SHEPPACK: A Fortran 95 package for Interpolation using the Modified Shepard Algorithm

Manjula A. Iyer and Layne T. Watson
Departments of Computer Science and Mathematics
Virginia Polytechnic Institute and State University
Blacksburg, Virginia - 24061

Michael W. Berry
Department of Computer Science
University of Tennessee
Knoxville, Tennessee - 37996

ABSTRACT

Scattered data interpolation and approximation problems arise in many applications. Shepard's method for constructing global interpolants by blending local interpolants using locally-supported weight functions usually creates reasonable approximations. This paper describes SHEPPACK, a Fortran 95 package containing five versions of the modified Shepard algorithm. These five versions include quadratic (TOMS Algorithm 660, 661, and 798), cubic (TOMS Algorithm 790), and linear variations of the original Shepard algorithm. The main goal of SHEPPACK is to provide users with a single consistent package consisting of all existing polynomial variations of Shepard's algorithm. The algorithms target data of different dimensions. The linear Shepard algorithm is the only algorithm in the package that is applicable to arbitrary dimensional data. The motivation is to enable researchers to experiment with different algorithms using their data and select one (or more) that is best suited to their needs, and to support interpolation for sparse, high dimensional data sets.

1. INTRODUCTION

Interpolation problems arise in many areas where there is a need to construct a continuous surface from irregularly spaced data points. These areas include cartography, geophysics, data mining, engineering, meteorology, landscape ecology, computer graphics, and scientific visualization. The problem is to find a surface that approximates a function defined in m-dimensional space R^m, from a finite set of data points.

Currently, there are a number of solutions to the scattered data interpolation problem. The choice of the interpolation technique depends on the distribution of points in the data set, application domain, approximating function, or the method that is prevalent in the discipline. The contribution of this work is SHEPPACK, a software package intended to provide users with a single consistent package of all existing polynomial variations of Shepard's algorithm plus a new linear version for high dimensional data.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.
ACM SE'06, March 10–12, 2006, Melbourne, Florida, USA.
Copyright 2006 1-59593-315-8/06/0004...$5.00.

SHEPPACK consists of five different Fortran 95 implementations of the modified Shepard algorithm. QSHEP2D, QSHEP3D, and QSHEP5D are quadratic Shepard methods. QSHEP2D [Renka, 1988b] and QSHEP3D [Renka, 1988c] are direct translations of the original TOMS Algorithm, written in Fortran 77, developed by Renka for two-dimensional and three-dimensional data interpolation. QSHEP5D (developed by Berry [1999] in C++ as an upgrade on QSHEP3D for five-dimensional interpolation) has been translated to Fortran 95. CSHEP2D is a cubic Shepard method and is also a direct translation of the original TOMS Algorithm, written in Fortran 77, developed by Renka [1999a]. LSHEP is a linear Shepard interpolation method for arbitrary dimension. Note that the new code LSHEP is the only one of the five that is applicable to data of arbitrary dimension $m > 5$.

The remaining sections of the paper are organized as follows. Section 2 presents some common interpolation techniques. Section 3 presents Shepard's algorithm and its existing polynomial variations. Section 4 presents results for two sample test functions in 2, 3, and 5 dimensions. Lastly, Section 5 concludes.

2. INTERPOLATION TECHNIQUES

Shepard [1968] proposed an interpolation method that created a surface based on a weighted average of values at data points. The weight function was an inverse distance function of the data points. All methods of this type can be viewed as generalizations of Shepard's method. It was later found that this form of weight function accorded too much influence to data points that were far away from the point of approximation. Franke and Neilson [1980] developed a modification in which the weight function was designed to have local support and localized the overall approximation. This method is called the local modified Shepard method. Several variations of the modified Shepard algorithm have been developed.

Coons [1967] proposed a method for describing free form curved surfaces of a very general kind. Following this work, NURBS (nonuniform rational B-splines) were proposed by Versprille in 1975. They are a generalization of tensor product B-splines, and have several useful properties that have contributed to their popularity and wide commercial use. As in the case of ordinary B-splines, the sum of rational basis

functions of NURBS is equal to unity. By varying the knots, NURBS can satisfy a variety of smoothness requirements. Using NURBS, it is possible to represent free form curves and surfaces as well as analytic surfaces (such as conics and quadrics). A large variety of shapes can be designed by changing the coordinates of the control points and the weights associated with them. See Piegl [1989a, 1989b] for a more detailed discussion of the properties of NURBS.

A DACE (design and analysis of computer experiments) model is an interpolating model based on Bayesian statistics. It uses the prior distribution mechanism in Bayesian statistics through which one applies past experiences, knowledge, and intuition when solving a problem. In DACE models, the unknown function is typically expressed as the sum of a known function and a Gaussian random function. The known function is usually a constant, which is estimated based on observed response values. The Gaussian random function is characterized by a covariance matrix that depends on a correlation function selected by the user. See Guinta [1997] for a good overview of DACE. A more detailed discussion of the fundamental statistical and mathematical concepts can be found in Sacks et al. [1989], Koehler and Owen [1996], Osio and Amon [1996], and Booker et al. [1995].

The MARS (multivariate adaptive regression splines) technique adaptively selects a set of spline basis functions for approximating the response function through a forward/backward iterative approach. The algorithm partitions the input space into regions, each with its own regression equation. It then constructs a relation between the predictor variables and dependent variables (the spline approximation) from a set of basis functions that are entirely based on the regression data. In general, the technique is popular because it does not assume any particular type of relationship between predictor and dependent variables, and thus is widely applicable. See Friedman [1991] for a complete description of MARS.

Another popular approximation method uses radial basis functions (RBF). The significance and usefulness of approximation by RBF follows from the theory developed in Schoenberg [1938]. The paper by Micchelli [1986] presents fundamental theory for approximation by RBF. This type of approximation works well with scattered data because the interpolant only depends on the distances from the interpolation points. A popular choice for an interpolant function is a polyharmonic spline, which is derived from a strictly positive definite function. More details about the theory behind the choice of the interpolant function can be found in Sibson and Stone [1991], who obtained results using the thin-plate spline radial basic functions together with linear polynomials.

An excellent source of modern approximation theory is Cheney and Light [1999], which contains references for many different methods of approximation.

3. SHEPARD ALGORITHMS

This section describes the original Shepard algorithm. It also presents the quadratic, cubic, and linear variations of the Shepard algorithm.

3.1 Original Shepard Algorithm

Local methods are attractive for very large data sets because the interpolation or approximation at any point can be achieved by considering only a local subset of the data. Many local methods can be characterized as weighted sums of local approximations $P_k(x)$, where the weights $W_k(x)$ form a partition of unity. In order for the overall method to be local, it is necessary that the weight functions have local support, that is, be nonzero over a bounded region, or at a limited number of the data points.

The original global inverse distance weighted interpolation method is due to Shepard [1968]. All methods of this type may be viewed as generalizations of Shepard's method.

Let E^m denote m-dimensional Euclidean space, $x = (x_1, \ldots, x_m) \in E^m$, and for real w let $w_+ = \max\{0, w\}$. The scattered data interpolation problem can be defined as: given a set of irregularly distributed points $x^{(i)} \in E^m$, $i = 1, \ldots, n$, and scalar values f_i associated with each point satisfying $f_i = f(x^{(i)})$ for some underlying function $f : E^m \to E$, look for an interpolating function $\tilde{f} \approx f$ such that $\tilde{f}(x^{(i)}) = f_i$. Assume that every m-simplex formed from the points $x^{(i)}$ is nondegenerate (has a nonempty interior).

Define an approximation to $f(x)$ by

$$\tilde{f}(x) = \frac{\sum_{k=1}^{n} W_k(x) f_k}{\sum_{i=1}^{n} W_i(x)},$$

where the weight functions $W_k(x)$ are defined in the original Shepard algorithm as

$$W_k(x) = \frac{1}{\left\| x - x^{(k)} \right\|_2^2}.$$

However, this form of the weight functions accords too much influence to data points that are far away from the point of approximation and may be unacceptable in some cases.

3.2 Modified Shepard Algorithm

Franke and Nielson [1980] developed a modification that eliminates the deficiencies of the original Shepard's method. They modified the weight function $W_k(x)$ to have local support and hence to localize the overall approximation, and replaced f_k with a suitable local

approximation $P_k(x)$. This method is called the *local modified Shepard method* and has the general form

$$\tilde{f}(x) = \frac{\sum_{k=1}^{n} W_k(x) P_k(x)}{\sum_{i=1}^{n} W_i(x)},$$

where $P_k(x)$ is a local approximant to the function $f(x)$ centered at $x^{(k)}$, with the property that $P_k\left(x^{(k)}\right) = f_k$. The choice for the weight functions $W_k(x)$ used by Renka [1988a] was suggested by Franke and Nielson [1980] and is of the form

$$W_k(x) = \left[\frac{\left(R_w^{(k)} - d_k(x)\right)_+}{R_w^{(k)} d_k(x)} \right]^2,$$

where $d_k(x) = \left\| x - x^{(k)} \right\|_2$ is the Euclidean distance between the points x and $x^{(k)}$, and the constant $R_w^{(k)} > 0$ is a radius of influence about the point $x^{(k)}$ chosen just large enough to include N_w points. The data at $x^{(k)}$ only influences $f(x)$ values within this radius.

There are several variations of the original Shepard algorithm based on polynomial and trigonometric functions for P_k.

The polynomial function P_k is written as a Taylor series about the point $x^{(k)}$ with constant term $f_k = P_k\left(x^{(k)}\right)$ and coefficients chosen to minimize the weighted square error

$$\sum_{\substack{i=1 \\ i \neq k}}^{n} \omega_i\left(x^{(k)}\right) \left[P_k\left(x^{(i)}\right) - f_i \right]^2,$$

with weights

$$\omega_i\left(x^{(k)}\right) = \left[\frac{\left(R_p^{(i)} - d_i\left(x^{(k)}\right)\right)_+}{R_p^{(i)} d_i\left(x^{(k)}\right)} \right]^2,$$

and $R_p^{(k)} > 0$ defining a radius about $x^{(k)}$ within which data is used for the least squares fit. R_w and R_p are taken by Franke and Nielson [1980] as

$$R_w = \frac{D}{2}\sqrt{\frac{N_w}{n}}, \quad R_p = \frac{D}{2}\sqrt{\frac{N_p}{n}},$$

where $D = \max_{i,j} \left\| x^{(i)} - x^{(j)} \right\|_2$ is the maximum distance between any two data points, and N_w and N_p are arbitrary constants. The constant values for R_w and R_p are appropriate assuming uniform data density.

3.3 Quadratic Shepard Algorithm for 2D data

QSHEP2D (quadratic Shepard algorithm for two-dimensional data), is TOMS Algorithm 660 developed by Renka [1988b]. The subroutine QSHEP2D in SHEPPACK is a direct Fortran 95 translation of Renka's Fortran 77 code of the same name.

The parameters N_p and N_w for QSHEP2D were chosen by computing all twenty-four error norms for all parameter values $N_p \in [9, 16]$ and $N_w \in [1, 30]$. The optimal pair of parameter values was found to vary widely with the choice of norm, test function, and data set, but in each case, the error norms vary smoothly with N_p and N_w, generally increasing monotonically and slowly with distance from the optimal pair. The recommended value for $N_p = 13$ and that for $N_w = 19$.

3.4 Quadratic Shepard Algorithm for 3D data

QSHEP3D (quadratic Shepard algorithm for three-dimensional data), is TOMS Algorithm 661 developed by Renka [1988c]. The subroutine QSHEP3D in SHEPPACK is a direct Fortran 95 translation of Renka's Fortran 77 code of the same name.

The parameters N_p and N_w for QSHEP3D were chosen by computing all twelve error norms for all parameter values $N_p \in [11, 18]$ and $N_w \in [6, 35]$. The choice of optimal value is less clear than in the two-dimensional case due to more rapid variation of some of the error norms with N_p values. The optimal range of parameters was found to be $N_p \in [13, 14]$ and $N_w \in [31, 35]$.

3.5 Quadratic Shepard Algorithm for higher dimensional data

QSHEP5D is a quadratic Shepard Algorithm for higher dimensional data (up to five dimensions). It is TOMS Algorithm 798 developed by Berry [1999]. The Fortran 95 package provided with this paper is a direct translation of Berry's C++ code. The visualization of results using netcdf file format have not been incorporated in the translation.

Following the recommendation of Renka [1988a], QSHEP5D chooses parameters N_p and N_w as follows. For an M-dimensional hypervolume with N known nodes, N_p was chosen as

$$\frac{M(M+3)}{2} \leq NP \leq \min(50, N-1).$$

For an M-dimensional hypervolume with N known nodes, N_w was chosen as

$$1 \leq NW \leq \min(50, N-1).$$

3.6 Cubic Shepard Algorithm for two-dimensional data

CSHEP2D (quadratic Shepard algorithm for two-dimensional data), is TOMS algorithm 790 developed by Renka [1999a]. The subroutine CSHEP2D in SHEPPACK is a direct Fortran 95 translation of Renka's Fortran 77 code of the same name.

The parameters N_p and N_w were chosen using Renka's recommendation [1988a]. The optimal values were found to be $N_p = 17$ and $N_w = 30$.

3.7 Linear Shepard Algorithm

The basis function $P_k(x)$ was the constant f_k in the original Shepard algorithm, and later variants used a quadratic polynomial, a cubic polynomial, and a cosine trigonometric polynomial. The primary disadvantages for large data sets is that a considerable amount of preprocessing is needed to determine closest points and calculate the local approximation. The second order polynomial models have $(m+2)(m+1)/2$ coefficients for m design variables, therefore the number of coefficients for $P_k(x)$ is at least $(m+2)(m+1)/2$ which is prohibitive for a problem in which $m >> 5$ and function values f_k are expensive. If $P_k(x)$ has degree d, the number of coefficients is $\binom{m+d}{d}$, requiring at least that many data points. Also the use of multivariate polynomials for the evaluation of the nodal functions leads to the loss of the main advantage of Shepard's original method, namely its independence in the evaluation phase from space dimension.

This consideration motivates the choice of $P_k(x)$ as linear, which would require less function values to be computed in order to construct the local least squares fit. The radii R_w and R_p vary with k and are taken to be

$$R_w^{(k)} = 2 \min_{\substack{1 \le i \le n \\ i \ne k}} ||x^{(k)} - x^{(i)}||_2,$$

$$R_p^{(k)} = \min \left\{ r \mid \overline{B(x^{(k)}, r)} \text{ contains at least } 3m/2 \text{ of the points } x^{(i)} \right\},$$

where $\overline{B(x,r)}$ is the closed ball of radius r with center x.

The Linear Shepard's method would choose $P_k(x)$ as

$$P_k(x) = f_k + \sum_{j=1}^{m} a_j^{(k)} \left(x_j - x_j^{(k)} \right)$$

Let $S = \{i_1, i_2, \ldots, i_{s_k}\} = \{i \mid i \ne k \text{ and } \omega_i(x^{(k)}) \ne 0\}$, the set of indices corresponding to points and weights $\omega_i(x^{(k)}) \ne 0$ that determine the local least squares approximation $P_k(x)$. Define the $s_k \times m$ matrix A and s_k-vector b by

$$A_{j\cdot} = \sqrt{\omega_{i_j}\left(x^{(k)}\right)} \left(x^{(i_j)} - x^{(k)} \right)^t,$$

$$b_j = \sqrt{\omega_{i_j}\left(x^{(k)}\right)} \left(f_{i_j} - f_k \right).$$

The coefficients $a^{(k)}$ of $P_k(x)$ are then the minimum norm solution of the linear least squares problem

$$\min_{a \in E^m} \|Aa - b\|_2,$$

found by using a complete orthogonal factorization of A via the LAPACK subroutine DGELSY.

4. SAMPLE FUNCTION RESULTS

Two continuous, piecewise smooth test functions, similar to functions occuring in many applications, are used to test and compare the SHEPPACK algorithms. Sample results have been compiled for problems with dimension $m = 2, 3, 5$.

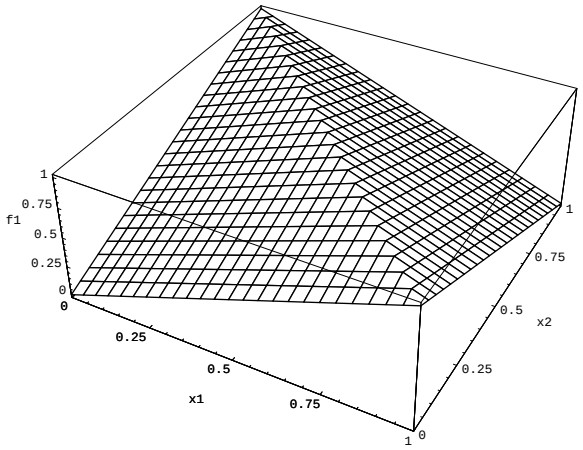

Figure 1. Test Function f_1.

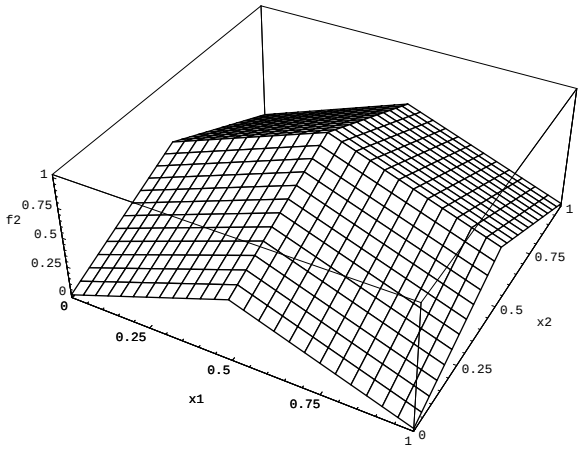

Figure 2. Test Function f_2.

4.1 Test Problems Formulation

The test functions f_1 and f_2 for $m = 2$ are shown in Figures 1 and 2 respectively.

The function f_1 in m dimensions is

$$f_1(x) = \begin{cases} \frac{2}{m} \sum_{i=1}^{m} x_i, & \sum_{i=1}^{m} x_i \le \frac{m}{2}, \\ -\frac{2}{m} \sum_{i=1}^{m} x_i + 2, & \sum_{i=1}^{m} x_i > \frac{m}{2}, \end{cases}$$

which has the maximum $f_1(x^*) = 1$ at $\sum_{i=1}^{m} x_i^* = \frac{m}{2}$.

The function f_2 in m dimensions is

$$f_2(x) = 1 - \frac{2}{m} \sum_{i=1}^{m} |x_i - 0.5|,$$

which has the maximum $f_2(x^*) = 1$ at $x_i^* = 0.5$, for $i = 1 \ldots m$.

4.2 Evaluation of Accuracy

The test function was sampled at n_0 random (uniformly distributed) scattered points of $Q = [0,1]^m$. The approximation error has been characterized using two error metrics. These are the maximum absolute error e_{max} and the root mean squared error e_r. The absolute approximation error is defined as

$$e_i = |\tilde{f}(x_i) - f(x_i)|,$$

where $\tilde{f}(x)$ is the interpolant function, $f(x)$ is the test function, and x_i are the points of a fine uniform $n_1 \times \ldots \times n_1$ grid $G \subset [0.1, 0.9]^m$ covering the domain Q. The total number of grid points is $n = n_1^m$.

Using this notation, the maximum absolute error is

$$e_{max} = \max_{1 \le i \le n} e_i,$$

and the root mean squared error is defined as

$$e_r = \sqrt{\frac{\sum_{i=1}^{n} e_i^2}{n}}.$$

4.3 Results

This section presents the approximation errors for the linear Shepard method, quadratic Shepard method, and cubic Shepard method for the sample test functions $f_1(x)$ and $f_2(x)$. The results for two, three, and five dimensional problems are summarized.

4.3.1 Two-dimensional test problem

For the two-dimensional test problem $n_0 = 20$, 40, 60, 80, 100, and $n = 121$. The 121 test sites are points of the 11×11 grid G. The approximation errors for the linear Shepard method (LSHEP), quadratic Shepard method (QSHEP), and cubic Shepard method (CSHEP) were calculated for the test functions f_1 and f_2 and are listed in Table 1.

Table 1. Interpolation errors for two-dimensional test functions f_1 and f_2.

		LSHEP		**QSHEP**		**CSHEP**	
	n_0	e_r	e_{max}	e_r	e_{max}	e_r	e_{max}
f_1	20	5.65E-2	1.47E-1	5.29E-2	1.70E-1	5.04E-2	1.56E-1
	40	4.77E-2	1.31E-1	3.02E-2	1.17E-1	2.97E-2	0.96E-1
	60	4.56E-2	1.13E-1	1.93E-2	7.72E-2	1.89E-2	6.82E-2
	80	4.18E-2	1.05E-1	1.47E-2	4.67E-2	1.18E-2	3.26E-2
	100	3.14E-2	9.82E-2	1.33E-2	3.61E-2	1.19E-2	1.81E-2
f_2	20	7.32E-2	2.30E-1	5.74E-2	1.77E-1	4.86E-2	1.61E-1
	40	5.87E-2	2.28E-1	3.12E-2	1.14E-1	2.93E-2	0.94E-1
	60	4.33E-2	1.28E-1	2.15E-2	7.46E-2	1.84E-2	6.38E-2
	80	3.53E-2	1.23E-1	1.62E-2	6.40E-2	1.51E-2	4.28E-2
	100	3.47E-2	1.01E-1	1.37E-2	5.10E-2	1.18E-2	3.93E-2

4.3.2 Three-dimensional test problem

For the three-dimensional test problem $n_0 = 100$, 200, 300, 400, 500, and $n = 1331$. The 1331 test sites are points of the $11 \times 11 \times 11$ grid G. The approximation errors for the linear Shepard method (LSHEP) and quadratic Shepard method (QSHEP) were calculated for the test functions f_1 and f_2 and are listed in Table 2.

Table 2. Interpolation errors for three-dimensional test functions f_1 and f_2.

		LSHEP		**QSHEP**	
	n_0	e_r	e_{max}	e_r	e_{max}
f_1	100	4.25E-2	1.37E-1	2.77E-2	1.06E-1
	200	4.12E-2	1.36E-1	2.13E-2	9.95E-2
	300	3.38E-2	1.28E-1	1.81E-2	9.74E-2
	400	3.14E-2	1.20E-1	1.55E-2	9.21E-2
	500	2.46E-2	1.03E-1	1.21E-2	8.35E-2
f_2	100	5.56E-2	1.69E-1	2.96E-2	1.24E-1
	200	3.77E-2	1.29E-1	2.04E-2	1.03E-1
	300	3.65E-2	1.17E-1	1.82E-2	9.40E-2
	400	3.14E-2	1.02E-1	1.59E-2	7.46E-2
	500	2.83E-2	9.51E-2	1.26E-2	5.44E-2

4.3.3 Five-dimensional test problem

For the five-dimensional test problem $n_0 = 100$, 200, 400, 800, 1600, and $n = 3125$. The 3125 test sites are points of the $5 \times 5 \times 5 \times 5 \times 5$ grid G. The approximation errors for the linear Shepard method (LSHEP) and quadratic Shepard method (QSHEP) were calculated for the test functions f_1 and f_2 and are listed in Table 3.

Table 3. Interpolation errors for five-dimensional test functions f_1 and f_2.

		LSHEP		**QSHEP**	
	n_0	e_r	e_{max}	e_r	e_{max}
f_1	100	5.89E-2	1.79E-1	3.52E-2	1.51E-1
	200	5.34E-2	1.72E-1	3.39E-2	1.47E-1
	400	5.16E-2	1.68E-1	3.31E-2	1.41E-1
	800	4.53E-2	1.51E-1	3.13E-2	1.37E-1
	1600	3.96E-2	1.38E-1	2.01E-2	1.26E-1
f_2	100	6.03E-2	2.02E-1	5.82E-2	1.95E-1
	200	5.44E-2	1.93E-1	5.04E-2	1.86E-1
	400	5.01E-2	1.81E-1	4.73E-2	1.73E-1
	800	4.63E-2	1.73E-1	4.49E-2	1.68E-1
	1600	3.99E-2	1.66E-1	3.63E-2	1.59E-1

5. CONCLUSIONS

The results summarized in Tables 1-6 show that the performance of the cubic Shepard algorithm is better than the quadratic Shepard algorithm, which

in turn is better than the linear Shepard algorithm. However, the linear Shepard algorithm gives reasonably good estimates and its performance is comparable to the other algorithms. The advantage of using the linear Shepard algorithm becomes more apparent as the dimension m increases. If $P_k(x)$ has degree d, the number of coefficients is $\binom{m+d}{d}$, requiring at least that many data points. This is a substantially bigger number in higher dimensions for higher order polynomials. In higher dimensions, it is therefore desirable to use the linear Shepard algorithm. In view of this consideration, only the linear Shepard method (LSHEP) in SHEPPACK is applicable to data in arbitrary dimension $m > 5$.

SHEPPACK is a single consistent package containing all existing polynomial variations of Shepard's algorithm. Using SHEPPACK, researchers can experiment with one or more algorithms using their data. After obtaining interpolation results from SHEPPACK, researchers can make an informed decision as to which algorithm is best suited to their needs for a given dataset.

ACKNOWLEDGEMENT

This work was supported in part by NSF Grant DMI-0422719.

REFERENCES

[1] M. W. Berry and K. S. Minser, "Algorithm 798: High-dimensional interpolation using the modified Shepard method", *ACM Transactions on Mathematical Software*, vol. 25, no. 3, pp. 353–366, 1999.

[2] A. J. Booker, A. R. Conn, J. E. Dennis, P. D. Frank, M. Trosset, and V. Torczon, "Global modeling for optimization: Boeing/IBM/Rice Collaborative Project", Text provided by Paul D. Frank of Boeing, 1995.

[3] W. Cheney and W. Light, "A course in approximation theory", The Brooks/Cole Series in Advanced Mathematics, 1st edition, 1999.

[4] S. Coons, "Surfaces for computer-aided design of space forms", Technical Report TR-41, Massachusetts Institute of Technology, Cambridge, MA, USA, 1967.

[5] R. Franke and G. Neilson, "Smooth interpolation of large sets of scattered data", *J. Numer. Methods Eng.*, vol. 15, pp. 1691–1704, 1980.

[6] J. H. Friedman, "Multivariate adaptive regression splines", *Annals of Statistics*, vol. 19, pp. 1–141, 1991.

[7] V. Gantovnik, Z. Gürdal, and L. T. Watson, "Linear Shepard interpolation for high dimensional piecewise smooth functions", in *Proc. 10th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conf.*, CD-ROM, Albany, NY, 2004, 15 pages.

[8] A. A. Giunta, "Aircraft multidisciplinary design optimization using design of experiments theory and response surface modeling methods", PhD thesis, Virginia Polytechnic Institute and State University, 1997.

[9] J. R. Koehler and A. B. Owen, "Computer experiments", *Handbook of Statistics*, S. Ghosh and C. R. Rao (eds.), Elsevier Science, vol. 13, pp. 261–308, 1996.

[10] C. A. Micchelli, "Interpolation of scattered data: Distance matrices and conditionally positive definite functions", *Constructive Approximation*, vol. 2, pp. 11–22, 1986.

[11] R. Myers, "Classical and Modern Regression with Applications", Duxbury Press, 2nd edition, 2000.

[12] I. G. Osio and C. H. Amon, "An engineering design methodology with multistage Bayesian surrogates and optimal sampling", *Research in Engineering Design*, vol. 8, pp. 189–206, 1996.

[13] L. Piegl, "Modifying the shape of rational B-splines, part 1: curves", *Computer-Aided Design*, vol. 21, no. 8, pp. 509–518, 1989a.

[14] L. Piegl, "Modifying the shape of rational B-splines, part 2: surfaces", *Computer-Aided Design*, vol. 21, no. 9, pp. 538–546, 1989b.

[15] R. J. Renka, "Multivariate interpolation of large sets of scattered data", *ACM Transactions on Mathematical Software*, vol. 14, no. 2, pp. 139–148, 1988a.

[16] R. J. Renka, "Algorithm 660: QSHEP2D: Quadratic method for bivariate interpolation of scattered data", *ACM Transactions on Mathematical Software*, vol. 14, no. 2, pp. 149–150, 1988b.

[17] R. J. Renka, "Algorithm 661: QSHEP3D: Quadratic method for trivariate interpolation of scattered data", *ACM Transactions on Mathematical Software*, vol. 14, no. 2, pp. 151–152, 1988c.

[18] R. J. Renka, "Algorithm 790: CSHEP2D: Cubic Shepard method for bivariate interpolation of scattered data", *ACM Transactions on Mathematical Software*, vol. 25, no. 1, pp. 70–73, 1999a.

[19] R. J. Renka, "Algorithm 791: TSHEP2D: Cosine series Shepard method for bivariate interpolation of scattered data", *ACM Transactions on Mathematical Software*, vol. 25, no. 1, pp. 74–77, 1999b.

[20] R. J. Renka and R. Brown, "Algorithm 792: Accuracy test of ACM algorithms for interpolation of scattered data in the plane", *ACM Transactions on Mathematical Software*, vol. 25, no. 1, pp. 78–94, 1999c.

[21] J. Sacks, W. J. Welch, T. J. Mitchell, and H. P. Wynn, "Design and analysis of computer experiments", *Statistical Science*, vol. 4, no. 4, pp. 409–435 1989.

[22] I. J. Schoenberg, "Metric spaces and completely monotone functions", *Ann. Math.*, vol. 49, pp. 811–841, 1938.

[23] D. Shepard, "A two-dimensional interpolation function for irregularly spaced data", *Proceedings of the 1968 23rd ACM national conference*, pp. 517–523, 1968.

[24] R. Sibson and G. Stone, "Computation of thin-plate splines", *SIAM Journal on Scientific and Statistical Computing*, vol. 12, pp. 1304–1313, 1991.

[25] K. J. Versprille, "Computer-aided design applications of the rational B-spline approximation form", PhD thesis, Syracuse University, 1975.