ROBUST SOLVER OF A SYSTEM OF NONLINEAR EQUATIONS

B. Růžek[1]*, P. Kolář*[2] *and M.Kvasnička*[3]

[1,2] Institute of Geophysics ASCR, [3] Nuclear Research Institute Řež, plc.

[1] b.ruzek@ig.cas.cz, [2] kolar@ig.cas.cz, [3] kva@ujv.cz

1.Motivation and basic ideas

The most frequent form of modelling physical events is based on parametric approach. The properties of the system are characterized by a vector of parameters $\boldsymbol{p} = (p_1,p_2,...p_m)$, the response of the system is characterized by a data vector $\boldsymbol{d} = (d_1,d_2,...,d_n)$. The vectors $\boldsymbol{p}$ fill up the vector space $\mathcal{P}^n$ with the dimension m, the vectors $\boldsymbol{d}$ fill up the vector space $\mathcal{D}^n$ with the dimension n and both vector spaces are related by forward mapping

$$\boldsymbol{d}=F(\boldsymbol{p}) \quad . \tag{1}$$

Eq. (1) represents a system of n non-linear equations for m variables in fact. The goal of inverse problems is to find (all) solutions $\boldsymbol{p}_0$ satisfying measured data $\boldsymbol{d}_0$:

$$\boldsymbol{p_0} : \boldsymbol{d_0}=F(\boldsymbol{p_0}) \quad . \tag{2}$$

Basically two approaches are applicable for the solution to (2): (i) inversion and (ii) optimization. Both methods should output the same result, but great differences may occur as regards the efficiency and robustness in favour either of the first or second method. Anyway the solution to (2) using both inversion and optimization is difficult job. Following features characterize inverse and optimization problems commonly solved in geophysics and in many other branches of physics and engineering:

- the mapping F is complex, the inverse mapping is not directly available or it may not exist at all
- the dimensions of $\mathcal{P}$ and $\mathcal{D}$ spaces are big ($m,n >> 1$)
- the problem is multivalued
- evaluating of the forward problem is time consuming

2.Inversion versus optimization

Let us consider the mapping $G(\boldsymbol{d})$ is inverse to the mapping $F(\boldsymbol{p})$ ($G = F^{-1}$). In cases that the inverse mapping does not exist in entire spaces $\mathcal{P}$ a $\mathcal{D}$, we can restrict ourselves to sufficiently small joint subspaces $p\mathcal{P}^m$ and $d\mathcal{D}^n$ respectively, inside which the mapping F is such smooth that the inverse to F does exist. Than the inverse problem to (1) can be formulated as follows

$$\boldsymbol{p_0}=F^{-1}(\boldsymbol{d_0})=G(\boldsymbol{d_0}) \quad . \tag{3}$$

In favourable situations Eq. (3) can be used immediately (e.g. if both G and F are linear). Unfortunately we do not know the exact form of G in general cases, and then Eq.(3) is of symbolic importance only and other specialized procedures are necessary. Inversion means mapping of particular data vector to particular model vector.

An alternate approach to direct inversion aiming at finding $\boldsymbol{p}_0$ according (3) is optimization - indirect method based on minimization of the misfit functional with respect to the vector $\boldsymbol{p}$:

$$min(Norm(\boldsymbol{d_0}-F(\boldsymbol{p})))=min(\boldsymbol{p}), \boldsymbol{p} \Rightarrow \boldsymbol{p_0} \tag{4}$$

where the most popular form of the norm is the standard L2 norm

$$\begin{aligned} Norm(\boldsymbol{d_0}-F(\boldsymbol{p})) &= \Delta \boldsymbol{d}^T C_d^{-1} \Delta \boldsymbol{d} \\ \Delta \boldsymbol{d} &= \boldsymbol{d_0}-F(\boldsymbol{p}) \end{aligned} \quad , \tag{5}$$

where C_d is the data covariance matrix. Great number of methods are known for finding the minimum of (4) (for an overview see e.g. *www3, Press et al. 2007*, etc.). Optimization includes the elementary mapping of the parameter vector to the norm, i.e. mapping the m-dimensional space to one-dimensional space. This mapping is not invertible (it is impossible to specify many parameters from a single norm), and definitely some loss of information takes place. All manipulations inside optimization are much simpler on the other hand, since it is not necessary to deal with great dimensions of the data space like inside inversions. Basic problem for optimization is not the existence or non-existence of the mapping F and its specifics, but the particular choice of the method used for searching in the space $\mathcal{P}$.

Important differences between inversion and optimization are summarized in the following Table 1.

Table 1

	Inversion	Optimization
basis of the method	mapping	searching
norm of the fit	L2 or not defined	arbitrary according the definition
dimensions of working spaces	*m,n*	*m,1*
multimodality[*]	not allowed	allowed

[*] multimodality = many formally equivalent solutions are possible

3.ANNIT Algorithm

The *ANNIT*[**] algorithm is an inverse (not optimizing) algorithm, utilizing numerical approximation of (3) in empirically constrained subspaces $\{p\mathcal{P}^m, d\mathcal{D}^n\}$. The existence of inverse mapping inside these subspaces is supposed, and thus the existence of its numerical approximation is also supposed. This numerical approximation is used for the prediction of the solution. Since this method is approximate, the algorithm is arranged into iterative cycles and the solution is gained successively. The appropriate block-diagram can be found in the Fig.1.

ANNIT is working simultaneously with a population of q models (individuals). Each model M is constituted from a parameter vector, data vector and the model error:

$$M_i=\{\boldsymbol{p}_i, \boldsymbol{d}_i, err_i\}, i=1,2,3,\ldots q \quad . \qquad (6)$$

The model error err_i can be in the form of Eq. (5), but any other form with the same qualitative content is also acceptable. The absolute magnitude of the error is not important, since *ANNIT* is using this quantity only for relative classification of distinct models within the population and for their sorting from the best to the worst model.

Such computer implementation of the algorithm is considered, which records already evaluated and tested models when these models can be used later. Repeated usage of some models generates the possibility of efficient inversion with minimum number of forward evaluations.

[**] *ANNIT* = **A**rtificial **N**eural **N**etwork **I**nversion **T**ool. *ANNIT* is a successor of previously developed optimizing algorithm *ANNO* (**A**rtificial **N**eural **N**etwork **O**ptimization). It is appearing that many applications are solved more efficiently by using inversion compared to optimization. Usually real measurements give right hand sides of non-linear equations in a natural way thus evoking to directly solve a set of non-linear equations. The first *ANNIT* version has been implemented with the only one predictor built from the network of radial basis functions, which belong to the class of Artificial Neural Network (*ANN*). Later on the range of predictors has been broadened even outside *ANN*. Therefore the *ANN* technology is nor unavoidable in *ANNIT* or a dominant component. From historical reasons the reference to *ANN* is retained in the name of the algorithm.

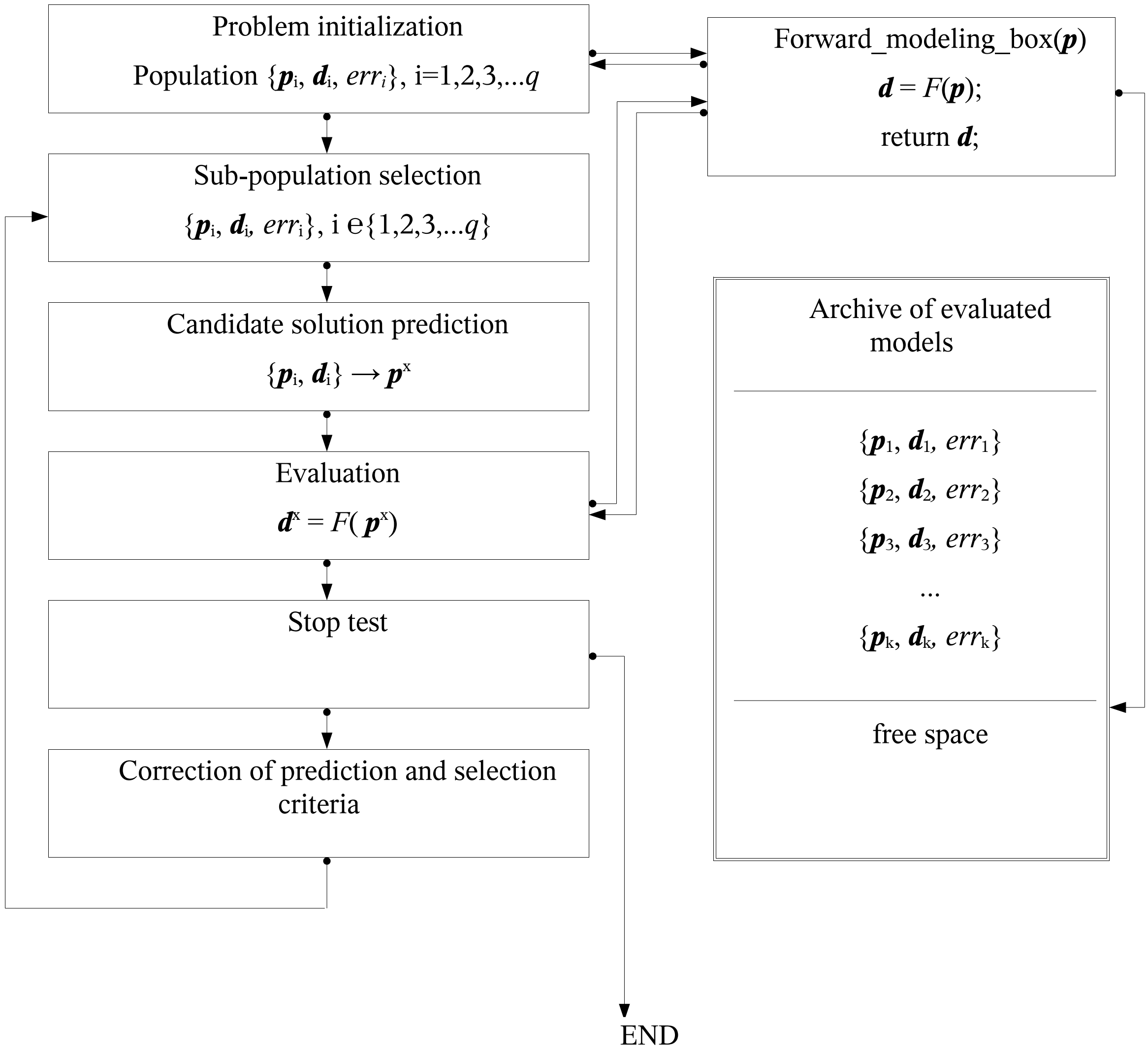

Fig.1. Block diagram of the ANNIT algorithm.

3.1.Initialization

All necessary control parameters are read, all necessary variables are initiated and all necessary arrays are allocated in the initialization part. Next computations will be limited to parameters lying inside the hypercube defined by the user:

$$p_i^{min} \le p_i \le p_i^{max}, i=1,2,3,...m \quad . \tag{7}$$

If the true solution is outside such hypercube, it will be never discovered. Starting population of models is generated in the allowed hypercube with an uniform probability. *ANNIT* does not allocate any specific model into the starting population, even if such possibility is easy applicable. More general approach without specifying starting model is preferred instead. Searching for the solution is fully in the competency of the algorithm itself and it follows naturally principles of the inverse process. The size of the starting population *q* is not very important (it will change during iterations), but suitable choice can be $5m \le q \le 10m$. Each model evaluated during the initialization stage is saved in the archive, so the archive contains *q* models in the beginning. Current population of models is sorted according individual errors of the models and candidate solution (i.e. the model with the least error) is called M^B:

$$M^B=\{\boldsymbol{p}^{\boldsymbol{B}}, \boldsymbol{d}^{\boldsymbol{B}}, err^B\}, err^B=min(err_i), i=1,2,3,...q. \tag{8}$$

The diameter of the population *R* a the index of the prediction function *ip* are important control parameters. The diameter of the population defines the size of a subregion, inside which next population of models will be generated, and index of the prediction function specifies prediction method used for predicting the candidate solution. Both parameters are tuned during the inversion, in the beginning they are set to $R = 1$, $ip = 1$. The way how these parameters are working will be explained later.

3.2.Geometry of the population

Individual populations of models generated in distinct iteration cycles follow the unique geometric criterion: one model is selected and declared as a centre of the population, and other q - 1 models are located randomly in the distance R measured from the centre of the population (with the only one exception in the beginning, when the initial population covers the parametric space $\mathcal{P}^m$ with uniform probability). It is therefore important to properly define all necessary geometric characteristics. Geometric relations are regarded only in the parametric space $\mathcal{P}^m$. The geometry in the data space $\mathcal{D}^n$ (it is different from the geometry in $\mathcal{P}^m$) is not regarded, since its adjustment is not immediately possible (we cannot freely adjust vectors $\boldsymbol{d}$, to which corresponding model patterns $\boldsymbol{p}$ are not known).

Metric tensor C^m is introduces in parametric space $\mathcal{P}^m$ as follows

$$\boldsymbol{C^m} = \begin{bmatrix} \Delta p_1^2 & 0 & \dots & 0 \\ 0 & \Delta p_2^2 & \dots & 0 \\ \dots & \dots & \dots & \dots \\ 0 & 0 & 0 & \Delta p_m^2 \end{bmatrix}, \Delta p_i = p_i^{max} - p_i^{min} \quad , \tag{9}$$

and distances s in $\mathcal{P}^m$ are evaluated according the formula

$$s^2 = |\boldsymbol{p}_1 - \boldsymbol{p}_2|^2 = |\Delta \boldsymbol{p}|^2 = \Delta \boldsymbol{p}.(C^m)^{-1} \Delta \boldsymbol{p}^{\boldsymbol{T}} \tag{10}$$

Distance s from Eq. (10) is a dimensionless quantity, so all possible differences in physical quantities represented in individual dimensions of the vector $\boldsymbol{p}$ are cleared. Distances depend on the size of the allowed parametric hypercube on the other hand. The length of each edge of the parametric hypercube is $s = 1$, and the length of its body diagonal is depending on the dimension of the parametric space $s = m^{1/2}$.

Any time the centre of the population $\boldsymbol{p}^C$ is defined relatively to the parameter space

$$\boldsymbol{p}^{\boldsymbol{C}} = \langle \boldsymbol{p}_{\boldsymbol{i}} \rangle, i = 1,2,3,\dots q \tag{11}$$

and the diameter of the population R as the mean distance between satellite models and the centre of the population

$$R = \langle |\boldsymbol{p}_{\boldsymbol{i}} - \boldsymbol{p}^{\boldsymbol{C}}| \rangle \quad . \tag{12}$$

It could be shown, that the hypersphere with the diameter $R = ½$ and with the centre $\boldsymbol{p}^C$ exactly in the middle of the parametric hypercube is a body escribed around this hypercube.

Generating a new population is controlled by the location of the centre $\boldsymbol{p}^C$ and the diameter R. Every newly generated satellite model has random position along the surface of a hypersphere $H := \{\boldsymbol{p}^C, R\}$. As a rule, the centre of the population $\boldsymbol{p}^C$ is set up to the so far best model $\boldsymbol{p}^C = \boldsymbol{p}^B$, see (8).

Next important geometric quality is the mean distance r_d between neighbouring satellite models (i.e. between models of the population except the centre). This distance depends on the diameter R, number of models constituting the population q and also on the dimension of the model space m. The appropriate formula sounds (see Appendix A.1)

$$r_d = 2.R.\left(\frac{m-1}{q}\right)^{\frac{1}{m-1}}.\pi^{\frac{1}{2}}.\frac{\Gamma\left(\frac{m-1}{2}\right)}{\Gamma\left(\frac{m}{2}\right)} \quad , \tag{13a}$$

and for m >> 1 quite simpler estimate can be used

$$r_d \approx 4 R \left(\frac{m}{q}\right)^{\frac{1}{m}} \quad . \tag{13b}$$

3.3.Predicting population

The prediction of the candidate solution (see the following text) can be done only using suitably configured population of models. The models of this prediction population should surround the searched solution in an ideal case (in order to work in interpolation regime and not in an extrapolation one), and the geometric size of the population should be small (in order to ensure the existence of local approximation (3) and maybe this approximation be close to linear mapping). The prediction population is therefore generated in such a way, that the centre is located close to the expected solution, and satellite models are located randomly along the surface of a hypersphere. The upper limit of the diameter of the hypersphere is estimated such that consecutively predicted candidate solution should lay still inside the hypersphere. Let the centre and diameter of the constituted population is defined $\{\boldsymbol{p}^C, R\}$. In the first cycle $\boldsymbol{p}^C$ is selected randomly and $R = 1$. How these control parameters are tuned in following iterations is explained in the part “Prediction corrector” later in this text. Provided the parameters $\{\boldsymbol{p}^C, R\}$ are known, new models of the predicting population are gained according the following steps.

i. The centre of the population $\{\boldsymbol{p}^C, \boldsymbol{d}^C, err^C\}$ is connected to the population as the first model.

ii. Metric tensor $\boldsymbol{C}^m$ is decomposed using Choleski decomposition $\boldsymbol{C}^m = \boldsymbol{L}.\boldsymbol{L}^T$.

iii. Following steps iv. - ix. are made q-1 times in order to get the population of the total size q.

iv. Random m-dimensional unit vector $\boldsymbol{g} = (g_1,g_2,...g_m)$, $\sum(g_i)^2 = 1$ is generated.

v. Candidate model $\boldsymbol{p}^g = \boldsymbol{p}^C + R.\boldsymbol{L}.\boldsymbol{g}$ is proposed.

vi. In case the candidate model $\boldsymbol{p}^g$ is outside the parametric hypercube, it is projected along the direction $(\boldsymbol{p}^g - \boldsymbol{p}^C)$ to the closest face of the parametric hypercube.

vii. The archive of already evaluated models is checked and the k-th model $\{\boldsymbol{p}_k, \boldsymbol{d}_k, err_k\}$ is selected. This k-th model is closest archive model to $\boldsymbol{p}^g$ and still not connected to the predicting population. The distance $s^g =|\boldsymbol{p}^g - \boldsymbol{p}^k|^2$ is computed according (10).

viii. If $s^g < r_d$, instead of $\boldsymbol{p}^g$ the archive model $\{\boldsymbol{p}_k, \boldsymbol{d}_k, err_k\}$ is connected to the population.

ix. If $s^g \geq r_d$, the model $\boldsymbol{p}^g$ is evaluated $\boldsymbol{d}^g = F(\boldsymbol{p}^g)$, model error err_g is computed and the completed model $\{\boldsymbol{p}_g, \boldsymbol{d}_g, err_g\}$ is both connected to the predicting population and copied to the archive for future usage.

Models generated this way surround the centre $\boldsymbol{p}^C$ in the distance R. Some models of the constituted prediction population may be close to some model already evaluated in the past. Probability of such event is increasing with growing size of the archive. In such case the proposed model can be substituted by archive model and it is no more necessary to evaluate forward problem. The efficiency of this substitution approach is monitored during inverse process. In other cases the evaluation of the forward problem is necessary.

3.4.Prediction of the candidate solution

Suitable population of models $\{\boldsymbol{p}_i, \boldsymbol{d}_i, err_i\}$, i=1,2,3,...$q$ is available in the prediction stage. The goal is now to use information in the population for predicting the solution (3). Prediction module can be arbitrary modified provided formal requirements put on input and output are satisfied:

$$\text{population of models}\{\boldsymbol{p_i}, \boldsymbol{d_i}, err^i\} \rightarrow [\![\text{prediction algorithm}]\!] \rightarrow \text{candidate solution } \boldsymbol{p_0} \tag{14}$$

Only the algorithm of Radial Basis Function Network (RBFN) was implemented in the original version of *ANNIT*. Following experiments have shown, that different prediction algorithm behave more or less occasionally in diverse environments. The best solution is therefore to use different prediction algorithm even inside each individual inverse problem. Current version of *ANNIT* is using three implemented prediction algorithms, and their selective efficiency is continuously monitored:

i. linear regression

ii. prediction by using RBFN

iii. linear prediction (also known as “Kriging”)

The *ANNIT* algorithm can be freely supplemented by other prediction methods in the future.

3.4.1.Prediction by using linear regression

This method is the least square method in fact. Linear problem is expected to be solved:

$$F(\boldsymbol{p})=\boldsymbol{A}\,\boldsymbol{p}+\boldsymbol{d}_c=\boldsymbol{d} \tag{15}$$

where ***A*** is a *n* x *m* matrix and $\boldsymbol{d}_c$ is vector with dimension *n*. Predicted solution is evidently

$$\boldsymbol{p}_0=pinv(\boldsymbol{A}).(\boldsymbol{d}_0-\boldsymbol{d}_c) \quad . \tag{16}$$

The function *pinv*(***A***) is pseudoinverse of matrix ***A***. It is necessary to know all elements both of ***A*** and $\boldsymbol{d}_c$ for evaluating (16). Appropriate computations of these elements is based again on linear algebra and for exact explanation see the Appendix A2. Incorporation of single linear regression into a priori non-linear difficult to solve problems is advantageous for several reasons:

- some configuration of models in a population can well approximate the solved problem even by using simple linear functions and resulting prediction can be good candidate solution thanks to fortune,
- the solved non-linear problem should be closer to linear one with advancing convergence if the true mapping *F* is "reasonably smooth" and linear regression has real chances for success in the final phase of the inversion,
- even essentially non-linear problem can behave like effectively linear one thanks to special configuration of measured data.

If some of the above supposition is true, using linear regression is extremely efficient. Strictly linear problem is solved exactly in one step. If the suppositions are false, no extra degradation of the efficiency takes place, since the internal overhead needed for linear regression is relatively small.

3.4.2.Prediction by using Radial Basis Function Network

Radial Basis Function Network (RBFN) is used in the *ANNIT* algorithm for local approximation of the inverse mapping (3). This is much more general approach compared to linear regression from the preceding text. Using RBFN flexible modelling of different configurations of parameters and data is possible, linearity is not a requirement. Standard form of RBFN, which theory is documented e.g. in v *Press et al.2007* or *Orr 1996* is used.

Individuals of the predicting population {$\boldsymbol{p}_i$, $\boldsymbol{d}_i$, err_i} are selected as the centres of radial base functions. A system of *q* base functions $h_i(\boldsymbol{d}_0)$ is then introduced

$$\begin{gathered} h_i(\boldsymbol{d}_0)=\frac{1}{\sqrt{(1+r_i^2)}} \\ r_i^2=(\boldsymbol{d}_0-\boldsymbol{d}_i)^T(\boldsymbol{C}^D)^{-1}(\boldsymbol{d}_0-\boldsymbol{d}_i) \end{gathered} \tag{17}$$

where $\boldsymbol{d}_0$ is "arbitrary" vector in data space (in fact the vector $\boldsymbol{d}_0$ can not be selected fully arbitrary, since it must be close enough to the centres of radial basis functions in order to be in their effective range), $\boldsymbol{d}_i$ is data vector of the *i*-th individual, $\boldsymbol{C}^D$ is data covariance matrix and r_i is the distance between $\boldsymbol{d}_0$ and $\boldsymbol{d}_i$. The function h_i in (16) is so called "inverse multiquadric function", but plenty of other function can be used as well (see Appendix A3). Particular choice of radial function has no significant influence on the inversion. Prediction of the parameter vector $\boldsymbol{p}_0$ corresponding to data vector $\boldsymbol{d}_0$ is made according the following scheme

$$\begin{gathered} \boldsymbol{p}_0=\boldsymbol{Hw} \\ \boldsymbol{w}=\begin{bmatrix} h_1(\boldsymbol{d}_0) \\ h_2(\boldsymbol{d}_0) \\ \dots \\ h_q(\boldsymbol{d}_0) \end{bmatrix} \end{gathered} \tag{18}$$

where ***w*** is weighting vector (*i*-th component gives the value of basis function which argument is the distance between $\boldsymbol{d}_0$ and the centre of *i*-th basis function), and ***H*** is variance matrix determined from the requirement that (18) is valid for all centres of basis functions as well. The way how the elements of ***H*** are evaluated is given in Appendix A4 and further detailed discussion can be found e.g. in *Orr 1996*.

Even though the RBFN method is relatively general interpolation tool, it has some restrictions limiting the applicability range. Fundamental problems can arise if the mapping (1) is not unique and thus the inverse mapping (3) does not exist. Such situation can not be easily and reliably detected from finite number of individuals constituting the prediction population. Even then the non-uniqueness is known in advance it is not possible in general terms to specify subspace where the inverse mapping is correct. *ANNIT* solves this problem by heuristic (empirical) changes of the size and location of the predicting population as described later.

3.4.3.Prediction by using linear prediction algorithm - "Kriging"

Kriging method (also known as "linear prediction method") exists in several slightly different variants. ANNIT uses standard formulation without considering errors, see e.g. *Press et al.2007*, or *www2*. Basic quantity, which is indispensable for realization of linear prediction, is the variance of parameter vector's coordinates as a function of distance measured in data space:

$$\begin{gathered}\boldsymbol{v}(r)=\langle[\boldsymbol{p_i}-\boldsymbol{p_j}]\otimes[\boldsymbol{p_i}-\boldsymbol{p_j}]\rangle \\ r=|\boldsymbol{d_i}-\boldsymbol{d_j}|\end{gathered} \quad . \tag{19}$$

The smooth function $\boldsymbol{v}(r)$ of smooth parameter r (so called variogram) is determined from population's individuals $\{\boldsymbol{p}_i, \boldsymbol{d}_i, err_i\}$. Simple approximation is considered according usual practice

$$\boldsymbol{v}(r)\approx\boldsymbol{\alpha}r^{1.5} \tag{20}$$

where m-dimensional vector $\boldsymbol{\alpha}$ is sufficient to determine (for each dimension of parameter space separately). It is possible to compute the distance in data space between any pair of individuals and also the variance $\boldsymbol{v}(r)$ according (19). Predicted vector $\boldsymbol{p}_0$ is given as a linear combination of all parameters in the population, for l-th component of the vector $\boldsymbol{p}_0$ holds:

$$\begin{gathered}p_{01}=\boldsymbol{v_0^T}\boldsymbol{V}^{-1}\boldsymbol{p_l} \\ \boldsymbol{p_l}=\begin{bmatrix}p_{11} & p_{21} & \cdots & p_{ql} & 0\end{bmatrix}^T\end{gathered} \quad . \tag{21}$$

Elements of the vector $\boldsymbol{p}_l$ are constituted from l-th components of all parameter vectors in the predicting population. The matrix ***V*** is variance matrix (different for each dimension of the parameter spacer) and corresponding to the geometry of the predicting population as seen from the data space:

$$\boldsymbol{V}=\begin{bmatrix} v_{11} & v_{12} & v_{13} & \cdots & v_{1\mathrm{q}} & 1 \\ v_{21} & v_{22} & v_{23} & \cdots & v_{2\mathrm{q}} & 1 \\ & & \cdots & & & 1 \\ v_{q1} & v_{q2} & v_{q3} & \cdots & v_{qq} & 1 \\ 1 & 1 & 1 & \cdots & 1 & 0 \end{bmatrix} \tag{22}$$

where $v_{ij} = v(|\boldsymbol{d}_i - \boldsymbol{d}_j|)$. Analogously the vector $\boldsymbol{v}_0$ is variance vector corresponding to the geometry of predicted vector as regards predicting population:

$$\boldsymbol{v_0^T}=\begin{bmatrix}v_{01} & v_{02} & \cdots & v_{0\mathrm{q}} & 0\end{bmatrix} \tag{23}$$

and $v_{0i} = v(|\boldsymbol{d}_0 - \boldsymbol{d}_j|)$, $\boldsymbol{d}_0$ is measured data.

All prediction methods have similar mathematical form. Also the results are similar if interpolation regime takes place. Totally different results are obtained in extrapolation regime and asymptotic behavior of distinct methods is also totally different. Simple one-dimensional example is presented in Appendix A4.

3.5.Prediction corrector

The quality of prediction can be controlled in principle due to the following adjustable items:

1. Prediction method currently used
2. Configuration of the predicting population, i.e.

2.1. Number of individuals constituting the population q

2.2. Geometric size of the population, i.e. the diameter R

2.3. Localization of the population, i.e. localization of its central individual $\boldsymbol{p}_c$

The measure how the error of the best individual min(err^B) is decreasing is intended under the prediction quality.

Add.1. *ANNIT* has implemented three prediction algorithms (linear regression, RBFN, linear prediction). These algorithms are regularly used in a cyclic manner according the variable *ip*:

Table 2

Iteration Nr.	*ip*	Prediction method
1	1	Linear regression
2	2	RBFN
3	3	Kriging
4	1	Linear regression
5	2	RBFN
6	3	Kriging
...	...	...

It has been shown empirically that using different prediction methods is advantageous. If complex problem is solved, probability of misbehaving of some method is growing with time. If this method is used exclusively, all the computations can crash. When solving simpler problem it does not matter which method is used on the other hand, since the efficiency of different prediction methods is comparable (the only exception is pure linear problem, which is solved by using linear regression exactly).

Simple cyclic changing of prediction methods can be replaced by more sophisticated irregular calling of selected methods. The frequency of calls can be proportional to the efficiency of individual prediction methods (the variable *ip* is set up based on statistical criteria). Appropriate testing did not show any significant improvement of this rather cumbersome approach.

Add.2. Let us recapitulate first the prediction process according the Fig.2. So far the best individual $\{\boldsymbol{p}_B, \boldsymbol{d}_B, err_B\}$ drawn in red is selected as the center of the predicting population. Satellite individuals are located in random positions around the center in the distance R from the center. The geometry in data space is generally not conserved: satellite individuals are no more on the surface of a hypersphere and $\boldsymbol{d}_B$ need not to be the center of the population. Prediction $\boldsymbol{p}_x$ is made for the measured data $\boldsymbol{d}_0$ (in violet). The position of $\boldsymbol{p}_x$ with respect to the population can be characterized by the distance from the center $R' = |\boldsymbol{p}_x - \boldsymbol{p}_c|$. If $R' \leq R$, interpolation regime is indicated (generally favorable situation), if $R' > R$, extrapolation regime is indicated (generally unfavorable situation). Since the procedure is approximate only, instead of data vector $\boldsymbol{d}_0$ other vector $\boldsymbol{d}_X$ correspond to the predicted parameter vector $\boldsymbol{p}_X$. The difference $\boldsymbol{d}_0 - \boldsymbol{d}_X$ characterizes the prediction error err_X.

Thus it is monitored in each iteration step, if “interpolation prediction” ($R' < R$), or “extrapolation prediction” ($R' \geq R$) has been made– see the first column of the Table 3. Other monitored quality is whether the predicted model becomes the currently best one ($err_X < err_B$) or not ($err_X \geq err_B$) – see the second column of the Table 3. The number of consecutive iterations during which no improvement has been reached (variable *nwait*) is also monitored. Based on these information 6 stages 1-6 are distinguished, which are further in two cases subdivided into two sub-stages 3a/b and 6a/b, resp. The parameters R a $\boldsymbol{p}_C$ responsible for generating the next population are modified in each iteration using following rules.

Case 1: Convergence indicated

This is the most promising situation from the point of view of the further advance. The true solution is expected to be somewhere inside the predicting population and it is believed that more precise result will be obtained by using smaller population surrounding the currently best model.

Case 4: Population movement indicated

The simplest explanation for this case is that the prediction population is located eccentrically the expected solution. Adequate response is therefore simple movement of the population like a rigid body, and its size will be conserved in the next iteration.

Cases 2 and 5: Stagnation indicated

Since no improvement occurred in these stages, suppositions for the correct functionality of the algorithms are probably not met. Simple changing the particular configuration of the predicting population using the same generating rules may help in less significant situations, so this stage is tolerated for a limited number of consecutive iterations. The allowed number of stagnation cycles is scaled according the model space dimension ($nwait \leq m$). Individuals are generated in next cycles randomly around the fixed centre and in the same distance from the center.

Cases 3 and 6: Reactivation of inverse process is necessary

If the stagnation regime persists too long ($nwait = m$), deeper intervention into the inverse process is necessary. This is achieved by increasing the geometric size of the population to maximum ($R = 1$) and also movement of the population center $\boldsymbol{p}_C$ to random position of the model space (with 50 % probability). The counter of stagnation cycles is cleared ($nwait = 0$).

Add.3. Since the efficiency of *ANNIT* can dramatically change depending on the size q (= number of individuals) of the predicting population, it is convenient to reflect this fact. No exact formula for optimum q is known due to the diversity of solved problems. *ANNIT* solves this dilemma in such a way, that individual value of q is set up randomly in each iteration cycle with a uniform probability in the range $m+1 < q < 10m$. The lower limit follows from the requirement for unique solution to the set of equations (A11 – A17). The upper limit corresponds to reasonable size of solved relations regarding the dimensions of the problem.

Table 3

R'	err_X	Adaptation of $\boldsymbol{p}_C$	Adaptation of R	Case
$R' < R$ Predicted model is inside the volume spanned by predicting individuals	$err_X < err_B$ Predicted model is the best one from now.	Predicted model set as the centre of the next population: $\boldsymbol{p}_C \longrightarrow \boldsymbol{p}_X$	Decrease the size of the next population, e.g. $R \rightarrow R/2$.	1
	$err_X \geq err_B$ Predicted model is not the best one.	Do nothing if this occurred no more than preselected number of cycles, e.g. $nwait{+}{+} < m$ and repeat with iterations.		2
		If this occurred exactly m-times, $nwait{+}{+} = m$: ■ $nwait = 0$; ■ Set the center of the next population with probability 50% to the so far best model: $\boldsymbol{p}_C \longrightarrow \boldsymbol{p}_B$ ■ otherwise select $\boldsymbol{p}_C$ randomly inside the allowed hypercube: $\boldsymbol{p}_C \longrightarrow \mathrm{rand}(\boldsymbol{p}_{\mathrm{min}}\ \mathrm{x}\boldsymbol{p}_{\mathrm{max}})$	Set the size of the next population to the full size, i.e. $R = 1$;	3a,b
$R' \geq R$ Predicted model falls outside the predicting population.	$err_X < err_B$ Predicted model is the best one from now.	Predicted model set as the center of the next prediction population: $\boldsymbol{p}_C \longrightarrow \boldsymbol{p}_X$	R unchanged	4
	$err_X \geq err_B$ Predicted model is not the best one.	Do nothing if this occurred no more than preselected number of cycles, e.g. $nwait{+}{+} < m$ and repeat with iterations.		5
		If this occurred exactly m-times, $nwait{+}{+} = m$: ■ $nwait = 0$; ■ Set the center of the next population with probability 50% to the so far best model: $\boldsymbol{p}_C \longrightarrow \boldsymbol{p}_B$ ■ otherwise select $\boldsymbol{p}_C$ randomly inside the allowed hypercube: $\boldsymbol{p}_C \longrightarrow \mathrm{rand}(\boldsymbol{p}_{\mathrm{min}}\ \mathrm{x}\ \boldsymbol{p}_{\mathrm{max}})$	R unchanged	6a,b

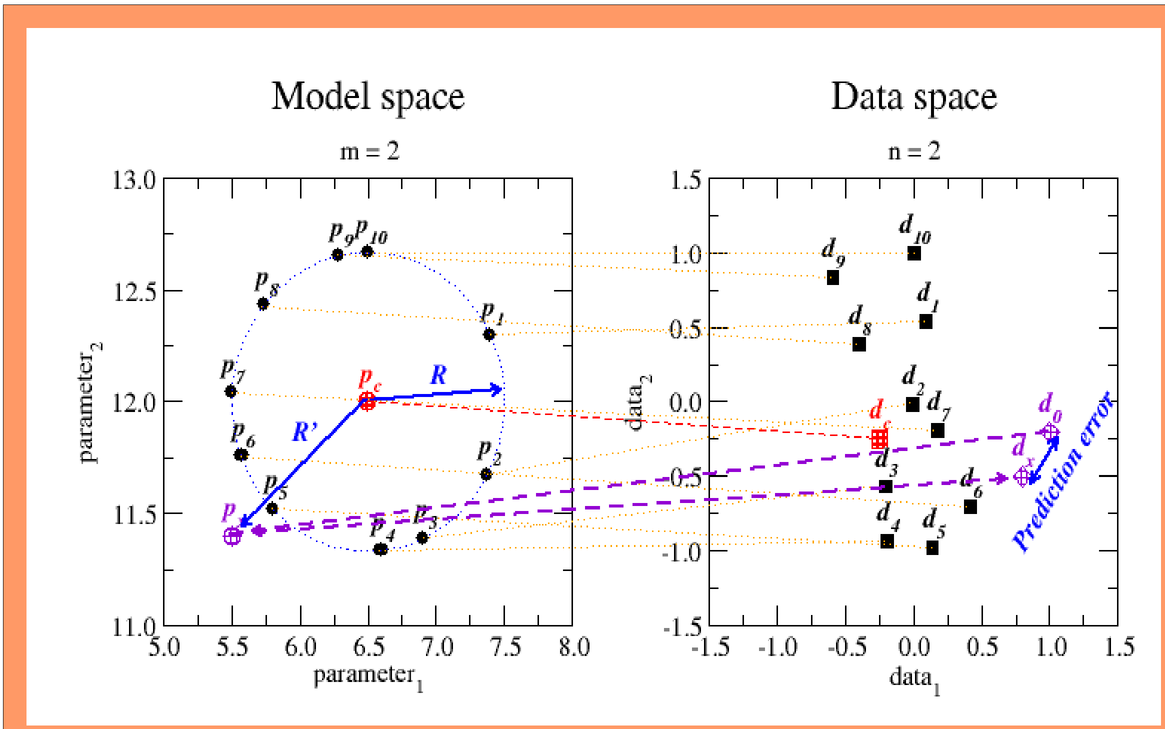

Fig.2. Schematic demonstration of the predicting population both in model and data spaces and the process of predicting the candidate solution. See also the appropriate text.

4.Numerical tests

The method similar as in (*Málek et al. 2007*) was used for testing the efficiency of *ANNIT*. Scalable set of polynomial equations enable to flexibly set up both model and data spaces dimensions and also the measure of non- linearity.

1. Linear case

$$a_{ij}\,p_j+r_i=d_i\,,\quad i=1,2,\cdots,n\,;\ j=1,2,\cdots,m \tag{24a}$$

2. Quadratic case

$$a_{ij}\,p_j+b_{ijk}\,p_j\,p_k+r_i=d_i\quad i=1,2,\cdots,n\,;\ j,k=1,2,\cdots,m \tag{24b}$$

3. Cubic case

$$a_{ij}\,p_j+b_{ijk}\,p_j\,p_k+c_{ijkl}\,p_j\,p_k\,p_l+r_i=d_i\quad i=1,2,\cdots,n\,;\ j,k,l=1,2,\cdots,m \tag{24c}$$

4. Biquadratic case

$$a_{ij}\,p_j+b_{ijk}\,p_j\,p_k+c_{ijkl}\,p_j\,p_k\,p_l+d_{ijklo}\,p_j\,p_k\,p_l\,p_o+r_i=d_i\quad i=1,2,\cdots,n\,;\ j,k,l,o=1,2,\cdots,m \tag{24d}$$

5. etc.

The coefficients a_{ij}, b_{ijk}, ...d_{ijklo}, r_i are adjusted in all cases (24a-d) as random numbers from the interval <−1; +1>, the solution to be found p_i is also random vector with components in the interval <−1; +1>, and forward problem (24a-d) gives synthetic data d_i. Alternatively right sides of all equations can be contaminated by synthetic noise.

The number of equations n is selected in order to ensure the uniqueness depending on the model space dimension as is documented in the Table 4.

Table 4

problem	number of equations	example A	example B
linear	$n = m$	$m = 5, n = 5$	$m = 10, n = 10$
quadratic	$n = 2m$	$m = 5, n = 10$	$m = 10, n = 20$
cubic	$n = 3m$	$m = 5, n = 15$	$m = 10, n = 30$
biquadratic	$n = 4m$	$m = 5, n = 20$	$m = 10, n = 40$

Convergence curves for all tested polynomial problems (except of linear problem which is solved even in the first iteration exactly) are depicted in the Fig.3 any time for 10 different randomly selected sets of coefficients. The correct solution was found in all cases, nevertheless sometimes was searching of the correct zone of attraction rather longer.

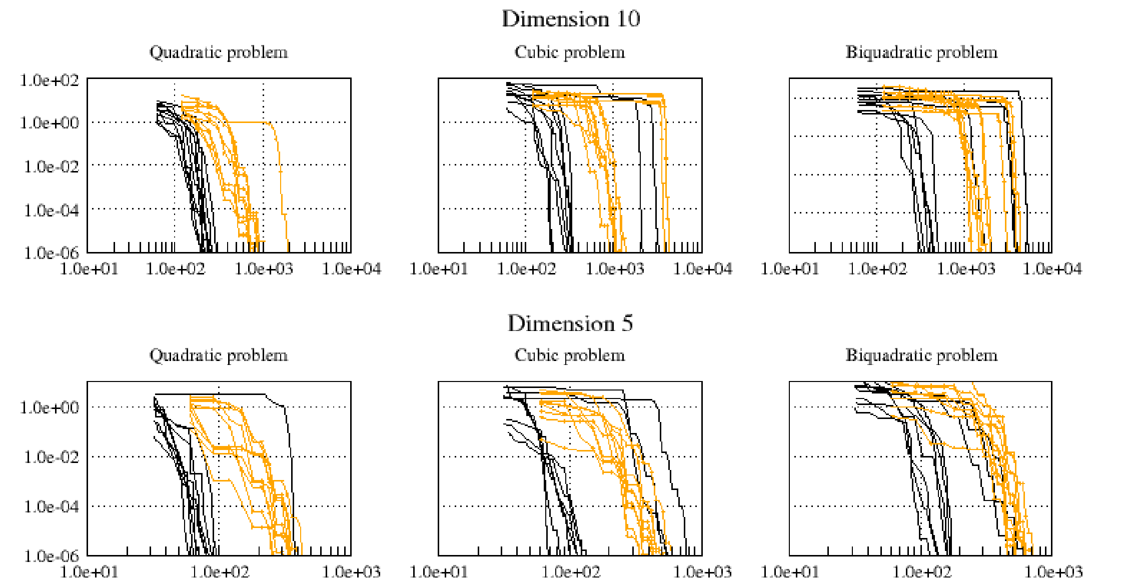

Fig.3. Convergence curves for a set of polynomial problems. Horizontal coordinates in each graph correspond to the number of forward evaluations, vertical coordinates correspond to the error of candidate solution. Black curves were obtained by using the ANNIT algorithm exactly as described in the text. Yellow curves correspond to modified computations where using of archive models were prohibited (equivalently $r_d = 0$ in (13)). Reusing of archive models is evidently advantageous since it speeds-up the convergence.

Sets of non-linear polynomial equations do not represent any particular physical problem. Nevertheless they are identical with first elements of the Taylor's expansion of multidimensional smooth functions, therefore they can be good approximation to many real physical problems depending on particular coefficients. Averaged convergence tests for randomly set up coefficients are then characteristic for the inverse algorithm and the specific features of the forward problem is of minor importance.

The efficiency of distinct prediction algorithms for the sequence of polynomial problems is summarized in the Fig.4. Any time better candidate solution was discovered, the method currently used was recorded. Apart from already described three prediction methods fourth way is possible how to generate candidate solution: some model randomly generated as a satellite individual is by random from now the best one model. Such cases are depicted in orange (RAND).

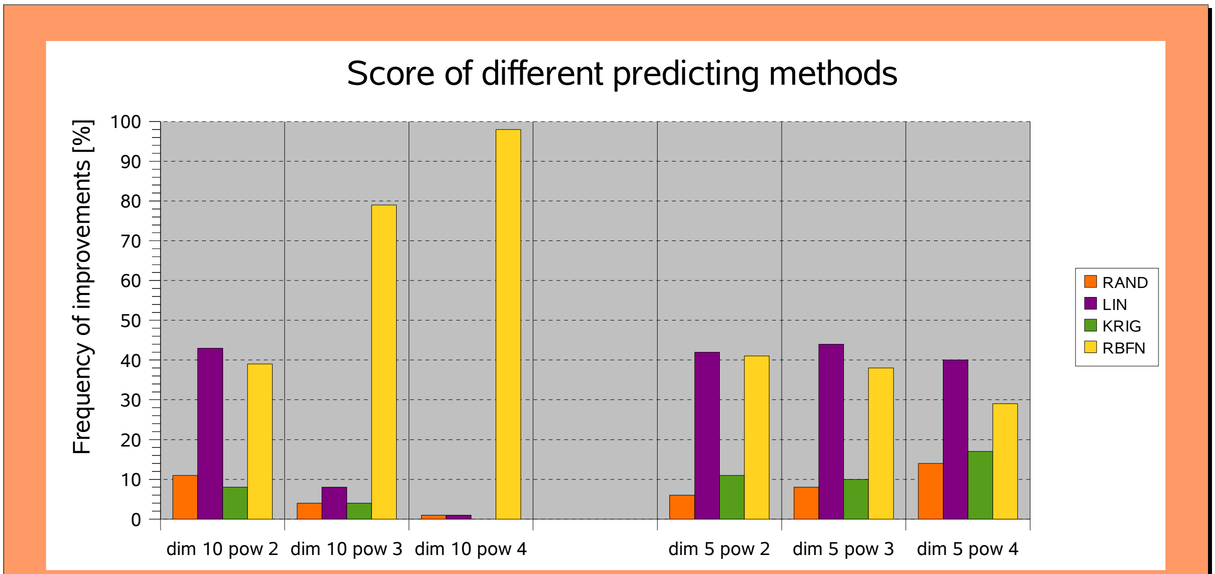

Fig.4. Comparison of the efficiency of individual predictors. Left part of the graph: polynomial problem of dimension 10. Right part of the graph: polynomial problem of dimension 5. Predictors are distinguished by colors.

Appendix A1. Mean distance between neighbouring points randomly distributed along the surface of a hypersphere

Let us consider a population of q models $\boldsymbol{p}_i$, $i = 1,2,3,...q$, generated by the algorithm *ANNIT*. Individual models of the population are distributed randomly with a uniform probability along the surface of an m-dimensional hypersphere. The origin of the hypersphere is considered in the coordinate origin without any loss of generality. The distance of each model from the hypersphere centre (origin) is R. The task is to determine the mean distance d_m from the collection of minimum distances between all doublets of models:

$$\begin{gathered} d_{ij}=|\boldsymbol{p}_i-\boldsymbol{p}_j|, i,j=1,2,3,...q, i\neq j \\ d_q=min(d_{qj}) \\ d_m=\langle d_q\rangle \end{gathered} \quad . \qquad \text{(A1)}$$

First the solution for the dimension $m = 3$ will be given for the sake of clearness. This task formulated in classical 3D space corresponds to random distribution of q points with uniform probability along the surface of a sphere. The area of a sphere is $S_3 = 4\pi R^2$. If this area is covered by q points, the area $S_2 = S_3/q = 4\pi R^2/q$ fall on each point. The sphere can be locally approximated by a plane in a proximity of each point for large q (the dimension is reduced to 2) and this part of the sphere can be represented by a circle with the diameter r: $S_2 = \pi r^2$. Now the area of the original sphere and sum of the areas of all circles should be the same, from which the characteristic distance between points $r_d \approx 2r$ can be estimated:

$$\begin{gathered} S_3=4\pi R^2, S_2=\pi r^2=S_3/q\rightarrow r^2=4R^2/q \\ r_d\approx 2r=2R/\sqrt{(q)} \end{gathered} \quad . \qquad \text{(A2)}$$

The above approach how to estimate the average span between points is used also in a general m-dimensional space:

$$S_m(R)=q.V_{m-1}(r) \qquad \text{(A3)}$$

where $S_m(R)$ is the area of m-dimensional hypersphere with the diameter R (this was area of a classical sphere in the preceding example), and $V_{m-1}(r)$ is the volume of $(m$-1)-dimensional hypersphere with the diameter *r* (area of a classical circle in the preceding example). The dependence between the volume and area of a hypersphere with the diameter R is evidently

$$\begin{gathered} V_m(R)=V_{m0}.R^m, resp. S_m(R)=S_{m0}.R^{m-1} \\ V_{m0}=V_m(1), S_{m0}=S_m(1) \end{gathered} \tag{A4}$$

where V_{m0} a S_{m0} define characteristic values for a unit hypersphere. Simple relation between area and volume holds for hypersphere with a general diameter:

$$\begin{gathered} V_m(R)=V_{m0}R^m=\int_0^R S_{m0}r^{m-1}dr=S_{m0}\left[\frac{r^m}{m}\right]_0^R=S_{m0}\frac{R^m}{m}=\frac{R}{m}S_m \quad . \\ \text{resp.}\, V_m=\frac{R}{m}S_m \end{gathered} \tag{A5}$$

Combining (A3) and (A4) only areas can be considered:

$$\begin{gathered} S_m(R)=q.V_{m-1}(r)=q\frac{r}{m-1}S_{m-1}(r) \quad . \\ \text{or}\, R^{m-1}S_{m0}=q\frac{r}{m-1}S_{(m-1)0}r^{m-2} \end{gathered} \tag{A6}$$

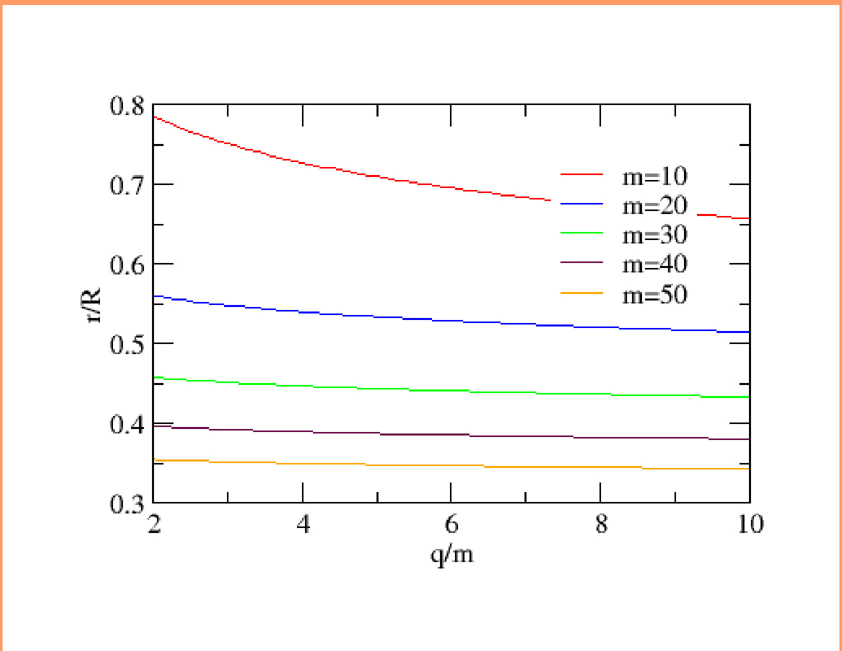

Fig.5. Relation between the mean relative distance between neighboring points r/R along the surface of a hypersphere. The number of points is q, radius of the hypersphere is R and the dimension of the space is m.

Now the general formula for the area of unit *m*-dimensional hypersphere S_{m0} has to be expressed:

$$S_m=\frac{2\pi^{\frac{m}{2}}}{\Gamma(\frac{m}{2})} \quad , \tag{A7}$$

(for proof see *www1*). Characteristic span r_d between *q*-points randomly distributed along the surface of the *m*-dimensional hypersphere is after rearrangement of (A6) and (A7)

$$r_d \approx 2r = 2.R.\left(\frac{m-1}{q}\right)^{\frac{1}{m-1}}.\pi^{\frac{1}{2}}.\frac{\Gamma\left(\frac{m-1}{2}\right)}{\Gamma\left(\frac{m}{2}\right)} \quad . \tag{A8}$$

It should be noted that strict physical realization of the presented approach is not possible, since the surface of *m*-dimensional hypersphere cannot be constructed exactly using a set of *m*-1 dimensional hyperspheres.

Appendix A2. Identification of parameters of general linear mapping

The purpose of this part is to show how to extract all necessary coefficients for representation of a linear problem by using population of models. Let us consider a linear system

$$F(\boldsymbol{p}) = \boldsymbol{Ap} + \boldsymbol{d_c} = \boldsymbol{d} \tag{A9}$$

which has straightforward solution for measured data $\boldsymbol{d}_0$

$$\boldsymbol{p_0} = pinv(\boldsymbol{A})(\boldsymbol{d_0} - \boldsymbol{d_c}) \quad . \tag{A10}$$

The *pinv* in (A10) means pseudoinversion, the solvability is subjected by knowing all elements of the matrix ***A*** and of the vector $\boldsymbol{d}_c$. The matrix ***A*** and vector $\boldsymbol{d}_c$ can be determined from the requirement that all models of the population $\{\boldsymbol{p}_i, \boldsymbol{d}_i\}$, i=1,2,3,...*q* fulfill the same linear relation (A9) like the searched for solution:

$$\begin{gathered} \boldsymbol{Ap_1} + \boldsymbol{d_c} = \boldsymbol{d_1} \\ \boldsymbol{Ap_2} + \boldsymbol{d_c} = \boldsymbol{d_2} \\ \cdots \\ \boldsymbol{Ap_q} + \boldsymbol{d_c} = \boldsymbol{d_q} \end{gathered} \quad . \tag{A11}$$

Averaging (A11) over rows $\boldsymbol{d}_c$ can be eliminated:

$$\boldsymbol{d_c} = \langle \boldsymbol{d_i} \rangle - A \langle \boldsymbol{p_i} \rangle \quad . \tag{A12}$$

Let us transform the vectors $\boldsymbol{p}_i$ a $\boldsymbol{d}_i$ as follows:

$$\boldsymbol{p_i'} = \boldsymbol{p_i} - \langle \boldsymbol{p_i} \rangle , \boldsymbol{d_i'} = \boldsymbol{d_i} - \langle \boldsymbol{d_i} \rangle \quad . \tag{A13}$$

Substituting centered vectors from (A13) into (A11) simpler equations not containing $\boldsymbol{d}_c$ are obtained

$$\begin{gathered} \boldsymbol{Ap_1'} = \boldsymbol{d_1'} \\ \boldsymbol{Ap_2'} = \boldsymbol{d_2'} \\ \cdots \\ \boldsymbol{Ap_q'} = \boldsymbol{d_q'} \end{gathered} \quad . \tag{A14}$$

Centered vectors ***d'*** and ***p'*** can be arranged into columns of matrices ***P'*** and ***D'***

$$\boldsymbol{P'} = \begin{bmatrix} \boldsymbol{p_1'} & \boldsymbol{p_2'} & \cdots & \boldsymbol{p_q'} \end{bmatrix} , \boldsymbol{D'} = \begin{bmatrix} \boldsymbol{d_1'} & \boldsymbol{d_2'} & \cdots & \boldsymbol{d_q'} \end{bmatrix} \tag{A15}$$

and equations (A14) can be equivalently expressed in one matrix equation

$$\boldsymbol{AP'} = \boldsymbol{D'} \tag{A16}$$

which solution for ***A*** is evidently

$$\boldsymbol{A} = \boldsymbol{D'}\, pinv(\boldsymbol{P'}) \quad . \tag{A17}$$

After substituting for ***A*** from (A17) into (A12) the vector $\boldsymbol{d}_c$ can be obtained. Knowing $\{\boldsymbol{A}, \boldsymbol{d}_c\}$ the forward problem, or its linear approximation, is completely defined.

Appendix A3. Typical radial basis functions

Radial basis function can be nearly any smooth function depending on distance. The differences by using different types of radial functions will be usually small for interpolation to points inside the convex hull of radial function centres. Significant differences will occur for extrapolation to points outside the convex hull, when the asymptotic behavior of the dominant radial function is decisive for *r* → inf. Usually the function value is +inf, -inf, or 0. This unfavorable feature of RBFN potentially causes instability. It is therefore desirable to avid extrapolation regime by using prediction population with suitable size and location.

ANNIT has been tested with following radial basis functions:

- Gaussian function

$$h(r)=e^{-r^2} \tag{A18}$$

- Quasilinear function

$$h(r)=1-r \tag{A19}$$

- Cauchy function

$$h(r)=e^{-r} \tag{A20}$$

- “Inverse Multiquadric” function

$$h(r)=\frac{1}{\sqrt{(1+r^2)}} \tag{A21}$$

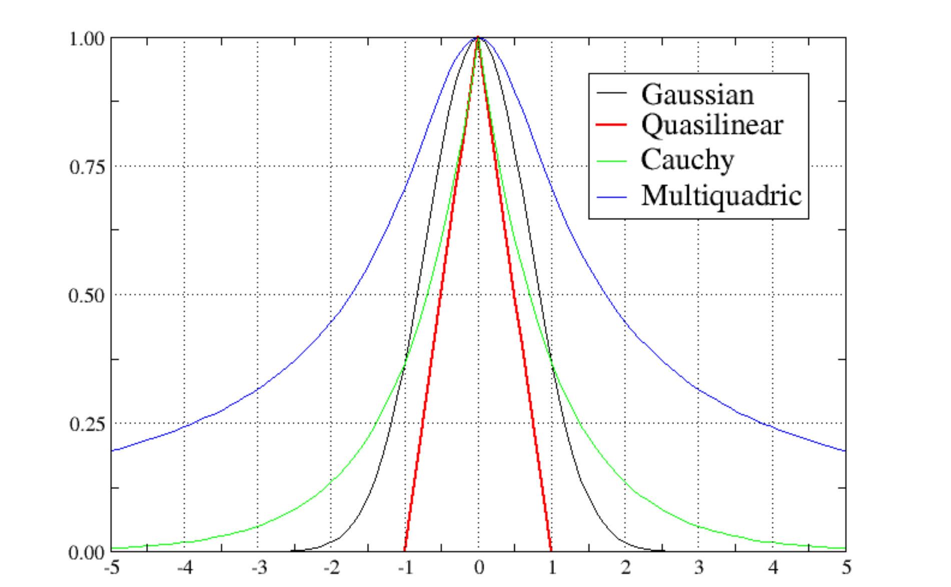

Fig.6. Example of typical radial basis functions. Quasilinear function is composed from linear sections connecting the centres of radial functions only in one-dimensional space, otherwise the shape of interpolated function is more complicated.

ANNIT is currently working with inverse multiquadric function. Using other types of radial functions is also possible.

Appendix A4. Interpolating formulae for the Radial Basis Function Network method

Prediction of the vector $\boldsymbol{p}_0$ corresponding to the data vector $\boldsymbol{d}_0$ according (17) is

$$\boldsymbol{p}_0^T=\boldsymbol{Hw}=\begin{bmatrix} h_{11} & h_{12} & \cdots & h_{1\mathrm{q}} \\ h_{21} & h_{22} & \cdots & h_{2\mathrm{q}} \\ & & \cdots & \\ h_{ql} & h_{q2} & \cdots & h_{qq} \end{bmatrix} \begin{bmatrix} h_1(\boldsymbol{d_0}) \\ h_2(\boldsymbol{d_0}) \\ \cdots \\ h_q(\boldsymbol{d_0}) \end{bmatrix} \tag{A22}$$

and the relation (A22) should fit the forward problem (1) so close as possible. The unknown coefficients h_{ij} can be determined using the requirement that (A22) holds for all individuals of the population. Let us use the following notation

$$\boldsymbol{w}_{\boldsymbol{i}}=\begin{bmatrix} h_1(\boldsymbol{d}_{\boldsymbol{i}}) \\ h_2(\boldsymbol{d}_{\boldsymbol{i}}) \\ \cdots \\ h_q(\boldsymbol{d}_{\boldsymbol{i}}) \end{bmatrix} \tag{A23}$$

then it is possible to write for any pair of vectors $\boldsymbol{p}_{\mathrm{i}}$ $\boldsymbol{d}_{\mathrm{i}}$, $i = 1,2,...q$

$$\boldsymbol{p}_1^T=\boldsymbol{Hw}_1,\ \boldsymbol{p}_2^T=\boldsymbol{Hw}_2,\ ...,\ \boldsymbol{p}_q^T=\boldsymbol{Hw}_q \tag{A24}$$

and if the matrices $\boldsymbol{P}$ and $\boldsymbol{W}$ will be constructed from vectors $\boldsymbol{p}$ and $\boldsymbol{w}$ as follows

$$\boldsymbol{P}=\begin{bmatrix}\boldsymbol{p}_1 & \boldsymbol{p}_2 & \cdots & \boldsymbol{p}_{\boldsymbol{q}}\end{bmatrix},$$

$$\boldsymbol{W}=\begin{bmatrix}\boldsymbol{w}_1 & \boldsymbol{w}_2 & \cdots & \boldsymbol{w}_{\boldsymbol{q}}\end{bmatrix}=\begin{bmatrix} h_1(\boldsymbol{d}_1) & h_1(\boldsymbol{d}_2) & \cdots & h_1(\boldsymbol{d}_{\boldsymbol{q}}) \\ h_2(\boldsymbol{d}_1) & h_2(\boldsymbol{d}_2) & \cdots & h_2(\boldsymbol{d}_{\boldsymbol{q}}) \\ & & \cdots & \\ h_q(\boldsymbol{d}_1) & h_q(\boldsymbol{d}_2) & \cdots & h_q(\boldsymbol{d}_{\boldsymbol{q}}) \end{bmatrix} \tag{A25}$$

then (A24) can be collected into one matrix equation

$$\boldsymbol{P}^T=\boldsymbol{HW} \tag{A26}$$

and coefficients in $\boldsymbol{H}$ to compute using the functions *inv* or *pinv*:

$$\boldsymbol{H}=\boldsymbol{P}^T\boldsymbol{W}^{-1}\ . \tag{A27}$$

The network of radial functions is chosen such in our case that the number of radial functions is the same as is the number of learning individuals. Therefore both the matrices $\boldsymbol{H}$ and $\boldsymbol{W}$ are square matrices of the q x q dimensions. More general cases can be solved analogously. It is also possible to include regularization, when the interpolated values do not reach exactly the learning points. See the bibliography at *www4* for deeper insight.

Appendix A5. Differences in interpolation/extrapolation by using different approximating methods

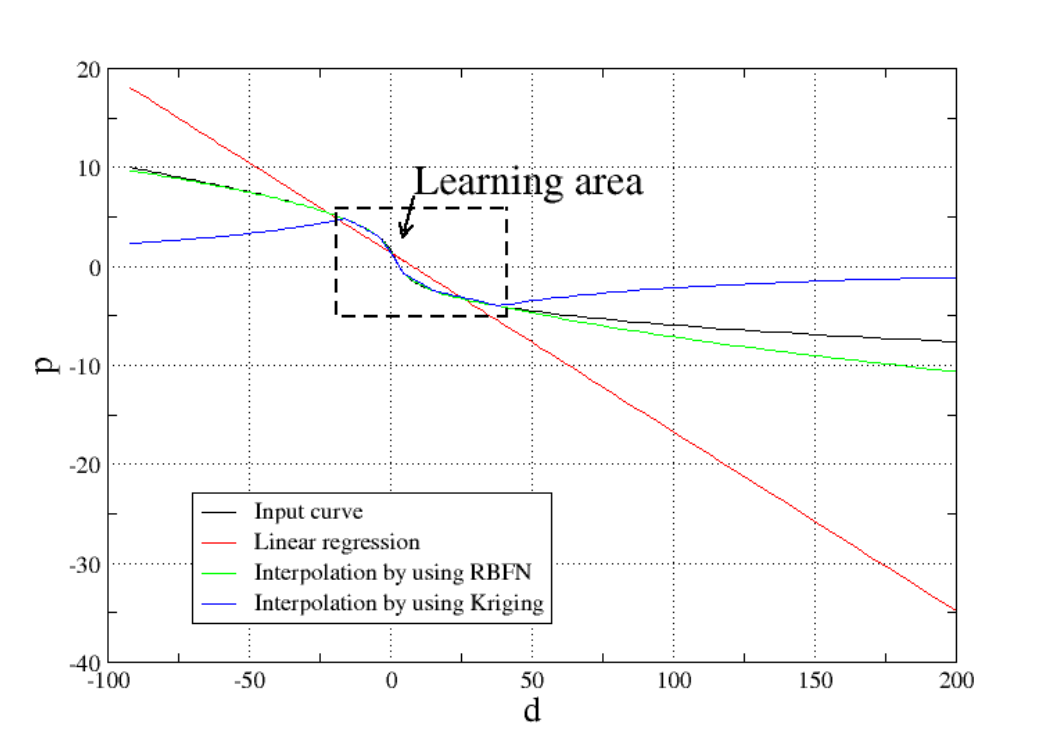

*Fig.7. The input functional dependence between **d** and **p** is given by polynomial relation*

$$\boldsymbol{d} = 3 - 2\boldsymbol{p} + 0.5\boldsymbol{p}^2 - 0.25\boldsymbol{p}^3 + 0.0125\boldsymbol{p}^4$$

*and since the problem considered is 1D only the vectors **d** and **p** and scalars in fact. True function is given by black line. Then using 10 random points lying on the black curve (not shown here) and inside the learning area were used as a population. Three predicting methods giving **p** for any **d** are documented by different colours. In this case, the RBFN predictor is clearly the best one, but in other circumstances the relations may be completely different. Linear predictor will give a straight line in any case, and Kriging will tend to output the average value at great distances from the learning area.*

References

(1) [www1] http://mathworld.wolfram.com/Hypersphere.html

(2) [www2] http://en.wikipedia.org/wiki/Kriging

(3) [www3] http://en.wikipedia.org/wiki/Optimization_(mathematics)

(4) Press, W. H.; Flannery, B. P.; Teukolsky, S. A.; and Vetterling, W. T. Numerical Recipes. The Art of Scientific Computing, 3rd ed., Cambridge University Press, 2007.

(5) Orr M.J.L. Introduction to Radial Basis Function Networks. Edinburgh 1996. http://anc.ed.ac.uk/rbf/intro/intro.html

(6) [www4] http://citeseerx.ist.psu.edu/

(7) [www5] http://en.wikipedia.org/wiki/Radial_basis_function

(8) [www6] http://www.aranz.com/research/modelling/theory/

(9) Málek, Jiří ; Růžek, Bohuslav ; Kolář, Petr. Isometric Method: Efficient tool for Solving Non-linear Inverse Problems. *Studia geophysica et geodaetica*. Roč. **51**, č. 4 (2007), s. 469-490. ISSN 0039-3169.

Acknowledgments

This work was accomplished due to the support of grant projects **IAA200120701** and **IAA300120805** funded by the Grant Agency of the Academy of Sciences of the Czech Republic. For project web pages see:

http://www.ig.cas.cz/en/personal-pages/bohuslav-ruzek/anno/

http://rebel.ig.cas.cz/seismo/Petka_strankyProWWW/StopPhase/Stop_Phase_main.htm