ECCOMAS Congress 2016
VII European Congress on Computational Methods in Applied Sciences and Engineering
M. Papadrakakis, V. Papadopoulos, G. Stefanou, V. Plevris (eds.)
Crete Island, Greece, 5–10 June 2016

A NEW OPTIMIZED ANISOTROPIC MOVING LEAST SQUARES SURROGATE MODEL WITH MAXIMIZED PROGNOSIS

Kevin Cremanns[1] **& Dirk Roos**[2]

Institute of Modelling and
High-Performance Computing
Niederrhein University of Applied Sciences, Germany
[1] Kevin.Cremanns@hs-niederrhein.de
[2] Dirk.Roos@hs-niederrhein.de

Keywords: Anisotropic Moving Least Squares, Weighted Least Squares, Local Regression, Surrogate Model, Metamodel, Maximized Prognosis Quality

Abstract. *Computational simulations or experiments in engineering applications are often time and resource consuming. This is especially so when performing optimization, robustness or reliability analyses, because they need hundreds or even thousands of design evaluations. A common solution for this are surrogate models, which are mathematical approximations of the existent data. Nevertheless, these surrogate models also need some design evaluations as a base to train them and are usually obtained by performing a design of experiments with an intelligent design plan to reduce the number of needed points.*

Moving least squares is a very simple but powerful surrogate model method. It performs a least squares regression with a subset of the data in the local area of the point to be approximated. This allows the method to approximate complex functions with a low order polynomial base (first or second order). This basically works with the help of a weighting function, which gives a high weight to points near the target point and decreasing weights to points with an increasing distance to this point. Therefore, in the standard moving least squares method, a free model parameter has to be estimated. This parameter describes the influence radius, in which points are considered to approximate the target point. In this standard formulation, the weights depend on the distances of all variables. So, variables with a minor or even non-influence, contribute to these weights and so to the accuracy of the approximation.

The proposed method works anisotropically and is similar to methods like the anisotropic Kriging, so that the free model parameters need to be defined for each variable individually and the possibility of suppressing unimportant variables and thus increasing the prognosis quality of the standard moving least squares method is given.

This work presents the mathematical description of this method and demonstrates its performance for different analytical test functions compared to the standard moving least squares method.

NOMENCLATURE

AMLS	anisotropic moving least squares
CFD	computational fluid dynamics
DOE	design of experiment
FSI	fluid structure interaction
LHS	latin hypercube sampling
MLS	moving least squares
CV	cross validation
α	constant for weighting function
μ	mean value
σ	variance
x_n	n-th variable of prediction point $\mathbf{x}$
n	number of input variables
N	number of samples
$\widetilde{y}$	approximated output
R^2_{pred}	prediction quality based on CV
w_n	Gaussian weighting function
r	radius of points used for approximation
w_i	weight for input point i
$\mathbf{x}$	prediction point
$\mathbf{v}_n$	training values for n-th variable
$\mathbf{p}(\mathbf{x})$	vector of basis polynomials
$\mathbf{a}(\mathbf{X}, \mathbf{x})$	vector of approximation coefficients
$\mathbf{y}(\mathbf{X})$	vector of output training values
$\mathbf{w}_{N,n}$	weighting vectors for the anisotropic formulation
$\widetilde{\mathbf{y}}$	vector of approximated output values
$\mathbf{X}$	matrix of training points
$\mathbf{P}(\mathbf{X})$	matrix of basis polynomials $N \times n$
$\mathbf{W}(\mathbf{X}, \mathbf{x})$	weighting matrix $N \times N$
$\mathbf{X}_{\text{test}}$	points used to calculate R^2_{pred} not used for training

SUBSCRIPTS

n	n-th variable
m	$(n-1)$-th variable
N	N-th design point
test	test point not used for training
iso	isotropic
aniso	anisotropic

1 INTRODUCTION

In engineering applications, a lot of time and resource consuming simulations (e.g. CFD / FSI - calculations) or experiments take place. Therefore, in most cases, optimization or robustness evaluations are too expensive in terms of time and resources. One outcome of this is the usage of surrogate models. These are functions based on given training data points to approximate new unknown function evaluations within the initial training data range.

Especially in the field of aerodynamic optimization, surrogate models are used, because of the high calculation time. In [26] used for example a simulated annealing optimization algorithm to optimize the shape of a three-finned, stepped labyrinth seal. This optimization took about 830 design evaluations with a calculation time of 25 minutes per design. Hence, the whole optimization took more than 14 days. Therefore, others like [25] and [22] showed optimizations of turbo machinery parts by using surrogate models trained on a specific number of CFD-calculations. Also in other fields, surrogate models are used for optimization such as aircraft design ([11]), engine cycle design ([3]) or structural reliability analysis ([18]). A general overview of surrogate based optimization is also given in [9].

A wide range of different methods to create such surrogate models exist. Some of the most sophisticated are radial basis functions, Kriging (also known as the Gaussian process), support vector regression, neural networks and moving least squares. Because of this variety of methods, a lot of benchmarks for these different methods also exist, as described in [6], [10], [15] and [4]. In most of these comparisons Kriging ([14]) was identified as one of the most accurate methods and it has been shown that this method works well for a whole range of problems. However, in some cases other methods, such as moving least squares (MLS), ([16], [17]) can be more sophisticated then Kriging, as [1] shows for non-convex functions. Especially in arbitrarily spaced data ([7]) or in 3D surface reconstruction presented in [27], [2] and [8], MLS showed good performance.

MLS regression has the advantage compared to methods like Kriging, that it does not experience difficulties when there are a large number of design points to train on, because of numerical instabilities. Also, if the design points show different local behaviour, such as clusters, MLS performs well. The main obstacle for local regression methods like MLS is to determine the points to consider for the local approximation, especially if the points have a large variance in a defined window around the prediction point.

Another point is the dimensionality of the problem to be approximated. A lot of surrogate methods experience problems with high dimensional data. In the standard isotropic MLS all dimensions have an influence on the variance of the data points used for the approximation. Therefore, with increasing dimensionality the approximation accuracy reduces. Two possible methods exist for working with high dimensional problems. The first method is to reduce the variables to only the most important parameters through the use of a sensitivity analysis such as the variance based sensitivity analysis from [12]. An example of the successful utilization of this type of procedure is presented in [21] or in [5]. In many engineering applications, the number of most important parameters for an individual output is up to 20 input variables. Therefore, these methods work very well for finding these few important parameters and thus dramatically increase the performance of the surrogate model used. The other possible method of working with high dimensional data is anisotropy. In a similar way to anisotropic Kriging, the surrogate method provides the opportunity to consider the different importance influence of the input variables in the prediction. In Kriging, different correlation lengths will be defined for each input parameter, leading to higher accuracy of the Kriging surrogate model, especially

if the variance is different for each input parameter. [24] showed this for the MLS method. Also, an anisotropy was introduced through the definition of different bandwidths for each input parameter. Then, only the points which fall in the intersection of the different bandwidths which define the prediction window, are used for the prediction. As will be shown in section 4, different bandwidths for each input parameter will be defined, but a different method of calculating the overall weights for each prediction point will be used.

In this paper, the concept of anisotropy is used and applied to the moving least squares method different to [24], which in most cases leads to higher prognosis quality. First, the standard MLS is described, followed by the extension to anisotropy. Subsequently, examples of different analytical test functions are shown to compare both methods.

2 DESIGN SAMPLING

To create the training data for the surrogate models, some simulation or experiments have to be evaluated as base information for different designs. These designs should be part of a sampling plan which covers the range of interest for the specific problem. There are many possible methods for creating such a design plan, such as full factorial, D-optimal or orthogonal arrays. Latin hypercube sampling is commonly used (LHS) ([20]), which creates *n* hypercubes for each variable. Within each of these hypercubes, a point is chosen randomly, or in the case of mid-point LHS the point is always set in the center of this hypercube. Optimized LHS also exist, with optimized distance between the points and minimal unwanted correlations as [13] shows. For the test problems shown here, an optimized LHS will be used.

3 REVIEW OF THE MLS APPROXIMATION METHOD

The basis for the anisotropic moving least squares (AMLS) approximation is the standard MLS. The formulation to approximate the value $\widetilde{y}$ of the prediction point $\mathbf{x}$ is:

$$\widetilde{y}(\mathbf{x}) = \mathbf{p}^T(\mathbf{x})\mathbf{a}(\mathbf{X}, \mathbf{x}) \tag{1}$$

The prediction point $\mathbf{x}$ consists of the *n* input variables of the problem to approximate (e.g. a simulation model, mathematical function or experiment):

$$\mathbf{x} = [x_1, x_2, ..., x_n] \tag{2}$$

The vector $\mathbf{p}(\mathbf{x})^T$ contains the basis polynomial for the approximation depending on $\mathbf{x}$. For example a second order polynomial with cross terms:

$$\mathbf{p}^T(\mathbf{x}) = [1, x_1, x_2, \ldots, x_n, x_1^2, x_2^2, \ldots, x_n^2, x_1x_2, \ldots, x_nx_m]^T \tag{3}$$

The vector $\mathbf{a}(\mathbf{X}, \mathbf{x})$ contains the approximation coefficients and is calculated through:

$$\mathbf{a}(\mathbf{X}, \mathbf{x}) = (\mathbf{P}^T(\mathbf{X})\mathbf{W}(\mathbf{X}, \mathbf{x})\mathbf{P}(\mathbf{X}))^{-1}\mathbf{P}^T(\mathbf{X})\mathbf{W}(\mathbf{X}, \mathbf{x})\mathbf{y}(\mathbf{X}) \tag{4}$$

with the matrix $\mathbf{P}^T(\mathbf{X})$ containing the basis polynomials for the $N \times n$ training points matrix $\mathbf{X}$ of *n* samples:

$$\mathbf{X} = \begin{Bmatrix} x_{1,1} & \ldots & x_{n,1} \\ \vdots & \ddots & \vdots \\ x_{1,N} & \ldots & x_{n,N} \end{Bmatrix} \tag{5}$$

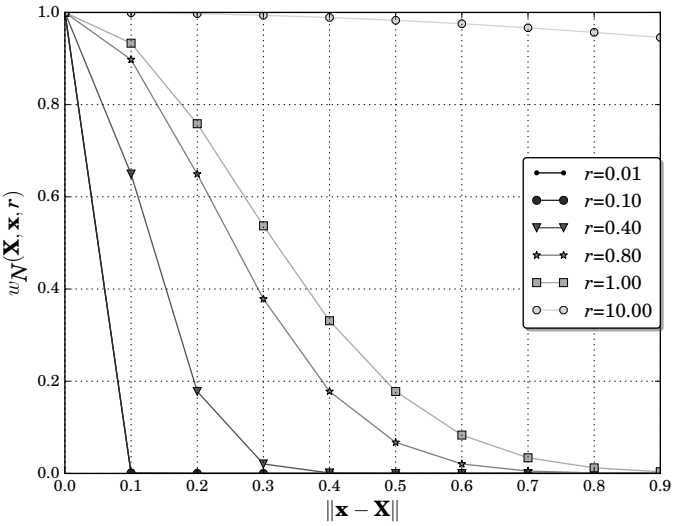


Figure 1: Gaussian weighting function depending on *r*

with $\mathbf{P}^T(\mathbf{X})$:

$$\mathbf{P}^T(\mathbf{X}) = \begin{Bmatrix} 1 & x_{1,1} & \dots & x_{n,1} & x_{1,1}^2 & \dots & x_{n,1}^2 & x_{1,1}x_{2,1} & \dots & x_{n,1}x_{n+1,1} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 1 & x_{1,N} & \dots & x_{n,N} & x_{1,N}^2 & \dots & x_{n,N}^2 & x_{1,N}x_{2,N} & \dots & x_{n,N}x_{n+1,N} \end{Bmatrix} \tag{6}$$

The matrix $\mathbf{W}(\mathbf{X}, \mathbf{x})$ is the $N \times N$ weighting diagonal matrix for the prediction point $\mathbf{x}$ and the main mathematical difference to normal polynomial regression. $\mathbf{W}(\mathbf{X}, \mathbf{x})$ contains the weights for each element of $\mathbf{X}$ regarding their distance to $\mathbf{x}$:

$$\mathbf{W}(\mathbf{X}, \mathbf{x}) = \begin{Bmatrix} w_1 & \dots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \dots & w_N \end{Bmatrix} \tag{7}$$

To estimate these weights, different types of weighting functions exist. Some of them can even force the MLS to interpolate through the training points ([19]). In this investigation a Gaussian weighting function is used like proposed in [17]:

$$w_N(\mathbf{X}, \mathbf{x}, r) = e^{-\left(\frac{\sum_1^N \|\mathbf{x} - \mathbf{X}\|}{r\alpha}\right)^2} \tag{8}$$

with the constant:

$$\alpha = \frac{1}{\sqrt{-log(0.001)}} \tag{9}$$

In this formulation the $\|\mathbf{x} - \mathbf{X}\|$ stands for the Euclidean distance calculated over all variable dimensions between the prediction point $\mathbf{x}$ and the trainings points $\mathbf{X}$. There is one free model parameter *r*, which represent the influence radius around the prediction point $\mathbf{x}$, in which the training points $\mathbf{X}$ are considered to approximate $\mathbf{x}$. As shown in Fig. 1 the weights tend to 0 if the training points lie outside of *r*. The last part to estimate $\mathbf{a}(\mathbf{X}, \mathbf{x})$ is the output vector $\mathbf{y}(\mathbf{X})$,

which contains the output values of the training points $\mathbf{X}$.

$$\mathbf{y}(\mathbf{X}) = [y_1, \ldots, y_N] \tag{10}$$

To estimate the free model parameter r different kinds of possibilities exist. A straightforward solution is to use an optimizer in combination with a stratified k-fold cross validation (CV). k-fold CV involves splitting the training points in k equal sized packages. Then the model is trained with $k-1$ packages and the remaining package is used as prediction points (test points). After each package has been evaluated the corresponding prognosis quality can be estimated with:

$$R^2_{\text{pred}} = \left(\frac{\sum_1^{N_{\text{test}}} (\mathbf{y}(\mathbf{X}_{\text{test}}) - \mu_{\mathbf{y}_{\text{test}}})(\widetilde{\mathbf{y}}(\mathbf{X}_{\text{test}}) - \mu_{\widetilde{\mathbf{y}}_{\text{test}}})}{(N_{\text{test}} - 1)\sigma_{\mathbf{y}_{\text{test}}}\sigma_{\widetilde{\mathbf{y}}_{\text{test}}}} \right)^2 \tag{11}$$

The objective of the optimization would be to maximize the prognosis quality $R^2_{\text{pred}} \in [0, 1]$.

4 EXTENSION TO ANISOTROPY

Having explained the standard MLS method, this section describes the extension of the method to the anisotropy. The key point of the MLS method is the weighting of training points, so that each prediction point is locally approximated. In the standard formulation, the Euclidean distance is calculated over all variable dimensions, as shown in 8. To give each variable an individual influence on the weights for the training points used, the anisotropic version calculates the weights for each variable separately. This leads to n weighting vectors $\mathbf{w}_{N,n}(\mathbf{v}_n, x_n, r_n)$ (the diagonal of the weighting matrix $\mathbf{W}(\mathbf{X}, \mathbf{x})$) with n elements and vector $\mathbf{v}_n$ describing the training data of the n-th variable:

$$w_{N,n}(\mathbf{v}_n, x_n, r_n) = e^{-\left(\frac{\sum_1^N \|\mathbf{v}_n - x_n\|}{r_n \alpha} \right)^2} \tag{12}$$

To combine these weighting vectors to one global and so to a corresponding weighting diagonal matrix $\mathbf{W}(\mathbf{X}, \mathbf{x})$, the geometric mean over all n weighting vectors $\mathbf{w}_{N,n}(\mathbf{v}_n, x_n, r_n)$ is calculated:

$$\mathbf{W}(\mathbf{X}, \mathbf{x}) = \text{diag}\left(\sqrt[n]{\prod_{i=1}^{n} \mathbf{w}_{N,n}(\mathbf{v}_n, x_n, r_n)} \right) \tag{13}$$

Compared to the standard MLS, one r_n per variable exists now. Because the geometric mean is used to combine the weighting vectors $\mathbf{w}_{N,n}(\mathbf{v}_n, x_n, r_n)$, variables with a minor or no influence on the output $\mathbf{y}(\mathbf{X})$, their r_n will tend to:

$$r_n \to \infty \tag{14}$$

because $r_n \to \infty$ gives $\mathbf{w}_{N,n}(\mathbf{v}_n, x_n, \infty) = \mathbf{1}$, meaning that all training points are considered with a weight of 1 regarding the non-important variables (see Fig. 1 with $r = 10$). Whereas variable with high influence will tend to:

$$r_n \to 0 \tag{15}$$

and so the weights are $\neq 0$ for training points in the local area of the prediction point $\mathbf{x}$. This means that only the r_n of the important variables have a significant influence on the final weighting matrix $\mathbf{W}(\mathbf{X}, \mathbf{x})$. Even though with this anisotropic approach of the MLS method it is not

necessary to scale the input parameters to equal ranges as is necessary for the isotropic MLS, but it is recommended that this scaling is done for numerical stability.

To estimate the free model parameters r_n the Powell optimization algorithm is used ([23]). This algorithm is especially suitable for this kind of optimization problem, because it iterates in a double loop over each optimization variable separately. The inner loop iterates over each free parameter and changes it individually until no further improvement is archived. If each free parameter has been optimized once, a new iteration begins (the outer loop). Furthermore, this algorithm does not need any gradients, just a start vector.

Additional as mentioned in 1, it is suggested that a sensitivity analysis should be used in applications with very high dimensions, to further increase the accuracy of the model and to decrease the optimization effort. So overall, it is recommended that this anisotropic MLS method is used in combination with a variance based sensitivity analysis ([12]) to receive the most accurate surrogate model.

5 EXAMPLES

In the following section, examples of comparing the standard MLS method to the anisotropic version are shown. For the comparison, the same Powell optimization algorithm is used for the isotropic MLS as for the anisotropic MLS. The optimization criterion is the already introduced R^2_{pred} (11) evaluated with k-fold CV. To train the surrogate model, 50 uniformly distributed samples were created, with optimized latin hypercube sampling as mentioned in 2. For these examples, no scaling was used, because all parameters are in the same value range. Further, each test function was approximated with a first order and second order polynomial base and the best result was taken. The k-fold CV was done with $k = 5$, with the same packages structure (same designs in each package) for both methods. For the comparisons 4 test functions assist in demonstrating how the AMLS works and how the free parameters r_n behave. They are defined as:

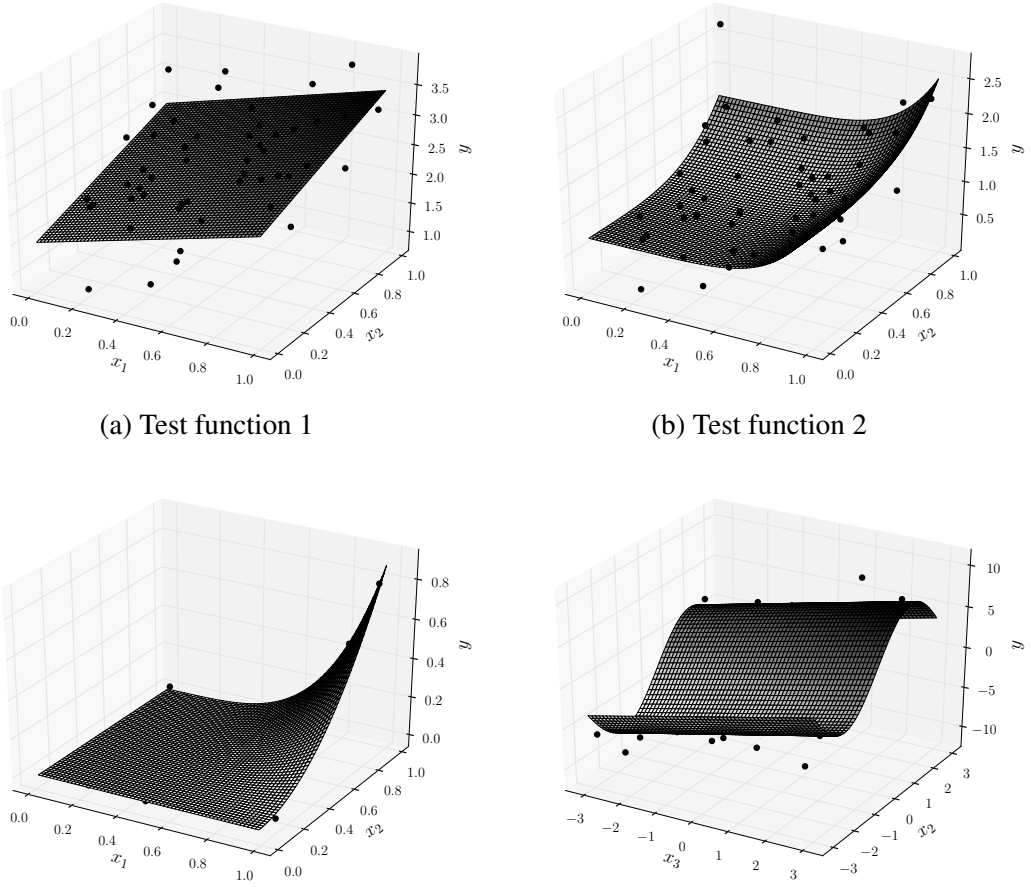
$$y(\mathbf{x}) = x_1 + x_2 + x_3 + x_4 + x_5, x_{1-5} \in [0, 1] \tag{16}$$

$$y(\mathbf{x}) = x_1^6 + x_2^4 + x_3^2 + x_4 + 0.001x_5, x_{1-5} \in [0, 1] \tag{17}$$

$$y(\mathbf{x}) = x_1^4 x_2^2 + 0.01x_3x_4 + 0.001x_5, x_{1-5} \in [0, 1] \tag{18}$$

$$y(\mathbf{x}) = 0.5x_1 + x_2 + 0.5x_1x_2 + 5sin(x_3) + 0.2x_4 + 0.1x_5, x_{1-5} \in [-\pi, \pi] \tag{19}$$

In Eq. 16 all variables have the same influence on $y(\mathbf{x})$ and should show equal results for the isotropic MLS and AMLS. Furthermore, all r_n of the AMLS should be equal because all variables have equal influence on the output $y(\mathbf{x})$. Eq. 17 includes decreasing influence from x_1 the highest and x_4 the lowest. Here it is intended, that the free parameter r_n should also show a decreasing value for the corresponding input parameters. The variance of each variable is different and so the power of the anisotropic version should result in a higher R^2_{pred} value for the AMLS. Eq. 18 contains influence of cross terms with a high variance in one corner of the data range and Eq. 19 has one dominant variable x_3 and 4 'noise' terms. The test functions 16 to 19 are illustrated in Fig. 2a to Fig. 2d. Tab. 1 shows the results for the free parameters of the MLS r_{iso} and for AMLS r_{1-5}. Additional the prognosis quality R^2_{pred} for both methods is given. For the first test function 16 the free model parameter of the AMLS were all equal as expected. The prognosis quality of both methods is maximal with $R^2_{\text{pred}} = 1$ because the test



(a) Test function 1

(b) Test function 2

(c) Test function 3

(d) Test function 4

Figure 2: Overview test functions

function is an additive simple example. The values of the free model parameters r_n of the AMLS method is higher than parameter the range of $[0, 1]$, which means all points are considered in the weighting and there is no local regression effect, in other words it works like a normal least squares approximation. The same applies to the MLS method for this specific example with parameter ranges of $[0, 1]$ with a maximum of $\sqrt{\sum_1^5 1^2} = \sqrt{5}$. The results for the second test function show the first difference between the MLS and the AMLS method regarding the prognosis quality R^2_{pred} of 5%. The free model parameters r_n of the AMLS are very low for the most important parameters x_1, x_2, x_3 and are very high for the unimportant parameters x_4, x_5 so that their weighting does not influence the overall weighting in the geometric mean described in 13. The values of the important parameters suggest that the approximation works locally. Additionally, there is a ranking of the parameters x_1, x_2, x_3 corresponding to the real ranking in the test function. The third test function 18 shows the highest difference of R^2_{pred} with 72% for all test functions. Here the AMLS seems to use all parameters, but with a focus on the first two parameters x_1, x_2 as important parameters for the overall weighting, corresponding to the test function part with $x_1^4 x_2^2$ which clearly dominate the values of the test function. The MLS is not capable to approximate this function with one single free parameter r. The last test function 19 yields a difference for R^2_{pred} of 34%. The dominant parameter x_3 has been identified in the AMLS, which results in the lowest free model parameter of $r_3 = 0.99$.

r_{iso}	r_1	r_2	r_3	r_4	r_5	$R^2_{\text{pred}_{\text{iso}}}$	$R^2_{\text{pred}_{\text{aniso}}}$
7.06	3.07	3.07	3.07	3.07	3.07	1.0	1.0
1.73	0.33	0.67	0.72	5494.53	2314.16	0.92	0.97
1.97	0.07	0.14	0.88	0.35	0.44	0.09	0.81
7.56	4762.98	5.17	0.99	3.71	2.85	0.54	0.88

Table 1: Overview of free model parameters r_n for test functions 16 to 19

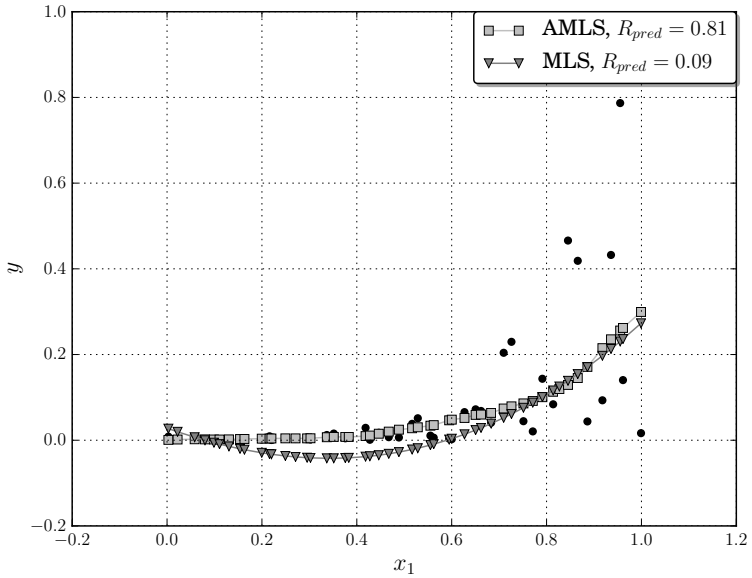


Figure 3: Surrogate model for test function 3, 1D view for parameter x_1

In Fig. 3 and 4 the 1D view of the surrogate models for the most important parameters of test function 18 and 19 is presented. For both pictures, it can be seen, that the trend is captured much better by the AMLS corresponding to the higher R^2_{pred}, especially for x_3 of test function 19, where it captured the $sin()$ term compared to the MLS, which seems to average through the data points.

6 CONCLUSION

The anisotropic moving least squares (AMLS) surrogate model method presented here uses, in contrast to the standard isotropic moving least squares (MLS), a free model parameter r_n for each input parameter and further the geometric mean to calculate the overall weighting (see 4). The AMLS method is more accurate than the MLS method for the examples with input parameters of differing strong influence as demonstrated with the test functions Eq. 17 to Eq. 19. Large differences could be seen in the example with high multivariate dependencies.

This is accomplished by setting the free model parameter r_n corresponding to the influence of the input parameter. Input parameters with a high influence result in lower values for r_n and vise versa. To find the right value for r_n, the Powel optimization algorithm was used. A possible start solution vector for this optimizer can be obtained by previously performing a sensitivity analysis. Otherwise, it is recommended that the mean value of the input parameter bounds for each r_n are used as a starting point. Additionally, if the input parameters are of different

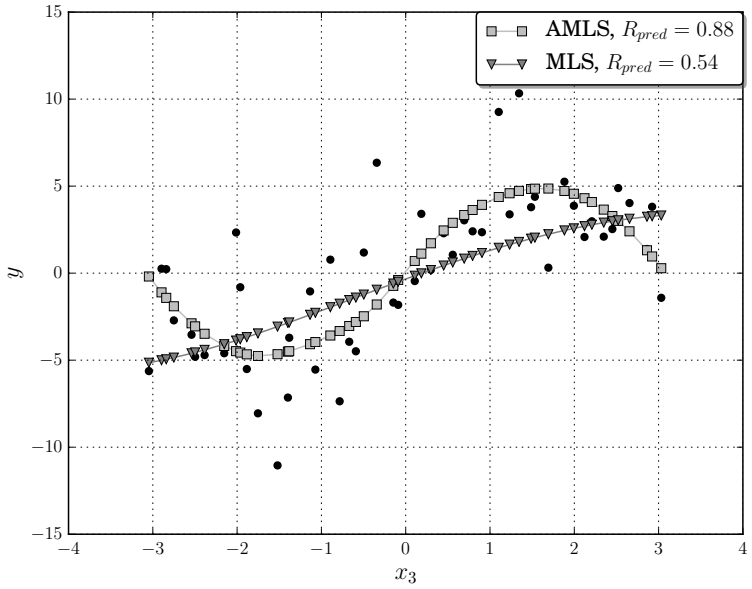


Figure 4: Surrogate model for test function 4, 1D view for parameter x_3

magnitudes it is recommended that they are rescaled.

If no sensitivity analysis has been performed, the AMLS optimization can also work as a simple kind of sensitivity analysis. Unimportant parameters can be identified through high values for the r_n parameters (similar to high correlation lengths in the anisotropic Kriging method). For all other parameters, the optimization also delivers a ranking of importance with the smallest r_n value correspond to the most important parameter etc.

Further improvements might be archived by testing different weighting functions. For example an interpolating weighting function or a function which avoids the high numerical numbers in the optimization for unimportant parameters by normalizing the maximum r_n with the maximum 1D distance for each specific input variable.

ACKNOWLEDGMENT

The authors would like to express their thanks to the German Federal Ministry for Economic Affairs and Energy, which supported this project, support identification no.: 03ET7071D.

References

[1] K. Beyong-Soo, L. Yong-Bin, and C. Dong-Hoon. Comparison study on the accuracy of metmetamodel technique for non-convex functions. *Journal of Mechanical Science and Technology*, 23:1175–1181, 2009.

[2] W. Y. Chang. Surface Reconstruction from Points. Technical report, Department of Computer Science and Engineering University of California, San Diego, 2009.

[3] J. A. Corman and B. J. German. A Comparison of Metamodeling Techniques for Engine Cycle Design Data. In *13th AIAA/ISSMO Multidisciplinary Analysis Optimization Conference*, 2010.

[4] J.-P. Costa, L. Pronzato, and E. Thierry. A comparison between Kriging and radial basis function network for nonlinear prediction. In *Conference: Proceedings of the IEEE-*

EURASIP Workshop on Nonlinear Signal and Image Processing (NSIP'99), Antalya, Turkey, June 20-23, 1999.

[5] K. Cremanns, D. Roos, and A. Grassmann. Sequential vs. Multidisciplinary Coupled Optimization and Efficient Surrogate Modelling of a Last Stage and the Successive Axial-radial Diffusor in a Low Pressure Steam Turbine. In *ASME Turbo Expo 2014: Turbomachinery Technical Conference and Exposition*, 2014.

[6] H. Fang, M. Rais-Rohani, Z. Liu, and M. F. Horstmeyer. A comparative study of metmetamodel methods for multiobjective crashworthiness optimiztaion. *Computers and Structures*, 83:2121–2136, 2005.

[7] R. Farwig. Multivariate interpolation of arbitrarily spaced data by moving least squares methods. *Journal of Computational and Applied Mathematics*, 16:79–93, 1986.

[8] S. Fleishman, D. Cohen-Or, and C. T. Silva. Robust moving least-squares fitting with sharp features. *ACM Transactions on Graphics (TOG) - Proceedings of ACM SIGGRAPH 2005*, 24:544–552, 2005.

[9] A. I. J. Forrester and A. J. Keane. Recent advances in surrogate-based optimization. *Progress in Aerospace Sciences*, 45:50–79, 2009.

[10] A. A. Giuanta and L. T. Watson. A comparison of approximation modeling techniques: polynomial versus interpolating models. Technical report, Multidisciplinary Analysis and Design (MAD) Center for Advanced Vehicles, Virginia Polytechnic Institute and State University, 1998.

[11] Z.-H. Han and K.-S. Zhang. Surrogate-Based Optimization. Technical report, School of Aeronautics, Northwestern Polytechnical University, Xian, P.R. China, 2012.

[12] T. Homma and A. Saltelli. Importance Measures in Global Sensitivity Analysis of Nonlinear Models. *Reliability Engineering and System Safety*, 52(1):1 – 17, 1996.

[13] R. V. Joseph and Y. Hung. Orthogonal-Maxmin Latin Hypercube Designs. *Statistica Sinica*, 18:171–186, 2008.

[14] G. D. Krige. A statistical approach to some basic mine valuation problems on the Witwatersrand. *Journal of the Chem., Metal. and Mining Socity of South Africa*, 52:119–139, 1951.

[15] T. Krishnamurty. Comparison of Response Surface Construction Methods for Derivative Estimation Using Moving Least Squares, Kriging and Radial Basis Functions. In *46th AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics & Materials Conference*, 2005.

[16] P. Lancaster and K. Salkauskas. Surfaces Generated by Moving Least Squares Methods. *Mathematics of Computation*, 37:141–158, 1981.

[17] D. Levin. The approximation power of moving least squares. *Mathematics of Computation*, 67:1517–1531, 1998.

[18] J. Li, H. Wang, and N. H. Kim. Doubly weighted moving least squares and its application to structural reliability analysis. *Structural and Multidisciplinary Optimization*, 46:69–82, 2012.

[19] G. G. Maisuradze and D. L. Thompson. Interpolating Moving Least Squares Methods for Fitting Potential Energy Surfaces. *The Journal of pyhsical chemistry A*, 37:7118–7124, 2003.

[20] M. McKay, R. Beckman, and W. Conover. A Comparison of Three Methods for Selecting Values of Input Variables in the Analysis of Output from a Computer Code. *Technometrics*, 21:239–245, 1979.

[21] T. Most and J. Will. Metamodel of optimal prognosis - an automatic approach for variable reduction and optimal metamodel selection. In *Proceedings of the 5th Weimar Optimization and Stochastic Days*, Weimar, Germany, November 20-21, 2008. DYNARDO GmbH.

[22] D. I. Papadimitriou, I. C. Kampolis, and K. C. Giannakoglou. Stochastic and deterministic optimization for turbomachinery applications based on the adjoint formulation. In *Design Optimization International Conference*, 2004.

[23] M. Powell. An efficient method for finding the minimum of a function of several variables without calculating derivatives. *Computer Journal*, 7:155–162, 1964.

[24] P. Salemi, B. L. Nelson, and J. Staum. Moving Least Squares Regression for High Dimensional Simulation Metamodelling. In *Proceedings of 2012 Winter Simulation Conference*, 2012.

[25] A. Samad and K. Kwang-Yong. Surrogate based optimiztaion techniques for aerodynamic design of turbomachinery. *International Journal of Fluid Machinery and Systems*, 2:179–188, 2009.

[26] V. Schramm, J. Denecke, S. Kim, and S. Wittig. Shape Optimization of a Labyrinth Seal Applying the Simulated Annealing Method. *International Journal of Rotating Machinery*, 10:365–371, 2004.

[27] A. Soon-Jeong, Y. Jaechil, L. Byung-Gook, and L. Joon-Jae. 3D surface reconstruction from scattered data using moving least square method. In *Image Analysis and Processing (ICIAP)*, 2005.