E-ISSN 2581 – 7957
P-ISSN 2277 – 3916

CVR Journal of Science and Technology, Volume 17, December 2019
DOI: 10.32377/cvrjst1712

Hooke and Jeeves Pattern Search Method and Global Optimal Solution

Dr. M. Raghava[1], B. Rambabu[2], V. Dattatreya[3]
[1]Proffesor, CVR College of Engineering/CSE Department, Hyderabad, India
Email: raghava.m@cvr.ac.in
[2]Assoc. Professor, CVR College of Engineering/CSE Department, Hyderabad, India
Email: b.rambabu@cvr.ac.in
[3]Assoc. Professor, CVR College of Engineering/CSE Department, Hyderabad, India
Email: v.dattatreya@cvr.ac.in

Abstract: **In Data Science, it is imperative to build a model that learns the parameters from the data itself to solve either predictive or prescriptive problems while ensuring improved fidelity of the solution. In this article, we propose to model the non-stationary present in the data in terms of spatial anisotropic interpolation which encapsulates the trend as polynomial regression and characterizes the associated error field as a Gaussian noise process. The fundamental emphasis of the paper lies in the minimization of anisotropic error by learning the model parameters using Hooke and Jeeve's pattern search algorithm, a gradient-free pattern search algorithm and works even in missing value scenarios. The Design and Analysis of the Computer Experiments (DACE) based metaphor are developed and the quality of results are demonstrated on benchmark functions. The proposed implementation essentially has a wide range of applications in Computer Vision, weather prediction, Ore mining, etc.**

Index Terms: **DACE, kriging, anisotropy, interpolation, regression, pattern search.**

I. Introduction

Data Engineering is a branch of Computer Science which deals with understanding the underlying process that generates data and fits a scientific or statistical model through data, analyzes it further explores the hidden patterns and uses them to prescribe a valid set of rules to resolve high-level decisions. Thus, it has changed the fundamental way in which the real-world engineering problems are addressed. The critical part is to identify the associated physical phenomenon and its realization through either statistical or mathematical model building.

Model building involves two equally important major phases namely, Design and Analysis of Data Space and selection of parameter space and it's Fine-tuning. Firstly, a selected model is fit through the data, followed by its execution and evaluation. But in real-world problems, the data generated out of a process may not fit conveniently to a closed function necessitating the application of Machine Learning algorithms [1]. In such cases, engineers alternatively try to develop a robust model that learns the underlying physical phenomenon from the data itself while ensuring certain constraints leading to optimization problems. The constraints often are confined to some standard values while building the model. Most of the times, as the parameters are not properly learned from the nature of the data, the developed models suffer from failures.

Coming to the statistical model building, it begins with collection the data under observation, classify them into dependent and independent variables and develop a function to solve classification or prediction sort of problems at hand with a minimum possible error. This aspect is addressed by answering three functional questions that constitute the data analytics pipeline: What are the variables involved in the scenarios, what kind of relationship can be modeled between the variables and finally what the external parameters are affecting these relations. The model implementation essentially requires evolving a suitable algorithm to realize the solution. Further, the model shall be translated into a computational model with the capability to handle large amounts of data.

Technologies are evolving rapidly into complete ecosystems that help to deal with Big Data. The maturity of statistical models and the latest data visualization techniques constitute the heart of this ecosystem. For example, understanding data properties in terms of moments and trend clearly help in prescriptive and predictive analytics aspects. Model building considers the division of data space into two distinct but important parts; Data Space- deals with data itself, and Parameter Space- tuning the selected parameters that can improve the performance of the solution. Initially, it is carried out by assuming the set of parameters with standard values and generates the model by following a sequence of steps. Select a domain-specific model, implement and execute the model and evaluate the correctness of the model using benchmark test beds. Very often, the model generated fails to offer the desired results reflecting its poor capabilities in capturing the latent relationship between the variables. In that case, the designer often tends to discard the model itself and work on alternatives and evaluate them. However, such a naive approach to juggle with models and flip-flop the solution set-based approach is not a good practice as the span of models available in the literature is really vast. Hence, it is imperative to shift the focus to the parameter space and learning the optimal values of parameters to yield better results from the selected model.

A very good example to illustrate this philosophy is regression. In regression we try to fit a trend surface through the points at which the responses of the system are available with least mean square error [2]. This function is then used to predict the responses at unknown locations. The quality of the results is assessed through the various cross-

E-ISSN 2581 – 7957
P-ISSN 2277 – 3916

DOI: 10.32377/cvrjst1712

validation techniques and error analysis methods. In case of high variance in the error we opt for higher-order polynomials, B-Splines, etc., as the basis functions to represent the trend. However, by its very nature regression is very sensitive to the outliers in the data space [2] and a mere change of basis functions may not result in desired quality predictions. Hence it is essential to switch the focus onto controlling the physical properties of the underlying system by incorporating few parameters and translate them into efficient implementations. One of the simpler means to work out and manipulate these parameters is to employ typical search algorithms that arrive at optimal values of the model. Literature is enriched with plenty of algorithms to predict the optimal value of the parameters. Many of these algorithms analyze the data and learn the parameters from the data itself which is widely referred to as Machine Learning. One such algorithm is Hooke and Jeeves method [3][7] which is a derivative-free method that can optimally search the parameter space and suggest ideal values of the parameters involved in the model. In the upcoming sections, we present different regression trend models and influence of values of parameters and efficient algorithms to arrive at ideal values.

II. KRIGING

Regression analysis involves taking the locations also called design sites at which the responses are available in the form of an array with location-value pairs. As there is an overwhelming amount of data, it becomes difficult to take all the data present into consideration. To alleviate this problem, we try to visualize design sites and design the experiments. While modeling the data it is not always possible to fit a sound surface through the data. This is because there are points farther away from the fit that we call them as outliers and they tend to pull the fit towards them, thus introducing high variance in the error. This error, also known as noise and can manifest in two different types depending on how we characterize it. If we try to characterize the error without a model, then it is called white noise [2]. In contrast if the same is done based on a model-driven by the spatial information then the result is called a stationary noise [3][4].

In regression models, quite often the error is considered to be white noise and disregards the correlation among the design sights across all the statistical moments. Thus, the white noise is an example of *i.i.d.* of random variables. This assumption leads the regression model to become sensitive to outliers. Stationary noise [2] is discrete signal and is similar to white noise but it is a vector that also considers the direction of the noisy data. It depicts what is the underlying shape of the change in error.

As mentioned previously, the best example to realize a prediction model is regression. Generally, there are two types of regressions, namely linear and spatial regression. In linear regression, we encounter and work on white noise whereas in spatial regression we deal with stationary noise. In real-world scenarios, spatial regression is able to solve wide varieties of problems ranging from weather prediction, ore quality assessment, epidemics, etc.

Spatial regression [2][4] deals with stationary noise which is the noise in which change in error depends on the direction. This stationary noise can be classified into three types: zeroth-order noise, first-order noise, and second-order noise. In zeroth-order assumes the trend as a constant, and first-order noise fits a general polynomial across the design space and the error feature has zero mean and in second-order stationary noise case the trend is modeled as a piecewise continuous functions value of mean is zero and variance are finite.

The second-order noise is sub-classified into isotropic and anisotropic noise [4]. Isotropic noise, as assumed to be equally distributed throughout the design space in all directions and can be modeled with relatively simpler efforts. Anisotropic noise varies not only with regard to the lag, i.e. the distance but also with the direction of the target data point. Kriging [3][5] is an implementation of spatial regression which offers a metaphor for the physical process and helps to solve the data prediction problems. The basic kriging model is well explained by the following two steps [3].

A. Model Building

Let $S=\{s_1,s_2,\ldots s_m\}$, $s_i \in \mathbb{R}^2$ contain the design sites and $Y=\{y_1\ y_2\ ,\ldots,y_m\} \in \mathbb{R}$ are associated responses. Let the trend polynomial fit through the design sites is f(s). Let $F \in \mathbb{R}^2$ is a matrix defined by $F= f(s_i)^T$ and R is the matrix representing the spatial correlation among all the design sites. The closed-form of $f(s_i)$ can be a 2-D polynomial involving the polynomial basis functions and cross-terms also. For example, the second-order polynomial is expressed as $f(s_i)= 1+ x_i + yi + x_i^2 + y_i^2 + x_iy_i$. Then the Kriging model [2] which accommodates the error is expressed as

$$F\beta + \varepsilon = Y \qquad (1)$$

For which the generalized least squares solution [2] is,

$$\beta = (F^T R^{-1} F)^{-1} F^T R^{-1} Y \qquad (2)$$

The corresponding variance estimate of the model is

$$\Sigma^2 = (Y - F\beta)^T R^{-1} (Y - F\beta)/m. \qquad (3)$$

Thus, we can observe β and σ^2 depend upon the correlation model.

B. Develop the Predictor.

The Kriging estimator at unknown design site x is given by

$$\hat{y}(x) = f(x)^T \beta + r^T R^{-1} (Y - F\beta). \qquad (5)$$

Here r stands for the vector representing the correlation between the target site and all the design sites. The estimated mean squared error is finite and depends only on the correlation kernel. We can gain control over the error model by imposing a second-order stationary field. And the stationary property is well defined through designing a spatial correlation model. The literature is enriched with the correlation models as listed in [3]. In the present study we are considering only exponential kernel which is expressed as

E-ISSN 2581 – 7957
P-ISSN 2277 – 3916

DOI: 10.32377/cvrjst1712

$$R(\Theta, d) = \exp(-\Theta|d|) \quad (6)$$

It is evident from the equation specified above the spatial correlation exponentially decreases with the lag(d) between the design sites and its behavior is further governed by the parameter Θ. If the components of the vector Θ are equal, then the spatial correlation is an instance of isotropic phenomenon whereas varying values of Θ refer to an anisotropy property. The significance of the vector Θ is it defines the shape of the error field either as a circle or an ellipse. Thus, theta influences the spatial correlation model and hence it is very essential to learn the theta value empirically from the data itself.

III. Hooke and Jeeves method and mult agent implementatio n

To optimally determine the value of Θ literature offers many algorithms such as convergence, gold-section search, Nelder-mead search, Luus–Jaakola[7] search, etc. In this work we have opted to use Hooke and Jeeves method for optimized pattern searching as it is a pattern search-based algorithm. The simplicity of this method lies in its exploratory capabilities and fast convergence. This method is a numerical procedure that is free from computation of gradients and thus avoids the operator selection policy from central, forward and backward differences. Further, if the objective function is not expressed in a closed-form but only the experimental responses are available then pattern search based algorithms are best suited to solve the unconstrained non-linear optimization problems [6]. The quality of the solution of the pattern search algorithm is dependent upon the heuristic rule selection, potentially a hybrid version and its implementation.

Hooke and Jeeves pattern search method [4] solves the optimization problem by generating the state space in explorative manner and proceeds to new state so long as the solution gets optimized. Otherwise, the algorithm retracts to the old state from the new state and proceeds in a different direction. The number of directions in which the exploration happens also can be configured. For example, in 2-D case the exploration directions can be either 4 or 8. In the case of higher-dimensional space exploration takes place sequentially along different directions.

While exploring along i^{th} dimension the optimal values along all the preceding dimensions are frozen. Once a local optimal has arrived then the algorithm anneals to new location, which we refer to as Pattern Move, along the direction in which partial optimal solution is realized. These two steps are repeated until optimal solution is realized. On the other hand, if the new pattern fails to generate further improved solution the solution backtracks to old state and explores for the optimal solution in a different direction that realizes the complete span of the search space. The algorithm strives to reach an optimal location from a random location in two types of moves Exploratory and Pattern Move as discussed below

1. Exploratory Search: A local move to seek an optimal solution. Given the current location X_c, one of the features of the location is perturbed in forward and backward directions and the feature is updated with the new value at which function evaluates to the maximum. Such a similar procedure is repeated with reference to each pattern, one at a time and new best location is configured. We characterize the exploratory move as a success if the new location is different from the initial location otherwise the move stands for failure. The outcome of the experiment is the new location.

Let X_c is the current state of the solution. Assume that the i^{th} component of X_c is perturbed by p while retaining the values of other features. Algorithm 1 lists out the behavior of Explore step for each feature X_i of the solution vector X_c. The function is evaluated at locations X_i, X_{i+} 1, and X_{i-} 1 and updates the component with the location at which the minimum value of the function is achieved. This process is repeated for all components of X_c. The algorithm returns success if new location X_n with optimal value of f is reached otherwise the Exploration returns a failure.

2. Pattern Move: Upon success from Exploration step the pattern move is taken with a leap in the direction of line joining $X_c$with X_n

A. Algorithm 1:

// Initialization $X_n := X_c$

Step 1: Calculate f:= f ($X_{c,i}$), the forward value f+ := f ($X_{c,i}$ + 1) and backward value f- := f ($X_{c,i}$, - 1).

Step 2: Find fmin := min (f, f+, f-).
Set $X_{n,i}$ that corresponds to f_{min}.

Step 3: Check i == N? If no, set i := i + 1 and go to Step 1;
Else X_n is the result and goto Step 4.

Step 4: If X_n != X_c, success; Else failure.

Step 5: Make pattern move
$X_p = X_n + \alpha(X_n - X_c)$

Step 6: Evaluate f at X_p. If it is evaluated to be better then control is transferred to Step 1. Otherwise, reduce the step size i.e. the value of α in Step 5.

Thus Hooke-Jeeves method switches between two steps Exploratory Search and Pattern Move. In the first step local optimal value is located in the vicinity and the latter step takes the solution to a new pattern in the direction of successful exploration with a long leap and repeats the Exploratory step. If the exploration at the new pattern is in vain then the quantum of leap is reduced.

In this paper, Hooke and Jeeves method is implemented to feature a parallel search mechanism by dividing the search space into disjoint regions and multiple parallel exploring agents are launched across these regions. Each agent is perceived as a random initial location in the given sub-region. The solutions generated by the exploration followed by pattern move from different agents are compared and the best solution is adapted. This parallel and divide and conquer strategy helps the empirical model to realize the global optimal solution and avoids the local optimal solution.

E-ISSN 2581 – 7957
P-ISSN 2277 – 3916

DOI: 10.32377/cvrjst1712

In the current work, a User Interface (UI)[7] had been designed by implementing a singleton pattern. The use-cases include actor being able to select a benchmark non-linear function, correlation kernel, and the number of orientations that are required by the program. Each input field is labeled with the corresponding name so that the user will find it easy to type or select the input. The other set of fields that are required are the x and y coordinates that refer to initial location of each agent, delta value, epsilon value, number of rotations and number of iterations.

The following experiments demonstrate the results of the proposed model on benchmark mathematical functions that are available in the public domain of the internet.

B. Experimental Results:

Function: Sum Square
EQUATION: $(x - 2)^2 + (y - 2)^2$ (7)
X - Value: 11
Y - Value: 13
EPSILON: 0.01
DELTA: 0.5
ITERATIONS: 150

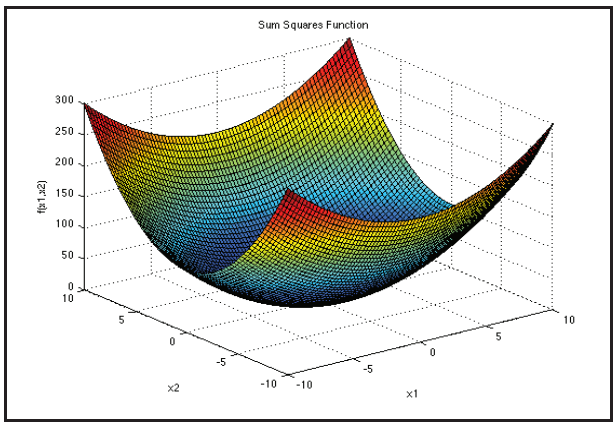

Figure1. 3-D plot

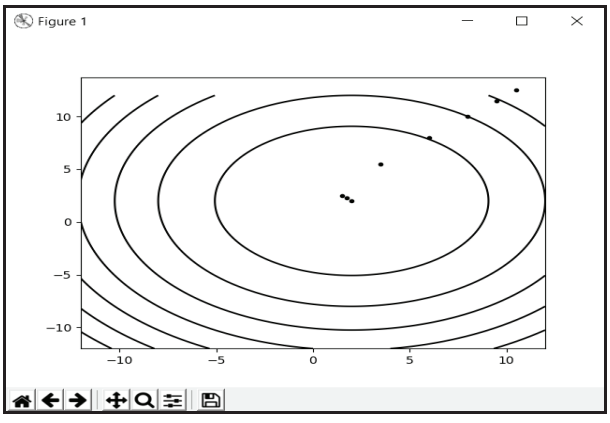

Figure 2. Contour Map

The 3D representation of the sum square function depicts the span of the function over the data space in Figure1, Figure2 presents the Contour Map to demonstrate the convergence of the algorithm in the form of traces of the solution and the benchmark function is presented in contour form. It clearly demonstrates that the solution which is started at a random location is eventually reaching the optimal value of the function by making the pattern moves. A similar analogy is applied to the subsequent benchmark function evaluations shown in Figure 4 and Figure 6.

Function: Matyas
EQUATION: $0.26(x^2 + y^2) - 0.48xy$
X - Value: 12
Y - Value: 14
EPSILON: 0.0001
DELTA: 0.5
ITERATIONS: 100

The 3D representation of the Matyas function is depicted in Figure 3 along with span of the function over the data space.

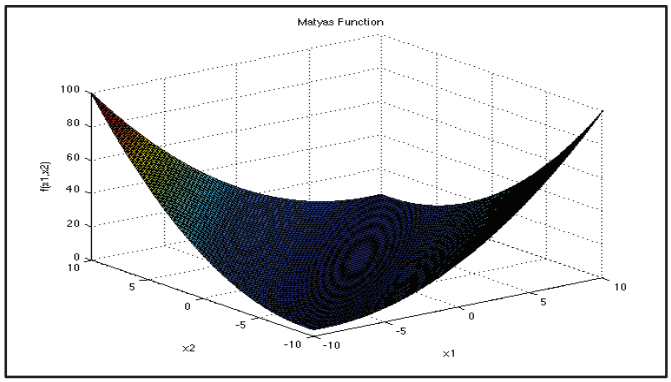

Figure 3. 3-D plot

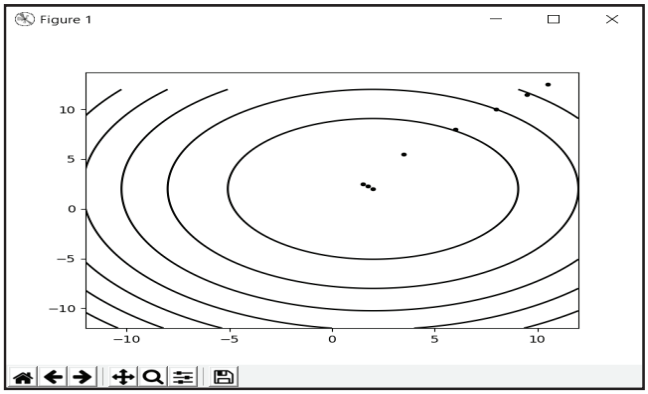

Figure 4. Contour Map.

Function: Beale
EQUATION:
$(1.5 - x + xy)^2 + (2.25 - x + xy^2)^2 + (2.625 - x + xy^3)^2$
X - Value: 5
Y - Value: 5
DELTA: 0.5
EPSILON: 0.01
ITERATIONS: 200

E-ISSN 2581 – 7957
P-ISSN 2277 – 3916

DOI: 10.32377/cvrjst1712

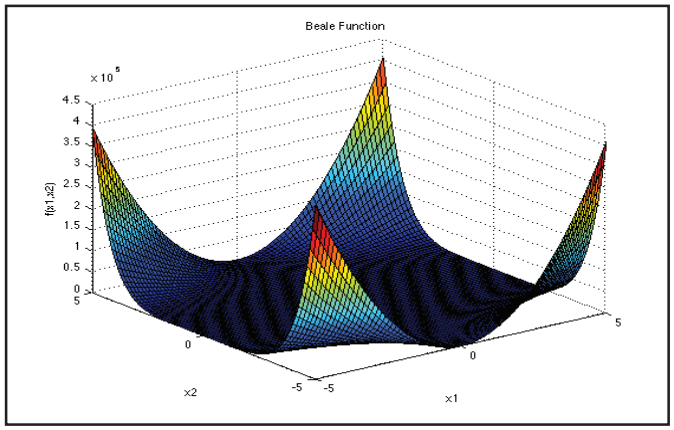

Figure 5. **3-D plot**

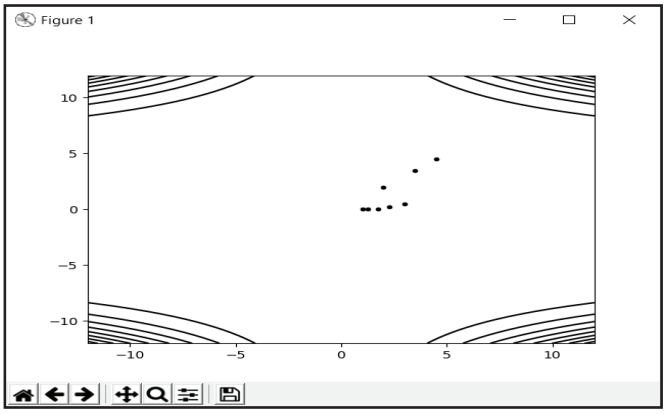

Figure 6. Contour Map.

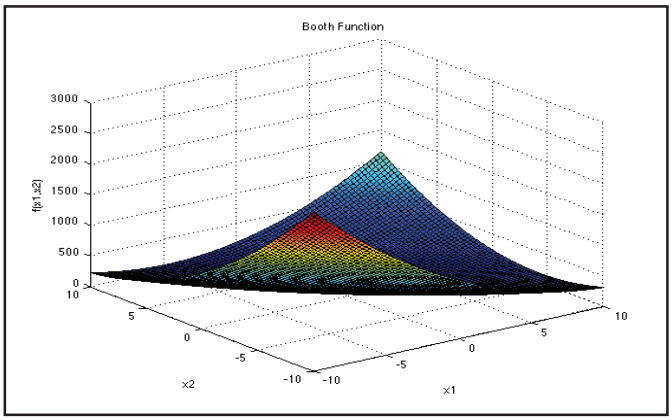

Figure 7. 3-D plot

The 3D representation of the Beale function is depicted in Figure 5. The function spans over the data space and has multiple optimal values.

In this testing process we choose a Beale function with starting coordinates as (5, 5) and having delta value 0.5 with epsilon value 0.01 with value check in eight directions.

Function: Booth

EQUATION: $(x + 2y - 7)^2 + (2x - y - 5)^2$

X - Value: 5

Y - Value: 5

DELTA: 0.5

EPSILON: 0.01

Figure 7 presents a 3D representation of the booth function and depicts the span of the function over the data space.

In this testing process, we choose a booth function with starting coordinates as (7,8) and having delta value 0.5 with epsilon value 0.001 with value check in four directions.

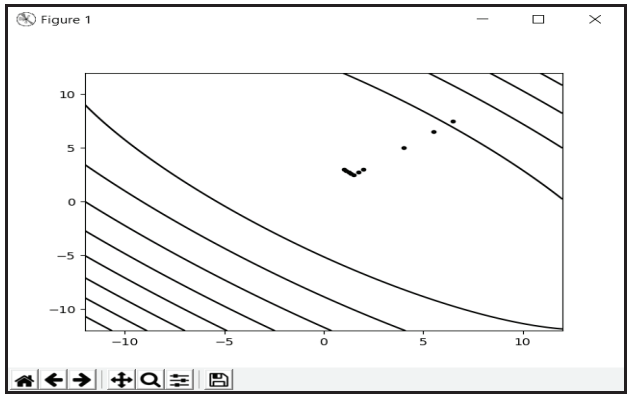

Figure 8. Contour Map.

TABLE I.
SHOWING THE CONVERGENCE OF THE PROPOSED ALGORITHM

Function	Initial Solution	Delta, Epsilon	Final Solution	Expected Solution
Circle	(11,13)	0.5, 0.01	(2,2)	(2,2)
Matyas	(12,14)	0.5, 0.0001	(0,0)	(0,0)
Beale	(5,5)	0.5,0.01	(3.0, 0.5)	(3.0,0.5)
Booth	(7,8)	0.5,0.001	(1.0, 3.0)	(1.0, 3.0)

Table 1 presents a summary of the results validating the performance of the algorithm on the benchmark functions.

IV. CONCLUSIONS OF THE PRESENT STUDY

This paper successfully validated optimization results for various benchmark functions. The analytically calculated optimum solution is best matched with closed-form solution. The paper also, demonstrated the parallel implementation of Hooke and Jeeves method that avoided the local optimal problem. The percentage error in analytical and python results depend on the number of iteration steps, length of data sets, objective function and constrained for optimization.

REFERENCES

[1]. Bishop, C. M. Pattern Recognition and Machine Learning, Springer, ISBN 978-0-387-31073-2, 2006.

[2]. Gentile, M., Frederic Courbin and Georges Meylan. "Interpolating point spread function anisotropy." Astronomy & Astrophysics manuscript, 2013.

[3]. Søren N. Lophaven, Hans Bruun Nielsen, Jacob Søndergaard, "*Correlation Models",* Aspects of the Matlab Tool DACE,Technical University of Denmark,DK-2800 Kongens Lyngby – Denmark, pg 10-39

[4]. Raghava, M., Arun, Agarwal., Raghavendra, Rao C. A Scalable Spatial Anisotropic Interpolation Approach for Object Removal from Images using Elastic Net

Regularization, MIWAI 2016, Thailand, LNCS, Vol 10053,126-140, 2016. (ISBN: 978-3-319-49396-1)

[5]. Couckuyt, I. and Forrester, A. and Gorissen, D. and De Turck, F. and Dhaene, T. Blind Kriging: Implementation and Performance Analysis, Adv. Eng. Softw.1-13, ISSN 0965-9978, 2012.

[6]. L. Armijo, Minimization of functions having Lipschitz continuous first partial derivatives, Pacic Journal of Mathematics, 16, pp. 1-3. 1966.

[7]. Hooke R & Jeeves T A. "Direct search" solution of numerical and statistical problems. J. Ass. Comput. Mach. 8:212-29, 1961.

[8]. Mark Summerfield, Rapid GUI programming with python and QT, 269 – 283, Prentice hall, ISBN-10: 0134393333, 2009.