil hydraulic properties. Water Resources Research **32**(9): 2697-2703.

Lebeau, M., and Konrad, J.-M. 2010. A new capillary and thin film flow model for predicting the hydraulic conductivity of unsaturated porous media. Water Resources Research **46**(12): W12554.

Levasseur, S., Malécot, Y., Boulon, M., and Flavigny, E. 2008. Soil parameter identification using a genetic algorithm. International Journal for Numerical and Analytical Methods in Geomechanics **32**(2): 189-213.

McCombie, P., and Wilkinson, P. 2002. The use of the simple genetic algorithm in finding the critical factor of safety in slope stability analysis. Computers and Geotechnics **29**(8): 699-714.

Nelder, J.A., and Mead, R. 1965. A simplex method for function minimization. The Computer Journal **7**(4): 308-313.

Ott, R.L., and Longnecker, M. 2010. An introduction to statistical methods and data analysis. 6th ed. Brooks/Cole. pp. 1273.

PDE Solutions. 2011. FlexPDE reference manual. Spokane Valley, WA.

Press, W.H., Teukolsky, S.A., Vetterling, W., T., and Flannery, B.P. 1996. Numerical recipes in Fortran 90. The art of parallel scientific computing. 2nd ed. Cambridge University Press.

Price, K.V. 1996. Differential evolution: a fast and simple numerical optimizer. *In* Biennial Conference of the North American Fuzzy Information Processing Society. pp. 524-527.

Qin, A.K., and Suganthan, P.N. 2005. Self-adaptive differential evolution algorithm for numerical optimization. *In* The 2005 IEEE Congress on Evolutionary Computation pp. 1785-1791.

Russo, D. 1988. Determining soil hydraulic properties by parameter estimation: On the selection of a model for the hydraulic properties. Water Resources Research **24**(3): 453-459.

Schroth, M.H., Ahearn, S.J., Selker, J.S., Istok, J.D. 1996. Characterization of Miller-similar silica sands for laboratory hydrologic studies. Soil Science Society of America Journal **60**(5): 1331-1339.

Storn, R., and Price, K. 1995. Differential evolution - A simple and efficient adaptive scheme for global optimization over continuous spaces. International Computer Science Institute, Berkeley, California. p. 12.

Storn, R., and Price, K. 1997. Differential evolution - A simple and efficient heuristic for global optimization over continuous spaces. Journal of Global Optimization **11**(4): 341-359.

Vrugt, J.A., Bouten, W., Gupta, H.V., and Hopmans, J.W. 2003. Toward improved identifiability of soil hydraulic parameters. Vadose Zone Journal **2**(1): 98-113.

Whisler, F.D., and Watson, K.K. 1968. One-dimensional gravity drainage of uniform columns of porous materials. Journal of Hydrology **6**(3): 277-296.

Zachmann, D.W., DuChateau, P.C., and Klute, A. 1981. The calibration of the Richards flow equation for a draining column by parameter identification. Soil Science Society of America Journal **45**(6): 1012-1015.

Zhao, B.D., Zhang, L.L., Jeng, D.S., Wang, J.H., and Chen, J.J. 2015. Inverse analysis of deep excavation using differential evolution algorithm. International Journal for Numerical and Analytical Methods in Geomechanics **39**(2): 115-134.