GLOBAL EXPLORATION OF OPTIMIZATION LANDSCAPES FOR NONLINEAR ILL POSED PARAMETER ESTIMATION PROBLEMS

Oliver Schweissgut[1], Wolfgang Wiechert[1]

[1] Institut für Biotechnologie 2, Forschungszentrum Jülich, 52425 Jülich

oliver.schweissgut@os-cillation.de (Oliver Schweissgut)

Abstract

Current approaches to parameter estimation concentrate on finding the global optima of the least squares problem of likelihood functions. Due to poorly determined parameters and noisy experimental data only finding the optimum does not meet the requirements of life science researchers. The shape of the optimum and the possibly other local optima which are almost as good as the global optimum are of key interest.

The algorithm described in the present contribution explores the parameter space of the objective function and produces a piecewise approximation of the objective function by quadratic functions, whereas the approximation quality can be adjusted at the cost of higher computational effort. In this way it is possible to gain more information on the global optimum, its shape and its neighborhood. On the other hand the algorithm is not able to guarantee the best solution and arbitrary fine details of the objective function get lost.

Keywords: Nonlinear Parameter Estimation, Global Function Exploration, Branch and Bound Algorithm, Ill posed Problems

Presenting Author's biography

OLIVER SCHWEISSGUT studied computer science at the University of Siegen and obtained his diploma in 2006. In 2000 he founded the software engineering company os-cillation in Siegen. Beside the position as CEO of os-cillation, he is a part-time Ph.D. student at Research Center Jülich since 2006.

1 Background

1.1 Motivation

Robust and efficient methods for parameter estimation are of key importance within the modeling and simulation workflow. This computational step is even dominant in the life sciences where it also has the purpose to find out if the measured data contains sufficient information to determine the model parameters at all. This contribution focuses on the nonlinear differential-algebraic (DAE) type of models. The parameter estimation step is then usually formulated as a nonlinear optimization problem, based on least squares of likelihood functions.

In the sketched situation, this leads to substantial problems with poorly (if at all) determined optima, poor scaling, convergence of standard algorithms to different solutions, or even multiple local optima. Here, standard linearization methods (e.g., based on an approximated covariance matrix) can be strongly misleading because of the non-ellipsoidal and flat neighborhood of the minimized nonlinear function (Fig. 1). A poorly determined optimum can have a multi-dimensional banana shape or even worse. In practice, the question remains unanswered what the true shape of the function close to the optimum is and whether multiple but separated solutions exist. A global overview of the shape of the minimized function is hard to obtain in dimension >3.

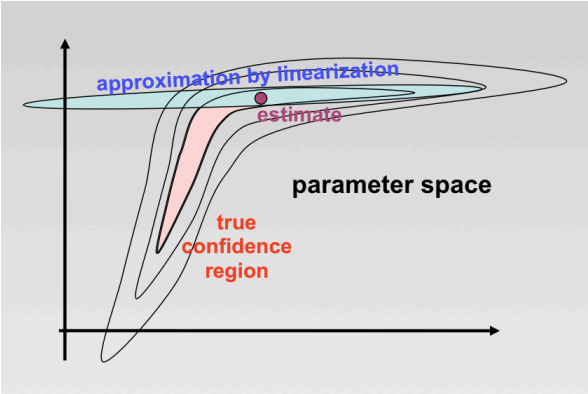

Fig. 1 Nonlinear confidence regions, typical situation for poorly determined parameters

These characteristics can hardly be addressed by traditional local or even global optimization methods and necessitate alternative approaches. The new approach presented in this contribution will be a modification of existing global optimization algorithms in such a way that the global shape of the function can be explored rather than focusing on the optimum itself.

1.2 Existing methods

There are mainly two different approaches described for global optimization in literature [4]. Algorithms with a heuristic approach are, e.g., evolution strategies and simulated annealing. On the other hand, there are exact methods like complete search [11] and several branch and bound algorithms [2,3,5,6,8]. Exact methods with a branch and bound approach split the parameter space into smaller regions and calculate a lower and upper bound for each region. If an occurrence of the optimum can be excluded, the region can be discarded. This is the case if the so far found minimal value or the smallest found upper bound is smaller than the minimum of the lower bound of a region. Otherwise, the region has to be split again.

There are different ways for generating lower and upper bounds. The upper bound for the optimum in the region may be a local minimum [3] or any function value in the region. The lower bound can be calculated by interval arithmetic [1] or several convex relaxation strategies [2,3,5]. Convex relaxation can only be done for optimization problems with special constraints or problems of special types. Most important, "puristic" (i.e., non heuristic) exact methods are not usable with black box models, which means models without any special structure of the underlying DAE system (e.g., pure AEs)[7]. Particularly, general differential equations still pose strong problems.

1.3 Required Characteristics for Function Exploration Algorithms

The new algorithm introduced and tested in this contribution deals with parameter estimation in arbitrary DAE models. It is able to produce more information on the optimum than just its location. An approximated reconstruction of the objective function also gives information on the neighborhood of the optimum. An exploration algorithm should be able to deal with typical ill conditioned and non-convex problems. Moreover, at least some more parameters should be manageable with an acceptable efficiency-to-robustness ratio than for typical exact global optimization algorithms.

Clearly, because no structural information on the model should be used (black box), a bit of the exactness and guaranteed success of global optimization algorithms has to be sacrificed. Thus, the new heuristics for function exploration will not reproduce arbitrary fine details of the original function. Nevertheless, it tries to resolve structural details with a size above some given threshold. For example, if the user does not expect strong fluctuations of the function for "small" parameter changes, this information will be used by the algorithm (cf. Fig. 3 for an example). As a drawback, it will not find deep holes with small diameter in an elsewhere plane region. For this reason, it should be pointed out that the new method cannot perform well on arbitrary test functions but rather on typical biological model functions with poorly determined nonlinear optima but rather well known qualitative influence of the parameters.

Last but not least, a function exploration algorithm should be parallelizable to scale well for cluster computing.

2 Algorithm

2.1 Overview

The algorithm described in the presented work is a hybrid between the exact branch and bound methods described in [2,3,5] and a stochastic approach. It uses the same branch and bound approach as the exact methods, but it introduces a new method for exploration of the regions to decide whether a branch step or a bound step is done. Instead of a convex relaxation, an approximation of the objective function is done by quadratic regression. This is the heuristic step which might miss some fine structures of the function.

By using an approximation for the objective function generated by quadratic regression, the algorithm must only be able to evaluate the model with a given set of parameters and input variables. The objective function is then reconstructed by approximations on the different branch-and-bound regions. By using adaptive branching, the resolution will be quite detailed close to the optimum. Thus, the approximations give extended information about its shape and multi-modality.

2.2 Quadratic Regression and Quality

Quadratic regression is used to approximate an objective function *f* on one every branch-and-bound region generated during the algorithm runtime. The objective function is calculated at several randomly chosen sample points (x_i,y_i) in parameter space. Eq. 1 shows a quadratic regression model in two dimensions:

$$b_i = a_0 + a_1 x_i + a_2 y_i + a_3 x_i^2 + a_4 x_i y_i + a_5 y_i^2 \quad (1)$$

This results in the following linear system matrix, sample vector and parameter vector:

$$A = \begin{bmatrix} 1 & x_1 & y_1 & x_1^2 & x_1 * y_1 & y_1^2 \\ 1 & x_2 & y_2 & x_2^2 & x_2 * y_2 & y_2^2 \\ \dots & & & & & \\ 1 & x_n & y_n & x_n^2 & x_n * y_n & y_n^2 \end{bmatrix} \quad (2)$$

$$\vec{b} = \begin{bmatrix} f(x_1, y_1) \\ f(x_2, y_2) \\ \dots \\ f(x_n, y_n) \end{bmatrix} \qquad \vec{a} = \begin{bmatrix} a_0 \\ a_1 \\ a_2 \\ a_3 \\ a_4 \\ a_5 \end{bmatrix} \quad (3)$$

The regression estimate then is well known to be:

$$\vec{b} = A\vec{a} \qquad \Rightarrow \qquad \hat{\vec{a}} = (A^T A)^{-1} A^T \vec{b} \quad (4)$$

With the estimated parameters of the regression model the approximated values for the points used as samples can be calculated. The quality of the approximation can be defined as sum of squares of the distances between approximated values and samples divided by the number of samples:

$$\vec{d} = A\hat{\vec{a}} - \vec{b} \qquad q_{sum} = \frac{1}{n} * \sum_{i=0}^{n} d_i^2 \quad (5)$$

Another possibility for defining the quality is the maximum relative distance between approximation and samples. To get a relative distance the absolute distance is divided by the distance from the global minimum. For least squares problems the value of the global minimum is known to be zero and a scaling constant α is necessary to avoid division by zero. α is chosen as 0.1. If the value of the global optimum is not known the best value m_{sff} found so far during algorithm runtime is used:

$$q_{max} = max_i\left(\frac{\sqrt{d_i^2}}{\sqrt{(b_i - m_{sff})^2} + \alpha}\right) \quad (6)$$

2.3 Algorithm - Step by Step

1. The Algorithm starts by choosing a region in parameter space from the list of not yet processed regions. In initial state there is only one entry for the whole examined region in this list.
2. The objective function is approximated in this region by quadratic regression with a number of stochastic samples calculated by solving the objective function at randomly chosen points and additionally all corners of the region. The number of samples depends on the number D_{model} of parameters of the objective function and the needed structural details resolution. The number of corners is:

   $$n_{corners} = 2^{D_{model}} \quad (7)$$

   The number of regression parameters can then be calculated by the following equation by using a binomial coefficient:

   $$n_{qreg} = 2 * D_{model} + \frac{D_{model}!}{2! * (D_{model} - 2)!} \quad (8)$$

   For the experiments described in this paper the number of stochastic samples for quadratic regression is chosen as:

   $$n_{samples} = n_{corners} + 2 * n_{qreg} \quad (9)$$

3. The minimum of the samples is compared to the so far found minimal value and chosen as new so far found minimal value if it is smaller.
4. The minimum of the approximation is calculated.
5. If the minimum of the approximation is outside the processed region or if it is a saddle point then the minimum on the border of the processed region is calculated by quadratic programming and used as minimum of the approximation.

6. The objective function is calculated at the minimum of the approximation and the value is compared to the so far found minimal value. If it is smaller, than it is used as new so far found minimal value.

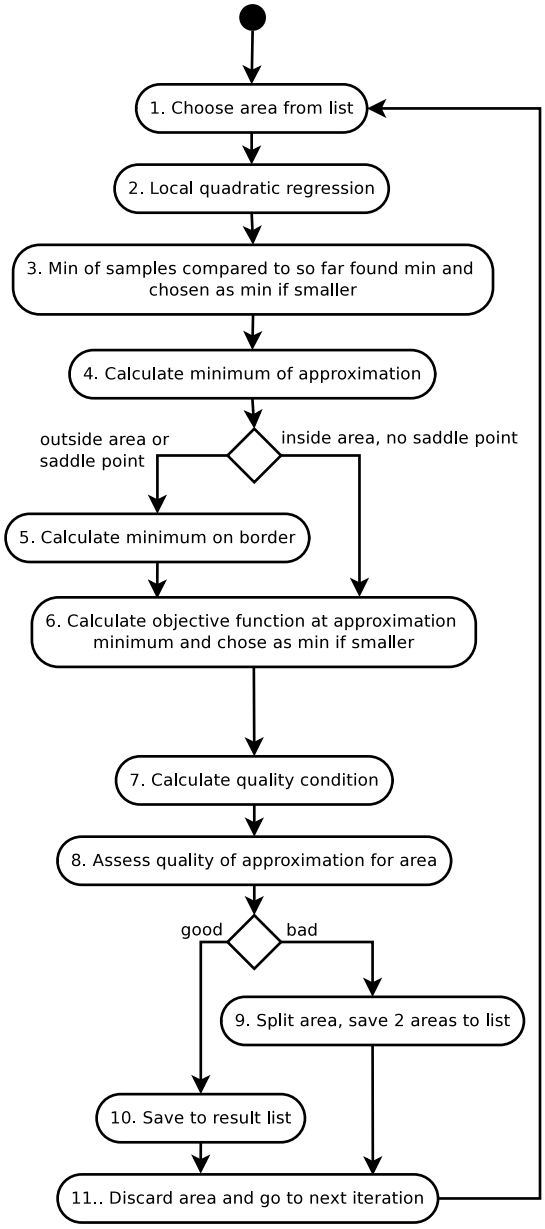

Fig. 2 Algorithm overview

7. The quality of the approximation is determined by finding the maximum relative distance between approximated values and corresponding samples (Eq. 6). Close to already found optima a better quality of the approximation is desired and far away a lesser quality suffice. This is done by dividing each distance between approximated value and corresponding sample by the distance between sample and the best solution found so far (Eq. 6).

8. The resulting quality is compared to a threshold. This threshold is an algorithm parameter and has to be chosen empirically. By this parameter the quality of the approximation and the necessary computational effort can be adjusted.

9. If the quality value is greater than the threshold this means the approximation is not good enough. The region has to be split into two new regions on the longest edge and saved to the list of not yet processed regions. The distances in parameter space from the the minimum of the approximation to the two center points of the new regions are calculated and the region with smaller distance is marked to be processed next. Then the algorithm starts at the beginning with choosing a region from the list.

10. If the quality value is smaller than the threshold, then the approximation is good enough, and the region and approximation are saved to the result list.

11. Afterwards, the region can be discarded because it is finally processed, and the algorithm starts again with choosing the next region from the list until the list is empty.

After exploring all regions from the list of not processed regions, the result is a list of regions in parameter space with corresponding approximations and the local minimal value found in this region. The entries with the smallest minimal values are candidates for the global optimum. The candidate with the smallest approximated optimum may be chosen as global optimum, or the regions with the smallest approximated optima may be searched by a local optimization algorithm to find the exact point. Most important, additional information generated by the algorithm is a piecewise approximation of the function by quadratic functions (Eq. 1) in such a way that structural details (like e.g., banana shape) close to the optimum can be inferred.

3 Examples

3.1 Visualizable Examples

With up to three parameters it is easy to check the correct functioning of the algorithm by standard 2D or 3D visualization tools. For this reason, the algorithm has first been tested with two and three dimensional test cases. In the present work Eq. 10 with the global minimum at (1,1) is used as test case:

$$f = \int (A sin(\omega x) - sin(x))^2 \qquad x = 0..8 * \pi \tag{10}$$

Eq. 10 is a continuous version of a well known least squares problem with many local optima. Fig. 3 shows the visualization of Eq. 10 with A and ω in [0,6] and x=2π. In the first step the approximation is calculated for the whole explored region. Clearly, the global optimum is obtained for A=1, ω=1.

Fig. 4A shows the objective function together with the approximation for the whole explored region. The approximation does not reproduce the details of the objective function and in this case, the approximation

is not good enough and the region has do be split into two. This is the first iteration of the algorithm. After some iterations Fig. 4B shows approximation and objective function in a later step. This approximation does not satisfy the quality criteria, too, and the region has to be split again.

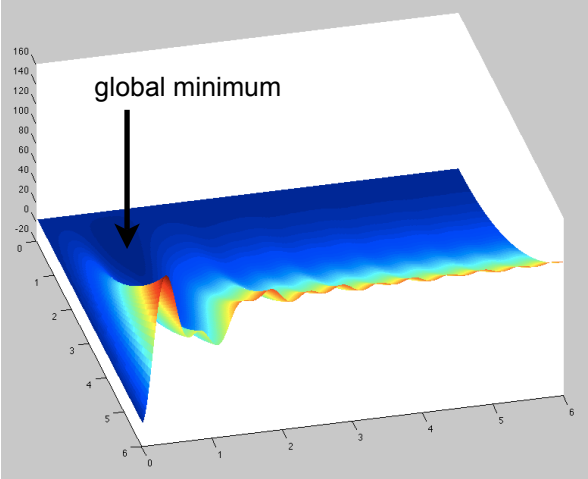

Fig. 3 Diagram of Eq. 10

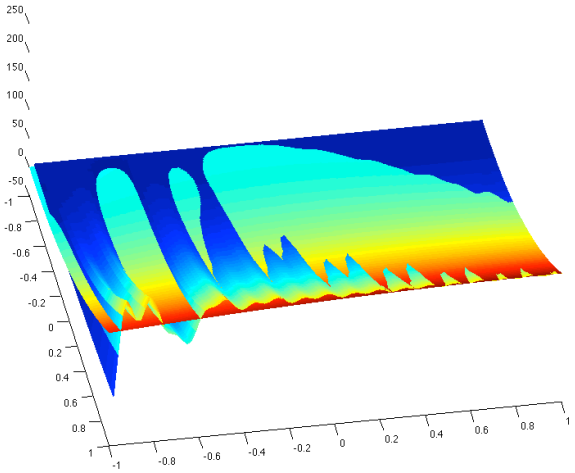

Fig. 4A Approximation in the first step, structural details are not resolved

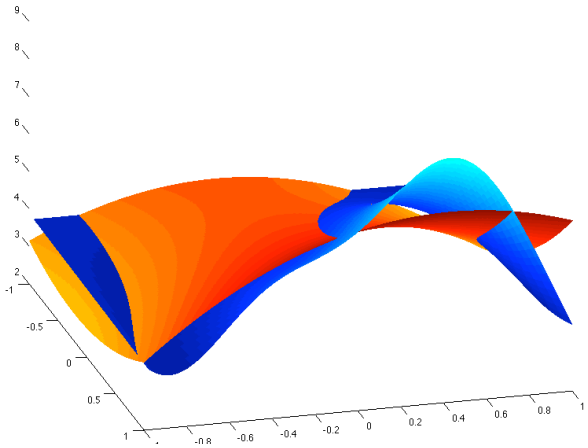

Fig. 4B Approximation in a later step

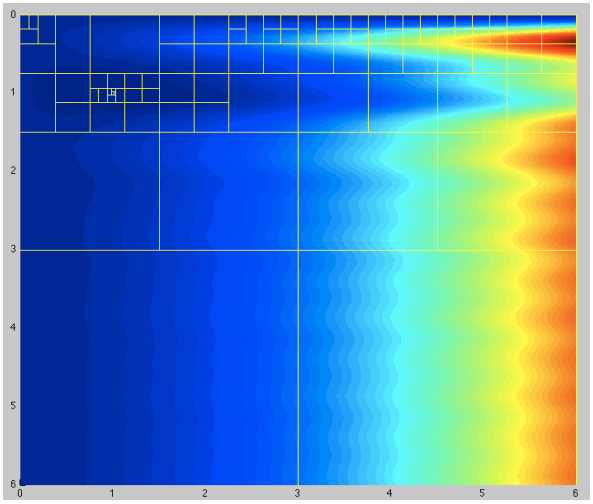

Fig. 5 Subdivided parameter space after algorithm run

After the algorithm is finished, the explored region in parameter space is subdivided into bigger and smaller regions (Fig. 5). The big regions have been discarded without further exploration because the global optimum is not expected to be in these regions. Closer to the optimum the regions become smaller, and the optimum is isolated in one region. By looking for small regions with a minimal value near the global optimum it is possible to describe the shape of the optimum or to find other optima that are almost as good as the global optimum.

Last but not least Fig. 6. is a surface diagram of all the approximations for processed and discarded regions. The objective function is reproduced, regions far away from the minimum lack the fine details and the regions around the minimum are reproduced much better.

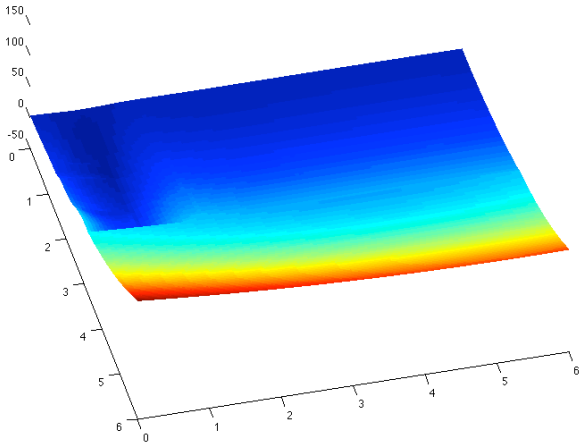

Fig. 6 All single region approximations: Reproduction of the objective function

Fig. 7 is the visualization of an algorithm run with a three dimensional generalization of Rosenbrock's banana function:

$$f = (z - x^2 - y^2)^2 + (x + y + z)^2 + 0.01(x^2 + y^2 + z^2) \quad (11)$$

The optimum at the bottom of the banana with very small function values is surrounded by smaller regions

again. The sphere outside has greater function values, and the discarded regions are bigger.

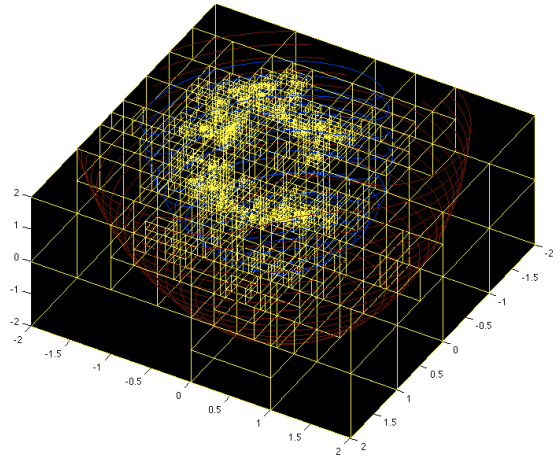

Fig. 7 Three dimensional Rosenbrock function

3.2 ODE System Example

After the algorithm is visually checked and found working a more complex objective function was used for further assessment. The model in Eq. 12 describes a reversible enzyme reaction. (Fig 8):

$$\begin{aligned}
\dot{S} &= -\alpha^+ * S * E + \alpha^- * ES && S_0 = 1 \\
\dot{ES} &= \alpha^+ * S * E - \alpha^- * ES \\
&\quad -\beta^+ * ES + \beta^- * EP && ES_0 = 0 \\
\dot{EP} &= -\gamma^+ * EP + \gamma^- * E * P \\
&\quad -\beta^+ * ES - \beta^- * EP && EP_0 = 0 \\
\dot{P} &= \gamma^+ * EP - \gamma^- * E * P && P_0 = 0 \\
\dot{E} &= -\alpha^+ * S * E + \alpha^- * ES \\
&\quad +\gamma^+ * EP - \gamma^- * E * P && E_0 = 1
\end{aligned} \tag{12}$$

With the two conservation equations Eq. 13 and Eq. 14 the model can be reduced so that the DGLs for $\dot{P}$ and $\dot{E}$ can be removed:

$$P = S_0 - ES - EP - S \tag{13}$$

$$E = E_0 - ES - EP \tag{14}$$

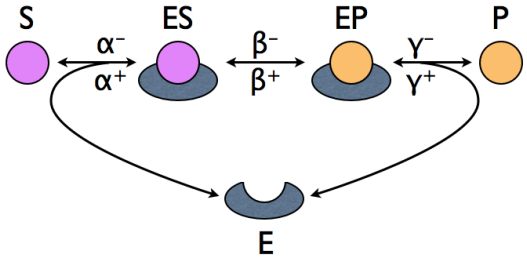

Fig. 8 Single substrate mechanism for enzyme reaction

The reason for choosing this example is because it is well known and ill conditioned.

The typical problem in life science is to find the parameters of the DAE model so that the solved DAE model fits the experimental data in the best way. To test the algorithm, the model is solved with a chosen set of parameters:

$$\begin{aligned}
\alpha^+ &= 2 & \beta^+ &= 10 & \gamma^+ &= 2 \\
\alpha^- &= 1 & \beta^- &= 10 & \gamma^- &= 1
\end{aligned} \tag{15}$$

The quality function is defined as the sum of least squares of the differences between values of the solution with chosen parameters and the solution with parameters set by the algorithm. The quality function is the objective function for the algorithm. The model solution produces concentration values for 121 point in time in the interval [0,12]. Fig. 9 shows the model solved by a standard Matlab ode solver (ode45) for the parameters in Eq. 15.

In addition a linearized regression analysis is done to show the correlations between the estimated parameters and to judge the ill conditioness of the example with the chosen set of parameters. The results of the regression analysis (Fig. 10/11) depends on the chosen parameters. With the parameters from Eq. 15 the two rate constants β^+ and β^- are correlated very strongly as it is obvious by the degenerated ellipse (Fig. 10). The other parameters are not correlated too much because the ellipses are not degenerated. From the linearized confidence regions the very large error bars for the correlated parameters (Fig. 11) can be derived. The parameters are poorly determined.

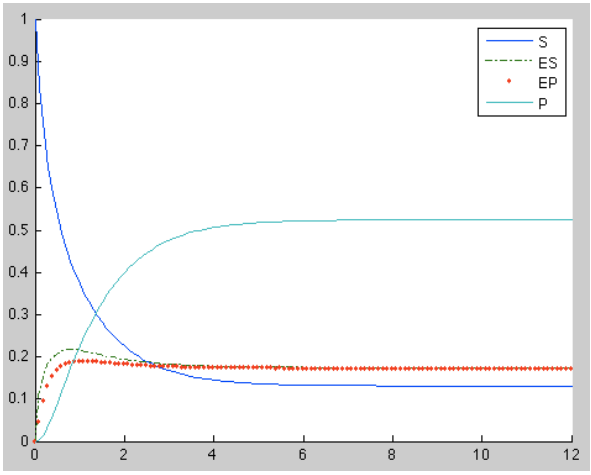

Fig. 9 Concentrations for the four pools S, ES, EP and P against time on the x axis

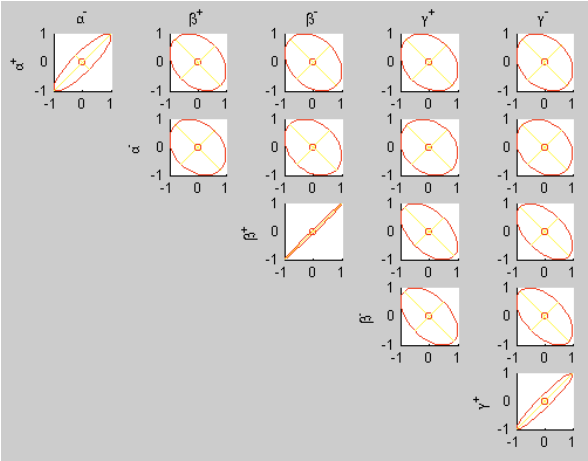

Fig. 10 Correlations between parameters by linearized regression analysis

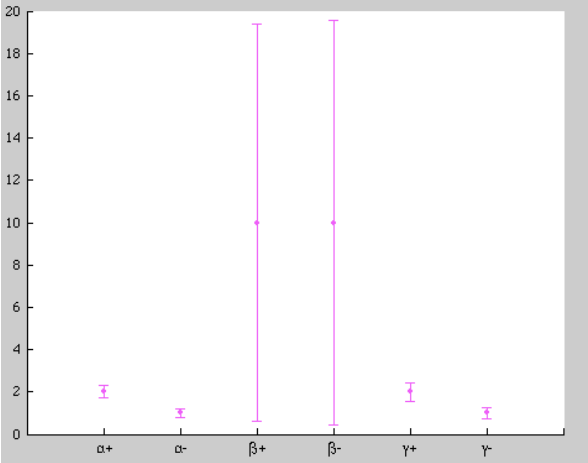

Fig. 11 Parameters with error bars from linearized regression analysis

After these preliminary stages the algorithm has explored the DAE model test case with different thresholds as algorithm parameters. It has produced the lists of regions with corresponding approximations as result. In case of a threshold of 0.3 the list contains 888 regions with corresponding approximations. With a threshold of 0.2 the list already contains 20806 regions. The runtime of the algorithm is not yet comparable to other algorithms because it is implemented in Matlab and thereby not optimized for speed. But the results for this example are produced with a normal quad core workstation computer. The longest algorithm runtime was about 8 hours.

To check the results in higher dimensions, the objective function is calculate at independent and randomly chosen sample points in parameter space. The approximation is calculated at the same points in parameter space by finding the regions containing the sample points and using the corresponding quadratic functions. After this, each value of the objective function is compared with the corresponding value of the approximation. The percentage differences of the two values of the objective function and the approximation are grouped to 60 intervals of the same size and illustrated as histogram (Fig. 12/13). Each bar in the histogram stands for the number of independent sample points with a percentage difference to the result of the objective function in the interval on the x-axis.

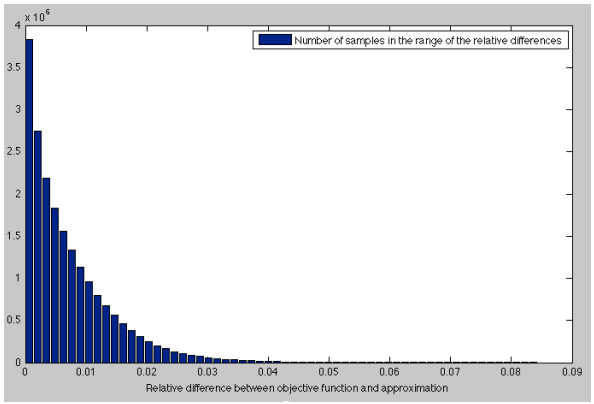

Fig. 12 Histogram for algorithm run with threshold 0.2

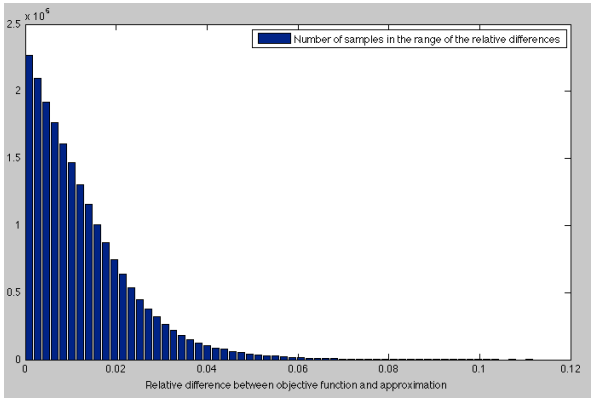

Fig. 13 Histogram for algorithm run with threshold 0.3

A smaller threshold should lead to a more precise approximation of the objective function. In Fig. 12 it is obvious that much more independent samples have a percentage deviation smaller than 0.01 which means 1 percent than in Fig. 13 with a higher threshold. This is what the threshold is expected to do as algorithm parameter.

4 Conclusion and Outlook

At least for dimensions with up to six parameters, the algorithm meets the requirements mentioned above. The algorithm is proofed working for three test cases.

It is a new approach for the bound step to decide the regions that have to be explored further. Contrary to global optimization algorithms, it is possible to gain more information about the objective function than only the location of the optimum. The approximations for each region are gathered to an approximation of the whole objective function with closer fit close to the optimum and loose fit far away from the optimum. The threshold as algorithm parameter can be used to adjust the quality of the approximation.

For the future it is intended to use the algorithm for higher dimensional problems with a practical background in life science. To achieve this, the

implementation of the algorithm has to be improved and and the algorithm has to be extended to fully utilize big cluster computers. The main focus is on an implementation in C++ to improve speed and scalability.

5 References

[1] H. Ratschek, J. Rokne. New Computer Methods for Global Optimization. Kluwer, 2007.

[2] A.B. Singer, J.W. Taylor, P.I. Barton, W.H. Green. Global Dynamic Optimization for Parameter Estimation in Chemical Kinetics. *J. Phys. Chem.* A 2006, 110, 971-6.

[3] A.B. Singer , P.I. Barton. Global Solution of Optimization Problems with Parameter-E,bedded Linear Dynamic Systems. *J. Optim. Theory App.* 2004, 121(3), 613-46.

[4] C.G. Moles, P. Mendes, J.R. Banga. Parameter Estimation in Biochemical Pathways: A Comparison of Global Optimization Methods. *Genome Res.* 2003, 13, 2467-74.

[5] Y. Lin, M.A. Stadtherr. Deterministic Global Optimization for Parameter Estimation of Dynamic Systems. *Ind. Eng. Chem. Res.* 2006, 45, 8438-8448.

[6] I. Papamichail, C.S. Adjiman. Global optimization of dynamic systems. *Computers and Chemical Engineering 28* 2004, 403-415.

[7] P.K. Polisetty, E.O. Voit, E.P. Gatzke. Identification of metabolic system parameters using global optimization methods. *Theoretical Biology and Medical Modeling* 2006, 3:4.

[8] I. Papamichail, C.S. Adjiman. A Rigorous Global Optimization Algorithm for Problems with Ordinary Differential Equations. *Journal of Global Optimization 24* 2002, 1-33.

[9] W. Wiechert, O. Schweissgut, H. Takanaga, W.B. Frommer. Fluxomics: mass spectrometry versus quantitative imaging. *Current Opinion in Plant Biology* 2007, 10:323–330

[10] S.E. Cox, W.E. Hart, R. Haftka, L. Watson. DIRCT Algorithm with Box Penetration for Improved Local Convergence. *AIAA/ISSMO Symposium of Multidisciplinary Analysis and Optimization* 9 2002

[11] R. Horst, H. Tuy. Global Optimization: Deterministic Approaches, 1996

[12] Mathworks. *Using Matlab version R2008a*. The Mathworks Inc., Natick, 1998.