Where am I?
A fast multidimensional point location test and its applications

Tanja Clees, Martin Hüttemann, Igor Nikitin, Lialia Nikitina, Daniela Steffes-lai
Department of High Performance Analytics
Fraunhofer Institute for Algorithms and Scientific Computing
Sankt Augustin, Germany
Tanja.Clees| Martin.Huettemann |Igor.Nikitin|Lialia.Nikitina|Daniela.Steffes-lai@scai.fraunhofer.de

***Abstract*—We present our recent advances in RBF metamodeling of multidimensional data. A rapid point location test in multidimensional data cloud distinguishing the cases of interpolation and extrapolation is proposed. A linear program detecting a containment of the probe point in a convex hull of the dataset is formulated, simplex and interior point solution methods are tested in different dimensionalities and densities of the data cloud, extensions of the approach to nonconvex datasets and various acceleration strategies are implemented. The resulting software module is integrated in our optimization tool DesParO and applied to several real life problems from the fields of automotive industry and chemical engineering.**

Keywords-complex computing in application domains, automotive industry, chemical engineering, energy optimization.

I. INTRODUCTION

Numerical simulations define a mapping $y=f(x)$: $R^n \rightarrow R^m$ from an n-dimensional space of simulation parameters to an m-dimensional space of simulation results. E.g. in automotive crash test simulation the dimensionality of simulation parameters x is moderate (n~10-30), while simulation results y are dynamical fields sampled on a large grid, typically containing $\sim 10^6$ nodes and ~100 time steps, resulting in values of $m \sim 10^8$. High computational complexity of crash test models restricts the number of simulations available for analysis (typically $Nexp<10^3$), and this number shall be as small as possible. Metamodeling is an approximation technique allowing efficient representation of these large datasets for the purpose of data analysis, robust optimization and real time visualization. The metamodeling naturally involves in the analysis the uncertainties in optimization variables and other control parameters influencing the simulation. In practice a metamodeling with radial basis functions (RBF) is often used, i.e. representation of the form:

$$f(x)=\sum_{i=1.. Nexp} c_i \Phi(|x-x_i|), \quad (1)$$

where x_i are the points with known function values $y_i = f(x_i)$. A suitable choice for the RBF is the multi-quadric function $\Phi(r)=(b^2+r^2)^{1/2}$, which provides non-degeneracy of interpolation matrix $\Phi_{ij}=\Phi(|x_i-x_j|)$ for all finite datasets of distinct points and all dimensions [1]. The result can be written in a form of weighted sum $f(x)=\sum_i w_i(x) y_i$, with the weights

$$w_i(x)=\sum_j \Phi^{-1}_{ij} \Phi(|x-x_j|). \quad (2)$$

RBF interpolation can be extended by adding polynomial terms, allowing reconstructing exactly polynomial (including linear) dependencies and generally improving precision of interpolation. Adaptive sampling and hierarchy of metamodels with appropriate transition rules are used for further precision improvement [2]. RBF metamodel is directly applicable for interpolation of high dimensional bulky data, e.g. complete simulation results can be interpolated at a rate linear in the size of data, and even faster in combination with PCA-based dimensional reduction techniques [3]. The precision can be controlled via the cross-validation procedure: the data point is removed, data are interpolated to this point and compared with the actual value at this point, which for an RBF metamodel leads to a direct formula [4]

$$err_i = f_{interpol}(x_i)-f_{actual}(x_i) = -c_i/(\Phi^{-1})_{ii}. \quad (3)$$

Metamodeling performed at controlled precision can replace simulation results or real experimental data in computationally intensive procedures, such as optimization, parameter studies, stochastic analysis (i.e. determination of probability distributions by Monte Carlo techniques). In our previous work [2-9] we use RBF metamodeling for solution of various applied problems. For correct metamodeling one should permanently control that the interpolated point is located inside the boundaries of the data cloud. In optimization process and data analysis it is necessary to avoid extrapolation or at least to warn the user about it, to ensure correct functionality of the metamodel. While it is straightforward for hypercube alike designs of experiments, the problem becomes non-trivial for more complex shapes. In this case, a bounding box provides a too loose estimator of the cloud.

The key contribution of this work is to provide metamodel with a precise and fast indicator whether a point belongs to a multidimensional data cloud. In general, testing whether a probe point belongs to a region is a classical (Where am I?) point location problem. We start with its

 ISBN: 978-1-61208-419-0

special type (Am I in a convex hull of data points?) and compare the efficiency of various algorithms. High dimensionality of the space (dim≥10) excludes the usage of standard tools (e.g. Qhull), since they perform direct segmentation of the region to simplices and the number of simplices explodes at high dimensions. For example, tessellation of an n-dimensional cube [10] produces the number of simplices $\geq (n+1)^{(n-1)/2}$, requiring memory [bytes] $\geq 4(n+1)^{(n+1)/2}$, which already at n=16 corresponds to 100 GB of memory.

Being reformulated as a linear program (LP), location test works also for high dimensions and can be implemented efficiently using state-of-the-art algorithms. Furthermore, restricting the test to a controlled neighbourhood of the probe point, the method can be extended to nonconvex clouds of data points. In the next sections we compare the performance of various algorithms for LP-based location test, discuss their extension to nonconvex data clouds and apply them to real life problems of chemical engineering and automotive industry.

II. LP-BASED LOCATION TEST

Let x_i be data points in R^n, conv{x_i} – their convex hull, conv'{x_i}= conv{x_i} \ ∂ conv{x_i} – the convex hull without its boundary, $x^* = \sum x_i$ /Nexp – center of mass, x – probe point. We consider non-degenerate datasets: conv'{x_i}≠∅. Let's define containment flag as Cflag(x)=0 iff (x_i-x) are contained in a half-space, i.e. there exists a separating hyperplane with a normal v≠0 satisfying $(x_i\text{-}x,v) \geq 0$ for all i, and Cflag(x)=1 otherwise, see Fig. 1a. In other words, x is located inside conv'{x_i} when Cflag=1 and outside of conv'{x_i} when Cflag=0. Practically, Cflag can be determined using the following linear program:

Algorithm LP(x,{x_i}):
find max(x^*-x,v) at (x_i-x,v) ≥ 0 for all i;
if solution is max=+∞ then Cflag=0;
else (v=0) Cflag=1.

For numerical solution of LP we compare two algorithms: simplex [11] and interior point [12], applying them to a randomly filled n-dim cube. The results of comparison are collected in Table 1. We see that for the given problem simplex method performs by factor 10^2-10^3 better than interior point method.

LP-algorithm can be accelerated by combining with two simple tests. Let's consider a (generally nonconvex) region Ω uniformly filled with random points {x_i}. Let BB=[min(x_i),max(x_i)] be bounding box of the dataset. Let r=|x_i-x_j| be inter-point distance and <r> - its average. One should better use quasi-random (low discrepancy) sequences with narrow r-histograms, see Fig. 2. Let B be a ball around the probe point x with radius c<r>, where c is an empiric safety factor (e.g c=3 for rnd2D, c=2 for Sobol2D). If this ball does not contain points from {x_i}, then x is surely outside Ω, see Fig. 1b.

The following algorithms can serve as simple conservative containment tests:

Algorithm BBox(x,{x_i}):
if x∈BB then Cflag=1; else Cflag=0.

Algorithm BT(x,{x_i}):
find {x_i}∩B;
if empty then Cflag=0; else Cflag=1.

Let's define a local convex hull as LCH=conv'({x_i}∩B). Differently from the global convex hull (GCH) it considers only a small portion of data points and is computationally much faster:

Algorithm LCH(x,{x_i}):
call BBox(x,{x_i});
if Cflag=1:
call BT(x,{x_i});
if Cflag=1:
call LP(x,{x_i}∩B).

If x is outside LCH (Cflag=0), it is also outside Ω. If x is inside LCH (Cflag=1), it is either inside Ω or at a distance ~<r> from its boundary, see Fig. 1c. LCH provides more tight location test then GCH and BT. Performance of LCH is slower than BBox/BT but much faster than GCH, since only a small portion of data points N/Nexp is contained in B. The number of data points passed to LCH can be additionally controlled by selecting N'>n nearest data points in B. Performance of BBox is O(n), BT is O(Nexp*n). LP-algorithms have theoretical worst case complexity exponential for simplex method and polynomial for ipopt, while in practice they show much better performance, especially at reduced N, see Table 1.

TABLE I. BENCHMARK OF SIMPLEX AND INTERIOR POINT METHODS IN LP-BASED LOCATION TEST*.

Nexp	dim	Simplex (ms)	Ipopt (ms)
100	10	0.020	18
250	10	0.106	42
500	10	0.470	77
100	20	0.096	36
250	20	0.181	67
500	20	0.260	258
100	30	0.070	54
250	30	0.206	310
500	30	0.546	266

*Resulting Cflag values for both methods are always identical. Timing per solution @ 3 GHz Intel i7 CPU.

 ISBN: 978-1-61208-419-0

III. Applications

The better performing method (simplex LCH) has been integrated in our software tool for design parameter optimization (DesParO [13]). It uses RBF metamodel to represent dependence between design parameters and optimization criteria. The graphical user interface allows to change interactively the parameters and to see immediately the variation of the criteria. Constraints can be set e.g. maximizing one objective and minimizing the other, in this way the constrained and multiobjective optimization problems can be investigated. A graphical representation of interdependencies between parameters and criteria allows to find most influencing parameters and most sensitive criteria. Also, the uncertainties of metamodeling found with cross-validation procedure are shown (the red bars under criteria sliders). Fig. 3 and Fig. 5 show screenshots of interface of DesParO tool in application to several industrial problems.

The first application is safety optimization in Audi B-pillar crash test. The model of B-pillar shown on Fig. 3 contains ten thousand nodes, 45 timesteps. Two parameters are varied representing thicknesses of two layers composing a part of a B-pillar, comprising 101 simulations. The purpose is to find a Pareto-optimal combination of parameters simultaneously minimizing the total mass of the part and crash intrusion in the contact area. Fig. 3 shows the optimization problem loaded in the DesParO Metamodel Explorer, where design variables (thicknesses1,2) are presented on the bottom image at the left and design objectives (intrusion and mass) at the right. First, the user imposes constraints on design objectives, trying to minimize intrusion and mass simultaneously, as indicated by red ovals. As a result, “islands” of available solutions become visible along the axes of design variables. Exploration of these islands by moving corresponding sliders shows an optimal configuration, shown on Fig. 3 (bottom). For this configuration both constraints on mass and intrusion are satisfied. For every criterion also its tolerance is shown corresponding to 1-sigma confidence limits, as indicated by horizontal bars under the corresponding slider as well as +/- errors in the value box. This indication allows satisfying constraints with 3-sigma (99.7%) confidence, as shown on the image. The Geometry Viewer, shown at the top of Fig. 3, allows to inspect the optimal design in full details. LCH algorithm has been used to indicate location of the probe point in the data cloud, interpolation (Cflag=1) and extrapolation (Cflag=0).

The second application is scatter analysis in Ford Taurus crash test simulation. As shown in our previous work [3], crash test simulations possess a random component, related to physical and numerical instabilities of the underlying simulation model. It can be triggered by microscopic variations of design variables (e.g. thicknesses of various parts in car body) and by the numerical process itself (e.g. propagation of round-off errors or by random scheduling in multiprocessing simulation). These microscopic sources are then amplified by inherent physical instabilities of the model related e.g. to buckling, contact phenomena or material failure. Stochastic analysis is used to track the sources of scatter, to reconstruct causal chains and to identify hidden parameters describing chaotic behavior of the model. The crash model shown on Fig. 4 contains 1 million nodes, 32 timesteps, 25 simulations. The simulation results have been processed by a method of temporal clustering [3], which decomposes the whole scatter in the model over a system of basis functions: $s=\sum_i \Psi_i c_i$. Every basis function Ψ_i is associated with elementary random process (bifurcation) and possesses a typical conic profile, originating from the corresponding bifurcation point and propagating forward in time. Fig. 4 right shows one of the major bifurcations, corresponding to a fold on the floor of the vehicle. In total, 15 bifurcation points have been identified, representing statistically independent sources of scatter. In this way the dimensionality of the problem is reduced to 15 variables (c-coefficients) completely describing stochastic behaviour of the model. One should only take care that reconstruction of scatter does not go beyond the boundary of simulated data cloud. The point location test by LCH algorithm at these values of dimensions (Nexp=25, dim=15) has typical performance 27 mks per query (inside BBox).

The third application is energy optimization for Polycarbonate production at Bayer MaterialScience AG. The purpose was to minimize consumption of various energy media, including electric power, gas, steam, water, etc. The optimization has been performed on the base of experimental measurements, collecting data from sensors on several production lines and comprising 1-year detailed records of plant performance. Optimization is performed in 10-dimensional parameter space sampled with ~8000 data points, using our software tool for design parameter optimization (DesParO). RBF interpolation has been used for continuous optimization, see Fig. 5. Optimization parameters (par1, par2, …) are displayed on the left, optimization objectives on the right: partial energy consumptions (E01, E02,…), total energy cost and production range used as a constraint. LCH algorithm has controlled location of point in the data cloud, ensuring applicability of the metamodel. Fig. 5 shows on the top an optimal point inside the data cloud (Cflag=1, interpolation), while the bottom image shows the middle point of bounding box located outside of the given data cloud (Cflag=0, extrapolation). Typical performance was 0.3 ms per query inside BBox, where complete LCH algorithm was involved; while outside BBox only O(n) part of the algorithm was active, showing an extremely fast performance of 20 ns per query.

Point location tests in all applications have been performed on 3 GHz Intel i7 CPU with 8 GB RAM.

IV. Conclusion

RBF metamodeling of multidimensional data supplemented by a rapid point location test for distinguishing the cases of interpolation and extrapolation is presented. A linear program detecting a containment of the probe point in a convex hull of the dataset is formulated. Comparing simplex and interior point methods for solution of this particular linear program, we see that simplex method performs by factor 10^2-10^3 better than interior point one. A concept of local convex hull allows to extend the approach to nonconvex datasets, while simple geometrical containment tests are used to accelerate the algorithm. The resulting software module is integrated in our optimization tool DesParO and applied to real life problems from the fields of automotive industry and chemical engineering. In a typical problem with dimension 10 and number of data points ~8000 the performance of location test was 0.3 ms per query (inside BBox) and 20 ns per query (outside BBox).

REFERENCES

[1] M.D.Buhmann, Radial Basis Functions: theory and implementations, Cambridge University Press, 2003.

[2] G. van Bühren, T.Clees, N.Hornung, L.Nikitina, Aspects of adaptive hierarchical RBF metamodels for optimization, Journal of computational methods in sciences and engineering JCMSE 12 (2012), Nr.1-2, pp.5-23.

[3] T.Clees, I.Nikitin, L.Nikitina, C.-A.Thole, Analysis of bulky crash simulation results: deterministic and stochastic aspects, in N.Pina et al (Eds.): Simulation and Modeling Methodologies, Technologies and Applications, AISC 197, Springer 2012, pp.225-237.

[4] T.Clees, I.Nikitin, L.Nikitina, Nonlinear metamodeling of bulky data and applications in automotive design, in M. Günther et al (eds), Progress in industrial mathematics at ECMI 2010, Mathematics in Industry (17), Springer, 2012, pp. 295-301.

[5] T.Clees, I.Nikitin, L.Nikitina, S.Pott, Efficient Quantile Estimators for River Bed Morphodynamics, in Tören et al, Eds, Proc. of SIMULTECH 2013, 3rd International Conference on Simulation and Modeling Methodologies, Technologies and Applications, Reykjavik, Iceland 29 - 31 July, 2013, pp.737-743, SCITEPRESS, 2013.

[6] T. Clees, N. Hornung, I. Nikitin, L. Nikitina and D. Steffes-lai, Multi-objective Optimization and Stochastic Analysis in Focused Ultrasonic Therapy Simulation, in Tören et al, Eds, Proc. of SIMULTECH 2013, 3rd International Conference on Simulation and Modeling Methodologies, Technologies and Applications, Reykjavik, Iceland 29 - 31 July, 2013, pp.43-48, SCITEPRESS, 2013.

[7] Clees, T., Nikitin, I., Nikitina, L., Kopmann, R., Reliability analysis of river bed simulation models, in J.Herskovits, Ed., CDROM Proc. EngOpt 2012, 3rd Int. Conf. on Engineering Optimization, Rio de Janeiro, Brazil, no. 267.

[8] T. Clees, N. Hornung, I. Nikitin, L. Nikitina, D. Steffes-lai, S. Klimenko, Focused ultrasonic therapy planning: Metamodeling, optimization, visualization, J. Comp. Sci. 5 (6), Elsevier 2014, pp 891-897.

[9] T. Clees, I. Nikitin, L. Nikitina, S. Pott, Quasi-Monte Carlo and RBF Metamodeling for Quantile Estimation in River Bed Morphodynamics, in Obaidat, M.S. et al (Eds.), Simulation and Modeling Methodologies, Technologies and Applications, Advances in Intelligent Systems and Computing 319, Springer 2014, pp 211-222.

[10] A.Glazyrin, Lower bounds for the simplexity of the n-cube. Discrete Math. 312 (2012), no. 24, 3656-3662.

[11] G.Dantzig, Linear programming and extensions. Princeton University Press and the RAND Corporation, 1963.

[12] A.Wächter, Short Tutorial: Getting Started With Ipopt in 90 Minutes, IBM Research Report, 2009.

[13] DesParO: the tool for design parameter optimization, www.scai.fraunhofer.de/en/business-research-areas/high-performance-analytics/products/desparo.htm

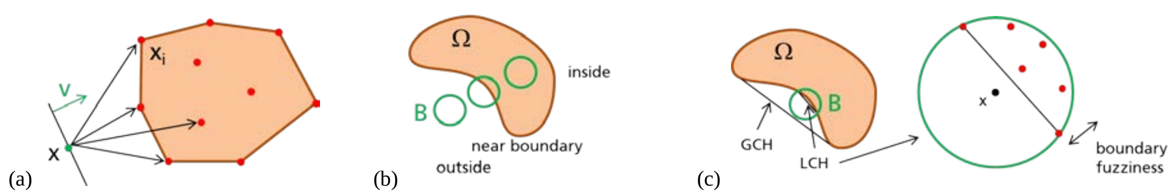

Figure 1. (a) to algorithm LP: data cloud, probe point and separating hyperplane; (b) to algorithm BT: data cloud and test ball; (c) to algorithm LCH: definition of local convex hull.

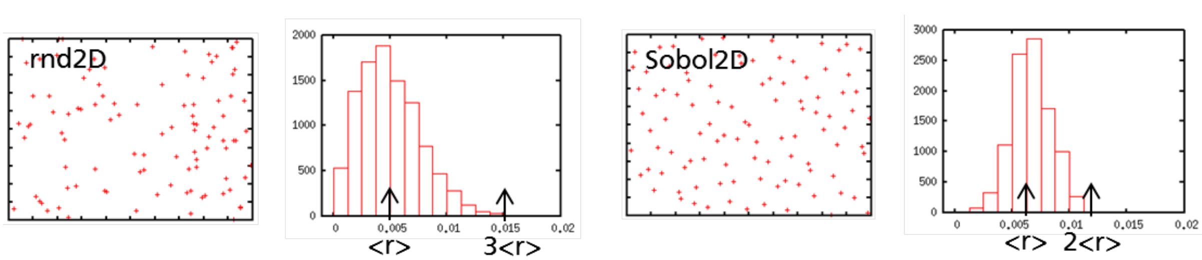

Figure 2. Pseudo-random (rnd2D) and quasi-random (Sobol2D) filling of a square and the corresponding r-histograms.

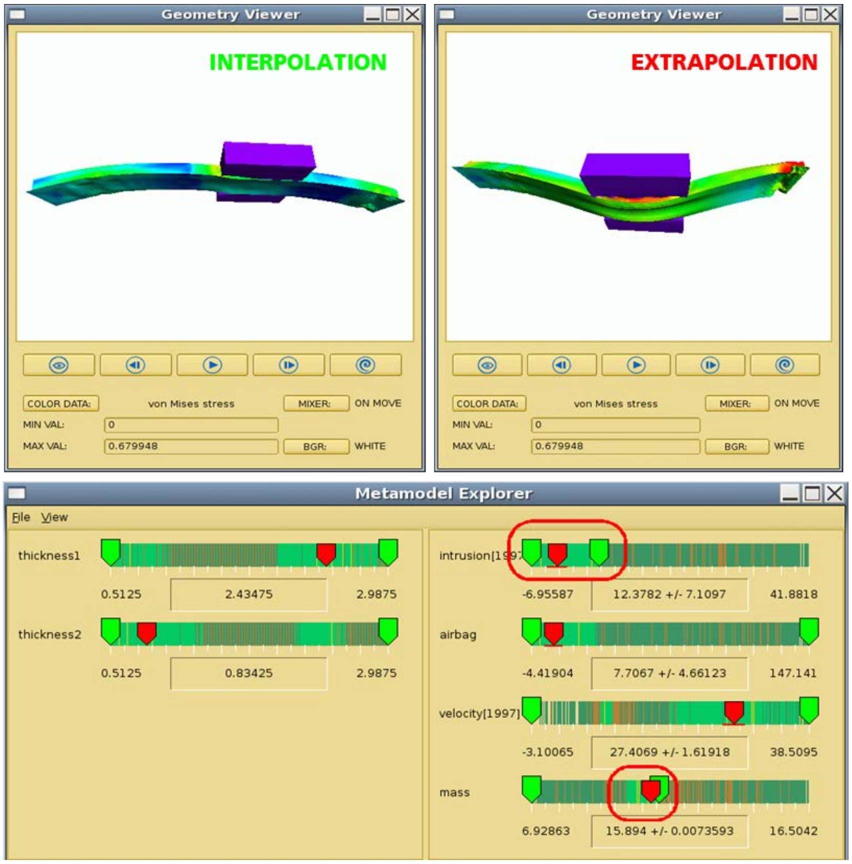

Figure 3. Metamodeling of B-pillar crash test simulation results. Point location test is used to distinguish cases of interpolation (point inside data cloud, in the image on the left) and extrapolation (point outside the data cloud, on the right). At the bottom: optimal design in DesParO Metamodel Explorer. Simulation model: courtesy of Audi.

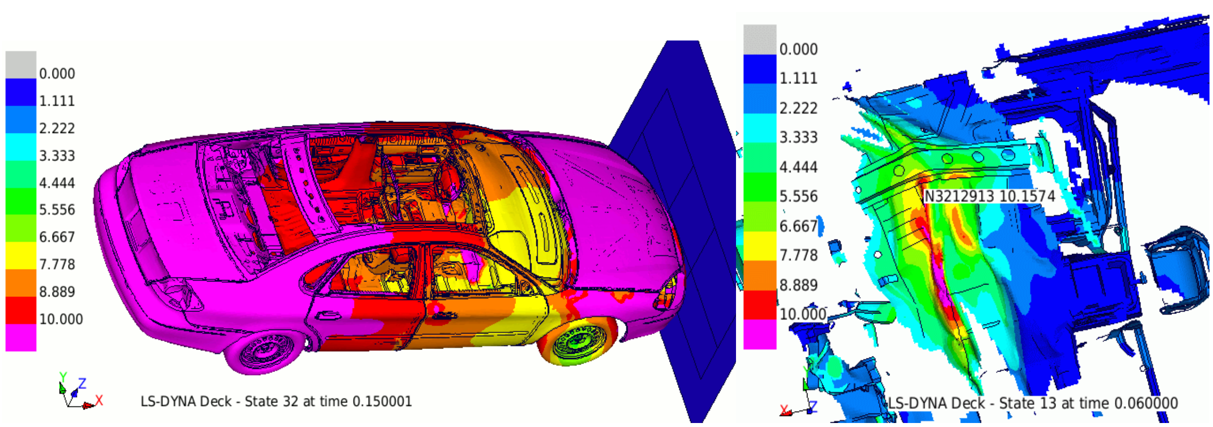

Figure 4. Scatter analysis in automotive crash test simulation. On the left: original scatter in mm. On the right: closeup to the main bifurcation, the source of scatter. Data courtesy of Ford.

 ISBN: 978-1-61208-419-0

Figure 5. Energy optimization for Polycarbonate production. Data courtesy of Bayer MaterialScience AG.