Comput Geosci (2017) 21:1325–1342
DOI 10.1007/s10596-017-9657-9

ORIGINAL PAPER

Distributed Gauss-Newton optimization method for history matching problems with multiple best matches

Guohua Gao[1] · Jeroen C. Vink[2] · Chaohui Chen[3] · Yaakoub El Khamra[1] · Mohammadali Tarrahi[1]

Received: 17 June 2016 / Accepted: 26 April 2017 / Published online: 8 May 2017

Abstract Minimizing a sum of squared data mismatches is a key ingredient in many assisted history matching (AHM) workflows. A novel approach is developed to efficiently find multiple local minima of a data mismatch objective function, by performing Gauss-Newton (GN) minimizations concurrently while sharing information between dispersed regions in the reduced parameter space dynamically. To start, a large number of different initial parameter values (i.e., model realizations) are randomly generated and are used as initial search points and base-cases for each subsequent optimization. Predicted data for all realizations are obtained by simulating these search points concurrently, and relevant simulation results for all successful simulation jobs are recorded in a training data set. A local quadratic model around each base-case is constructed using the GN formulation, where the required sensitivity matrix is approximated by linear regression of nondegenerated points, collected in the training data set, that are closest to the given base-case. A new search point for each base-case is generated by minimizing the local quadratic approximate model within a trust region, and the training data set is updated accordingly once the simulation job corresponding to each search point is successfully completed. The base-cases are updated iteratively if their corresponding search points improve the data mismatch. Finally, each base-case will converge to a local minimum in the region of attraction of the initial base-case. The proposed approach is applied to different test problems with uncertain parameters being limited to hundreds or fewer. Most local minima of these test problems are found with both satisfactory accuracy and efficiency.

Keywords Gauss-Newton method · Distributed computing · History matching · Sensitivity matrix · Trust region method

Mathematics Subject Classification (2010) 90C39 · Nonlinear programming

✉ Guohua Gao
Guohua.gao@shell.com

[1] Shell Global Solutions (US) Inc., 150 N. Dairy Ashford Rd., Houston, TX 77079, USA

[2] Shell Global Solutions International B.V., Kessler Park 1, 2288 GS Rijswijk, the Netherlands

[3] Shell International Exploration & Production Inc., 3333 HWY 6 S, Houston, TX 77082, USA

1 Introduction

1.1 Formulating history matching within the Bayesian framework

Input properties of a reservoir simulation model such as permeability and porosity, potentially in each gridblock, are uncertain parameters that have to be adjusted to honor production data using an automatic (or assisted) history matching (AHM) technique [51, 52]. Generally, a history matching problem can be formulated within the Bayesian framework [52, 64, 65], where the posterior probability density function (PDF) of uncertain model parameters (x), conditioned to production data (d_{obs}), can be expressed as

$$P\left(x|d_{\text{obs}}\right) = cP\left(d_{\text{obs}}|x\right) P\left(x\right). \tag{1}$$

In Eq. 1, $P(x)$ is the prior PDF of uncertain model parameters (x), $P\left(d_{\text{obs}}|x\right)$ is the likelihood of d_{obs} for given x, and $c = 1/P(d_{\text{obs}})$is a normalization constant. Here, x is an

$n-$dimensional vector that contains all uncertainty parameters to be tuned during the process of history matching, and d_{obs} is an N_d-dimensional vector that contains all observed data. If we assume that the prior PDF of x is Gaussian with a mean x_{prior} and covariance matrix C_M and the measurement errors for d_{obs} are also Gaussian with zero mean and covariance matrix C_D, then the posterior PDF of Eq. 1 can be rewritten as

$$P\left(x|d_{\text{obs}}\right) = c\exp\left[-f\left(x\right)\right],$$

where $f\left(x\right)$ is an objective function defined as

$$f\left(x\right) = \frac{1}{2}\left(x - x_{\text{prior}}\right)^T C_M^{-1}\left(x - x_{\text{prior}}\right) + \frac{1}{2}\left(y(x) - d_{\text{obs}}\right)^T C_D^{-1}\left(y(x) - d_{\text{obs}}\right). \quad (2)$$

In Eq. 2, $y\left(x\right)$ represents the simulated data responses from a reservoir model with parameters x.

To quantify the impact of uncertainty on simulated forecast results, the posterior probability distribution of Eq. 1 has to be properly sampled in AHM workflows. One approach is to compute the (global) maximum a posteriori (MAP) estimate by minimizing the objective function defined in Eq. 2 and then generate an ensemble of approximate conditional realizations by linearization of the reservoir responses around the MAP estimate [2, 15]. More recently, we proposed to use all (local) MAP points and apply linearization locally such that the posterior is approximated by a "Gaussian mixture model" (GMM), i.e., a superposition of Gaussian distributions [27].

Alternatively, the randomized maximum likelihood (RML) method [37, 49] can be applied to generate approximate conditional realizations by minimizing a large number of perturbed objective functions,

$$f_{\text{RML}}^{(i)}\left(x\right) = \frac{1}{2}\left(x - x_{uc}^{(i)}\right)^T C_M^{-1}\left(x - x_{uc}^{(i)}\right) + \frac{1}{2}\left(y(x) - d_{uc}^{(i)}\right)^T C_D^{-1}\left(y(x) - d_{uc}^{(i)}\right) \quad (3)$$

In Eq. 3, $x_{uc}^{(i)} \sim N\left[x_{\text{prior}}, C_M\right]$ and $d_{uc}^{(i)} \sim N[d_{\text{obs}}, C_D]$ for $i = 1, 2, \ldots, N_e$, where N_e is the number of realizations to be generated.

1.2 Gauss-Newton formulation

As discussed by Christie et al. [13], inverse problems in which the data response is (highly) nonlinear may have nonunique best-match solutions. However, most optimization methods, especially local search methods, are designed for finding one optimum. Even stochastic optimization methods, which may have a larger chance of finding a global optimum than local search methods, will converge to a single optimum. Obviously, it is very challenging to design a generic method to find a large number of (local) optima of an arbitrary objective function. Since our ultimate objective is to develop efficient AHM and uncertainty quantification workflows, we shall limit ourselves to least-squares-type objective functions of the form shown in Eq. 2. The Gauss-Newton (GN) formulation, either with line search methods such as in the Levenberg-Marquardt (LM) method or with trust region search methods, is tailored for this type of problems. However, the only straightforward way to find multiple minima with these standard (or traditional) GN methods is to repeat the optimization with a large number of different starting or initial values. In this paper, we will develop a much more efficient strategy.

A brief summary of the GN formulation is provided here for clarity. Let $J(x)$ be the sensitivity matrix evaluated at x. The l-th row and m-th column entry of the sensitivity matrix is defined as the partial derivative of the l-th data (for $l = 1,2,..., N_d$) with respect to the m-th ($m = 1,2,..., n$) parameter, i.e.,

$$J_{l,m} = \frac{\partial y_l}{\partial x_m}. \quad (4)$$

The gradient of the objective function defined in Eqs. 2 and 3 can be expressed as

$$\nabla f(x) = C_M^{-1}[x - x_{\text{prior}}] + J^T(x)C_D^{-1}(y(x) - d_{\text{obs}}). \quad (5)$$

$$\nabla f_{\text{RML}}^{(i)}(x) = C_M^{-1}[x - x_{uc}^{(i)}] + J^T(x)C_D^{-1}(y(x) - d_{uc}^{(i)}). \quad (6)$$

If the second- and higher-order derivatives of the data responses $y(x)$ are neglected, the Hessian of the objective function can be approximated by the following GN formulation:

$$H(x) = C_M^{-1} + J^T(x)C_D^{-1}J(x). \quad (7)$$

1.3 Review of derivative-free optimization methods

It has been shown that gradient-based quasi-Newton optimization algorithms are quite efficient and robust for large-scale history matching problems [22, 23] and production optimization problems [40, 41] when the adjoint method is available to compute gradients or sensitivities [5, 8, 39]. Unfortunately, most commercial reservoir simulators do not have an adjoint method implementation to compute gradients and sensitivities. In these situations, derivative-free optimization methods that do not require analytical evaluation of the gradient of the objective function or the sensitivity of a datum are necessary. Recently, different derivative-free optimization methods suitable for history matching approaches have been developed. These derivative-free optimization (DFO) algorithms can be roughly divided into three classes: stochastic global search, direct search, and model-based local search.

Stochastic global search methods include genetic algorithms [32], simulated annealing [36], particle swarming

[35], etc. By introducing probabilistic factors in the search process that encourage global exploration, global search methods may escape from a local optimum. Because our major objective is to efficiently find multiple local minima of the objective function, global stochastic search algorithms do not fit our purpose and will not be discussed in this paper; interested researchers may see Ouenes and Bhagavan [53], Cheng et al. [12], Hajizadeh et al. [31], and Mohamed et al. [42, 43], for references.

Direct search methods include the simplex method [46], direct pattern search [29, 34], and the mesh adaptive direct search [3]. Because direct search methods do not use the smooth features of an objective function to guide the search direction, they generally converge quite slowly. As benchmarked by More and Wild [45], model-based local search methods, e.g., NEWUOA developed by Powell [55], perform much better than direct search methods for problems with smooth objective functions. However, as shown by Gao et al. [26], direct search performs more robustly than model-based methods for nonsmooth problems, especially when numerical noise is present. A hybrid method that integrates direct pattern search with quasi-Newton and/or Gauss-Newton methods using trust region search strategy was proposed by Gao et al. [26].

Model-based DFO local search algorithms exploit the smooth features of the objective function by constructing an analytical model to approximate the actual objective function in the neighbourhood of the best solution found in the current iteration. Although other types of local analytical models are also feasible, e.g., radial-basis function [66], a local quadratic model is the most popular model [54, 55, 66, 67]. The Hessian matrix and the gradient of the quadratic model are approximated by interpolating or fitting values of the objective function at different points that have been evaluated in previous iterations. In addition to the aforementioned interpolation methods, the gradient of the objective function can also be estimated by stochastic approaches, such as the stochastic noise reaction method proposed by Okano and Koda [48], the simultaneous perturbation and stochastic approximation (SPSA) method [24, 38, 62, 67], and the stochastic simplex approximate gradient (StoSAG) method [21], or by ensemble-based approaches such as the EnOpt method [10]. As shown by Do and Reynolds [16], different algorithms to estimate the gradient such as the simplex, preconditioned simplex, and EnOpt algorithms are theoretically connected, and they can be derived directly from a modified SPSA-type algorithm.

Once a local quadratic model is constructed, a new search point in the next iteration can be generated using either a line search strategy or a trust region search strategy.

In a line search strategy, a search direction is first selected and then a proper step size along the search direction is determined. If the Hessian of the quadratic model, $H^{(k)}$, is positive definite, the search direction can be determined by $p = -\left[H^{(k)}\right]^{-1} \nabla f^{(k)}$, where $\nabla f^{(k)}$ is the gradient of the objective function evaluated at the current solution $x^{(k)}$. A proper step size can then be determined by a backtracking method or quadratic fitting method. Generally, the step size has to satisfy the Wolfe conditions to guarantee its convergence; see Nocedal and Wright [47] for more details about line search.

In a trust region search strategy, a new search point is generated by finding the global minimum of the quadratic model within a given ball-shaped trust region by solving the following trust region subproblem:

$$\min\ q^{(k)}(s) = \left\{ \frac{1}{2} s^T H^{(k)} s + s^T \nabla f^{(k)} + f^{(k)} \text{ s.t. } \|s\|_2 \le \Delta^{(k)} \right\}. \tag{8}$$

In Eq. 8, $\Delta^{(k)}$ is the radius of the ball-shaped trust region. Different methods have been developed to solve the trust region subproblem (TRS) defined in Eq. 8. Direct trust region solvers are based on the Cholesky decomposition of the Hessian matrix; see More and Sorensen [44] and Gould et al. [28]. Iterative trust region solvers apply conjugate-gradient, subspace minimization [19] or Krylov-based methods [58, 61].

As indicated by Eq. 4 through Eq. 7, for history matching problems, a quadratic model can be constructed using the GN formulation to approximate the objective function defined in Eq. 2 or 3 by linear approximation of simulated data responses. Similarly, the sensitivity matrix $J^{(k)}$ at $x^{(k)}$ can be estimated. Coats et al. [14] proposed a method to estimate the sensitivity matrix by fitting training data points using a least squares method. The training data points were randomly sampled within the lower and upper bounds for each parameter. Gao et al. [26] applied a two-sided finite difference equation to estimate the sensitivity matrix.

As benchmarked by Gao et al. [26], the GN trust region search method together with a linear approximation of responses using the approximated sensitivity matrix performs much better than other quasi-Newton counterparts. Zhou and Zhang [69] observed similar results, and their benchmarking results also indicate that the GN method performs the best, whereas the LM method performs the worst, when applied to least squares problems. Even worse, line search methods are not robust for problems with numerical noise [26]. Because numerical noise is unavoidable when reservoir responses are predicted from running reservoir simulations, we believe that the GN trust region search method is the preferred choice for optimization methods in the context of history matching.

Ensemble-based methods such as the ensemble Kalman filter (EnKF), the ensemble smoother (ES), and some of their variants (iterative EnKF or ES, localization, and clustering) have been proved quite efficient for linear and near-linear history matching problems; see Gu and Oliver [30], Evensen [20], Smith [60], Aanonsen et al. [1], Stordal et al. [63], Chen and Oliver [9], Elsheikh et al. [18], and Reynolds et al. [56], to mention only a few of them. As pointed out by Reynolds [57], these ensemble-based methods can be regarded as a combination of local GN or LM line search optimizer with a RML sampler, where the sensitivity matrix required in the GN formulation of Eq. 7 is estimated by averaging over all realizations obtained in the current iteration. Reynolds [57] also showed that the degree of freedom (DOF) of ensemble-based methods is reduced to $N_e - 1$. Therefore, ensemble-based approaches can also be regarded as model-based DFO methods combined with a specifically designed built-in parameter reduction technique.

As addressed by different authors [18, 60], using the globally averaged sensitivity matrix (GSM) is the major cause that results in failure of convergence for ensemble-based approaches. Different approaches, such as space localizations [63] or clustering [18, 60], have been developed to improve the performance of ensemble-based methods, especially for problems with multiple modes. Smith [60] proposed to "use cluster analysis to identify a GMM describing the forecast ensemble" and integrated it with EnKF. Elsheikh et al. [18] proposed a stochastic ensemble method and augmented it with k-means clustering technique, where the model update equation "resembles the update step of EnKF."

In addition to the ensemble Kalman filter, particle filters, a genetic-type sampling approach, can also be applied to generate a set of particles to represent the posterior distribution [17, 59].

Generation of correct, unbiased samples of the posterior PDF defined in Eq. 1 and quantifying uncertainty associated with model parameters and production forecasts are very important and very interesting topics, which are beyond the scope of this paper. The focus of this paper will be developing a new, parallel DFO method to find multiple local minima (or local MAP estimates) of the objective function defined in Eq. 1.

1.4 New strategies

Computational cost (in terms of the number of function evaluations, or equivalently, the number of simulation runs) required for a traditional model-based DFO algorithm to converge increases at least linearly with the number of uncertain parameters. Therefore, DFO algorithms become impractical when the number of parameters that have to be optimized becomes very large. As will become clear below, most of the required function evaluations (i.e., simulations) can be performed concurrently; hence, the maximum manageable number of parameters depends on the size of the available compute cluster. When a large compute cluster is available, the number of manageable parameters may be O(1000). If the history matching problem requires tuning some reservoir or other properties (e.g., permeability, porosity, facies type) in each gridblock, some type of parameter reduction techniques has to be applied upfront. For example, the pluri-PCA method [7] can be applied to reduce the number of uncertain parameters for a real history matching problem. Even after reducing the number of parameters to a few hundreds, the extremely high computational cost makes it impractical to apply the traditional DFO methods to find multiple local minima of the objective function defined in Eq. 2 or to generate multiple RML realizations by minimizing the objective function defined in Eq. 3.

A major cause for high computational cost of traditional model-based DFO algorithms is the lack of information sharing among different optimization tasks starting from different initial guesses. Most of the available model-based DFO algorithms find multiple local optimum independently, i.e., useful information (e.g., simulation results) from one optimization task is not shared with others. In this paper, we develop effective information sharing techniques applicable for model-based DFO algorithms. If optimization tasks that start from different initial guesses finally converge to the same local optimum, then simulation results obtained from one task are very useful for another one, especially, when these points of model parameters used for simulations get closer to each other. If results of different tasks that will finally converge to the same local optimum can be dynamically shared with each other during the process of history matching, fewer number of function evaluations (or equivalently, simulation jobs) should be used for convergence when compared with the independently executed history matching tasks in which no information is shared.

The idea of sharing or communicating information as part of a Gauss-Newton optimization has been applied to node localization in wireless sensor networks. Cheng et al. [11] proposed distributed algorithms for sensor localization based on computation of the Gauss-Newton step for a local cost function and selection of a proper step size with line search. Zhao and Nehorai [68] studied the distributed implementation of the Gauss-Newton method in the maximum likelihood estimation in wireless sensor networks. Bejar et al. [4] presented a distributed consensus-based Gauss-Newton method for localization in ad hoc networks. Although the optimization method proposed in this paper has the same name, "distributed Gauss-Newton," as proposed by Cheng et al. [11], the methodology and the theoretical formulation presented in this paper are completely different from those presented in their papers and other related papers [4, 11, 68].

We should note that the information sharing mechanism of these distributed Gauss-Newton (DGN) approaches applied to node localization in wireless sensor networks is similar to, but also substantially different from, the information sharing mechanism among different realizations of ensemble-based methods [1, 9, 20, 56]. The similarity is that the same sensitivity matrix is estimated from and used for different sensors or realizations. In ensemble-based methods, all realizations assume the same sensitivity matrix. In contrast, in DGN approaches, only sensors that are close enough to each other (but not all of them) use the same sensitivity matrix.

Inspired by the ideas of information communication and sharing among different nodes, a novel distributed, parallelized Gauss-Newton (DGN) optimization method is developed in this paper to find the multiple local minima of an objective function that can be written as a sum of squares. A summary of the main idea for the proposed method is as follows: N_e different initial starting points are randomly generated and they are used as the N_e initial base-cases and initial search points. Production data for the N_e search points in each iteration are predicted by running reservoir simulation jobs concurrently, e.g., by submitting them to N_e nodes in high-performance computer (HPC) clusters. The relevant simulation results for all successful simulation jobs are recorded in a training data set. Based on simulation results collected in the training data set, the sensitivity matrix for each base-case is estimated by linear regression of n nondegenerated points in the training data set that are closest to the given base-case. A local quadratic model for each base-case is then constructed using the Gauss-Newton formulation of Eq. 5 through Eq. 7. A new searching point for each base-case is generated by minimizing the local quadratic model within a reasonably small trust region. If the searching point improves the objective function, then it is accepted as the new base-case and the trust region size will be updated (it may be expanded). If not, the base-case remains unchanged, but the trust region size will be reduced. By updating the local quadratic models and the base-cases iteratively, different local minima will be found if the number of initial base-cases is chosen large enough.

The proposed DGN history matching approach also has some similarity to iterative ensemble-based methods, especially when applied to generate multiple RML samples using the objective function defined in Eq. 3. For example, they both apply parameter reduction techniques to reduce the degree of freedom and computational cost, share information among different realizations to compute the sensitivity matrix, use the GN formulation in Eq. 5 through Eq. 7 to build an approximate quadratic model of the actual objective function, and apply a local search optimization method to find a local minimum of the objective function. However, the parameter reduction methods, the information sharing mechanisms, and the optimization algorithms implemented in the DGN approach are quite different from those of ensemble-based approaches. Further in-depth analysis and detailed discussions about their similarity and difference and a thorough performance comparison or benchmarking study are beyond the scope of this paper and will be presented elsewhere in the future.

The remainder of this paper is organized as follows: More details about the proposed DGN methodology are described in Section 2. The DGN method is validated with synthetic toy problems in Section 3. In Section 4, the proposed DGN method is applied to finding multiple MAP points in a real history matching problem. Finally, some conclusions are drawn in Section 5, based on our theoretical formulations and numerical tests.

2 Methodology

2.1 Illustration of the basic ideas using a 2-D example

Assume that N_e initial starting points (or base-cases) are generated, e.g., randomly, as shown by open circles in Fig. 1a, where $N_e = 10$; alternatively, the initial points could be distributed according to a space-filling design, for a maximally even spread in parameter space. The sensitivity

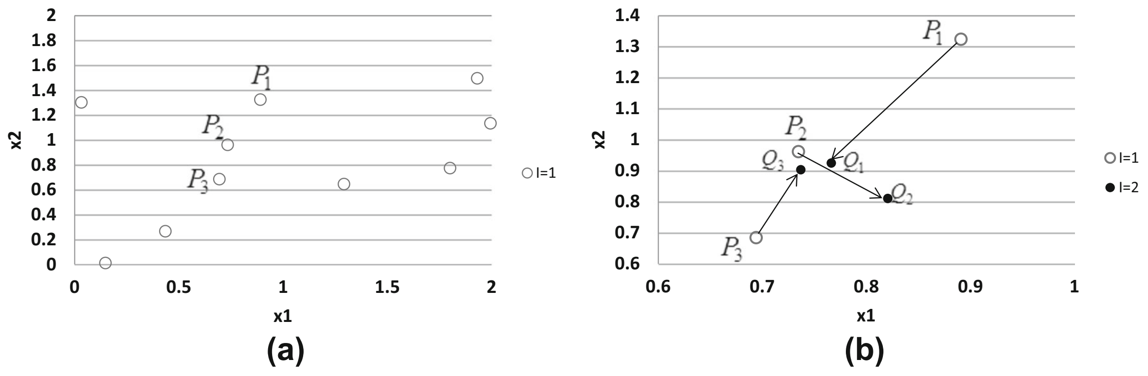

Fig. 1 Illustration of sharing information between different base-cases. **a** Starting locations of 10 base-cases. **b** A zoom-in around the three starting points of P_1, P_2, and P_3; the *arrows* indicate how these base-cases are updated

matrix at one of the N_e base-cases, e.g., at P_1 in Fig. 1a, can be estimated by regression techniques. The simplest approach is based on linear regression.

In the first iteration, there are N_e available points in the training data set. Among all available points, the linear regression through the three points that are closest to P_1, as shown by P_1, P_2, and P_3 in Fig. 1a, yields the best estimate of the sensitivity matrix. Using the estimated sensitivity matrix, an approximate quadratic model of the objective function is constructed using the Gauss-Newton formulation of Eq. 5 through Eq. 7. A trial searching point is generated by minimizing the approximate quadratic model within a given trust region; see the point Q_1 in Fig. 1b as an example. If Q_1 improves the actual objective function, then the corresponding base-case in the current iteration (P_1) is replaced by Q_1. Otherwise, the corresponding base-case remains unchanged, but the trust region size for the same base-case will be reduced (e.g., cut by half) in the next iteration.

In the second iteration, N_e additional points become available (assuming simulation jobs for all trial searching points are successfully completed), as shown by solid black dots in Fig. 1b. If the objective function evaluated at Q_1 is smaller than P_1, then Q_1 becomes the new base-case. Taking Q_1 as an example, the two points that are closest to Q_1 in the second iteration are P_2 and Q_3; see Fig. 1b. Because these three new points in the second iteration, Q_1, P_2, and Q_3, are closer to each other than the old three points in the first iteration, P_1, P_2, and P_3, it is expected that the new sensitivity matrix estimated in the second iteration is more accurate, and therefore, minimizing the corresponding quadratic model will probably generate a better searching point in the next iteration. The same procedure is repeated until suitable convergence criteria are satisfied, and a local minimum of the objective function is found. More detailed discussions on how to update the training data set, how to estimate the sensitivity matrix through linear regression, how to construct an approximate quadratic model, and how to generate a new searching point will be presented in the following subsections.

2.2 Share available information through dynamically updating the training data set

Before turning to the details of our DGN method, we would like to contrast the information sharing mechanisms sketched above and illustrated in Fig. 1 with the information sharing employed in iterative ensemble-based approaches. For iterative ensemble-based methods, a global sensitivity matrix is estimated by averaging over all data points that have been evaluated in the current iteration, and it is applied to all realizations. For example, in the second iteration, it is estimated by averaging over Q_1, Q_2, Q_3, etc. Such a globally estimated sensitivity matrix may become inaccurate either because the underlying points are spread too far away from each other or because they are located in different local basins. Obviously, using the same sensitivity matrix for all realizations is not a good choice. Although the clustered iterative ensemble-based method proposed by Elsheikh et al. [18] uses the same sensitivity matrix for realizations that are in the same cluster, the sensitivity is estimated using some of those realizations obtained in the current iteration, and simulation results evaluated in the previous iterations are not reused. Theoretically, a better choice is to estimate the sensitivity matrix locally for each realization using all available points that have been evaluated in the current iteration and previous iterations, but not limited to the current iteration only. For example, the sensitivity matrix estimated by fitting the three points Q_1, P_2, and Q_3 that are closest to Q_1 will definitely yield a more accurate estimation of the sensitivity matrix for Q_1. To reuse the point P_2, we have to record the simulation results of P_2.

A training data set is designed to store all relevant simulation results of all successfully completed simulation jobs. In each iteration, the training data set is dynamically updated by adding results of newly completed simulation jobs. Before starting the iterations, the training data set is empty because no simulation jobs have been successfully completed yet. During subsequent iterations, any successful simulation job is added to the training data set if it satisfies the following requirement: the distance between the new point and any point in the training data set must be greater than a given minimum distance $d_{\min}$ to prevent from generating identical points in the training data set. When all parameters are normalized by their standard deviations, we recommend using 0.001 to 0.00001 for $d_{\min}$. We use 0.0001 for the test problems in this paper. Additionally, the size of the training data set is also limited by imposing a maximum number of training data $N_{T\max}$ to prevent out-of-memory issues. For simplicity, we assume in this paper that $N_{T\max}$ is a very large number and that the number of training data needed to reach convergence is always smaller than $N_{T\max}$.

We note in passing that a new training point may also be a point that has been simulated successfully for other tasks, e.g., in an experimental design or sensitivity analysis study, or points generated by other kinds of history matching algorithms or workflows. When relevant simulation results are available, they can be used as additional points in the training data set for the first iteration. In this way, all relevant simulation results can be (re)used to speed up the process of finding multiple MAP points.

A point in the training data set (or equivalently, a successful simulation job) has the following three attributes: a set of values of each unknown parameter $x = [x_1, x_2, ..., x_n]^T$, where n is the number of unknown parameters, a value of the objective function $f(x)$, and a set of values of each predicted datum $y(x) = [y_1(x), y_2(x), ..., y_{N_d}(x)]^T$ where N_d is the number of observed data to be matched.

Let $N_T \le N_{T,\max}$ denote the number of points in the training data set that have been run in the previous iterations. If the training data set is not empty ($N_T > 0$), it will be updated using the following algorithm for given n, N_d, $d_{\min}$ and a successful simulation job that used parameters x:

(1) Calculate the distance between the new point (x) and all training points in the current training data set, $d^{(l)} = \left\| x - x_T^{(l)} \right\|_2$ for $l = 1, 2, ..., N_T$;

(2) If Min $\left\{d^{(l)}, l = 1, 2, ..., N_T\right\} > d_{\min}$, then update the training data set by

a) Increase N_T by 1;

b) Record the value of each unknown parameter, $x_{T,j}^{(N_T)} = x_j$ for $j = 1, 2, ..., n$;

c) Record the value of the objective function, $f_T^{(N_T)} = f(x)$;

d) Record the value of each predicted datum, $y_{T,j}^{(N_T)} = y_j(x)$ for $j = 1, 2, ..., N_d$;

e) Record the distances between the new point and all other points in the training data set, $d_T^{(N_T,l)} = d^{(l)}$ for $l = 1, 2, ..., N_T - 1$.

Here, the subscript "T" denotes "training," the subscript "j" denotes the j-th element of a vector, and the superscript "(l)" indicates the l-th point in the training data set. Throughout this paper, a 2-norm is used to determine the distance between any two points or the norm of a vector. To mitigate the effect of different scales or units for different parameters, all parameters are normalized by their standard deviations.

If the training data set is empty ($N_T = 0$), it can be initialized by executing steps (a) through (e) in the algorithm discussed above. The recorded distances $d_T^{(N_T,l)}$ will be used to select the n nondegenerated points that are used to determine the local quadratic model for each base-case.

As illustrated in the training data set updating algorithm, the values of $n$uncertain parameters, the value of the objective function, the values of N_d predicted data, and the $N_T - 1$distances for each successfully completed simulation case will be stored in the training data set. After parameter reduction, n is generally much smaller than the number of gridblocks of a reservoir simulation model. Therefore, the memory used by the training data set is negligible when compared to the memory used by performing a reservoir simulation.

2.3 Approximate the sensitivity matrix using a linear regression approach

When analytical derivatives computed with the adjoint method are unavailable, we have to estimate the sensitivity matrix. A natural approach is to estimate the partial derivatives that feed into the sensitivity matrix using linear regression. We assume that the N_d data $Y = [y_1, y_2, ..., y_{N_d}]^T$ have been evaluated at n sufficiently well separated points, where n is the number of uncertain parameters. In this paper, these n points are chosen as those nondegenerated points that are closest to a given point $x^{(*)}$. The N_d data evaluated at the j-th point $x^{(j)} = [x_1^{(j)}, x_2^{(j)}, ..., x_n^{(j)}]^T$ are given by $Y^{(j)} = [y_1^{(j)}, y_2^{(j)}, ..., y_{N_d}^{(j)}]^T$. Then, the sensitivity matrix evaluated at the given point $x^{(*)} = [x_1^{(*)}, x_2^{(*)}, ..., x_n^{(*)}]^T$ can be approximated by solving the following linear regression equation:

$$\begin{bmatrix} Dx_1^{(1)}, & Dx_2^{(1)}, & ..., & Dx_n^{(1)} \\ Dx_1^{(2)}, & Dx_2^{(2)}, & ..., & Dx_n^{(2)} \\ \multicolumn{4}{c}{\dots\dots\dots\dots\dots\dots} \\ Dx_1^{(n)}, & Dx_2^{(n)}, & ..., & Dx_n^{(n)} \end{bmatrix} \begin{bmatrix} J_{1,1}, & J_{2,1}, & \dots, & J_{N_d,1} \\ J_{1,2}, & J_{2,2}, & \dots, & J_{N_d,2} \\ \multicolumn{4}{c}{\dots\dots\dots\dots\dots\dots} \\ J_{1,n}, & J_{2,n}, & \dots, & J_{N_d,n} \end{bmatrix} = \begin{bmatrix} Dy_1^{(1)}, & Dy_2^{(1)}, & \dots, & Dy_{N_d}^{(1)} \\ Dy_1^{(2)}, & Dy_2^{(2)}, & \dots, & Dy_{N_d}^{(2)} \\ \multicolumn{4}{c}{\dots\dots\dots\dots\dots\dots} \\ Dy_1^{(n)}, & Dy_2^{(n)}, & \dots, & Dy_{N_d}^{(n)} \end{bmatrix}, \tag{9}$$

where $Dx^{(j)} = [x_1^{(j)} - x_1^{(*)}, x_2^{(j)} - x_2^{(*)}, ..., x_n^{(j)} - x_n^{(*)}]^T$ and $Dy^{(j)} = [y_1^{(j)} - y_1^{(*)}, y_2^{(j)} - y_2^{(*)}, ..., y_{N_d}^{(j)} - y_{N_d}^{(*)}]^T$. Equation 9 can be rewritten in matrix format as $D_x^T J^T = D_y^T$.

If the regression matrix D_x has rank n and is not ill-conditioned, Eq. 9 has a unique and robust solution. However, it is not straightforward to ensure that the n shift vectors are linearly independent. In this paper, we shall assume that for $N_T > n$ the space spanned by the N_T training data points has a dimension of at least n, i.e., there are at least n points in the training set that are linearly independent, such that it is possible to find n independent shift vectors from the N_T training data points.

2.4 Update each realization through minimizing a local quadratic model within a trust region

If we assume that the regression method has been successful in determining the local sensitivity matrix $J^{(i,k)}$, where the superscript "(i, k)" represents the i-th base-case in the k-th iteration, then a local quadratic model can be built for the i-th base-case [26]:

$$q^{(i,k)}(s) = f^{(i,k)} + s^T \nabla f^{(i,k)} + \frac{1}{2} s^T H^{(i,k)} s. \tag{10}$$

In Eq. 10, $f^{(i,k)}$, $\nabla f^{(i,k)}$ and $H^{(i,k)}$ are the value, the gradient, and the Hessian of the objective function $f(x)$ evaluated at $x^{(i,k)}$.

In each iteration $k > 1$, a new search point, $x^{(i,k)} + s^{(i)}$, for the i-th base-case is generated by solving the trust region subproblem [28, 44]:

$$\min\ q^{(i,k)}(s) = \left\{ \frac{1}{2} s^T H^{(i,k)} s + s^T \nabla f^{(i,k)} + f^{(i,k)} \text{ s.t. } \|s\|_2 \le \Delta^{(i,k)} \right\}. \tag{11}$$

The i-th base-case in the following iteration is then updated as $x^{(i,k+1)} = x^{(i,k)} + s^{(i)}$ if $f(x^{(i,k)} + s^{(i)}) < f(x^{(i,k)})$. If no improvement is found, the i-th base-case remains unchanged, but the radius of the trust region $\Delta^{(i,k)}$ shrinks, and the trial step computation is repeated. To avoid jumping from one basin to another basin, the trust region size $\Delta^{(i,k)}$ is also limited by a maximum allowable value. The ratio of actual to predicted decrease is computed as

$$\rho^{(i,k)} = \frac{f^{(i,k)} - f(x^{(i,k)} + s^{(i)})}{f^{(i,k)} - q^{(i,k)}(s^{(i)})}. \tag{12}$$

The basic trust region algorithm can be summarized as follows:

Given $k = 0$, i, $\Delta^{(i,0)} > 0$, $x^{(i,0)}$, $0 < \eta_v < 1$, $0 < \eta_s < \eta_v < 1$, $\gamma_e > 1$, and $0 < \gamma_d < 1$, until convergence do:

(1) Build the local model $q^{(i,k)}(s)$ to approximate $f(x^{(i,k)} + s)$;
(2) Solve the trust-region subproblem in Eq. 11 and compute $\rho^{(i,k)}$ with Eq. 12;
(3) If $\rho_{(i,k)} \ge \eta_v$ (very successful) and $\left\|s^{(i)}\right\| > 0.5\Delta^{(i,k)}$, set $x^{(i,k+1)} = x^{(i,k)} + s^{(i)}$ and $\Delta^{(i,k+1)} = \gamma_e \Delta^{(i,k)}$; else if $\rho^{(i,k)} \ge \eta_s$ (successful) or $\rho^{(i,k)} \ge \eta_v$ but $\left\|s^{(i)}\right\| \le 0.5\Delta^{(i,k)}$, set $x^{(i,k+1)} = x^{(i,k)} + s^{(i)}$ and $\Delta^{(i,k+1)} = \Delta^{(i,k)}$; else (unsuccessful), set $x^{(i,k+1)} = x^{(i,k)}$ and $\Delta^{(i,k+1)} = \gamma_d \Delta^{(i,k)}$.
(4) Increase k by 1.

As recommended by Gao et al. [26], the range for the initial trust region size ($\Delta^{(i,0)}$) is 0.1 to 0.5, if all uncertain parameters are normalized by their standard deviations.

2.5 The distributed Gauss-Newton optimization algorithm

The distributed Gauss-Newton optimization algorithm can now be summarized as:

Given $k = 0$, $N_T = 0$, n, N_d, N_e, $d_{\min}$; for $i = 1, 2, ..., N_e$, initialize $\Delta^{(i,0)} > 0$ and repeat the following steps until convergence:

(1) Generate N_e different base-cases $x^{(i,0)}$ (for $i = 1, 2, ..., N_e$) when $k = 0$, by randomly sampling them from either their prior probability distribution or uniform distribution.
(2) Evaluate the value of the objective function $f(x)$ and the simulated data $y(x)$ at the i-th base-case when $k = 0$, or at the search point of the i-th base-case when $k > 0$;
(3) Initialize (when $N_T = 0$) or update (when $N_T > 0$) the training data set;
(4) Update the trust region size $\Delta^{(i,k)}$ and the i-th base-case when $k > 0$, using the basic trust region algorithm discussed above;
(5) Evaluate the sensitivity matrix of $y(x)$ at the i-th base-case;
(6) Construct the quadratic model $q^{(i,k)}(s)$;
(7) Generate a search point by solving the trust-region subproblem in Eq. 11 using the quadratic model $q^{(i,k)}(s)$ with trust region size of $\Delta^{(i,k)}$;
(8) Increase k by 1.

2.6 Convergence criteria

Some convergence criteria have to be applied to terminate the distributed Gauss-Newton optimization algorithm. For gradient-based optimization algorithms, the following two convergence criteria are conventionally applied: (1) The relative improvement of the objective function in two successive iterations (denoted by ε_1) should be smaller than a user-specified threshold ($\varepsilon_1 \le \varepsilon_{cr1}$), and (2) the normalized norm of the gradient (denoted by ε_2) is smaller than a user-specified threshold ($\varepsilon_2 \le \varepsilon_{cr2}$). Here, the normalized norm of the gradient is the 2-norm of the gradient divided by the absolute value of the objective function plus a small positive number (e.g., 0.0001). Since the estimated sensitivity matrix is based on linear regression, its accurate evaluation may be hampered by numerical noise in simulation results; hence, the second criterion may be difficult to achieve. Instead, we shall use the requirement that the step size (denoted by $\varepsilon_3 = \left\|s^{(i,k+1)}\right\|$) is sufficiently small, i.e., $\varepsilon_3 \le \varepsilon_{cr3}$, as the third stopping criterion. Since we perform multiple minimizations in parallel, these criteria have to be applied per base-case. For example, a base-case is regarded as converged and no new search point for this base-case will be generated in the future iterations when the first and the second (or third) convergence criteria are satisfied. The distributed Gauss-Newton optimization process stops when all base-cases are converged or when the iteration number reaches the specified maximum iteration number.

3 Validation

3.1 Validate the proposed approach with a 2-D nonlinear toy problem

The proposed distributed Gauss-Newton approach is first validated with a 2-D nonlinear toy problem with two unknown parameters. In this toy problem, the response

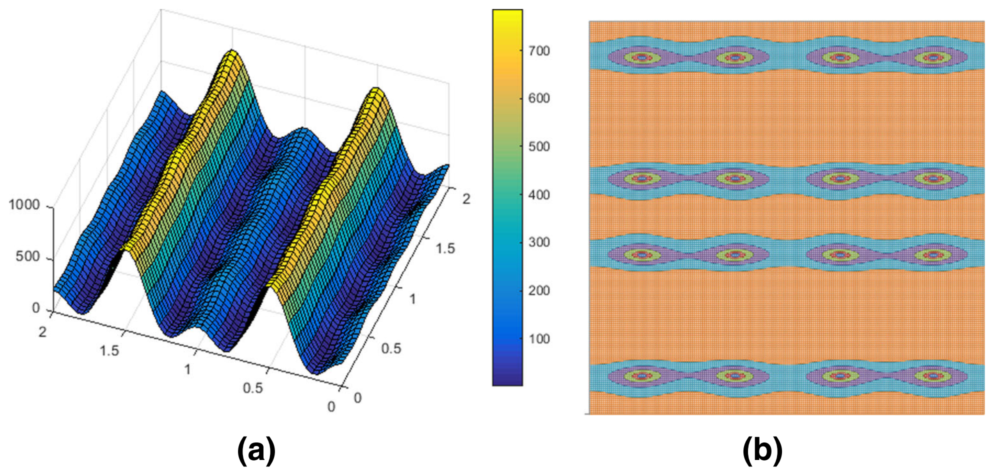

Fig. 2 The surface map (**a**) and the illustration of 16 local minima (**b**) for the 2-D toy nonlinear problem with $N_d = 10$

of the system is a nonlinear function of two unknown parameters given by

$$d(x, t) = 2\sin^2(\pi x_1)\sin t + 6\cos^2(\pi x_2)\cos t. \qquad (13)$$

In Eq. 13, $x = [x_1, x_2]^T$ represents the two unknown parameters and t (in radians) represents time. Observations $d_{\text{obs},j} = d(x_{\text{True}}, t_j)$ are generated at $t_j = j$ for $j = 1, 2, ..., 10$ with $x_{\text{True}} = (0.25,\ 0.195913)^T$. The two unknowns are estimated by minimizing the following objective function of data mismatches:

$$f(x) = 9\sum_{j=1}^{N_d}\left[d(x, t_j) - d_{\text{obs},j}\right]^2. \qquad (14)$$

The surface map of the objective function defined in Eq. 14 with $N_d = 10$ in the searching domain of $[0, 2] \times [0, 2]$ is shown in Fig. 2a. There are 16 local minima in the allowed parameter region, as illustrated in Fig. 2b. For the purpose of validating the capability of the proposed DGN approach to find multiple local minima and to identify valleys, the objective function defined in Eq. 14 does not have the regularization term (or the model mismatch term).

Figure 3a illustrates how the 10 base-cases starting from 10 initial starting points, as shown by open circles in Fig. 3a, converge to 7 local minima after 11 iterations, as indicated by solid purple triangles in Fig. 3a. Figure 3b shows a zoom-in of Fig. 3a, and it illustrates how the three different base-cases starting from three different initial points, denoted by P_1, P_2, and P_3 in Fig. 3a, b, converge to the same local optimum, denoted by the solid purple triangle (overlapped with one of the red circles and one of the black stars) in Fig. 3b. In Fig. 3a, b, different symbols with different colors represent the base-cases obtained in different iterations, e.g., black stars with the legend "I=3" are the base-cases obtained in the third iteration. When a base-case remains the same, e.g., the black solid dot at Q_2 that is identical to a "*" point obtained in the third iteration, this indicates that the new search point does not improve the objective function in the given iteration, so the corresponding base-case is not updated, but the trust region size for that base-case is reduced by half.

Obviously, increasing the number of base-cases (i.e., increasing N_e) will help to find more local optima, as shown in Fig. 4a–d, where the blue ellipses indicate the local

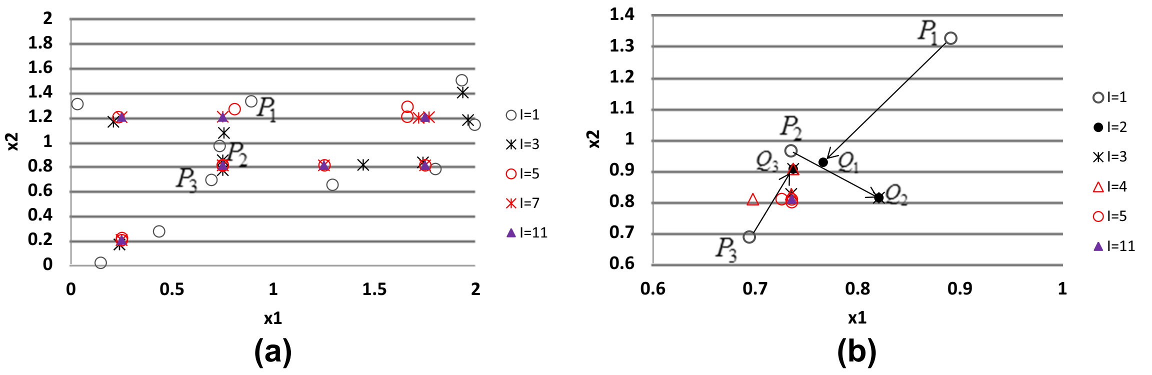

Fig. 3 Illustration of the convergence process to different local minima when the distributed Gauss-Newton approach is applied to the 2-D nonlinear toy problem, starting from 10 different randomly generated initial points. **a** Updating all 10 base-cases. **b** A zoom-in around the three starting points P_1, P_2, and P_3

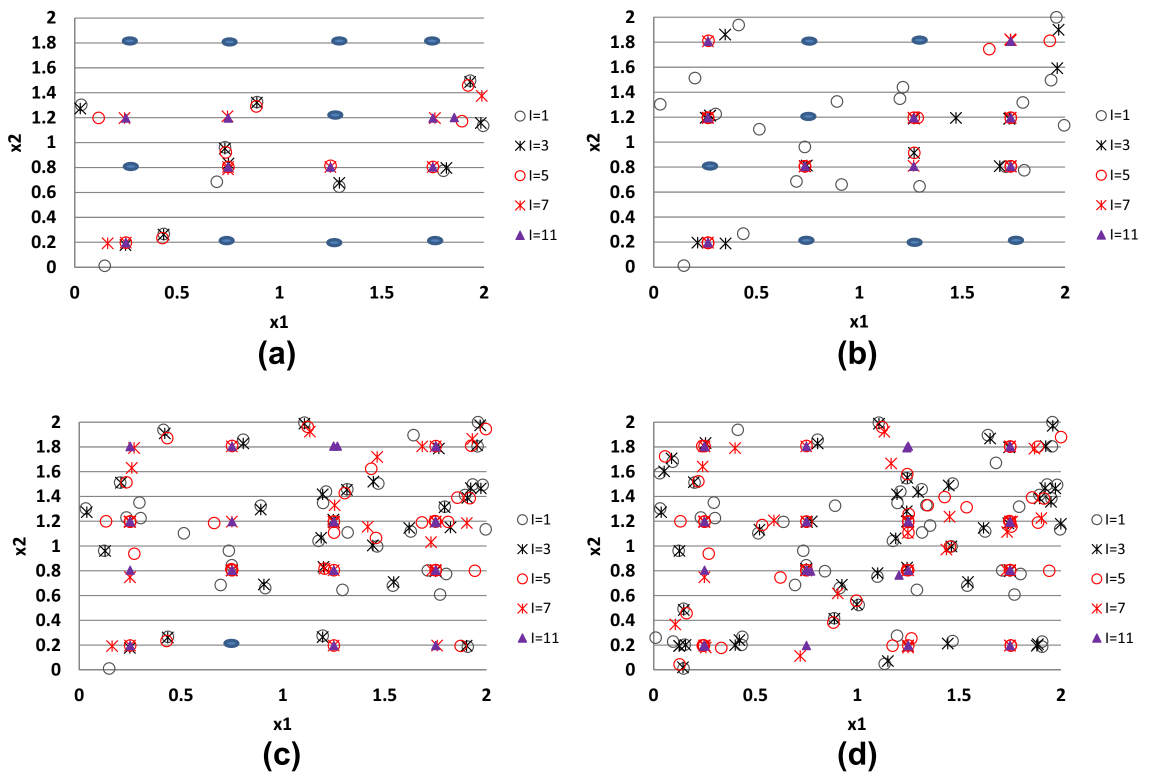

Fig. 4 Impact of number of base-cases on the performance of the distributed Gauss-Newton approach to find more local optima. **a** 10 base-cases. **b** 20 base-cases. **c** 40 base-cases. **d** 60 base-cases

optima that are not found by the distributed Gauss-Newton approach. When $N_e = 10$, only 7 among the 16 local minima are found, as shown in Fig. 4a. As N_e increases to 20, 40, and 60, the distributed Gauss-Newton approach can find 9, 15, and all 16 local minima, respectively.

Throughout this paper, the normalized objective function is defined as the objective function divided by the number of observed data (N_d). Figure 5 illustrates how the values of the normalized objective function for the 60 base-cases decrease as the number of iterations increases. In Fig. 5, red open circles represent the values of the normalized objective function evaluated at the 60 initial points, whereas solid black dots are the values evaluated after 2 iterations in Fig. 5a, 4 iteration in Fig. 5b, 8 iterations in Fig. 5c, and 11 iterations in Fig. 5d. After 11 iterations, the values of the normalized objective function for 50 base-cases are reduced to a value smaller than 0.00001. In fact, all 16 local minima are found after 11 iterations, which required at most $N_F^{\mathrm{DGN}} = 60 \times 11 = 660$ function evaluations in total.

Figure 6a shows the plots of the three convergence criteria vs. iteration. For this toy problem, the maximum values of the three convergence criteria (ε_1, ε_2, and ε_3) over all base-cases are shown, denoted by $\varepsilon_{1,\max}$, $\varepsilon_{2,\max}$, and $\varepsilon_{3,\max}$ in Fig. 6a. In this example, we set $\varepsilon_{cr1} = \varepsilon_{cr2} = \varepsilon_{cr3} = 0.0001$. After about 16 iterations, both the first and the second convergence criteria are satisfied, and the distributed Gauss-Newton optimization algorithm is terminated accordingly. Figure 6b shows the plot of the number of running base-cases vs. iteration. After 4 iterations, the number of running base-cases decreases as the number of iterations increases. After 16 iterations, no base-case is running, i.e., all base-cases converge. The plot shown in Fig. 6b also tells us that 524 (summation of running base-cases over all 16 iterations) function evaluations in total are required to find all 16 local minima. On average, it requires about 33 (=524/16) function evaluations to find 1 local minimum.

3.2 Compare with the traditional AHM approach using the GN-DPS optimizer

Most available model-based DFO approaches are sequential optimizers, where only one or a few searching points are generated in each iteration, e.g., the NEWUA developed by Powell [55]. One of the well-parallelized optimizers is the simultaneous perturbation and multivariate interpolation (SPMI) developed by Gao et al. [25, 26], which includes the hybrid Gauss-Newton or quasi-Newton (e.g., BFGS, SR1) method with direct pattern search methods (DPS), denoted by GN-DPS, BFGS-DPS, and SR1-DPS, respectively. Chen et al. [6] applied the SPMI optimizer using the SR1-DPS option to several history matching problems and compared its performances with those of some other DFO methods,

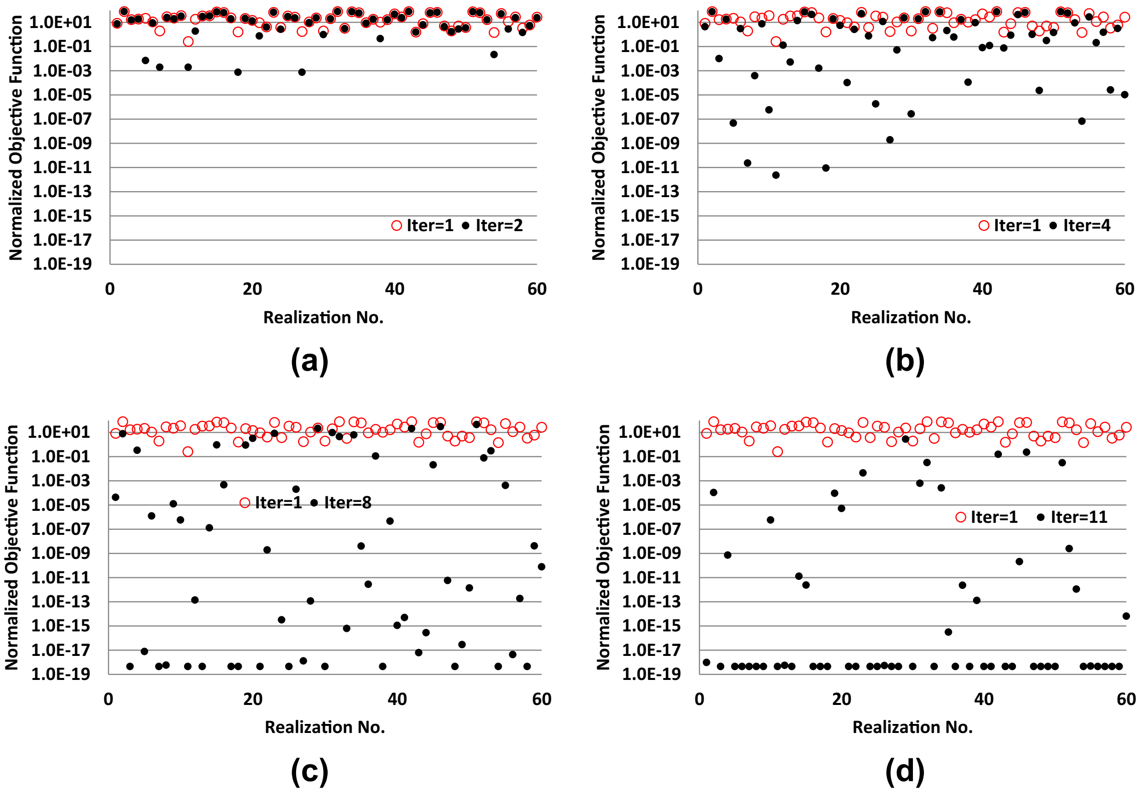

Fig. 5 Illustration of objective function reduction for all 60 base-cases after certain iterations. **a** 2nd iteration. **b** 4th iteration. **c** 8th iteration. **d** 11th iteration

including particle swarm (PSO), very fast simulated annealing (VFSA), and different quadratic interpolation models (QIM) developed by Zhao et al. [67]. Their numerical comparison indicates that SR1-DPS converges faster (in terms of the number of iterations required for convergence) to better solutions (in terms of the objective function value evaluated at the best solution found) when compared with others. Gao et al. [25] compared the performance of SR1-DPS in SPMI with that of other model-based DFO methods, using data and results published by other researchers. Because SPMI is well parallelized and is able to converge in 15 to 30 iterations for problems with different numbers of parameters, the elapse time of an optimization is relatively modest. In contrast, other DFO methods suffer from very long elapse times, since the others need hundreds to thousands (or even more) of iterations. In terms of the number of function evaluations, SPMI is at least as good as other model-based DFO methods on average. Recently, Gao et al. [26] developed the new option of GN-DPS and compared it with SR1-DPS, BFGS-DPS, the well-known LBFGS-B, and the stochastic noise reaction method proposed by Okano and Koda [48]. Among them, GN-DPS is the best performer. In this paper,

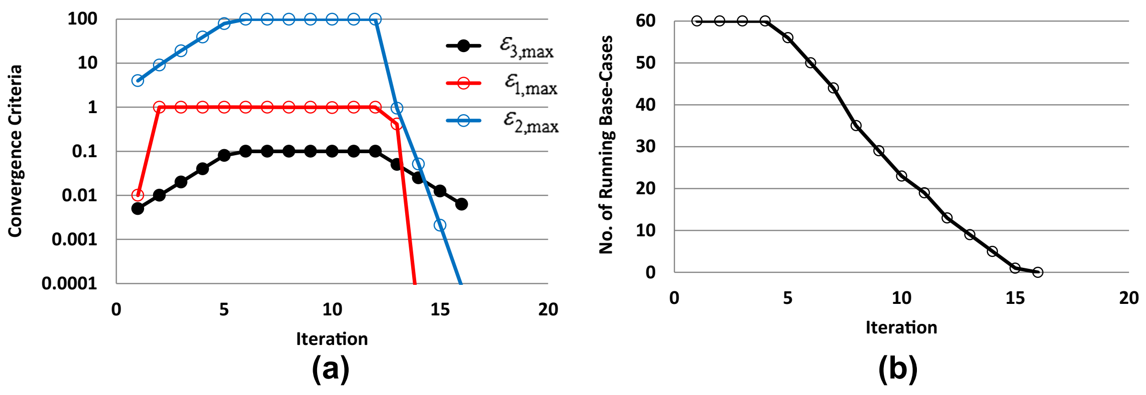

Fig. 6 Convergence criteria (**a**) and no. of running base-cases (**b**) vs. iteration for the 2-D toy problem

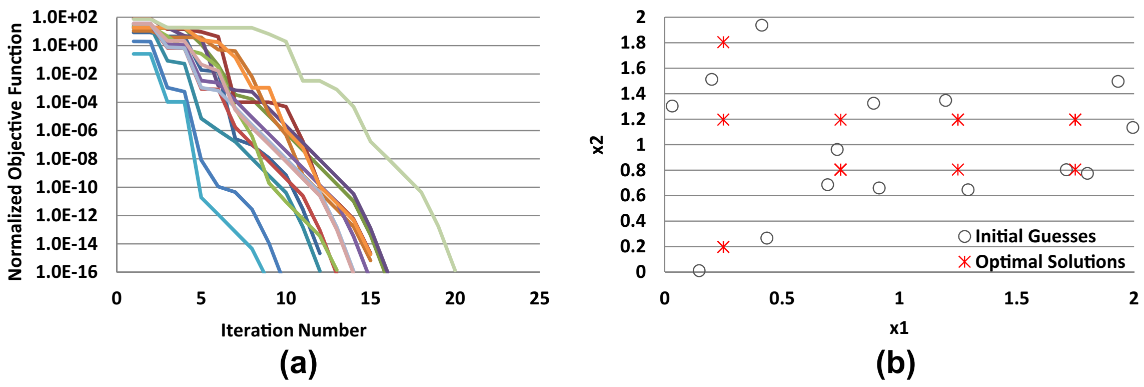

Fig. 7 Results of traditional GN-DPS method when applied to the same 2-D nonlinear toy problem. **a** Convergence performance. **b** Initial guesses and the optimal solutions that were found

we will not repeat the tedious work of comparing DGN with different DFO methods. Instead, we will use GN-DPS as a representative of traditional model-based DFO methods and compare DGN with GN-DPS only. The major reason is that other DFO methods are too computational expensive when applied to history matching problems with hundreds of uncertain parameters, e.g., the real field case discussed below, because they are not well parallelized.

For the purpose of comparison, the GN-DPS is also applied to the same 2-D nonlinear toy problem. Convergence performances of the GN-DPS method for the first 15 base-cases are shown in Fig. 7a. Starting from 15 different initial guesses, the final solutions converge to 9 different local minima of the problem; see Fig. 7b.

As shown in Fig. 7a, it requires 14 iterations for the GN-DPS method to converge on average. In each iteration, the GN-DPS method requires to evaluate the objective function at five different points, when the two-sided finite difference approximation is applied to estimate the gradient of the objective function. In total, $N_F^{GN} = 15 \times 14 \times 5 = 1050$ function evaluations are required to find 9 different local minima. On average, it takes about 117 (=1050/9) function evaluations for the traditional GN-DPS to find 1 local minimum, which is 3.6 times the computational cost required by the DGN method.

3.3 Identify valleys

In many history matching cases, in particular when the model geology is parameterized in a realistic manner, the number of available observed data may be less than the number of unknown parameters. Even if the number of data is larger than the number of parameters, it may be the case that some parameters do not impact the data. Without any regularization term (i.e., without a model mismatch term), instead of having multiple local minima, the objective function will then have one or more null spaces. For example, when using $N_d = 1$, the surface map of the objective function defined in Eq. 14 is shown in Fig. 8a. Interestingly, the proposed distributed Gauss-Newton approach can also

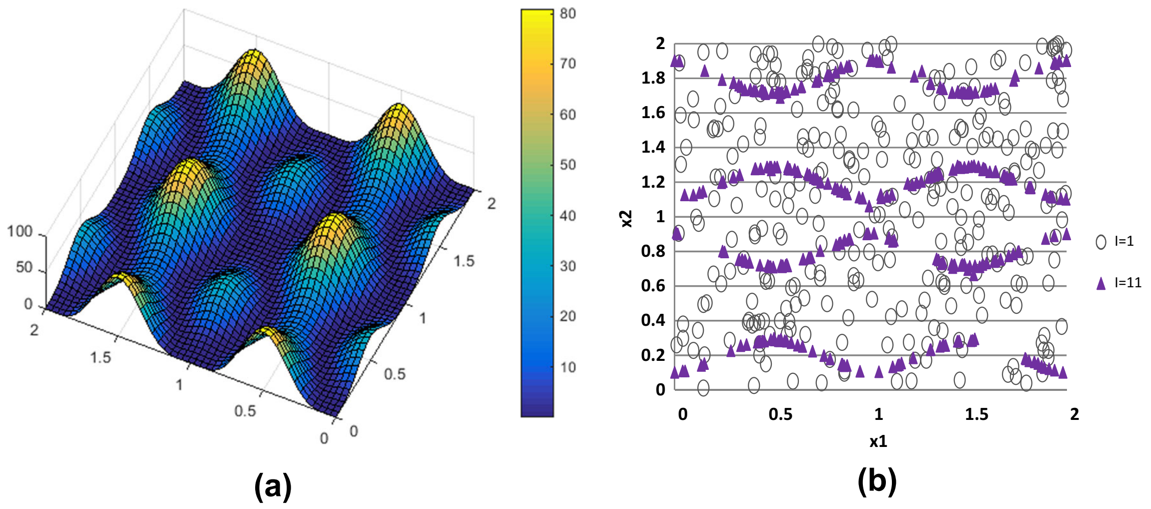

Fig. 8 Illustration of valleys (**a**) and feasibility of identifying valleys using the proposed DGN approach (**b**)

be applied to identify valleys of the objective function, as shown in Fig. 8b.

4 Application to a real history matching problem

In this example, the proposed distributed Gauss-Newton approach is applied in the context of a history matching case with synthetic data, but using a real field model, which is a channelized reservoir with three facies: channel sand facies, levee facies, and shale facies. The channels are populated with an object-based modeling approach. The porosity and permeability in each grid are then modeled with sequential Gaussian simulation. The reservoir simulation model contains $77 \times 78 \times 30 = 180180$ gridblocks and 106,680 active gridblocks, and there are seven producers and one gas injector in the reservoir [7, 26].

Chen et al. [7] proposed a re-parameterization approach, through the integration of truncated pluri-Gaussian simulation technique with principal component analysis (PCA), called pluri-PCA. This approach is applied to reduce the number of uncertain parameters from 426,720 (facies indicator, porosity, permeability, and net-to-gross ratio in each gridblock) to 200 principal component coefficients. An ensemble of 1410 unconditional realizations is generated with the object-based modelling approach, and 1400 of them are used as training realizations for pluri-PCA; see Honorio et al. [33] for different realizations of facies models. One unconditional realization, which does not belong to the 1400 training realizations, is used as the true model. Figure 9a, b shows the areal view and cross-sectional view of the facies distribution in the true model. In addition to the 200 PCA coefficients that drive the facies and reservoir property distribution, 18 parameters to define the rock-type rule (RTR) and 17 global parameters such as aquifer strength and fault transmissibility are also used as uncertain parameters to be tuned for history matching.

In this synthetic history matching example, "true" measurement data, including gas and water production rates in all producers and bottom-hole pressures (BHP), are generated by running the true reservoir simulation model. During the process of history matching, the liquid rate (oil rate plus water rate) in each producer and the gas injection rates in the gas injector obtained from the truth model are used as constraints for reservoir simulation. The monthly water production rate and gas rate in each producer and BHP in each well are the simulation results that have to be matched with the observed data. The observed data (d_{obs}) to be used for history matching are obtained from the simulation results of the truth case model by adding independent Gaussian measurement errors with zero mean and specified standard deviations: 100 bbl/day for water rate, 500,000 SCF/day for gas rate, and 100 psi for BHP, respectively. There are 1701 observed data in total for this example.

After re-parameterization with pluri-PCA and normalization, all PCA coefficients and other normalized uncertain parameters are independent Gaussian random variables with zero mean and unit covariance matrix. The normalized objective function can be written as

$$O(\xi) = \frac{1}{N_d}\left[\xi^T\xi + (g^*(\xi) - d_{\mathrm{obs}})^T C_D^{-1}(g^*(\xi) - d_{\mathrm{obs}})\right], \tag{15}$$

where ξ is the vector with the reduced number of (pluri-PCA coefficients and other) parameters and g^* is the simulated data expressed as a function of these reduced parameters. To start the DGN optimization, $N_e = 300$ initial base-cases are generated by randomly sampling from the Gaussian prior distribution, $P(\xi) \propto e^{-\frac{1}{2}\xi^T\xi}$. Figure 10 illustrates how the values of the objective function for the 300 base-cases decrease as the number of iterations increases. In Fig. 10, red open circles represent the values of the objective function evaluated at the 300 initial base-cases, whereas solid black dots are those evaluated for the 300 base-cases updated after 4 iterations in Fig. 10a, 8 iterations in Fig. 10b, 16 iterations in Fig. 10c, and 32 iterations in Fig. 10d. Initially, the values of the normalized objective function for most of the 300 unconditional realizations are larger

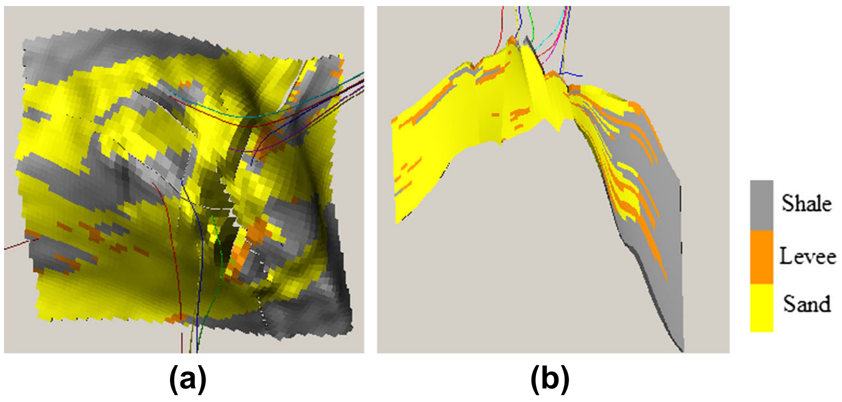

Fig. 9 Areal view (**a**) and cross-sectional view (**b**) of the facies distribution for the true model

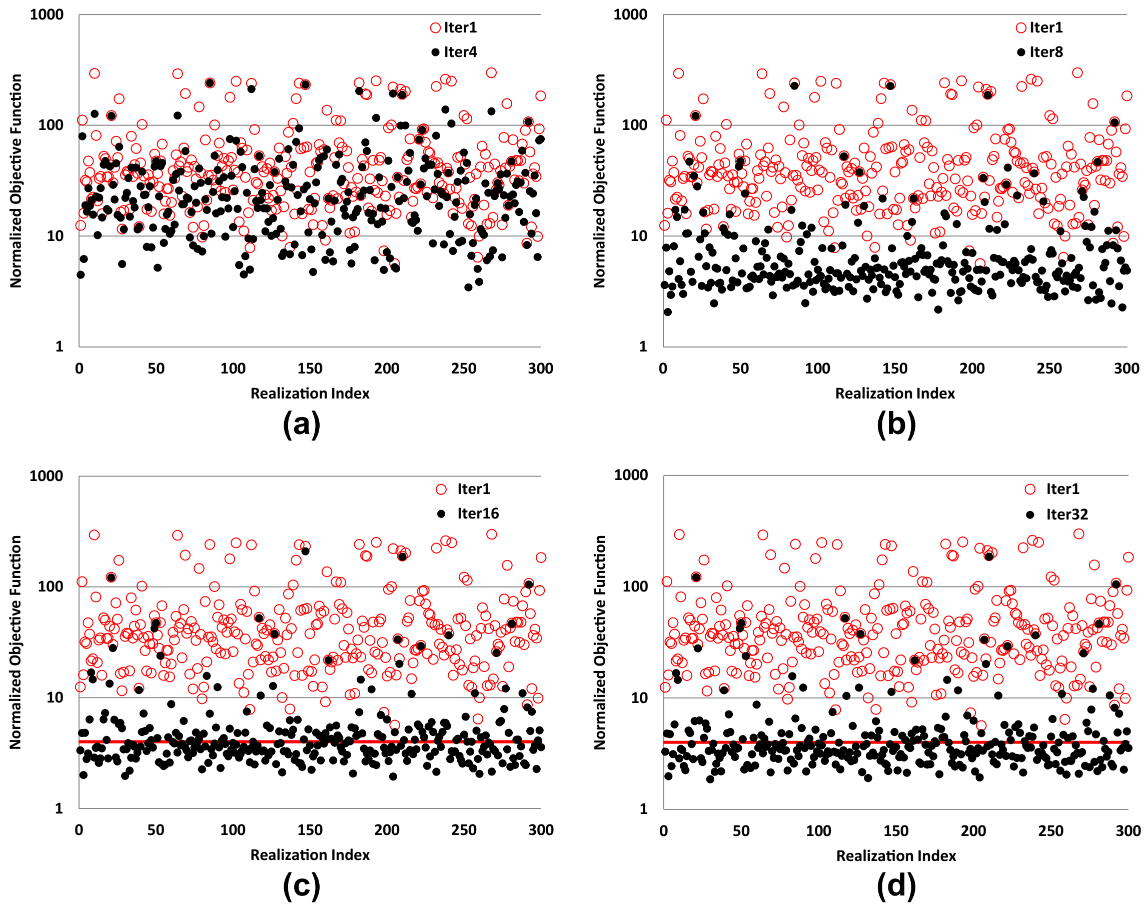

Fig. 10 Illustration of objective function reduction for all 300 base-cases after certain iterations. **a** 2nd iteration. **b** 4th iteration. **c** 16th iteration. **d** 32th iteration

than 50. After 32 iterations, values of the objective function for more than 200 base-cases are reduced to a value smaller than 4.0 (represented by the red horizontal line in Fig. 10c, d).

Figure 11a is similar to Fig. 6a, and it shows the plots of the three convergence criteria vs. iteration. After about 32 iterations, the $\varepsilon_{1,\max}$ (maximum change in objective function) and $\varepsilon_{3,\max}$ (maximum step size) convergence criteria are satisfied, and the distributed Gauss-Newton algorithm is terminated. For the real field model, numerical noise may make it more difficult to converge; therefore, a larger value of $\varepsilon_{cr1} = \varepsilon_{cr2} = \varepsilon_{cr3} = 0.01$ is used instead. As expected, because of numerical noise due to reservoir simulation, the value of $\varepsilon_{2,\max}$ (the maximum

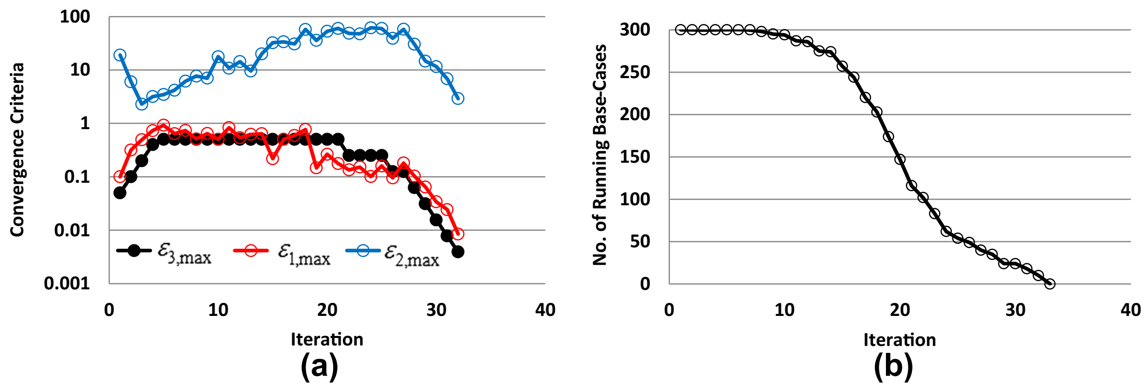

Fig. 11 Convergence criteria (**a**) and no. of running base-cases (**b**) vs. iteration for the real history matching problem

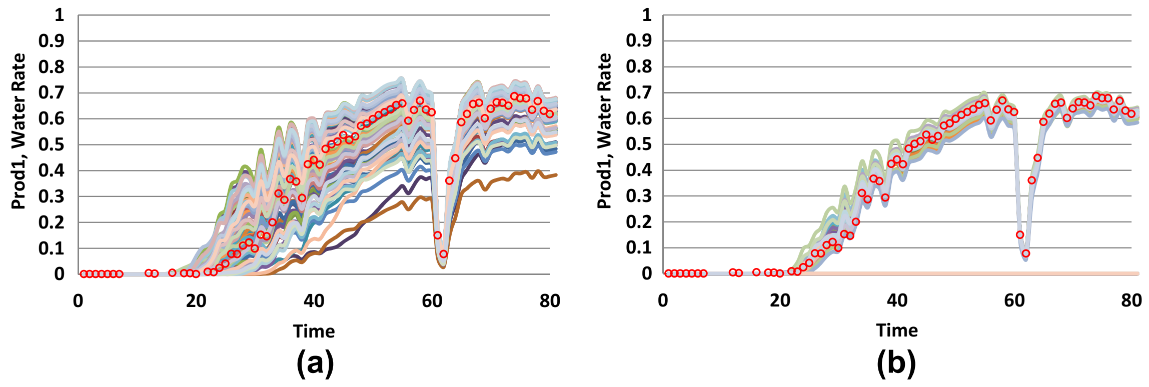

Fig. 13 Comparison of normalized water production profiles for well Prod-1. **a** Generated from initial realizations. **b** Generated from realizations after history matching

norm of the approximated gradient) cannot be reduced to a very small value even after 32 iterations. Comparison between Fig. 10c and d indicates that further reduction of the objective function value after 16 iterations is not significant, and data matches are reasonably good at iteration 16. Figure 11b shows the plot of the number of running base-cases vs. iteration. After 7 iterations, the number of running base-cases decreases as the number of iterations increases. After 32 iterations, no base-case is running, i.e., all base-cases converge. Summation of running base-cases over all 32 iterations gives the total number of function evaluations, 5970, to find 200 local minima with the converged objective function being smaller than 4.0. On average, it requires about 30 (=5970/200) function evaluations to find 1 local minimum.

As shown in Fig. 12, the traditional approach to find local MAP points using, e.g., the GN-DPS method proposed by Gao et al. [26], needs at least 10 iterations to converge on average for this synthetic case. In each iteration, the traditional approach requires 470 simulation jobs to evaluate the numerical gradients, when two-sided finite difference approximation is applied. Therefore, it requires at least $N_F^{\mathrm{GN-DPS}} = 10 \times 200 \times 470 = 940,000$ simulation jobs when the GN-DPS method is applied to find 200 local MAP estimates. In contrast, the total computational cost for the distributed Gauss-Newton approach to find 200 local MAP estimates within 32 iterations is equivalent to running $N_F^{\mathrm{DGN}} = 5970$ simulation jobs. Roughly, the distributed Gauss-Newton approach is able to reduce the computational cost by a factor of 157 when applied to this real field example. We should note that the computational cost reduction factor for the 2-D toy problem is 3.6. Obviously, the benefit of the proposed DGN approach will be problem dependent, but these results suggest that DGN is able to achieve a larger computational cost reduction factor for problems with more parameters. We will continue to benchmark performance of DGN (by integration with GMM) with iterative ensemble-based methods in the future.

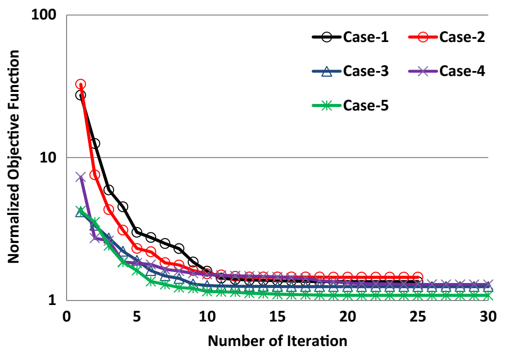

Fig. 12 Normalized objective function vs. iteration profiles when applying the hybrid GN-DPS method to minimize the objective function (copied from Gao et al. [26])

Although the proposed distributed Gauss-Newton approach can significantly reduce the high-performance computer (HPC) usage, we need to address some of its disadvantages. As shown in Fig. 12, the GN-DPS method can reduce the normalized objective function to a value very close to 1, e.g., less than 1.5. In contrast, as shown in Fig. 10c, d, the distributed Gauss-Newton algorithm can only reduce the normalized objective function to a value between 2 and 4. We believe that this difference is due to the way both methods deal with the numerical noise introduced by numerical simulation, which is the major cause for not being able to reduce the normalized objective function to smaller values. Gao et al. [25, 26] discussed some challenges for simulation-based history matching. When gradually changing model parameters, small numerical discontinuities in simulation results are unavoidable for any numerical simulator. These small discontinuities lead to an inaccurate estimation of the gradient, which may cause failure of convergence for model-based search strategies. Re-parameterization of discrete properties (e.g., facies indicators in each gridblock for a reservoir model with multiple facies types) makes the situation even worse, because the objective function becomes even more discontinuous.

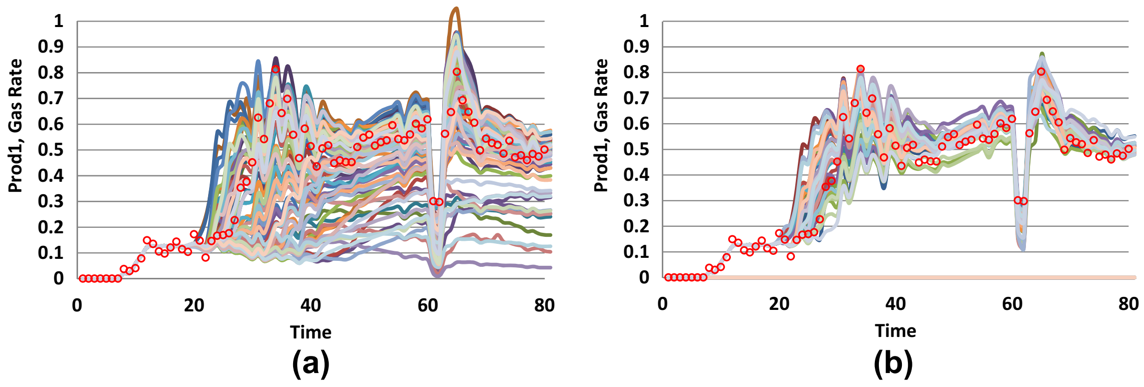

Fig. 14 Comparison of normalized gas production profiles for well Prod-1. **a** Generated from initial realizations. **b** Generated from realizations after history matching

When numerical noise is absent, the distributed Gauss-Newton algorithm is able to reduce the normalized objective function to a value as low as 10^{-18}, very close to the actual minimum value of zero; see results shown in Fig. 5c, d. The auto-adaptive pattern size updating algorithm of the GN-DPS method guarantees finding a proper pattern size such that the effect of numerical noise can be effectively mitigated [26]. In the future, we need to investigate effective methods to mitigate the negative effect of numerical noise on the performance of the distributed Gauss-Newton algorithm too, e.g., by implementing support-vector machine learning to build a more accurate proxy model.

Normalized production profiles of water rate, gas rate, and BHP in Prod-1 for the first 100 initial realizations of the real field case model are shown in Figs. 13a, 14a, and 15a. For the purpose of comparison, normalized production profiles of water rate, gas rate, and BHP in Prod-1 for the first 100 history matched realizations obtained after 16 iterations are illustrated in Figs. 13b, 14b, and 15b. In these figures, we removed a few cases that do not match the production data well, e.g., cases with values of the normalized objective function being larger than 4.0. Results shown in Figs. 13b, 14b, and 15b indicate that using 4.0 as the criterion of selecting a matched model is satisfactory, i.e., the history matched models with a normalized objective function smaller than 4.0 can generate production profiles that match observed data reasonably well. Although we did not show all plots, other wells show similar results.

Here, we should reemphasize that the multiple history matched models obtained by the DGN approach are not conditional realizations of the posterior PDF defined in Eq. 1. To further quantify the uncertainty of model parameters and production forecasts after conditioning to production data, we need to integrate DGN either with the Gaussian mixture model (GMM) as proposed by Gao et al. [27] or with the RML methods with some modifications [50]. In the GMM approach for uncertainty quantification, the DGN approach proposed in this paper is applied to find multiple local MAP estimates by minimizing the objective function defined in Eq. 2. Then, the posterior PDF defined in Eq. 1 can be properly approximated by superposition of multiple Gaussian models called GMM. More details about how to construct an accurate enough GMM are discussed in the paper of Gao et al. [27].

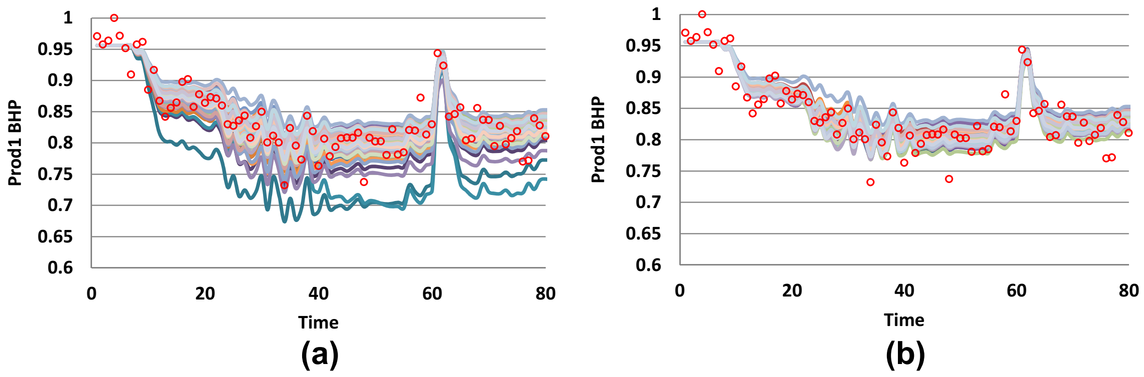

Fig. 15 Comparison of normalized BHP profiles for well Prod-1. **a** Generated from initial realizations. **b** Generated from realizations after history matching

5 Conclusions

In the traditional local GN minimization method with line search or trust region search methods, an approximate quadratic model of the actual objective function is constructed by fitting some subset of the data points evaluated in the current and previous iterations that are close enough to the current best solution. However, in this traditional approach, there is no sharing of information among different optimization tasks, and generally, it is required to run a large number of simulations when the method is applied to find multiple best matches. To improve the efficiency of GN for this type of optimization problems, a DGN approach is proposed in this paper. The DGN method is validated with a nonlinear toy problem and applied to a real history matching problem. According to our theoretical discussions and numerical validations, we can draw the following conclusions:

1. DGN efficiently and effectively shares information among different realizations, and therefore, it performs more efficiently than traditional DFO methods (such as GN-DPS). In addition to results obtained in the current iteration, DGN can also reuse available results obtained in previous iterations.
2. DGN only fits points that are locally clustered within the same basin, which is similar to the traditional local GN approaches. Therefore, it also inherits the robustness of the traditional local GN approaches of being able to converge to a local minimum.
3. DGN has the capability of finding multiple local best matches (or MAP estimates) and may identify valleys (or null spaces) of the objective function.

As shown by Gao et al. [27], when properly combined with parameter reduction techniques and appropriately integrated with the GMM approach, DGN has the potential to generate unbiased, acceptable conditional realizations for nonlinear problems. In the future, we will continue our research, e.g., to further validate the applicability of DGN for real history matching problems, especially uncertainty quantification of model parameters and production forecasting after conditioning to production data, and to benchmark its performance (both in terms of robustness and efficiency) against other methods, e.g., ensemble-based approaches.

References

1. Aanonsen, S.I., et al.: The ensemble Kalman filter in reservoir engineering—a review. SPE J. **14**(3), 393–412 (2009)
2. Alabert, F.: The practice of fast conditional simulations through the LU decomposition of the covariance matrix. Math. Geol. **19**(5), 369–386 (1987)
3. Audet, C., Dennis, J.E.: Mesh adaptive direct search algorithms for constrained optimization. SIAM J. Optim. **17**(1), 188–217 (2006)
4. Bejar, B., Belanovic, P., Zazo, S.: Distributed Gauss-Newton method for localization in ad-hoc networks. In: 44th Asilomar Conference on Signals, Systems and Computers held in Pacific Grove. doi:10.1109/ACSSC.2010.5757776 (2010)
5. Chavent, G., Dupuy, M., Lemonnier, P.: History matching by use of optimal theory. SPE J. **15**(1), 74–86 (1975)
6. Chen, C., et al.: Assisted history matching using three derivative-free optimization algorithms. Paper SPE-154112-MS presented at the SPE Europec/EAGE annual conference held at Copenhagen (2012)
7. Chen, C., et al.: Assisted history matching of channelized models using pluri-principal component analysis. SPE J. doi:10.2118/173192-PA (2016)
8. Chen, W.H., et al.: A new algorithm for history matching. SPE J. **14**(6), 593–608 (1974)
9. Chen, Y., Oliver, D.: Levenberg-Marquardt forms of the iterative ensemble smoother for efficient history matching and uncertainty quantification. Comput. Geosci. **17**(4), 689–703 (2013)
10. Chen, Y., Oliver, D.: Ensemble-based closed-loop optimization applied to Brugge field. SPE Reserv. Eval. Eng. **13**(1), 56–71 (2010)
11. Cheng, B., et al.: Distributed Gauss-Newton methodology for node localization in wireless sensor networks. In: IEEE 6th Workshop on Signal Processing Advances in Wireless Communication (SPAWC) held in New York. doi:10.1109/SPAWC.2005.1506273 (2005)
12. Cheng, H., Dehghani, K., Billiter, T.C.: A structured approach for probabilistic-assisted history matching using evolutionary algorithms: Tengiz field applications. Paper SPE-116212-MS presented at the SPE annual technical conference and exhibition held in Denver (2008)
13. Christie, M.A., Demyanov, V., Erbsa, D.: Uncertainty quantification for porous media flows. J. Comput. Phys. **217**, 143–158 (2006)
14. Coats, K.H., Dempsey, J.R., Henderson, J.H.: A new technique for determining reservoir description from field performance data. SPE J. **10**(1), 66–74 (1970)
15. Davis, M.: Production of conditional simulations via the LU decomposition of the covariance matrix. Math. Geol. **19**(2), 91–98 (1987)
16. Do, S.T., Reynolds, A.C.: Theoretical connections between optimization algorithms based on an approximate gradient. Comput. Geosci. **17**(6), 959–973 (2013)
17. Doucet, A., et al.: Rao-black wellised particle filtering for dynamic bayesian networks. In: Proceedings of the Sixteenth Conference on Uncertainty in Artificial Intelligence, pp. 176–183. Morgan Kaufmann Publishers Inc. (2000)
18. Elsheikh, A.H., Wheeler, M.F., Hoteit, I.: Clustered iterative stochastic ensemble method for multi-modal calibration of subsurface flow models. J. Hydrol. **491**, 40–55 (2013)
19. Erway, J.B., Gill, P.E., Griffin, J.D.: Iterative methods for finding a trust-region step. SIAM J. Optim. **20**(2), 1110–1131 (2009)
20. Evensen, G.: Data Assimilation: The Ensemble Kalman Filter. Springer, New York (2007)
21. Fonseca, R.R., et al.: A stochastic simplex approximate gradient (StoSAG) for optimization under uncertainty. Int. J. Numer. Meth. Engng. doi:10.1002/nme.5342
22. Gao, G., Reynolds, A.C.: An improved implementation of the LBFGS algorithm for automatic history matching. SPE J. **11**(1), 1–17 (2006)
23. Gao, G., Zafari, M., Reynolds, A.C.: Quantifying uncertainty for the PUNQ-S3 problem in a Bayesian setting with RML and EnKF. SPE J. **11**(4), 506–515 (2006)

24. Gao, G., Li, G., Reynolds, A.C.: A stochastic optimization algorithm for automatic history matching. SPE J. **12**(2), 196–208 (2007)
25. Gao, G., et al.: An efficient optimization work flow for field-scale in-situ upgrading developments. SPE J. **20**(4), 701–716 (2015)
26. Gao, G., et al.: A parallelized and hybrid data-integration algorithm for history matching of geologically complex reservoirs. SPE J. doi:10.2118/175039-PA (2016)
27. Gao, G., et al.: Uncertainty quantification for history matching problems with multiple best matches using a distributed Gauss-Newton method. SPE-181611-MS, SPE Annual Technical Conference and Exhibition held in Dubai (2016)
28. Gould, N.I.M., Robinson, D., Thorne, H.S.: On solving trust-region and other regularized subproblems in optimization. Math. Prog. Comp. **2**(1), 21–57 (2010)
29. Gray, G.A., Kolda, T.G.: Algorithm 856: APPSPACK 4.0: Asynchronous parallel pattern search for derivative-free optimization. ACM Trans. Math. Softw. **32**(3), 485–507 (2006)
30. Gu, Y., Oliver, D.S.: An iterative ensemble Kalman filter for multiphase fluid flow data assimilation. SPE J. **12**(4), 438–446 (2007)
31. Hajizadeh, Y., Christie, M., Demyanov, V.: Comparative study of novel population-based optimization algorithms for history matching and uncertainty quantification: PUNQ-S3 revisited. Paper SPE-136861-MS presented at the international petroleum exhibition & conference held in Abu Dhabi (2010)
32. Holland, J.H.: Adaptation in Natural and Artificial System, p. 1992. The University of Michigan Press, Ann Arbor (1975). ISBN: 978-0-26258-111. Reprinted by MIT Press
33. Honorio, J., et al.: Integration of PCA with a novel machine learning method for reparameterization and assisted history matching geologically complex reservoirs. Paper SPE-175038 presented at SPE annual technical conference and exhibition held in Houston (2015)
34. Hooke, R., Jeeves, T.A.: Direct search solution of numerical and statical problems. J. Assoc. Comput. Mach. **8**, 212–229 (1961)
35. Kennedy, J., Eberhart, R.: Particle swarm optimization. In: Proceedings of the IEEE International Conference on Neural Networks, pp. 1942–1945. Perth (1995)
36. Kirkpatrick, S., et al.: Optimization by simulated annealing. Science **220**(4598), 671–680 (1983)
37. Kitanidis, P.K.: Quasi-linear geostatistical theory for inversing. Water Resour. **31**(10), 2411–2419 (1995)
38. Li, G., Reynolds, A.C.: Uncertainty quantification of reservoir performance predictions using stochastic optimization algorithm. Comput. Geosci. **15**(3), 451–462 (2011)
39. Li, R., Reynolds, A.C., Oliver, D.S.: Simultaneous estimation of absolute and relative permeability by automatic history matching of three-phase flow production data. JCPT **43**(3), 37–46 (2004)
40. Liu, X., Reynolds, A.C.: Gradient-based multiobjective optimization with application to waterflooding. Optim. Comput. Geosci. **20**, 677-693 (2016)
41. Liu, X., Reynolds, A.C.: Gradient-based multiobjective optimization for maximizing expectation and minimizing uncertainty or risk with application to optimal well control problem with only bound constrtaints. SPEJ. doi:10.2118/173216-PA (2016)
42. Mohamed, L., et al.: History matching and uncertainty quantification: multiobjective particle swarm optimisation approach. Paper SPE-143067-MS presented at the SPE EUROPEC/EAGE annual conference and exhibition held in Vienna (2011)
43. Mohamed, L., et al.: Population MCMC methods for history matching and uncertainty quantification. Comput. Geosci. **16**, 423–436 (2012)
44. More, J.J., Sorensen, D.C.: Computing a trust region step. SIAM J. Sci. Stat. Comput. **4**(3), 553–572 (1983)
45. More, J.J., Wild, S.M.: Benchmarking derivative-free optimization algorithms. SIAM J. Optim. **20**(1), 172–191 (2009)
46. Nelder, J.A., Mead, R.: A simplex method for function minimization. Comput. J. **7**, 308–313 (1965)
47. Nocedal, J., Wright, M.: Numerical Optimisation. Springer Verlag (1999)
48. Okano, H., Koda, M.: An optimization algorithm based on stochastic sensitivity analysis for noisy landscapes. Reliab. Eng. Syst. Safety **79**, 245–252 (2003)
49. Oliver, D.S.: On conditional simulation to inaccurate data. Math. Geol. **28**, 811–817 (1996)
50. Oliver, D.S.: Metropolized randomized maximum likelihood for sampling from multimodal distributions. arXiv:1507.08563 (2015)
51. Oliver, D.S., Chen, Y.: Recent progress on reservoir history matching: A review. Comput. Geosci. **15**, 185–211 (2011)
52. Oliver, D.S., Reynolds, A.C., Liu: Inverse Theory for Petroleum Reservoir Characterization and History Matching. Cambridge University Press (2008)
53. Ouenes, A., Bhagavan, S.: Application of simulated annealing and other global optimization methods to reservoir description: myths and realities. Paper SPE-28415-MS presented at the SPE annual technical conference and exhibition held in New Orleans (1994)
54. Powell, M.J.D.: Least Frobenius norm updating of quadratic models that satisfy interpolation conditions. Math. Program. **100**(1), 183–215 (2004)
55. Powell, M.J.D.: Developments of NEWUOA for minimization without derivatives. IMA J. Numer. Anal. (2008)
56. Reynolds, A.C., Le, D.H., Emerick, A.A.: Ensemble-based methods for data integration and uncertainty quantification. In: 2nd EAGE Integrated Reservoir Modeling Conference (2014)
57. Reynolds, A.C.: My decade-long journey through the field of ensemble-based data assimilation. In: Proceedings of the Ninth International EnKF Workshop. Bergen (2014)
58. Rojas, M., Sorensen, D.C.: A Trust-region approach to the regularization of large-scale discrete form of ill-posed problems. SIAM J. Sci. Comput. **23**(6), 1842–1860 (2002)
59. Schön, T., Gustafsson, F., Nordlund, P.J.: Marginalized Particle Filters for Nonlinear State-Space Models. Linköping University Electronic Press (2003)
60. Smith, K.W.: Cluster ensemble Kalman filter. Tellus A **59**(5), 749–757 (2007)
61. Sorensen, D.C.: Minimization of a large-scale quadratic function subject to a spherical constraint. SIAM J. Optim. **7**(1), 141–161 (1997)
62. Spall, J.C.: Adaptive stochastic approximation by the simultaneous perturbation method. IEEE Trans. Autom. Control **45**(10), 1839–1853 (2000)
63. Stordal, A.S., Karlsen, H.A., Neavdal, G., Oliver, D.S., Skaug, H.J.: Filtering with state space localized Kalman gain. Phys. D: Nonlin. Phenom. **241**(13), 1123–1135 (2012)
64. Stuart, A.M.: Inverse problem: A Bayesian perspective. Acta Numerica **19**, 45–559 (2010)
65. Tarantola, A.: Inverse problem theory and methods for model parameter estimation. SIAM (2005)
66. Wild, S.M.: Derivative free optimization algorithms for computationally expensive functions. Ph.D dissertation, Cornell University Ithaca, New York (2009)
67. Zhao, H., et al.: Large-scale history matching with quadratic interpolation models. Comput. Geosci. **17**(1), 117–138 (2013)
68. Zhao, T., Nehorai, A.: Information-driven distributed maximum likelihood estimation based on Gauss-Newton method in wireless sensor network. IEEE Trans. Signal Process. **55**(9), 4669–4682 (2007)
69. Zhou, W., Zhang, L.: Global convergence of a regularized factorized quasi-Newton method for nonlinear least squares problems. Comput. Appl. Math. **29**(2), 195–214 (2010)