11th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference, 6-8 September 2006

Comparison of Three Surrogate Modeling Techniques: Datascape®, Kriging, and Second Order Regression

Shawn E. Gano* Harold Kim † and Don E. Brown II ‡

Lockheed Martin - Integrated Systems and Solutions, Valley Forge, Pennsylvania, USA

Using surrogate models in place of high fidelity engineering simulations can help reduce design cycle times and cost by enabling rapid analysis of alternative designs. Surrogate models can also be used in a deliverable product as an efficient replacement for large lookup tables or as a soft sensor to predict quantities than cannot be directly measured. Many different surrogate modeling techniques exist, including new commercial technologies, each with different capabilities and pitfalls. The goal of this research is to aid the designer in selecting the appropriate surrogate model by comparing two popular techniques, second order regression and kriging, along with a new commercial application called Datascape. The three different modeling techniques are compared on model accuracy, computational efficiency, robustness, transparency, and ease of use. The comparisons were done using three test problems: an Earth-Mars transfer orbit problem, the analytic Shekel function, and a low Earth orbit three-satellite constellation design problem. It was found that kriging models performed the best when the sample data used to build the models was sparse, when larger sample sets were used Datascape produced more accurate models.

I. Introduction

In competitive, technically challenging environments, surrogate models can reduce program cost and increase the efficiency of the design process. These approximation models can be applied in different phases and aspects of the engineering design process. Design optimization can use surrogate models when the computational expense of high fidelity simulations is prohibitive or when limited physical experiments can be conducted. Executable architecture simulations can also leverage surrogate models to provide rapid feedback concerning system architecture tradeoffs. A deliverable product, such as an embedded system or soft sensor, can even use these models to replace large lookup tables or complicated calculations. Another non-standard way in which surrogate models can be used is to reduce software licensing costs by capturing the response of expensive simulation tools for use by other engineers and decision makers in the preliminary design phases.

Many different surrogate modeling techniques have been developed for, or applied to, multidisciplinary analysis and optimization. Probably the most widely used methods are first and second order polynomial regression techniques.[1–5] Regression models are created by performing a least squares fit of data to a linear or quadratic function. Such methods are relatively computationally efficient and produce a unimodal smooth fit to data that may contain random errors found in physical or stochastic experiments. However, this type of model requires the assertion that the data being modeled behaves linearly or quadratically. There is also some debate as to whether or not these methods should be applied to deterministic computer experimental data.[6] To address this debate Sacks *et al.*,[7] proposed an interpolation technique called design and analysis of computer experiments (DACE) to interpolate data from deterministic computer simulations. These interpolation models were based on work, known as kriging, done in the spatial statistics and geostatics. Kriging models have the ability to model multi-modal data but generally require more computational power

*Ph.D., Senior Systems Engineer, Modeling and Analysis Group, Operations Command and Control, e-mail: shawn.gano@lmco.com, AIAA Member.

†Systems Engineer, Enterprise Architecture Integration Engineering, e-mail: harold.kim@lmco.com.

‡Senior Systems Engineer, Enterprise Architecture Integration Engineering, e-mail: don.e.brown.ii@lmco.com.

to develop. Kriging models have been gaining in popularity in recent years for use in engineering design.[8–13] They have been used to model the response of many engineering systems, including the design of a low-boom business jet by Chung and Alonso[14] and the design of an aerospike nozzle by Simpson *et al.*[15] Martin and Simpson conducted a study on using kriging models to approximate deterministic computer models and discussed the applicability of various kriging variants.[16,17]

Commercial off-the-shelf (COTS) surrogate modeling software has also begun to emerge as a viable approach for industrial use. A COTS solution has two main advantages for a company: they are available for immediate use with no development and testing costs and for the support that is traditionally supplied along with the software. Datascape® is one such COTS tool that has recently been released. The surrogate modeling method used by the Datascape tool was originally developed in the 1990's at Lockheed Martin. Third Millennium Products (TMP) won a bid for exclusive rights to develop and license Lockheed Martin's unique modeling method in December of 2002. The method has been used on many projects within Lockheed Martin but no formal comparison of the method to other surrogate modeling techniques such as second order regression or kriging has been published. Datascape has the ability, like the regression techniques, to handle data from stochastic experiments, including multiple evaluations at the same trial point, without any assumptions as to the form of the model; furthermore, it can produce multi-modal approximations.

Previous studies have compared kriging and second order regression models. Giunta and Watson[10] did an initial exploration of the accuracy and modeling capabilities of kriging and second order regression models using a mathematical test problem that included high and low amplitude sinusoidal functions to simulate numerical noise often found in realistic engineering design spaces. Chung and Alonso[18] compared the accuracy of the same two methods for use in approximating the objective function in design optimization and concluded that both methods are practically applicable to realistic design optimization, using a supersonic business jet as a test case. Jin *et al.*[19] did a comparative study of second order regression, kriging, multivariate adaptive regression splines (MARS), and radial basis functions for mainly analytic test functions. In this paper kriging and second order regression methods are compared along with the COTS tool Datascape, using a two engineering and one mathematical test problem. The comparison not only includes the accuracy of the three methods but also the computational time required for constructing the different surrogate models, the applicability, and their ease of use.

In this paper an overview of each of the three surrogate modeling methods is given. Section III describes the different statistics and criteria used to compare the three techniques. Then three test problems are introduced and their respective results are discussed in Section IV. The test problems include modeling Earth-Mars transfer orbits, a standard analytic challenge function, and a three-satellite constellation. Finally, section VI provides conclusions and recommendations for future work.

II. Surrogate Modeling Techniques

A. Datascape®

Datascape is a COTS surrogate modeling application. The methodology used to create the surrogate models is proprietary; however, according to the user's manual, it uses features from fuzzy logic, non-linear regression and numerical optimization and presents them in a hybrid format.[20] Datascape was developed by Third Millennium Productions, Incorporated, which was founded by two Lockheed Martin engineers in 1990 and was originally given the rights to commercialize Lockheed Martins finite element post-processing technology. They were subsequently approached to commercialize and develop Lockheed Martin Aeronautical Company's unique modeling method in 2002. It was from this method that Datascape was developed. Datascape's models are based on a small number of coefficients, which allow users to update only the coefficients, rather than replace the entire embedded model. Datascape uses an iterative learning process to build a model from the data points. The graphical user interface allows for dynamic visualization of the surrogate model via a 3D surface view which can be used during the training processes. Appendix A shows screenshots of the application. The application also allows users to observe the evolution of the model in the form of statistical information and 2D correlation plots[a]. Datascape models have been used on development and operations of the F-16, which saved an estimated $36 million.

Currently Datascape supports building a surrogate model with up to 10 design variables at a time (as of version 1.9k); this, however, is to be increased in future versions. The application can handle over a million training points on a standard PC. Creating a surrogate model using Datascape is a primarily straightforward and intuitive process. The user must select the number of points on the influence functions (in the range of 4-180, default of 20), which are essentially functions used to characterize each dimension of the design space. The higher the number of points on the influence functions, the greater the flexibility Datascape has in fitting the data; however with this higher flexibility, the chance of capturing noise in the model becomes greater and generally requires more CPU time for convergence. A choice of using a smaller number of points on the influence function results in model that captures the general regressive trend of the data. Different problems require different selections of this parameter, and in this work a few preliminary runs were preformed for each test problem to determine a suitable number. The modeling learning process can also be stopped at any point in time and still produce a usable model. By default, Datascape uses a squared error fit function to evaluate the quality of its model using a training data set. Then it iteratively refines the model to minimize the error. The resulting surrogate models can be evaluated in the user interface, a C++ API, or with an Excel plug-in. Another result of creating a surrogate model using Datascape is that the method produces a ranking of all the design variables so the designer can understand the relative importance of each variable over the entire design space.

B. Kriging

The second surrogate modeling technique compared in this paper is kriging. Kriging presupposes that the true unknown function can be modeled as a combination of a fixed and known trend function, $B(\mathbf{x})$, and a departure from that function that is a Gaussian random function, $Z(\mathbf{x})$, with mean of zero and non-zero variance σ^2. Therefore, the unknown function is expressed as,

$$\hat{y}(\mathbf{x}) = B(\mathbf{x}) + Z(\mathbf{x}). \tag{1}$$

This unknown function is then estimated using a set of sample points, $\mathbf{y}$, which have been evaluated using the truth function or the original simulation. The kriging approximation is typically formulated as follows:[21–23]

$$\hat{y} = \mathbf{f}\hat{B} + \mathbf{r}^T(\mathbf{x})\mathbf{R}^{-1}(\mathbf{y} - \mathbf{f}\hat{B}), \tag{2}$$

where $\mathbf{f}$ is a constant vector typically of all ones,[17] $\mathbf{R}$ is the correlation matrix between the sample points, and $\mathbf{r}$ is the correlation vector between any new point and the original sample points. A variety of correlation functions can be used.[24] The one used in this investigation, and most frequently in engineering applications, is the Gaussian function,

$$R(x^i, x^j) = e^{-\sum_{k=1}^{n_v} \theta_k |x_k^i - x_k^j|^2}, \tag{3}$$

proposed by Sacks[7] which includes a spacial correlation parameter vector θ. The spacial correlation parameters are found by solving an optimization problem using maximum likelihood estimation,[10,15] which can be computationally intensive.[25]

Kriging is a very flexible method due to the various correlation functions that can be used both as a regressive or an interpolative model based on the choice of $\mathbf{f}$. The main drawbacks of the method are: if any of the sample points used to construct the model are too close, the correlation matrix can become singular and large amounts of computer memory and CPU time are needed for constructing models with high dimensionality or large sample sets. In this work the DACE MATLAB toolbox is used to construct the kriging surrogate models.[26]

[a]Datascape website: http://www.tmpinc.com/datascape_overview.html

C. Second Order Regression

The third and final surrogate modeling technique compared in this research is second order regression (SOR). The general equation that describes a second order polynomial model is of the general form:

$$\hat{y} = c_0 + \sum_{j=1}^{n_v} c_j x_j + \sum_{j=1}^{n_v} \sum_{k=j}^{n_v} c_{n_v - 1 + j + k} \; x_j \; x_k, \tag{4}$$

where n_v is the number of variables. Or in matrix notation

$$\hat{y} = \mathbf{c}^T \bar{\mathbf{x}}. \tag{5}$$

where

$$\bar{\mathbf{x}} = [1, x_1, x_2, \ldots, x_1^2, x_1 x_2, x_1 x_3, \ldots, x_{n_v}^2]. \tag{6}$$

To construct the second order model, the coefficients, $\mathbf{c}$, must be solved for using a set of data points sampled throughout the design space. This set of points are stored in a matrix of the form

$$A = \begin{pmatrix} 1 & x_1^{(1)} & x_2^{(1)} & \ldots & (x_{n_v}^{(1)})^2 \\ . & . & . & \cdots & . \\ . & . & . & \cdots & . \\ 1 & x_1^{(n)} & x_2^{(n)} & \ldots & (x_{n_v}^{(n)})^2 \end{pmatrix} \epsilon \; \mathbb{R}^{n \times m}, \tag{7}$$

where n is the number of sampled data points and $m = (n_v + 1)(n_v + 2)/2$. Each of the points are evaluated using the truth model,

$$\mathbf{y} = [y^{(1)}, \ldots, y^{(n)}]^T \; \epsilon \; \mathbb{R}^n. \tag{8}$$

The sampled data set stored in this polynomial format yields a linear set of equations,

$$\mathbf{Ac} = \mathbf{y}, \tag{9}$$

with the SOR coefficients as the only unknowns. The set of equations has a unique least squares solution,

$$\mathbf{c} = (A^T A)^{-1} A^T \mathbf{y}, \tag{10}$$

when the inverse, $(A^T A)^{-1}$, exists. Note that the solution requires $n \geq m$, and if $n > m$ then the system is overdetermined and the model is regressive.

Sevreal of the major benefits of using SOR as a surrogate model are that it is a quite computationally efficient method, it can smooth out noisy functions, and the resulting model coefficients provide a clear between the design variables. The main drawbacks of the method are that it cannot model multimodal or non-symmetric design spaces. For this work, the SOR method has been implemented in MATLAB.

III. Model Comparison Measures and Data Sampling

The three different surrogate modeling techniques described in the previous section were compared using multiple performance measures. The following five aspects, which Jin *et. al*[19] proposed, were used to compare the methods:

Accuracy: the ability of the surrogate model in predicting the response of the truth function over the design space.

Robustness: the ability of the surrogate model to accurately predict the response of the truth function for different problem sizes and types.

Efficiency: the computational time required to construct the surrogate model.

Transparency: the ability of a surrogate modeling method to provide information about the contributions of different design variables and the interactions among different variables.

Conceptual Simplicity: ease of implementation or use of the surrogate modeling technique.

In order to compare the accuracy, a validation set of data points was evaluated using the different surrogate models and their predictions were contrasted with the true response. Three statistics, Equations (11)-(13), were used to measure the accuracy using the validation set errors.[19]
R Squared:

$$R^2 = 1 - \frac{\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}{\sum_{i=1}^{n}(y_i - \bar{y}_i)^2} = 1 - \frac{MSE}{Variance}, \tag{11}$$

where $\bar{y}_i$ is the mean of the truth function responses. The larger the value of R^2 the more accurate the surrogate model is; this is the main measure of accuracy.

Average Absolute Error (AAE):

$$AAE = \frac{\sum_{i=1}^{n}|y_i - \hat{y}_i|}{n}, \tag{12}$$

is another measure of accuracy. AAE is highly correlated with the MSE and, therefore, with R^2. The lower the value of AAE the more accurate the model.

Maximum Absolute Error (MAE):

$$MAE = \max(|y_1 - \hat{y}_1|, |y_2 - \hat{y}_2|, ..., |y_n - \hat{y}_n|). \tag{13}$$

MAE is generally not correlated with R^2 or AAE. A small value of MAE is preferred, while a large value could indicate that the model inaccurate in a region or regions of the design space depending on the values of the first two accuracy measures.

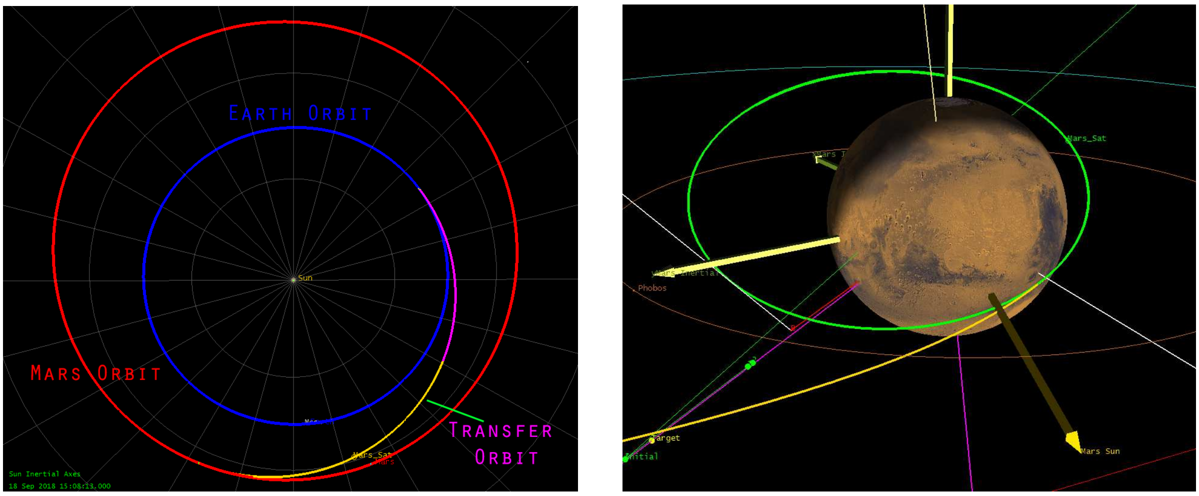
All of the sample sets for the three different test problems were done using Latin Hypercubes[27] and the same data set was used for each of the different surrogate modeling techniques. For each of the test problems an array of different sized sample sets was used to compare the robustness and efficiency of the surrogate models for sparse to dense data sets. The validation set, used to calculate the three aforementioned accuracy statistics, consisted of a much larger sample size than the largest data set used used to create the surrogate models. For the Mars Transfer Orbit Design 1,000 validation points were used, and for the Shekel and satellite constellation design problems 20,000 data points were used.

IV. Test Problem Descriptions and Results

In this section three test problems are described and their respective results are discussed. For each problem a set of data points was evaluated using different numbers of sample points. Then for each of the different sized sample sets the different surrogate modeling techniques were used to predict the response for a larger validation set. For each trial the accuracy statistics R^2, AAE, and MAE were calculated along with the time it took each surrogate modeling technique to build the surrogate model. The four statistics are then plotted to allow for easier analysis of the results; the raw data for each of the three test problems can be found in Appendices B-D.

A. Earth-Mars Transfer Orbit Design

In this problem a surrogate model is desired to predict the total change in velocity, ΔV, to transfer a spacecraft from low Earth orbit (LEO) to an orbit around Mars with an inclination of 25^o, eccentricity of 0.4, and an altitude of periapsis of 250km (see Figure 1). We wish to model the total ΔV in order to predict the mass of fuel required for the mission. The two input parameters considered in this model are the launch date between 1 Jan to 1 Oct 2018 of the satellite from LEO at an parking altitude of 407 km and the time



(a) Earth-Mars transfer orbit (b) Mars capture orbit and B-plane geometry

Figure 1. The Earth-Mars transfer and capture orbit geometries.

of flight (TOF) between Earth and Mars. The calculation of ΔV includes the initial departure maneuver, a mid-course correction, and the Mars orbit insertion burn.

This problem was modeled using Satellite Toolkit (STK) using the Astrogator propagator developed by Analytic Graphics, Inc. in cooperation with the NASA Goddard Space Flight Center (GSFC) Flight Dynamics Analysis Branch (FDAB). Astrogator is a commercialized and updated version of Swingby[28] simulator developed by NASA and was used in analyzing and planning maneuvers operationally for Clementine,[29] Lunar Prospector,[30] SOHO,[31] and others.[32]

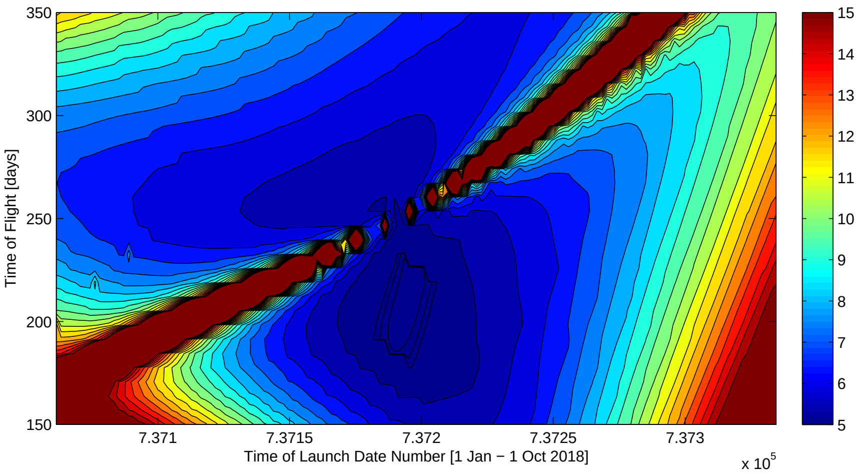


Figure 2. Contour plot of the total ΔV required to send a satellite from Earth Orbit to a Mars capture orbit as a function of launch date and time of flight.

The design space for this problem is very nonlinear and does contain some noise. A contour plot of the total ΔV is shown in Figure 2. Six different sized sample sets were used for this test problem, $n = \{10, 25, 50, 100, 200, 500\}$.

1. *Earth-Mars Transfer Orbit Results*

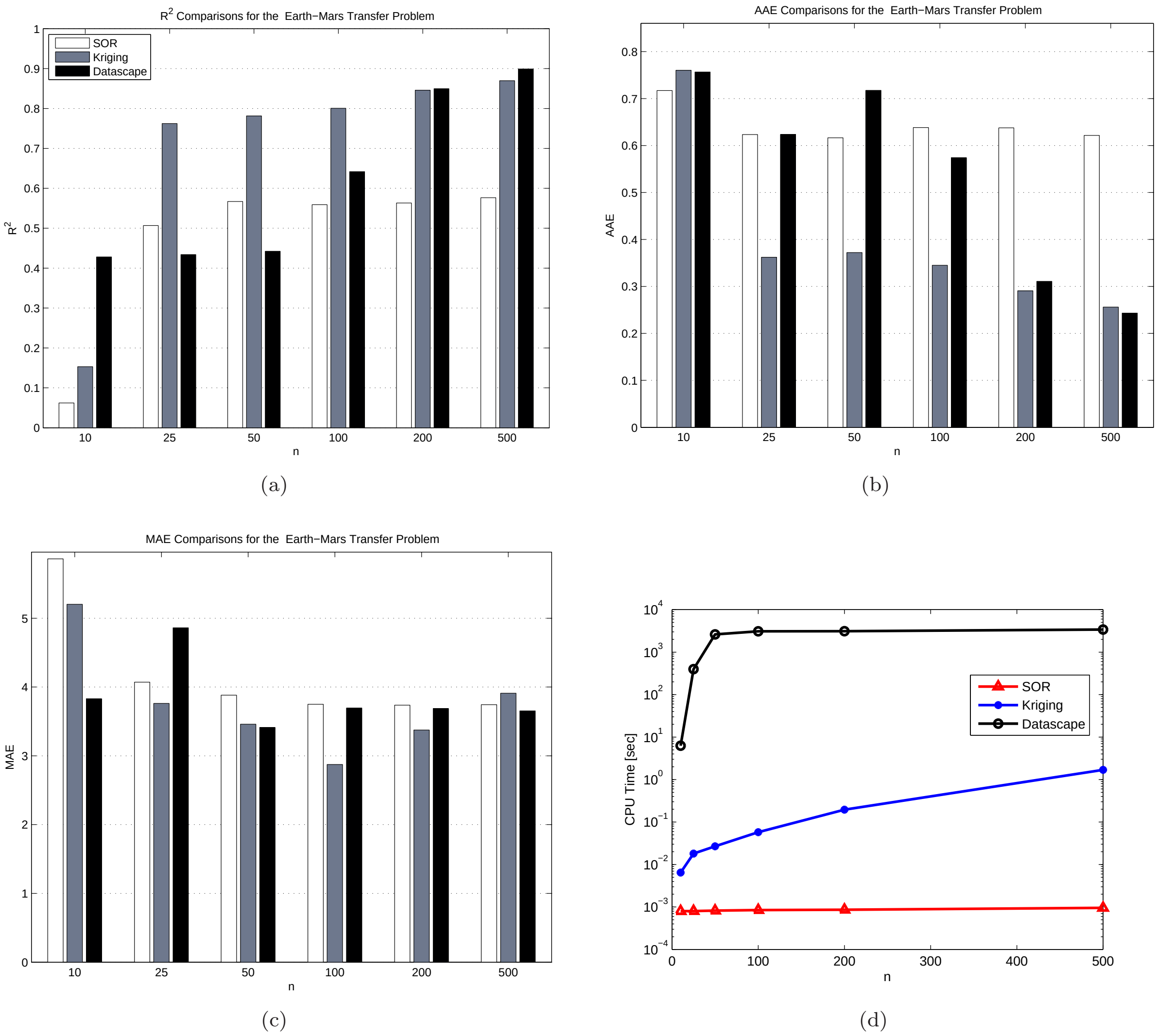


Figure 3. Accuracy and efficiency results for the Earth-Mars transfer orbit problem.

Using the six different data sets of the Earth-Mars transfer orbit problem, the three different surrogate models produced the results shown in Figure 3. Sub-figures (a)-(c) are bar plots for the R^2, AAE, and MAE accuracy statistics and (d) is a semi-log plot of the CPU time required to construct the different surrogate models on a PC with a 2.0 GHz Intel Pentium M processor and 1.0 gigabytes of RAM. The same PC was used for all the results presented in this paper. A tabularized version of the results from this problem can be found in Appendix B.

A comparison of the accuracy metrics in Figure 3 reveals many interesting trends. Datascape seems to have a lower R^2 value for sparse data sets, excluding the $n = 10$ case, than both SOR and kriging, and as the sample sizes increase the Datascape models becomes more accurate, and in fact, produce the most accurate results for the largest data set. Kriging, on the other hand, is the most accurate for the $n = 25$ trial by a considerable margin, but then improves at a much lower rate as the sample size increases. The SOR method seems to quickly converge to a steady state R^2 value but is outperformed by both kriging and Datascape at moderate samples sizes.

The AAE results show a similar trend, in which the kriging models perform well at lower sample sizes and Datascape improves faster as the samples sizes increase. The maximum absolute errors of all three methods had similar results overall, SOR being slightly worse than the other two methods.

Computationally, SOR was by far the most efficient method; it was orders of magnitude faster than kriging, which in turn was orders of magnitude faster than Datascape. Kriging seemed to show the highest rate of increase in CPU power required for each increase in sample size; though the $n = 500$ trial only took 1.7 seconds while Datascape took almost an hour.

B. Shekel Function

The second test problem is an analytic function named after its inventor J. Shekel.[33] This function is generally used in testing global optimization algorithms due to its non-linearity and many local optima. In this work we are using the Shekel function variant that has design variables, $n_v = 4$, and each variable is restricted in range between 2.0 and 6.0. The Shekel function is:

$$f(\mathbf{x}) = -\sum_{i=1}^{5} \frac{1}{(\mathbf{x} - A_i)(\mathbf{x} - A_i)^T + c_i}, \tag{14}$$

where

$$A = \begin{bmatrix} 4 & 4 & 4 & 4 \\ 1 & 1 & 1 & 1 \\ 8 & 8 & 8 & 8 \\ 6 & 6 & 6 & 6 \\ 3 & 7 & 3 & 7 \end{bmatrix} \tag{15}$$

$$c = [0.1\ 0.2\ 0.2\ 0.4\ 0.4]^T. \tag{16}$$

For this problem nine different sized sample sets were used, $n = \{25, 50, 100, 250, 500, 1000, 2500, 5000, 10000\}$, in comparing the different surrogate modeling methods.

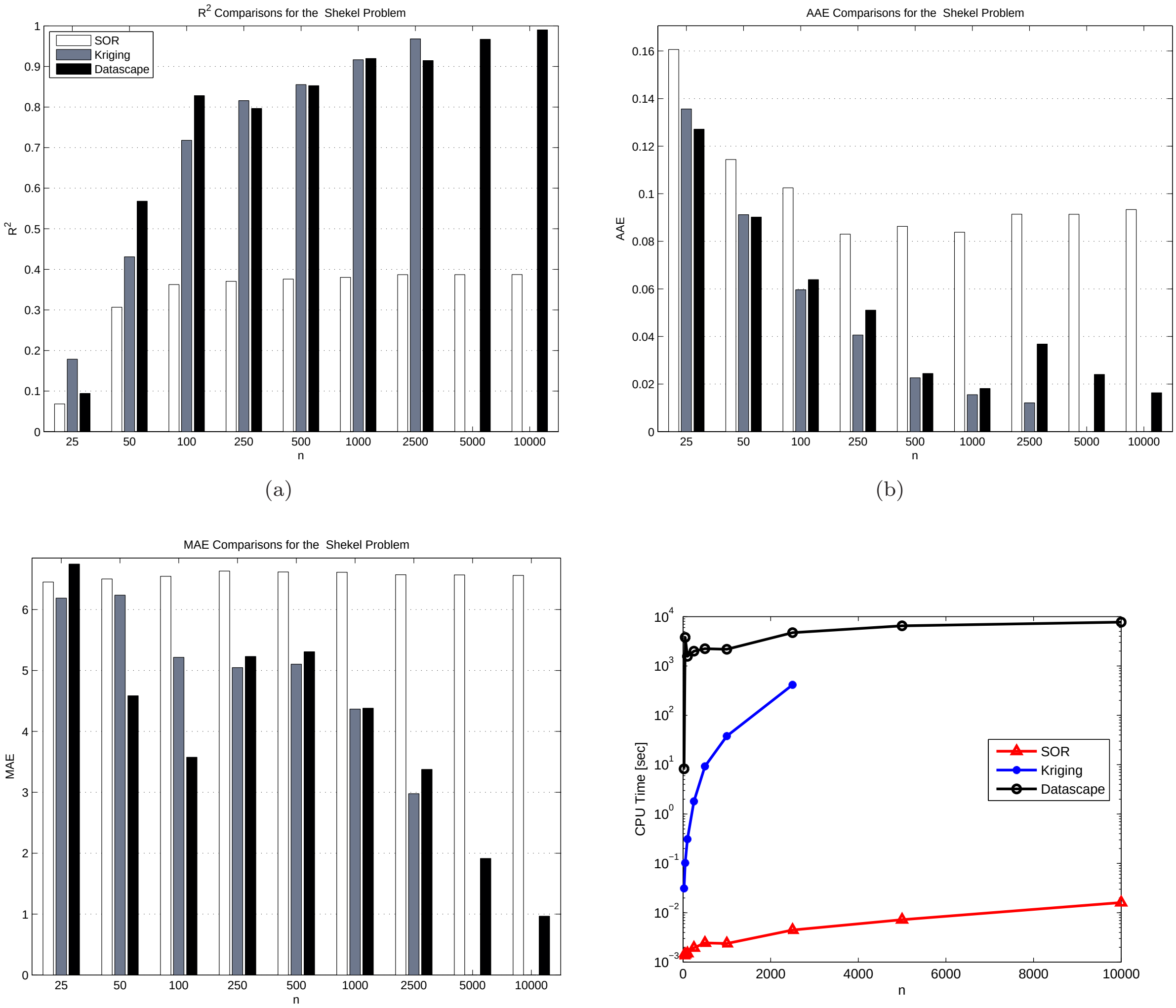
1. *Shekel Problem Results*

The results for the Shekel problem are plotted in Figure 4. A tabularized version of the results from this problem can be found in Appendix C. For the two largest sample sets, kriging was unable to create a surrogate model. This was because building a kriging model with such large data sets would require additional computer memory. Neither SOR or Datascape had this memory issue.

The R^2 results, given in Figure 4a, show that the SOR method produced the least accurate model for all of the sample sizes. In fact the SOR method seemed to converge to a value around 0.4 where as the other two methods obtained maximum R^2 values over 0.95. Datascape had the highest overall value of 0.99 when n=10000, a trial in which kriging failed to construct a model.

The SOR method also had the highest AAE and MAE. Kriging and Datascape followed similar trends in the AAE and MAE results with kriging generally having a lower AAE. Datascape produced the lowest overall MAE, but again this was for a case in which a kriging model was not built due to memory limitations.

Datascape took, by a considerable margin, the longest to construct surrogate models for each sample size and took just over 2 hours to complete the model for n=10000. At the opposite end of the spectrum, the SOR method took only 0.02 seconds to build the a model for the n=10000 case. The kriging models again had the highest sensitivity in CPU time required as a function of the number of samples, taking 0.03 seconds to build a model for n=25 and 7 minutes for n=2500.



(a) (b) (c) (d)

Figure 4. Accuracy and efficiency results for the Shekel problem.

V. Satellite Constellation Design

In the third and final test problem of this research, surrogate models were constructed to predict the line-of-sight access fraction to Valley Forge, Pennsylvania over a twenty four hour period for a constellation of three satellites, each in a near circular orbit (e=0.024) with semimajor axis of 7173.47km (perigee altitude of 621.9km). Figure 5 depicts an example of one such constellation. Line-of-sight access fraction is defined here as the total time at least one of the three satellites has direct view of Valley Forge divided by the 24 hour period. Nine orbital design variables were considered: the inclination, i_n, of satellites 1-3, the right ascension of the ascending node (RAAN), Ω_n, of satellites 2-3, the argument of perigee, ω_n, of satellites 2-3, and mean anomaly of satellites 2-3, ν_n. The simulations were performed with AGI's Satellite Toolkit using a J4 Perturbation propagator; which is an analytic propagator with corrections for the Earth's oblateness and asymmetric gravitational field. For this problem eight different sized sample sets were used, $n = \{50, 100, 250, 500, 1000, 2500, 5000, 10000\}$.

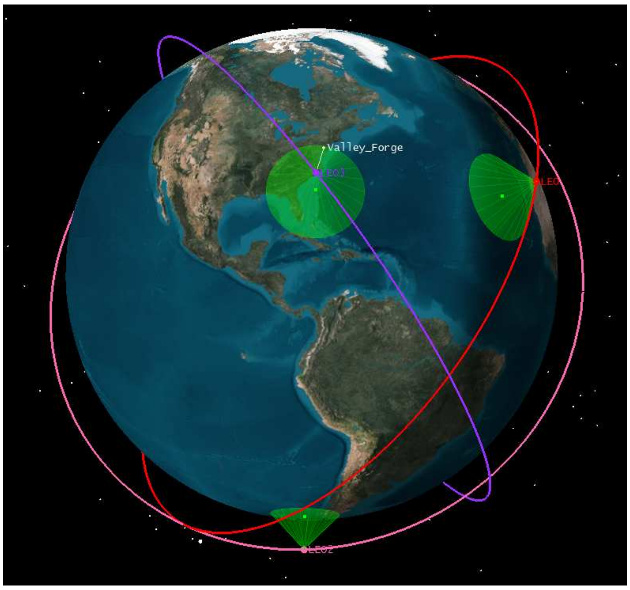


Figure 5. A three-satellite constellation with access cones.

2. Satellite Constellation Problem Results

The three satellite constellation problem results are plotted in Figure 6 along with a tabularized version in Appendix D. As in the Shekel problem kriging was unable to create a surrogate model for the largest two sample sets due to memory issues. Likewise, neither SOR or Datascape had this problem. Contour plots of the access fraction are plotted as a function of the inclinations of Satellites 1 and 2 as predicted by the STK simulation and of the three surrogate modeling techniques (for the n=1000 case) are given Figure 7. The contour plots are shown here to allow for a visual comparison of the surrogate models produced by the three techniques being studied. As expected, the SOR model showed the "bulls-eye" shape characteristic of a quadratic function, and didn't capture the different local optima of the design space. Both kriging and Datascape were able to capture the four local maxima; the kriging model is smooth, while the Datascape is piecewise linear with sharp corners.

Datascape results for $n = 15$ and $n = 100$ and SOR results for $n = 50$ are not shown in the R^2 bar graph in Figure 6 because they had negative values, and therefore didn't have enough data points to construct a usable model. Both the the R^2 and AAE results show that the kriging model performed the best for the four smallest data sets and Datascape preformed the best for the larger sets.

The maximum absolute error results, shown in Figure 6, show that Datascape has the highest single point error from the validation set for the trials of less 5,000 samples. For the higher sample sizes Datascape's maximum error become much smaller and similar to the results of the other models. This signifies that while the Datascape models were relatively accurate on average over the design space, for the lower data sets, they had at least one region in which they predicted the access fraction poorly.

The CPU time results for the SOR and Datascape trials were consistent with the previous test problems, taking into account the added number of design variables. Kriging, however, required a larger time to construct a model than Datascape did for $n = 2,500$ and showed a sharp increase in CPU time required for each larger data set.

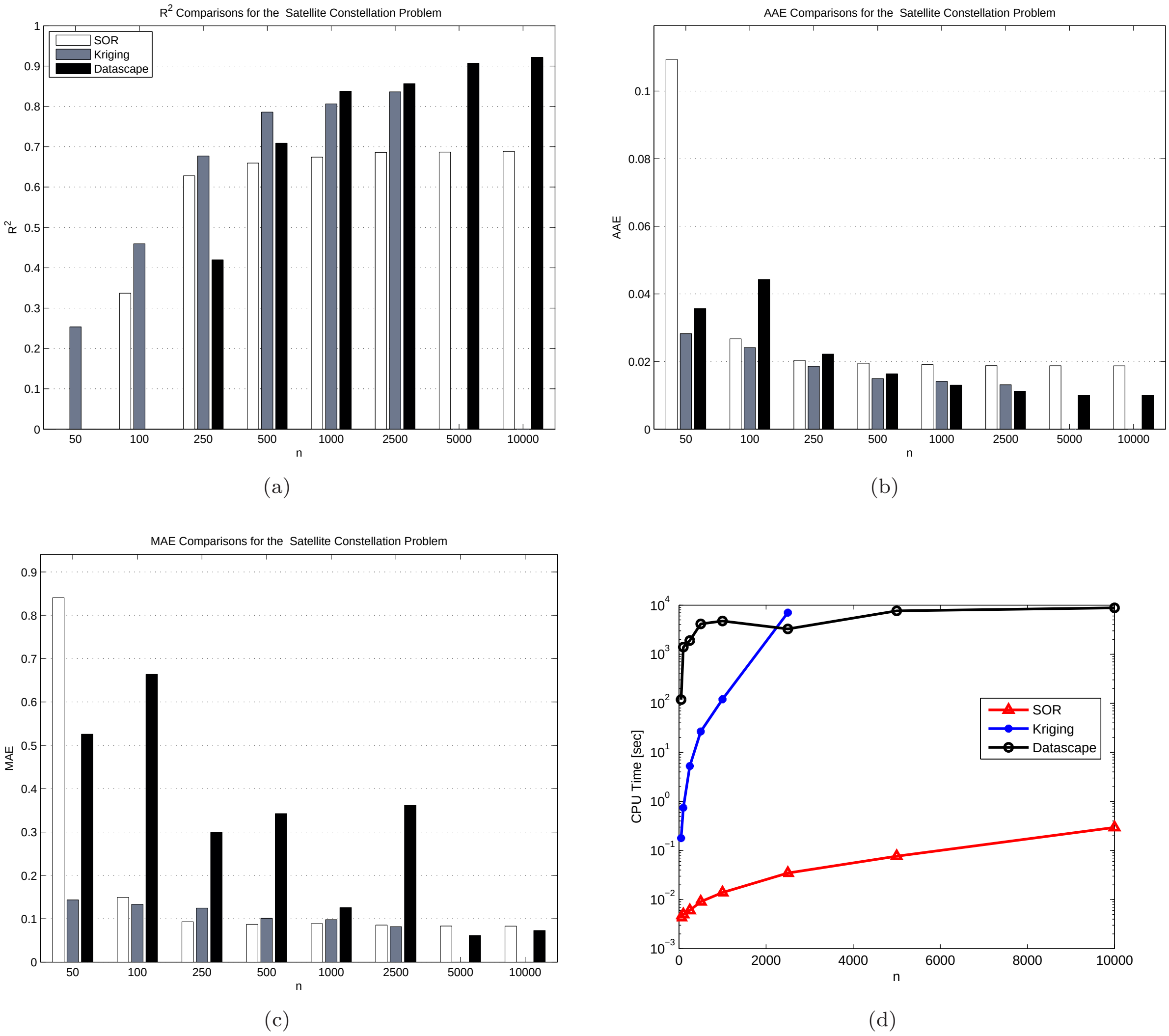


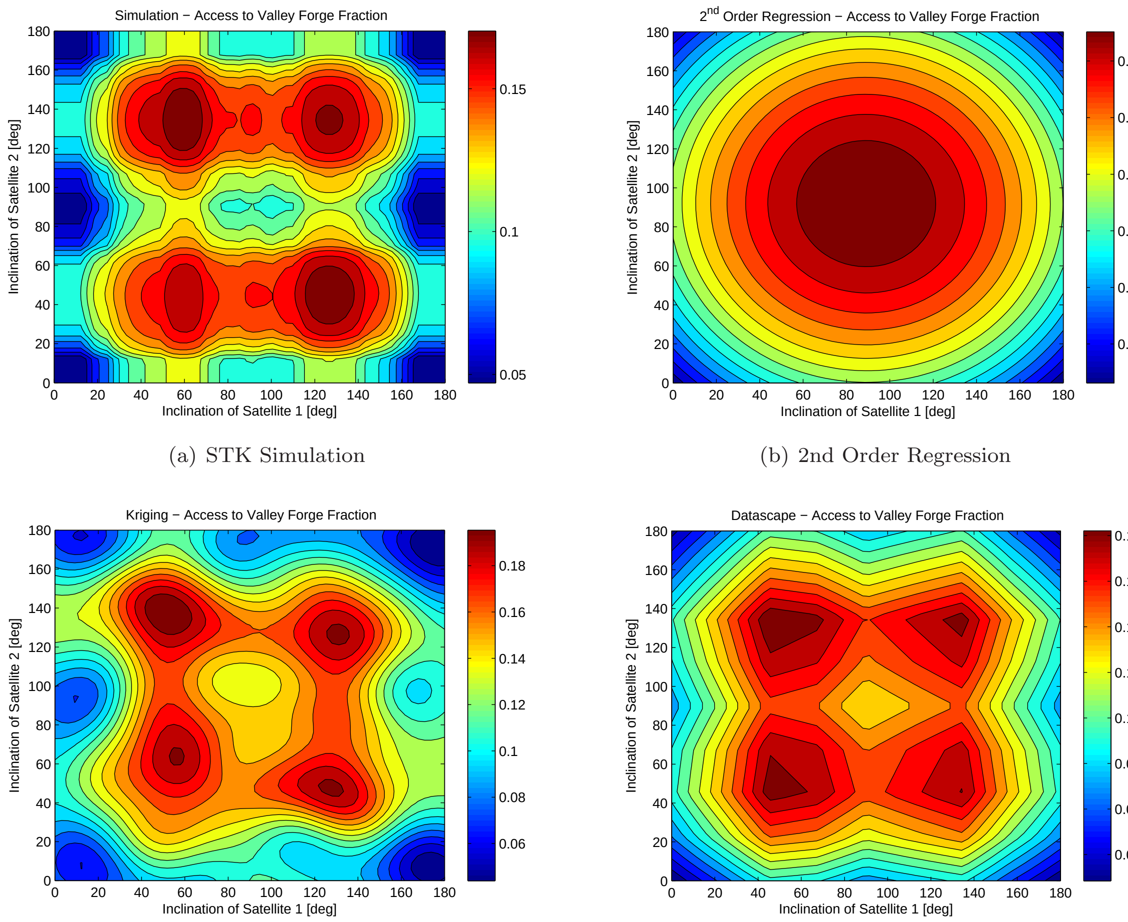
Figure 6. Accuracy and efficiency results for the Satellite Constellation problem.

VI. Conclusions and Future Work

In this research three surrogate modeling methods were compared: second order regression, kriging, and a COTS application called Datascape. They were compared by constructing a metamodel for a wide range of sample sizes, from sparse to dense, for three different test problems each having different numbers of design variables. Then a validation set of data points was used to measure the accuracy of each model by comparing the predictions of the models to the output of the simulation of test function. The CPU time required by the different modeling technique to create the surrogate models was also compared.

The accuracy of the surrogate modeling techniques was measured using three statistics, R^2, AAE, and MAE, to quantify the error between the surrogate models predictions and the validation responses. It was observed that when the surrogate models were created with the smaller sample sets, kriging had the lowest error. However, for the larger sample sets the Datascape performed the best. SOR was not found to be a suitable surrogate modeling technique for modeling the entire design space.

Computationally, SOR required the least amount of CPU time; in fact, it required orders of magnitude less than either kriging or Datascape. While kriging had the highest sensitivity to the number of design variables



(a) STK Simulation

(b) 2nd Order Regression

(c) Kriging

(d) Datascape

Figure 7. Access fraction contour plots showing the difference between the simulation and the surrogate modeling techniques using n=1000 sample points.

as well as the number of sample points used in building the models, Datascape was much more CPU intensive taking tens of minutes to converge even for smaller data sets. Kriging was unable to produce a surrogate model for the $n = 5,000$ and $n = 10,000$ cases due to memory requirements in constructing a single model for the entire design space. Conversely, SOR and Datascape had no such limitation; both methods could easily handle much larger data sets. All of the methods are very efficient at predicting new data points; SOR and Datascape could predict 20,000 points for the satellite design problem in less than a second, and kriging took just under two seconds.

The robustness of the three modeling techniques was difficult to quantify in this work because of the limited number of application types. It was observed that as the sample sizes increased, both SOR and kriging had a more monotonic decrease in prediction error. The Datascape models tended to be less robust in handling different sized data sets, especially for the smaller sample sets.

SOR and Datascape models provided excellent transparency, as they both provide direct contribution measures of the different design variables. Kriging provides some information in terms of its spatial correlation parameters, but they are harder to interpret.

SOR and kriging are both very easy methods to use and many free as well as commercial packages are readily available. In the kriging model there are many choices of correlation functions that can be used; though, the

Gaussian function is generally used in most engineering applications. Datascape generally didn't perform well as a "black-box" surrogate modeling technique. The optimal choice of points on the influence functions were very problem dependent and took a few iterations for each problem.

In future work, more test problems should be used to broaden the conclusions made in this study. Also a comparison of other surrogate modeling techniques could be included. Another interesting comparison between the different surrogate models could be the accuracy with which they predict local optima locations and values.

Acknowledgments

The authors would like to thank Craig Coburn, Angie Wrubel, and Rebekah Gano for their technical help and for reviewing this manuscript. Special thanks are also due to Mike Loucks of Space Exploration Engineering for his help with Astrogator and Mark Lovrich of TMP Inc. for help in using Datascape. Funding for this Lockheed Martin IS&S strategic initiative research project was graciously approved by Charlie Fisk and Dan Littley.

References

[1]Myers, R. H. and Montgomery, D. C., *Response Surface Methodology: Process and Product Optimization Using Designed Experiments*, Wiley, 1995.

[2]Giunta, A. A., Balabanov, V., Haim, D., Grossman, B., Mason, W. H., Watson, L. T., and Haftka, R. T., "Multidisciplinary Optimisation of a Supersonic Transport Using Design of Experiments Theory and Response Surface Modeling," *Aeronautical J.*, Vol. 101, No. 1008, 1997, pp. 347–356.

[3]Healy, M. J., Kowalik, J. S., and Ramsay, J. W., "Airplane Engine Selection by Optimization on Surface Fit Approximations," *J. Aircraft*, Vol. 12, No. 7, 1975, pp. 593–599.

[4]Sellar, R. S., Stelmack, M. A., Batil, S. M., and Renaud, J. E., "Response Surface Approximations for Discipline Coordination in Multidisciplinary Design Optimization," No. AIAA-96-1383, 1996.

[5]Pérez, V. M., Renaud, J. E., and Watson, L. T., "Reduced Sampling for Construction of Quadratic Response Surface Approximations Using Adaptive Experimental Design," *Proceedings of the 43rd AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference*, No. AIAA 2002-1587, Denver, CO, 22-25 April 2002.

[6]Simpson, T. W., Pepliski, J., Koch, P. N., and Allen, J. K., "On the Use of Statistics in Design and the Implications for Deterministic Computer Experiments," *in Design Theory and Methodology - DTM'C97*, No. No. DETC97/DTM-3881, Sacramento, CA, 1997.

[7]Sacks, J., Welch, W. J., Mitchell, T. J., and Wynn, H. P., "Design and Analysis of Computer Experiments," *Statistical Science*, Vol. 4, No. 4, 1989, pp. 409–423.

[8]Booker, A. J., Dennis, J. E., Frank, P. D., Serafini, D. B., Torczon, V., and Trosset, M. W., "A Rigorous Framework for Optimization of Expensive Function by Surrogates," *Structural Optimization*, Vol. 17, No. 1, 1999, pp. 1–13.

[9]Gano, S. E., Renuad, J. E., and Sanders, B., "Hybrid Variable Fidelity Optimization Using a Kriging-Based Scaling Function," *AIAA Journal*, Vol. 43, No. 11, 2005, pp. 2422–2430.

[10]Giunta, A. A. and Watson, L. T., "A Comparison of Approximation Modeling Techniques: Polynomial Versus Interpolating Models," No. AIAA-98-4758, 1998.

[11]Chung, H.-S. and Alonso, J. J., "Design of a Low-Boom Supersonic Business Jet Using Cokriging Approximation Models," *9th AIAA/ISSMO Symposium on Multidisciplinary Analysis and Optimization*, No. AIAA 2002-5598, Atlanta, GA, 4-6 September 2002.

[12]Simpson, T. W., Peplinski, J. D., Koch, P. N., and Allen, J. K., "Metamodels for Computer-based Engineering Design: Survey and recommendations," *Engineering with Computers*, Vol. 17, 2001, pp. 129–150.

[13]Simpson, T. W., Booker, A. J., Gosh, D., and Gunita, A. A., "Approximation Methods in Multidisciplinary Analysis and Optimization: A Panel Discussion," *Structural and Multidisciplinary Optimization*, Vol. 27, No. 5, 2004, pp. 302–313.

[14]Chung, H.-S. and Alonso, J. J., "Using Gradients to Construct Cokriging Approximation Models for High-Dimensional Design Optimization Problems," *40th AIAA Aerospace Sciences Meeting and Exhibit*, No. AIAA 2002-0317, Reno, NV, 14-17 January 2002.

[15]Simpson, T. W., Mauery, T. M., Korte, J. J., and Mistree, F., "Kriging Metamodels for Global Approximation in Simulation-Based Multidisciplinary Design Optimization," *AIAA Journal*, Vol. 39, No. 12, 2001, pp. 2233–2241.

[16]Martin, J. D. and Simpson, T. W., "A Study on the Use of Kriging Models to Approximate Deterministic Computer Models," *Proceedings of ASME 2003 Design Engineering Technical Conferences and Computers and Information in Engineering Conference*, No. DET2003/DAC-48762, Chicago, IL, 2-6 September 2003.

[17]Martin, J. D. and Simpson, T. W., "On the Use of Kriging Models to Approximate Deterministic Computer Models," *AIAA Journal*, 2005, In Press.

[18]Chung, H. S. and Alonso, J. J., "Comparison of Approximation Models with Merit Functions for Design Optimization,"

Proceedings of the 8th AIAA/USAF/NASA/ISSMO Symposium on Multidisciplinary Analyisis and Optimization, No. AIAA-2000-4754, Long Beach, CA, 6-8 September 2000.

[19]Jin, R., Chen, W., and Simpson, T. W., "Comparative Studies of Metamodelling Techniques under Mutliple Modelling Criteria," *Structural and Multidisciplinary Optimization*, Vol. 23, 2001, pp. 1–13.

[20]Third Millennium Productions, I., "Datascape User Manual," Tech. rep., 303 Main Street, Suite 300, Fort Worth, Texas 76102, 2004.

[21]Denison, D. G. T., Holmes, C. C., Mallick, B. K., and Smith, A. F. M., *Bayesian Methods for Nonlinear Classification and Regression*, John Wiley & Sons, Inc., New York, 2002.

[22]Stein, M. L., *Interpolation of Spatial Data: Some Theory for Kriging*, Springer-Verlag, 1999.

[23]Jones, D. R., Schonlau, M., and Welch, W. J., "Efficient Global Optimization of Expensive Black-Box Functions," *Journal of Global Optimization*, Vol. 13, No. 4, 1998, pp. 455–492.

[24]Lophaven, S. N., Nielsen, H. B., and Søndergaard, J., "DACE a Matlab Kriging Toolbox," Tech. Rep. IMM-REP-2002-12, Technical University of Denmark, DK-2800 Kongens Lyngby - Denmark, 1 August 2002, version 2.0.

[25]Gano, S. E., Renaud, J. E., Martin, J. D., and Simpson, T. W., "Update Strategies for Kriging Models used in Variable Fidelity Optimization," *Structural and Multidisciplinary Optimization*, Vol. 32, No. 3, 2006.

[26]Lophaven, S. N., Nielsen, H. B., and Søndergaard, J., "Aspects of the Matlab Toolbox DACE," Tech. Rep. IMM-REP-2002-13, Technical University of Denmark, DK-2800 Kongens Lyngby - Denmark, August 2002.

[27]McKay, M. D., Beckhan, R. J., and Conover, W. J., "A Comparison of Three Methods for Selecting Values of Input Variables in the Analysis of Output from a Computer Code," *Technometrics*, Vol. 21, 1979, pp. 239–245.

[28]Carrico, J., Schiff, C., Roszman, L., Hooper, H. L., Folta, D., and Richon, K., "An Interactive Tool for Design and Support of Lunar, Gravity Assist, and Libration Point Trajectories," *Proceedings of the AIAA/AHS/ASEE Aerospace Design Conference*, No. AIAA 93-1126, Irvine, California, February 1993.

[29]Carrico, J., Conway, D., Ginn, D., Folta, D., and Richon, K., "Operational Use of Swingbyan Interactive Trajectory Design and Maneuver Planning Toolfor Missions to the Moon and Beyond," *Proceedings of the AAS/AIAA Astrodynamics Specialist Conference*, No. AAS 95-323, Halifax, Nova Scotia, Canada, August 1995.

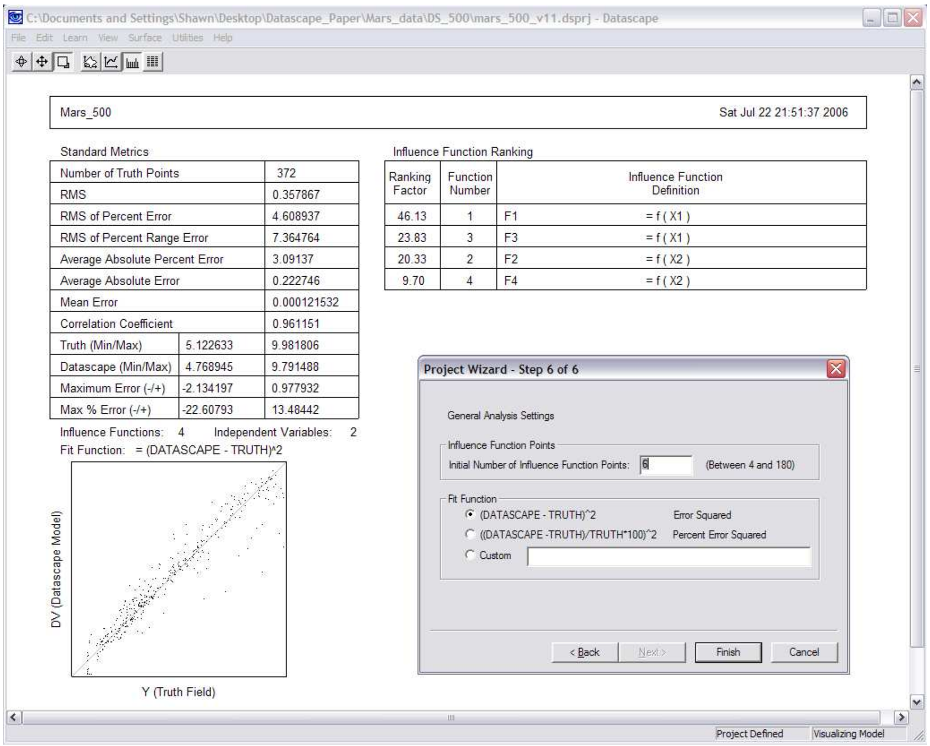
[30]Lozier, L., Galal, K., Folta, D., and Beckman, M., "Lunar Prospector Mission Design and Trajectory Support," *Proceedings of the AAS/AIAA Astrodynamics Specialist Conference*, No. AAS 98-323, 1998.

[31]Dunham, D. W., Jen, S. J., Roberts, C. E., and Seacord, A. W., "Transfer Trajectory Design for the SOHO Libration-Point Mission," *Proceedings of the 43rd Congress of the International Astronautical Federation*, No. IAF-92-0066, Washington, D.C., USA, August/September 1992.

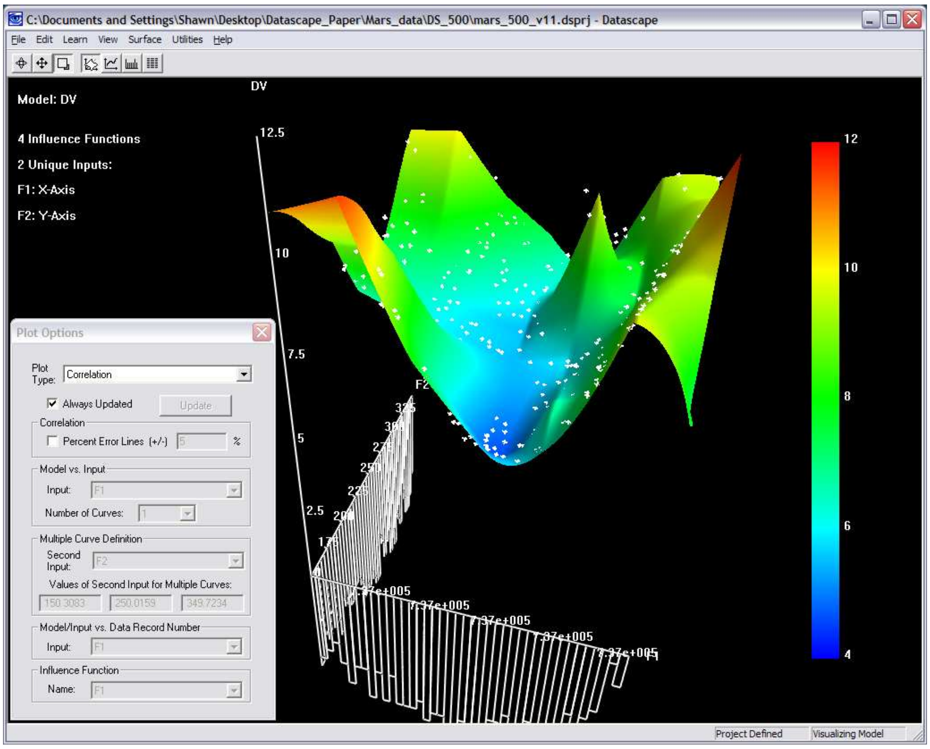
[32]Loucks, M., Carrico, J., Carrico, T., and Deiterich, C., "A Comparison of Lunar Landing Trajectory Strategies Using Numerical Simulations," *Proceedings of the International Lunar Conference*, Toronto, Canada, 18-23 September 2005.

[33]Shekel, J., "Test Functions for Multimodal Search Techniques," *Proceedings of the Fifth Annual Princeton Conference on Information Science and Systems*, 1971.

Appendix A: Datascape Screenshots



(a) Metrics Summary Screen.



(b) Surrogate Surface View.

Figure 8. Screen shots of the Datascape Application.

Appendix B: Earth-Mars Transfer Orbit Problem Tabularized Results

Method	CPU time [sec]	RMS	R^2	AAE	MAE	σ
n= 10						
SOR	7.9898e-004	1.2150e+000	6.2247e-002	7.1730e-001	5.8629e+000	1.1766e+000
Kriging	6.4732e-003	1.1547e+000	1.5296e-001	7.6048e-001	5.2034e+000	1.1282e+000
Datascape	6.2500e+000	9.4894e-001	4.2797e-001	7.5668e-001	3.8276e+000	9.4947e-001
n= 25						
SOR	8.0569e-004	8.8119e-001	5.0674e-001	6.2345e-001	4.0711e+000	8.7712e-001
Kriging	1.8102e-002	6.1178e-001	7.6224e-001	3.6223e-001	3.7614e+000	5.8490e-001
Datascape	3.9800e+002	9.4387e-001	4.3407e-001	6.2374e-001	4.8619e+000	9.4397e-001
n= 50						
SOR	8.2189e-004	8.2559e-001	5.6702e-001	6.1667e-001	3.8814e+000	8.2332e-001
Kriging	2.6861e-002	5.8649e-001	7.8150e-001	3.7211e-001	3.4602e+000	5.6873e-001
Datascape	2.6070e+003	9.3694e-001	4.4235e-001	7.1758e-001	3.4116e+000	9.3245e-001
n= 100						
SOR	8.4676e-004	8.3317e-001	5.5903e-001	6.3835e-001	3.7493e+000	8.3216e-001
Kriging	5.7672e-002	5.6039e-001	8.0051e-001	3.4507e-001	2.8737e+000	5.6075e-001
Datascape	3.0850e+003	7.5102e-001	6.4170e-001	5.7411e-001	3.6947e+000	7.4985e-001
n= 200						
SOR	8.6268e-004	8.2917e-001	5.6326e-001	6.3768e-001	3.7362e+000	8.2807e-001
Kriging	1.9527e-001	4.9264e-001	8.4583e-001	2.9080e-001	3.3755e+000	4.8801e-001
Datascape	3.1140e+003	4.8651e-001	8.4964e-001	3.1094e-001	3.6877e+000	4.8587e-001
n= 500						
SOR	9.5627e-004	8.1656e-001	5.7644e-001	6.2166e-001	3.7438e+000	8.1608e-001
Kriging	1.6969e+000	4.5289e-001	8.6971e-001	2.5611e-001	3.9101e+000	4.4922e-001
Datascape	3.4040e+003	3.9858e-001	8.9908e-001	2.4326e-001	3.6526e+000	3.9787e-001

Appendix C: Shekel Problem Tabularized Results

Method	CPU time [sec]	RMS	R^2	AAE	MAE	σ
n= 25						
SOR	1.3728e-003	2.5466e-001	6.8221e-002	1.6061e-001	6.4510e+000	2.5466e-001
Kriging	3.1208e-002	2.3911e-001	1.7852e-001	1.3563e-001	6.1874e+000	2.3261e-001
Datascape	8.2200e+000	2.5113e-001	9.3887e-002	1.2713e-001	6.7460e+000	2.5094e-001
n= 50						
SOR	1.4398e-003	2.1969e-001	3.0652e-001	1.1439e-001	6.5015e+000	2.1915e-001
Kriging	1.0234e-001	1.9904e-001	4.3077e-001	9.1213e-002	6.2355e+000	1.9904e-001
Datascape	3.7920e+003	1.7786e-001	5.6804e-001	9.0188e-002	4.5847e+000	1.7755e-001
n= 100						
SOR	1.5114e-003	2.1064e-001	3.6252e-001	1.0249e-001	6.5452e+000	2.1063e-001
Kriging	3.1040e-001	1.4012e-001	7.1791e-001	5.9605e-002	5.2159e+000	1.3872e-001
Datascape	1.5670e+003	1.0943e-001	8.2795e-001	6.3914e-002	3.5746e+000	1.0890e-001
n= 250						
SOR	1.9477e-003	2.0930e-001	3.7060e-001	8.3015e-002	6.6307e+000	2.0889e-001
Kriging	1.8041e+000	1.1323e-001	8.1578e-001	4.0624e-002	5.0472e+000	1.1301e-001
Datascape	1.9880e+003	1.1905e-001	7.9635e-001	5.1069e-002	5.2291e+000	1.1856e-001
n= 500						
SOR	2.4579e-003	2.0842e-001	3.7586e-001	8.6290e-002	6.6160e+000	2.0838e-001
Kriging	9.2302e+000	1.0037e-001	8.5524e-001	2.2634e-002	5.1029e+000	1.0031e-001
Datascape	2.2370e+003	1.0130e-001	8.5257e-001	2.4433e-002	5.3072e+000	1.0127e-001
n= 1000						
SOR	2.3844e-003	2.0767e-001	3.8034e-001	8.3814e-002	6.6119e+000	2.0743e-001
Kriging	3.7943e+001	7.6209e-002	9.1655e-001	1.5525e-002	4.3666e+000	7.6210e-002
Datascape	2.1730e+003	7.4761e-002	9.1969e-001	1.8127e-002	4.3801e+000	7.4745e-002
n= 2500						
SOR	4.4785e-003	2.0657e-001	3.8687e-001	9.1407e-002	6.5706e+000	2.0658e-001
Kriging	4.1510e+002	4.7329e-002	9.6781e-001	1.2083e-002	2.9771e+000	4.7320e-002
Datascape	4.7260e+003	7.7213e-002	9.1434e-001	3.6827e-002	3.3759e+000	7.7214e-002
n= 5000						
SOR	7.2269e-003	2.0656e-001	3.8696e-001	9.1370e-002	6.5680e+000	2.0656e-001
Kriging	-	-	-	-	-	-
Datascape	6.4790e+003	4.8019e-002	9.6687e-001	2.4057e-002	1.9141e+000	4.8020e-002
n= 10000						
SOR	1.6120e-002	2.0654e-001	3.8707e-001	9.3357e-002	6.5598e+000	2.0654e-001
Kriging	-	-	-	-	-	-
Datascape	7.7280e+003	2.6128e-002	9.9019e-001	1.6316e-002	9.6655e-001	2.6127e-002

Appendix D: Satellite Constellation Problem Tabularized Results

Method	CPU time [sec]	RMS	R^2	AAE	MAE	σ
n= 50						
SOR	4.3480e-003	1.4632e-001	-1.1526e+001	1.0938e-001	8.4057e-001	1.4015e-001
Kriging	1.7888e-001	3.5723e-002	2.5332e-001	2.8241e-002	1.4344e-001	3.5622e-002
Datascape	1.1900e+002	4.9622e-002	-4.4068e-001	3.5661e-002	5.2576e-001	4.8276e-002
n= 100						
SOR	5.0015e-003	3.3663e-002	3.3698e-001	2.6720e-002	1.4900e-001	3.3630e-002
Kriging	7.4305e-001	3.0391e-002	4.5960e-001	2.4124e-002	1.3315e-001	2.9980e-002
Datascape	1.3920e+003	6.3590e-002	-1.3659e+000	4.4300e-002	6.6373e-001	6.0625e-002
n= 250						
SOR	6.0223e-003	2.5212e-002	6.2809e-001	2.0343e-002	9.3021e-002	2.5195e-002
Kriging	5.2523e+000	2.3496e-002	6.7698e-001	1.8591e-002	1.2451e-001	2.3485e-002
Datascape	1.9010e+003	3.1494e-002	4.1968e-001	2.2191e-002	2.9919e-001	3.1353e-002
n= 500						
SOR	9.0827e-003	2.4122e-002	6.5955e-001	1.9483e-002	8.7066e-002	2.4113e-002
Kriging	2.6542e+001	1.9128e-002	7.8592e-001	1.4975e-002	1.0106e-001	1.9050e-002
Datascape	4.1520e+003	2.2300e-002	7.0904e-001	1.6360e-002	3.4254e-001	2.2271e-002
n= 1000						
SOR	1.3917e-002	2.3598e-002	6.7417e-001	1.9117e-002	8.8453e-002	2.3598e-002
Kriging	1.2055e+002	1.8196e-002	8.0627e-001	1.4121e-002	9.7872e-002	1.8195e-002
Datascape	4.7355e+003	1.6668e-002	8.3744e-001	1.3000e-002	1.2574e-001	1.6669e-002
n= 2500						
SOR	3.4768e-002	2.3165e-002	6.8603e-001	1.8782e-002	8.5438e-002	2.3162e-002
Kriging	7.0156e+003	1.6735e-002	8.3614e-001	1.3135e-002	8.1664e-002	1.6735e-002
Datascape	3.2760e+003	1.5663e-002	8.5645e-001	1.1193e-002	3.6191e-001	1.5660e-002
n= 5000						
SOR	7.6776e-002	2.3137e-002	6.8680e-001	1.8738e-002	8.3144e-002	2.3137e-002
Kriging	-	-	-	-	-	-
Datascape	7.6500e+003	1.2594e-002	9.0720e-001	9.9849e-003	6.1294e-002	1.2589e-002
n= 10000						
SOR	2.9610e-001	2.3066e-002	6.8870e-001	1.8709e-002	8.2957e-002	2.3066e-002
Kriging	-	-	-	-	-	-
Datascape	8.8200e+003	1.1573e-002	9.2163e-001	1.0055e-002	7.2945e-002	1.2840e-002