NAFTA-GAZ, ROK LXXII, Nr 10 / 2016

DOI: 10.18668/NG.2016.10.06

Piotr Łętkowski, Wiesław Szott

Oil and Gas Institute – National Research Institute

Determination of Basic Reservoir Parameters in Shale Formations as a Solution of Inverse Problem in the Computer Assisted History Matching of their Simulation Models. Part II – Hybrid Optimization Algorithm

The paper presents a hybrid optimization algorithm as a practical method to solve the inverse problem of simulation model calibration process. The method is applied to determine basic reservoir parameters in shale formations as a result of the process carried out for their models. Due to some specific features of the problem including its non-linearity and the large size of the solution space, an algorithm that can be employed in the process of model automatic calibration has to run fast and be effective in finding acceptable solution using limited number of simulations. The selection of an appropriate global optimization method is crucial in the situation of many expected local minima of the problem. One of the stochastic sampling method used and presented in the paper is the method of Particle Swamp Optimization (PSO). Such a method is characterized by a simple concept, fast convergence, and intelligent balance between searching and testing of the solution space. Besides the PSO method three other elements are combined to result in the effective solution of the problem. They include: search with stable Levy distribution of iteration step size, Latin hypercube sampling and response surface. The combination of the elements employs both deterministic and stochastic approaches that make the proposed solution both robust and effective. The algorithm was positively tested for convergence and performance using special functions that are commonly applied for such purposes.

Key words: reservoir simulation models, inverse problem, optimization methods, particle swarm optimization.

Rozpoznanie parametrów formacji łupkowych poprzez rozwiązanie problemu odwrotnego metodą wspomaganej komputerowo kalibracji modeli symulacyjnych. Część 2 – hybrydowy algorytm optymalizacyjny

W pracy przedstawiono hybrydowy algorytm optymalizacyjny, stanowiący metodę praktycznego rozwiązania kalibracyjnego problemu odwrotnego. Ze względu na rodzaj problemu, jego nieliniowość oraz rozmiar przestrzeni rozwiązań, algorytm stosowany w procesie automatycznej kalibracji modelu symulacyjnego musi szybko operować w wielowymiarowej przestrzeni rozwiązań oraz skutecznie poszukiwać dobrych rozwiązań, przy ograniczonej liczbie symulacji. Wybór odpowiedniej metody optymalizacji ma szczególne znaczenie w sytuacji, gdy poszukujemy oszacowania wielu parametrów, przy obecności wielokrotnych minimów lokalnych. Właśnie umiejętność radzenia sobie z obecnością minimów lokalnych była jedną z głównych przyczyn rozwoju metod optymalizacji, opartych na próbkowaniu stochastycznym. Jedną z odmian próbkowania stochastycznego są metody oparte na tzw. inteligencji roju, do których należy przedstawiona w artykule metoda optymalizacji rojem cząstek. Metody te ze względu na prostotę idei, zbieżność oraz zachowanie równowagi pomiędzy eksploracją i eksploatacją przestrzeni rozwiązań są powszechnie stosowane. Zaproponowany algorytm stanowi element komputerowo wspomaganej kalibracji modelu symulacyjnego formacji łupkowej, celem określenia jej istotnych parametrów. W prezentowanej metodzie zaimplementowano kombinację nowoczesnych technik, tj. optymalizacja rojem cząstek, przeszukiwanie przestrzeni rozwiązań z wykorzystaniem stabilnego rozkładu Levy'ego, próbkowanie hipersześcianu łacińskiego oraz funkcja powierzchni odpowiedzi. W celu rozwiązania postawionego problemu, algorytm łączy w sobie techniki deterministyczne i stochastyczne, co pozwala na wyeliminowanie wad każdej z metod. Ponadto przedstawiono wyniki testów zbieżności zbudowanego algorytmu, potwierdzając przy tym jego efektywność przy przeszukiwaniu przestrzeni rozwiązań.

Słowa kluczowe: symulacyjne modele złożowe, problem odwrotny, metody optymalizacyjne, optymalizacja rojem cząstek.

Introduction

This paper contains a detailed description of the method introduced in [12] to determine significant parameters of shale formations (such as permeability and porosity of secondary fractures, whose measurement is difficult or simply impossible) via the application of simulation models in the process of their calibration. Paper [12] provides general principles of the applied method and an example of its use for realistic examples of domestic shale formations.

From a theoretical point of view a simulation reservoir model is a complex mathematical system, defined by a large number of parameters and described by a complicated system of non-linear differential equations, while the calibration of the simulation model is an attempt to resolve the inverse problem, that is such choice of its parameters that the obtained results would guarantee matching the observation data. That means that when calibrating the simulation model an attempt is made to modify the input data (e.g. reservoir rock properties) in such a way as to obtain the pre-set values of output quantities (e.g. the bottom hole pressure) [4, 15]. However, attempts to resolve the problem presented in this way encounter a number of difficulties. First of all, the inverse calibration problem is ill posed, which means that the calibration does not have a unique solution, that is various combinations of changes in the model can give very similar results and frequently it is difficult to decide, which solution is better. The solution space size is another problem. Simulation models have frequently a few thousand blocks with defined quantities (porosity, permeability etc.). Moreover, the model defines discontinuities of the reservoir rock (faults), contacts of deposit fluids and their properties, water-bearing layers properties and many other quantities. That means that the number of possible combinations of those parameters is difficult to define. The simulation model calibration consists just in finding among those combinations of such one, which ensures matching of the model with the observation data. Because of the nature of phenomena described by the simulation model, the solution space size (possible parameter combinations) and couplings between them, the objective function as a measure of matching between the simulation results and the observation data usually features many local minima, additionally complicating the problem complicated anyway.

So it seems natural to make an attempt to formulate the calibration problem as a problem of global optimisation and to apply optimisation methods to resolve it, which would allow, at least partly, automating the calibration process. The algorithm used in the process of automatic calibration of a simulation model must operate fast in a multidimensional solution space and effectively search for good solutions at a limited number of simulations. The choice of appropriate optimisation method is especially important in the situation, when we are looking for estimation of many parameters at the existence of multiple local minima – that is in the situation, which we encounter in the case of reservoir model calibration. One of first attempts to build a mechanism supporting the simulation model calibration consisted in formulating the problem as the optimum control issue [3]. A few categories of methods supporting the simulation model calibration may be distinguished today. These are e.g. gradient methods [1], particle filtering methods [6, 8, 9] and stochastic sampling methods. In recent years the stochastic sampling method became very popular, which resulted in its fast development, which may be observed in the form of numerous alternative implementations [13]. The particle swarm optimisation and so-called ant algorithm are among most popular optimisation methods included in this group [2, 7, 10, 11]. The presented algorithm is based on four numerical techniques creating a hybrid optimisation method. These are: the particle swarm optimisation, the solution space searching with the use of stable Levy distribution, the sampling of Latin hypercube, and the response surface function. The algorithm effectiveness was verified using the De Jong test environment, carrying out tests of its convergence for selected test functions.

Latin Hypercube Sampling

The Latin hypercube sampling belongs to the group of experimental design techniques and is one of most effective methods to generate random samples based on the probability density function [5]. The idea of the method boils down to the division of the solution space into separate sub-areas via the division of variability intervals for each component in the N-dimensional solution space into a specified number of sub-intervals with identical probability of the variable assuming values from each sub-interval. Then a specified number of samples is generated in such a way as to obtain one sample, built based on the combinatorics between all dimensions in the solution space, from each sub-space (Fig. 1).

In the presented algorithm the Latin hypercube sampling is responsible for preliminary searching of the solution space to determine promising directions of searching for the global minimum. Initial conditions for the optimisation algorithm are determined based on solutions generated by means of hypercube sampling.

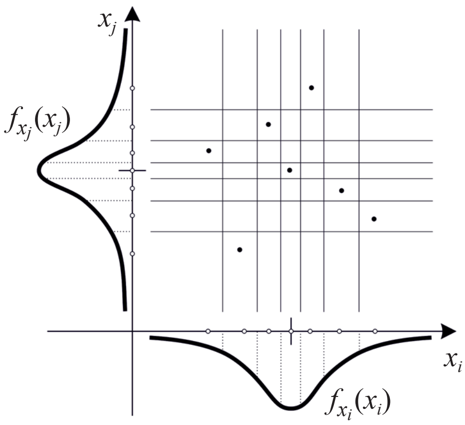

Fig. 1. The principle of random samples generation by means of Latin hypercube sampling using the example of two random variables

Particle Swarm Optimisation

In the particle swarm optimisation the solutions, referred to as particles, cooperate to find the optimum solution [7]. During the optimisation each particle changes its position in the solution space via determination of a velocity pseudo-vector (position change vector). This vector is modified using the information on the search history both for the given particle and for the other particles of the swarm.

Such way of solution space searching may be described using the following formulae:

position:
$$x^i_{t+1} = x^i_t + v^i_{t+1} \tag{1}$$

where time is dimensionless and discrete, $\Delta t = 1$
pseudo-velocity:

$$v^i_{t+1} = \omega v^i_t + c_1 r_1(p^i_t - x^i_t) + c_2 r_2(p^g_t - x^i_t) \tag{2}$$

In each iteration step, t the position of the particle in the multidimensional space, x^i_t is modified in accordance with formula (1) via the velocity vector, v^i_{t+1} (2). The velocity vector in step $t + 1$ is a function of the previous velocity vector value (v^i_t), position x^i_t, the best solution found by this particle, p^i_t and the best solution found by all particles, p^g_t.

The other quantities existing in formula (2) have been defined below:
ω – constant or variable coefficient of inertia,
r_1, r_2 – random numbers from the (0, 1) interval,
c_1, c_2 – scaling parameters (variable or constant).

Parameter ω referred to as a coefficient of inertia specifies the influence of particle velocity in the previous iteration step on its current position. This is significant for the algorithm convergence and in most applications exists in a form depending on the iteration number:

$$\omega = \omega_{\min} + (\omega_{\max} - \omega_{\min}) \cdot \left(1 - \frac{k-1}{k_{\max} - 1}\right)$$

where:
$\omega_{\max}$, $\omega_{\min}$ – maximum and minimum parameter value,
k, $k_{\max}$ – iteration number and the planned number of iterations,

Values $\omega_{\max}$, $\omega_{\min}$ depend on the problem being resolved and usually the following are assumed:

$$\omega_{\max} \in (0.9,\ 1.1)$$
$$\omega_{\min} \in (0.1,\ 0.5)$$

In accordance with the above formulae each particle of the swarm searches the solution space modifying the position based on its best solution, p^i_t and using the information about the best swarm solution, p^g_t. Scaling coefficients c_1, c_2 allow to control the influence of individual terms of the velocity vector on the solution. If $c_1 = 0$, the particle uses only the information about the best swarm solution. For $c_2 = 0$, the particle searches the solution on its own, not taking into consideration the solutions obtained by other particles. Usually constant values of coefficients are assumed, $c_1 = c_2 = 2.05$. In the form presented above the particle swarm optimisation is the basic optimisation method of the presented algorithm.

Levy Flight

So-called Levy flight has been applied in the used optimisation algorithm to prevent a premature convergence of the basic optimisation algorithm [7]. Its job is to introduce to the algorithm of an additional stochastic element preventing too early finding of a local solution, which usually means the lack of algorithm convergence. 'Levy flight' is a stochastic Markov process consisting in searching the variables space, where after a sufficiently large number of steps their lengths may be described by a stable Levy distribution. This distribution is usually determined by means of so-called Mantegna's algorithm.

According to this algorithm the step length is described by the formula:

$$v = \frac{x}{|y|^{\frac{1}{\alpha}}}$$

where:
x, y – random variables with normal distribution,
α – coefficient defining the way of searching, $\alpha \in (0.3, 2.0)$.

Standard deviations for random variables distributions, x, y have been defined below:

$$\sigma_x = \left[\frac{\Gamma(1+\alpha)\sin(\frac{\pi\alpha}{2})}{\Gamma\left(\frac{\alpha+1}{2}\right)\alpha\, 2^{\left(\frac{\alpha-1}{2}\right)}}\right]^{\frac{1}{\alpha}}$$

$$\sigma_y = 1$$

where Γ stands for the Gamma Euler function.

The distribution defined in such a way is asymptotically convergent to a stable Levy distribution for large values of the random variable ($|v| \gg 0$). In the original Mantegna's algorithm a non-linear transformation is used at this stage to achieve quick convergence of the defined distribution with the Levy distribution. However, to determine the searching step length it is possible to apply a simplified algorithm used successfully in the Cuckoo Search optimisation method [16]. The modification of optimisation by the particle swarm used in the described algorithm consists in introducing to the solution of a disturbance generated in accordance with the Levy distribution.

Response Surface Function

To improve convergence a response surface function module was introduced to the algorithm, allowing to interpolate the simulation optimisation results based on the already obtained values. The response surface function was taken in the form of the second degree polynomial, presented below:

$$y = a_{0,0} + a_{0,1}x_1 + \ldots + a_{0,N}x_N + a_{1,1}x_1^2 + \ldots + a_{N,N}x_N^2 + a_{1,2}x_1x_2 + \ldots + \\ + a_{1N}x_1x_N + a_{2,3}x_2x_3 + a_{N-1,N}x_{N-1,N}x_N$$

where:
y – objective function,
x_i – model parameters, $i = 1, \ldots N$,
N – the number of optimised model parameters,
$a_{i,j}$ – unknown coefficients.

The suggested shape of the response surface function contains a free term, linear, quadratic, and mixed terms, which allows both to consider a non-linear dependence of the response function, y on the model parameters and interactions between them. However, for unknown coefficients $a_{i,j}$ the adopted model is linear, which substantially facilitates its application.

Having performed $M\left(\geq \frac{(N+1)(N+2)}{2}\right)$ simulations for various values of model parameters it is possible to build and resolve a system of equations for coefficients a_{ij}.

$$\vec{x}^{(j)} = \begin{pmatrix} x_1^{(j)} \\ \ldots \\ x_N^{(j)} \end{pmatrix} \xrightarrow{\text{simulation}} y^{(j)}$$

In the above formula each vector $\vec{x}^{(j)}$ stands for a set of optimised parameters, identifying the simulation model, while $y^{(j)}$ is the value of the objective function at point $\vec{x}^{(j)}$. As a result we obtain a linear system of equations in a matrix form:

$$\vec{y} = \overleftrightarrow{X} \cdot \vec{a}$$

We obtain coefficients $\vec{a}$ after the application of the greatest reliability method, which is an equivalent of the least squares method.

$$\left(\vec{y} - \overleftrightarrow{X} \cdot \vec{a}\right)^T \cdot \overleftrightarrow{G_y} \cdot \left(\vec{y} - \overleftrightarrow{X} \cdot \vec{a}\right) = \min$$

where $\overleftrightarrow{G_y}$ is the matrix of weights – an inverse matrix of objective function values covariances $y^{(j)}$

$$\left(G_y\right)_{ii} = \frac{1}{\sigma_{y_i}^2}$$

$$\left(G_y\right)_{ij} = 0 \text{ for } i \neq j$$

In the above formula $\sigma_{y_i}^2$ is the variance of the function value y_i

As a result we obtain the equation for the vector of $\vec{a}$ coefficients in the form of

$$\overleftrightarrow{X}^T \cdot \overleftrightarrow{G_y} \cdot \left(\vec{y} - \overleftrightarrow{X} \cdot \vec{a}\right) = 0$$

which solution is as follows

$$\vec{a} = -\overleftrightarrow{C_a} \cdot \overleftrightarrow{X}^T \cdot \overleftrightarrow{G_y} \cdot \vec{y}$$

The matrix of the surface coefficients covariances $\overleftrightarrow{C_a}$ existing in the above formula is defined in the following way

$$\overleftrightarrow{C_a} = \left(\overleftrightarrow{X}^T \cdot \overleftrightarrow{\mathrm{G_y}} \cdot \overleftrightarrow{X}\right)^{-1}$$

The determination of the response surface function coefficients, $\vec{a}$ enables determination of its minimum.

Having applied the condition of minimum for the response surface function

$$\left.\frac{\partial \mathrm{y}}{\partial \mathrm{x_i}}\right|_{x_i = x_{oi}} = 0, \quad i = 1, \dots, N$$

we obtain the following system of equations

$$\vec{0} = \overrightarrow{A_1} + \overleftrightarrow{A_2} \cdot \overrightarrow{X_0}$$

the solution of which is

$$\overrightarrow{x_0} = -\left(\overleftrightarrow{A_2}\right)^{-1} \overrightarrow{A_1}$$

where:

$$\overrightarrow{A_1} = \begin{pmatrix} a_{0,1} \\ a_{0,2} \\ \cdots \\ \cdots \\ a_{0,N} \end{pmatrix} \qquad \overleftrightarrow{A_2} = \begin{pmatrix} a_{1,1} & a_{1,2} & \cdots\cdots & a_{1,N} \\ a_{1,2} & a_{2,2} & \cdots\cdots & a_{2,N} \\ \cdots & \cdots & \cdots\cdots & \cdots \\ \cdots & \cdots & \cdots\cdots & a_{N-1,N} \\ a_{1,N} & a_{2,N} & \cdots\cdots & a_{N,N} \end{pmatrix}$$

Moreover, the determination of non-liner transformation Jacobian in the form of

$$x_{oi} = f_i(a_{jk})$$

$$(J)_{ij} = \left(\frac{\partial x_{oj}}{\partial a_i}\right)$$

allows to determine the matrix of coordinates covariances $\overrightarrow{x_0}$

$$\overleftrightarrow{C_{xo}} = \overleftrightarrow{J}^T \cdot \overleftrightarrow{C_a} \cdot \overleftrightarrow{J}$$

where:

$\sigma^2_{x_i} = (C_{xo})_{ii}$ – variance of x_{oi},

$\rho_{x_i x_j} = \dfrac{(C_{xo})_{ij}}{\sigma_{x_i}\,\sigma_{x_j}}$ – correlation of x_{oi} and x_{oj}.

The implementation of the above reasoning in the existing algorithm allows to determine, based on results of reservoir simulations, the minimum of the response surface, and to determine its error measure (variance) and relationships between the matched model parameters.

Hybrid Algorithm

The finally constructed hybrid algorithm operates acc. to the following scheme:

1. The reading in of programme operation parameters (controlling parameters, optimisation method parameters, optimised model parameters, their variability intervals and definitions of control quantities).
2. The reading in of observation data (values of control quantities for specified simulation times).
3. Drawing 'the swarm' of solutions (group of models) for various combinations of optimised model parameters values. Parameter values are determined by the method of Latin hypercube sampling within the pre-set variability intervals. A rule has been adopted that at this stage the algorithm operates on $2N + 1$ models, where N is the number of optimised model parameters.
4. Determination of the objective function value, OF (matching function) for each 'swarm' model. The function used for quantitative assessment of simulation results matching to the measurement data is described by the following formula:

$$\mathrm{OF} = \sum_{i=1}^{N} \frac{w_i}{n_i} \sum_{j=1}^{n_i} u_{ij} \left(\frac{O_{ij} - S_{ij}}{O_{ij}}\right)^2$$

N – number of control quantities, w_i – weight coefficients for the adopted control quantities, n_i – number of measurements for the i-th control quantity, u_{ij} – weight coefficients for measurement points, O_{ij}, S_{ij} – observation data and simulation results, respectively, for the i-th control quantity and the j-th measurement point.

5. Saving in the 'memory' the values of optimised parameters and values of response function obtained for the initial, $(2N + 1)$ – element of the 'solutions swarm'.
6. Saving the best global solution and the best solutions for each element of the 'swarm'.
7. Ranking the models. Ordering models (solutions) acc. to matching with the observation data. The selection $N + 1$ of best solutions (models). From this stage on the algorithm operates on $N + 1$ solutions.
8. Initialisation of the main optimisation loop.
9. Modification of solutions in accordance with the formula taking into account optimisation via a particle swarm or the Levy steps method (with a specified probability).
10. Determination of the objective function value for each 'swarm' model.
11. Updating the best global solution and the best solutions for each element of the 'swarm'.
12. Saving in the 'memory' the values of optimised parameters and values of response function obtained in each iteration for the $N + 1$ element of the 'solutions swarm'. Saving in the 'memory' is performed in such a way that the algorithm would always remember $M = \left(\dfrac{(N+1)(N+2)}{2}\right)$ best solutions from all performed simulations.

13. Starting the procedure of the response surface function if the number of 'memory' elements is equal to *M*. Determination of the response surface function minimum. If the obtained solution is better than the worst solution existing in the 'memory' then the new solution replaces the hitherto worst solution.
14. Checking the conditions of the optimisation completion. If the optimisation completion condition has not been met – return to point 8.

The presented procedure is performed till achieving the condition for computation completion, e.g. reaching a specified number of performed simulations (objective function calls) or achieving the pre-set precision of the measurement data reconstruction.

Algorithm Convergence Tests

The developed algorithm was subject to convergence tests using the De Jong test environment [14]. De Jong tests consist in checking the convergence of optimisation algorithms based on example of defined analytical functions with known extrema. The algorithm convergence was verified using the following set of test functions.

F_1: The first De Jong function.

F_2: Generalised Rosenbrock function.

F_4: Quadratic function with noise.

F_5: Two-dimensional Shekel function (so-called fox's burrows).

F_6: Generalised Rastrigin function.

F_7: Schwefel function.

Fig. 2 presents F_6 – a generalised Rastrigin function in a two-dimensional case.

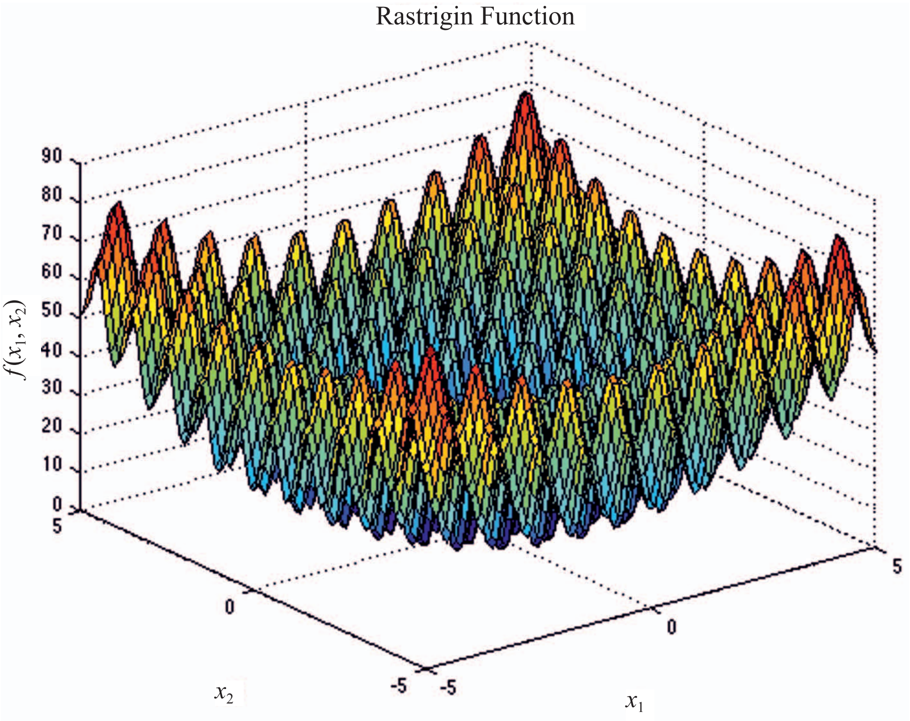

Fig. 2. Two-dimensional Rastrigin function (http://www.sfu.ca/~ssurjano/rastr.html)

The algorithm capability to improve the result was taken as the method convergence measure for individual De Jong test functions. To unify the way of results presentation, the measure of matching (objective function) was adopted in the following form:

$$OF_k = \frac{\left|min_k^{opt} - min^{teor}\right|}{\left|min_0^{opt} - min^{teor}\right|}$$

where:

OF_k – value of the objective function for the *k*-th iteration,
min^{teor} – theoretical minimum of the test function,
min_0^{opt} – minimum obtained in the 0-th iteration,
min_k^{opt} – minimum obtained in the *k*-th iteration.

The objective function adopted in such a way makes that the convergence graph for each test function takes always the value of 1 for the zeroth iteration, where the result of the zeroth iteration should be understood as the minimum obtained in point 6 of the algorithm. Tests, except for function F_5, were carried out for $N = 10$ (solution space dimension) assuming maximum 500 calls of the objective function. Because of a probabilistic nature of the proposed method the optimisation process for each test function was repeated 100 times and the results were averaged. Results are presented in Fig. 3.

The worst results were obtained for function F_6 and F_7, which proves a significant dependence of the algorithm convergence on the number of local minima. For F_7, which in the tested variability interval has only 4 minima, a more than 4 times better result was obtained compared to a multi-modal Rastrigin function (F_6). For the remaining test functions good and very good convergences of the tested algorithm were obtained. In particular good results were obtained for a generalised Rosenbrock function, which as the only used test function features a coupling between variables.

It is necessary to draw attention to the fact that for all functions the process of the objective function minimisation proceeds continuously during all the time of algorithm operation, which appears in the lack of long horizontal sections in the convergence graphs. This proves a good effectiveness of the algorithm both at a global searching of the solution space and at a local optimisation.

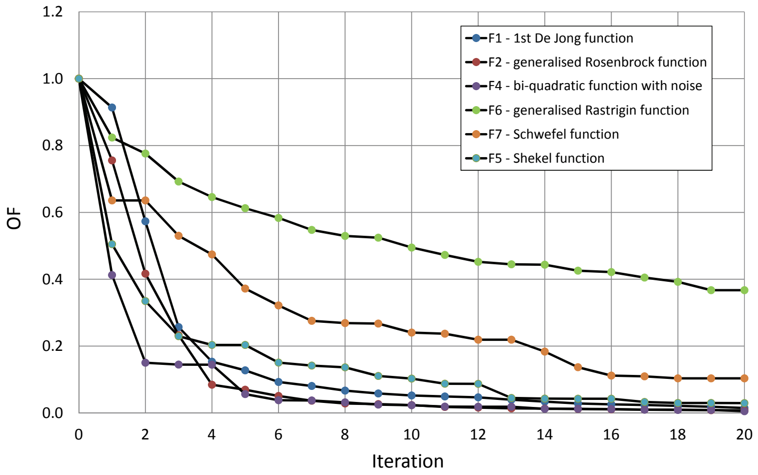

Fig. 3. Effectiveness test of the hybrid algorithm. The De Jong test environment

Summary

The paper presents theoretical foundations, convergence tests, and the scheme of hybrid optimisation method operation. The proposed method to resolve an inverse calibration problem was based on known numerical techniques allowing to search effectively a multi-dimensional solution space. The applied techniques include: Latin hypercube sampling, particle swarm optimisation, Levy flight method and the response surface function. The performed tests of the developed algorithm operation and convergence entitle to formulate the following comments and conclusions:

- Selected numerical techniques creating the presented hybrid optimisation method have turned out to be effective. Each of them is responsible for another element of the algorithm and together they create an effective optimisation algorithm.
- The Latin hypercube sampling guarantees an effective initial searching of the solution space at a relatively small number of performed simulations. The carried out tests have shown that the performance, at the initial stage of the solution space recognition, of $2N + 1$ simulations (N – the number of optimisation variables) ensures a satisfactory convergence of the algorithm. What's more, an increased number of simulations at this stage does not significantly improve the subsequent algorithm convergence.
- The applied particle swarm optimisation was properly chosen. This method worked very well as the basic optimisation method. It is relatively simple to implement and it may be easily combined with other optimisation methods creating hybrid algorithms.
- The solution space searching using a stable Levy distribution has substantially improved the effectiveness of the presented optimisation method. Fig. 4 presents a comparison of the optimisation algorithm convergence in two versions: particle swarm optimisation algorithm (PSO), and the hybrid algorithm. Making more than 30 iterations the PSO algorithm changed the model matching to the observation data only twice, correcting the initial match only 1.3 times (0.24/0.18). At the same time the hybrid algorithm corrected the initial result eight times (0.24/0.03). Differences in the convergence for both variants of the method are clearly seen in the presented graph. For the PSO algorithm we have long horizontal sections proving the algorithm stagnation, while the hybrid algorithm convergence features permanent improving of the already obtained matching.

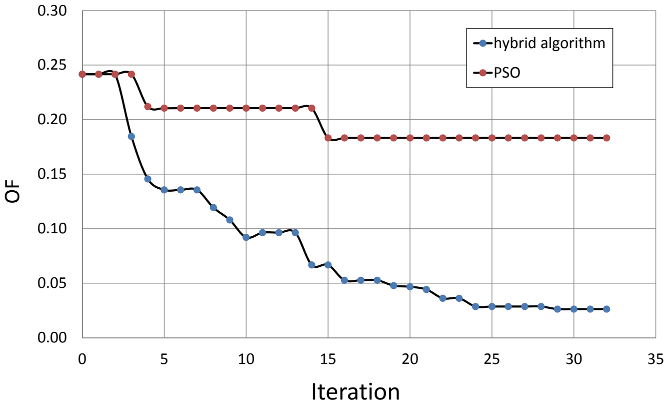

Fig. 4. Optimisation algorithm convergence – the influence of method hybridisation. Convergence tests were performed on a synthetic model of a shale formation, for 10 model parameters

- The theoretical foundations of response surface function presented in the paper were applied in the hybrid algorithm, which enabled interpolation of the objective function based on results of simulation obtained during optimisation. However, attention should be drawn to the fact that the response surface function is used in close surroundings of the global minimum, i.e. having completed the proper optimisation process. The mechanism for algorithm convergence improvement using the response surface function applied in the presented method has a very little impact on the algorithm convergence.

Please cite as: Nafta-Gaz 2016, no. 10, pp. 814–821, DOI: 10.18668/NG.2016.10.06

Article contributed to the Editor 4.04.2016. Approved for publication 18.07.2016 r.

The work was carried out by the authors under the Blue Gas project entitled *Integrated Model of Reservoir Engineering for the Gas Reservoirs Extraction in Shale*, the task *Development of Gas Reservoirs Extraction Models in Shale Formations for Better Determination of Significant Parameters of the Formation and Its Development and for Reliable Forecasting the Gas Extraction Process in the Future* – the INiG work to the order of NCBiR, archive No.: DK-0601/9/13, order No.: 6112/KZ/13.

Literature

[1] Anterion F., Eymard F.: *Use of parameter gradients for reservoir history matching. SPE Symposium on Reservoir Simulation*. Houston, Texas 6–8 February 1989.

[2] Bonabeau B., Dorigo M., Theraukaz G.: *Inspiration for optimization from social insect behavior*. Nature 2000, 406 (6791), 39–42.

[3] Chen W. H., Gavalas G. R., Seinfelt J. H., Wasserman M. L.: *A New algorithm for automatic history matching*. SPE 4545. SPE J. 1974, vol. 14(6), pp. 593–608.

[4] Dean S. Oliver, Albert C. Reynolds, Ning Liu: *Inverse Theory for Petroleum Reservoir Characterization and History Matching*. Cambridge University Press, May 2008.

[5] Goda T., Sato Kozo: *History matching with iterative Latin hypercube samplings and parameterization of reservoir heterogeneity*. Journal of Petroleum Science and Engineering, February 2014, vol. 114, pp. 61–73.

[6] Jafarpour B., Mc Laughlin D. B.: *History matching with an ensemble kalman filter and discrete cosine parametrization*. SPE 108761. SPE Annual Technical Conference and Exhibition, Anaheim, California 11–14 November 2007.

[7] Kathrada M.: *Uncertainty evaluation of reservoir simulation models using particle swarms and hierarchical clustering*. PhD thesis 2009, Institute of Petroleum Engineering, Heriot Watt University, Edinburgh, United Kingdom.

[8] Liu N., Olivier D. S.: *Critical evaluation of the Ensemble Kalman Filter on History Matching of Geologic Facies*. SPE 92867. SPE Reservoir Simulation Symposium. The Woodlands, Texas 31 January – 2 February 2005.

[9] Lodoen O. P., More H.: *Scale-corrected ensemble kalman filter applied to production-history conditioning in reservoir evaluation*. SPE 111374. SPE J. 2008, vol. 13(2), pp. 177–194.

[10] Łętkowski P.: *Zastosowanie algorytmu mrówkowego w procesie kalibracji symulacyjnego modelu złożowego*. Nafta-Gaz 2012, no. 2, pp. 98–104.

[11] Łętkowski P.: *Zastosowanie hybrydowej metody optymalizacji rojem cząstek w procesie automatycznej kalibracji modeli złożowych*. Nafta-Gaz 2014, no. 11, pp. 784–793.

[12] Łętkowski P., Szott W.: *Determination of Basic Reservoir Parameters in Shale Formation as a solution of Inverse Problem in the Computer Assisted History Matching of their Simulation models. Part I – Introduction to Methodology and Initial Tests*. Nafta-Gaz 2015, no. 11, pp. 870–876.

[13] Mohamed L., Christie M., Demyanov V.: *Comparison of stochastic sampling algorithms for uncertainty quantification*. SPE 119139. SPE Reservoir Simulation Symposium, Woodlands, Texas 2–4 February 2009.

[14] Molga M., Smutnicki C.: *Test functions for optimization needs*. 2005 (unpublished).

[15] Np.: *Basic elements of a reservoir characterization study*. PetroWiki – Published by SPE International. http://petrowiki.org/Basic_elements_of_a_reservoir_characterization_study (acces on: 31.03.2015).

[16] Yang X.-S., Deb S.: *Cuckoo search via Lévy flights*. [in:] Proceedings of the IEEE World Congress on Nature & Biologically Inspired Computing (NaBIC) 2009, pp. 201–214.

Dr. Eng. Piotr ŁĘTKOWSKI PhD.
Assistant Professor
Department of Hydrocarbon Deposits and UGS Facilities Simulation
Oil and Gas Institute – National Research Institute
ul. Lubicz 25 A, 31-503 Kraków
E-mail: *piotr.letkowski@inig.pl*

Dr. Wiesław SZOTT PhD.
Assistant Professor
Head of the Department of Hydrocarbon Deposits and UGS Facilities Simulation
Oil and Gas Institute – National Research Institute
ul. Lubicz 25 A, 31-503 Kraków
E-mail: *wieslaw.szott@inig.pl*