EVALUATION OF SOME DATA TRANSFER ALGORITHMS FOR NONCONTIGUOUS MESHES

Marilyn J. Smith
Assistant Professor, School of Aerospace Engineering, Georgia Institute of Technology, Atlanta, GA 30332-0150

Carlos E. S. Cesnik
Boeing Assistant Professor, Dept. of Aeronautics and Astronautics, MIT, Cambridge, MA

Dewey H. Hodges
Professor, School of Aerospace Engineering, Georgia Institute of Technology, Atlanta, GA

KEYWORDS

Interpolation, Extrapolation, CFD, CSD, Numerical Methods, Unstructured Grids, Structured Grids

ABSTRACT

The objective of this research was to identify and evaluate selected suitable methods to transfer information between computational methods with noncontiguous meshes or grids. This data transfer can easily be the limiting factor in the accuracy of computational simulations in a variety of applications. The data to be transferred can include point variables, such as deflections, pressure, and temperature; area-based variables, such as loads; and rates such as heat flux. A method should provide smooth, yet accurate, transfer of data for a wide variety of functional forms. An extensive literature survey was completed that identified current algorithms in aerospace applications, as well as other candidate algorithms from different disciplines, such as mapping and CAD/CAM. The performance of the various methods was assessed by a series of test cases, including the mapping of constant and linear functions, as well as sinusoidal functions with varying numbers of oscillations within the domain. Accuracy, computational memory requirements, and computational time requirements were all evaluated.

BACKGROUND

Many computational methodologies have become relatively mature, so that utilization of these methodologies in more complex, interdisciplinary problems is a current area of interest. These interdisciplinary applications include, but are not limited to, thermal signature or loads evaluation, computational elasticity, and particulate flows. In these applications, single-application codes are many times utilized together. Data must be transferred from one application to the other, and the grids or surfaces of the different methodologies are unlikely to coincide. Thus, an interpolation method to interface between the meshes must be employed. Although the transfer of data between grids may seem at first to be trivial, this is far from the case. Because of inherent differences in the physical equations, some simulations may require meshes to be very fine in certain areas, while other methodologies may require more coarse meshes. In addition, if the meshes are not totally coincident, the transfer of data between the two grids may require both extrapolation and interpolation.

Early efforts in the development of interpolation algorithms centered on the application of one-dimensional (1-D) splines (Done 1965, Conte and de Boor 1980) for both one- and two-dimensional structural panels. Harder and Desmarais (1972) developed a method of surface splines for plates known as the infinite-plate spline (IPS) method. These surface splines are the basis of interpolation schemes used today in many finite element methods including MSC/NASTRAN (MacNeal-Schwindler, 1987).

Many linear and surface splines were developed for two-dimensional models and are not suitable in many instances for applications to three-dimensional structures (or data) which are required in current state-of-the-art codes. A systematic study is necessary to examine some of these existing interpolation schemes, assess their strengths and weaknesses, and analytically assess their applicability for a wide range of data types.

This paper extends from research that was designed to isolate appropriate interpolation/extrapolation algorithms for aeroelastic applications on aircraft. These results, including many applications to current aeroelastic problems of interest, can be found in Smith et al. (1995 and 2000). This paper presents all of the algorithms examined in the study, including those that were not selected for final evaluation on aeroelastic aircraft test cases. These algorithms may be suitable for applications other than fluid-structure interactions, in particular those in civil and mechanical engineering.

APPROACH

As part of the development of a generic interface method, the manner through which information is passed from one mesh to another was examined. A literature search was performed to: 1) identify or eliminate possible interpolation schemes based on previous research, and 2) aid in and reduce the amount of investigation which must be done to determine the suitability of a potential scheme. This literature search encompassed not only methods applied to aerospace-related interpolation, but also to other engineering disciplines as well as mathematical or scientific (physics, etc.) applications. An excellent review of some of these methods was accomplished by Franke (1982).

The selection of candidate algorithms was made on the basis of the results of the literature search, as well as two algorithms at the request of the sponsoring agency (see Acknowledgments). Six selections were made: infinite-plate splines, finite-plate splines, thin-plate splines, multiquadric-biharmonic, inverse isoparametric mapping, and Non-Uniform B-Splines.

Analytical tests were then performed to examine the behavior of the functions for data that may be encountered in a variety of applications, and to evaluate specific behaviors, such as smoothness and extrapolation. These methods were evaluated in a generic sense; that is, there was no data from actual applications at this point in the analysis. Instead, the general behavior of the algorithm to different types of data was evaluated. Thus, to simulate randomly varying data and data with directional bias, a combination of linear and sinusoidal functions were generated as a subset of the test cases. By means of these tests, the functions were analyzed for their characteristics on two- and three-dimensional meshes. Since these functions should provide both interpolation and extrapolation, the characteristics of their behavior and limits of operation were examined. Using these metrics, combinations of these functions were used to develop over 260 different test cases against which to compare the interpolation and extrapolation characteristics of each algorithm.

Both quantitative and qualitative assessments were made. Quantitative assessments of the accuracy of each method were developed, as well as for each method's computational requirements. Quantitative assessment was made using relative errors between the original and interpolated data. The relative errors were computed using the algorithm:

$$\text{Relative Error (\%)} = (\text{Computed Value - Actual Value}) / \text{Actual Value} \times 0.01 \qquad (1)$$

The "Actual Value" is defined as the actual function value when computed directly on the second grid (that is, the grid to which the values were interpolated). In addition, it was impossible to quantitatively evaluate the extrapolation capabilities of an algorithm since there were not discrete values with which to compute an error. Thus, a qualitative assessment of extrapolation was made to see how the extrapolated points behave in the extrapolated region. Although some algorithms may extrapolate points differently than other algorithms, the extrapolation may not be "wrong"; the interpretation of the results must lie with the application to which it is being applied. Thus, only behavior such as oscillations was noted as being poor extrapolation characteristics. Plots demonstrating the character of each method with respect to these quantitative results are presented. Only partial results are shown for clarity of the plots. Three-dimensional (shell) results are shown only if they vary from the two-dimensional (plate) results.

Qualitative assessments also consisted of visual inspection of the interpolation and/or extrapolation results. Smoothness of the interpolated function and evidence of directional bias in the interpolated data were noted, and examples of these are presented herein. An assessment of the robustness of each method was made by noting how often the method failed (crashed).

ALGORITHM DESCRIPTIONS

Full technical descriptions of each method are contained in Smith et al. (1995). A short summary of each technique is included herein.

Infinite-Plate Splines (IPS) - The method of infinite-plate splines (Harder and Desmarais, 1972) is one of the most popular methods of interpolation, currently used in programs such as MSC/NASTRAN. This method is based on a superposition of the solutions of the partial differential equation of equilibrium for an infinite-plate. Consider a set of *N* discrete "grid points" lying within a two-dimensional domain with Cartesian coordinates x_1 and x_2. Each grid point has associated with it a "deflection" *H* that defines the vertical position coordinate of the surface on which both sets of mesh points are presumed to lie. For a one-dimensional problem, this equation is

$$H(x) = \sum_{i=1}^{N} \{ A_i + B_i (x - x_i)^2 + F_i (x - x_i)^2 \ln (x - x_i)^2 \} \qquad (2)$$

where *H(x)* is the deflection, A_i, B_i, and F_i are undetermined coefficients, and x_i are the surface locations of the known function.

Using solutions of the infinite-plate equation, one calculates the values of a set of concentrated loads, all presumed to act at the known data points that give rise to the required deflections. Those concentrated forces are then substituted back into the solution, thus providing a smooth surface that passes through the data.

Some advantages to this method are that the grid is not restricted to a rectangular array and that the interpolated function is differentiable everywhere. Areas far from known points are extrapolated nearly linearly. A minimum of three points is required, since three points are necessary to define a plane.

Finite-Plate Splines (FPS) - The original method of Appa (1989) employs uniform plate bending elements to represent a given planform by a number of quadrilateral or triangular elements. A virtual surface is defined and constrained to pass through both sets of mesh points. These constraints are imposed at the element level, and a proper choice of shape functions is required. These shape functions define a virtual surface that relates the data for both sets of grid points. The node points of the virtual surface do not have to coincide with those of either grid. Usually, however, the number of virtual surface grid points is less than or equal to the number of the coarser grid. The governing equation in one dimension can be expressed as:

$$\left\{ r(x) \right\} = \left[\Omega(x) \right] \left\{ q^e \right\} \tag{3}$$

where $r(x)$ is the displacement and rotation at any x, $\Omega(x)$ are the shape functions at x, and q^e is the vector of local element displacement and rotation.

The finite-plate approach has the advantage of accommodating changes in meshes or models easily. In addition, this approach conserves the work done by external forces (loads) when obtaining the global nodal force vector.

Multiquadric-Biharmonic (MQ) - The multiquadrics method is an interpolation technique that represents an irregular surface. More recently named the Multiquadric-Biharmonic method, it was used to perform interpolation of various topographies (Hardy, 1971). The original name reflects the method's use of quadratic basis functions; note that a "quadric" surface is one whose geometry is described by quadratic equations. The quadric surface used in most cases is a hyperboloid of revolution in two sheets. The addition of "biharmonic" to the name is due to an important proof that the equations governing the method can always be solved (Hardy, 1971). The interpolation equation investigated is

$$H(x) = \sum_{i=1}^{N} \alpha_i \left[(x - x_i)^2 + r^2 \right]^{1/2} \tag{4}$$

The MQ method is stable and consistent with respect to the user-defined parameter *r* that controls the shape of the basis functions. A large *r* gives a flat sheet-like function, while a small *r* gives a narrow cone-like function. For non-zero values of *r*, MQ produces an infinitely differentiable function that preserves monotonicity and convexity. Later development and implementation by Kansa (1990) and by Smith et al. (1995 and 2000) show that the method's conditioning, accuracy, and general numerical performance are improved by (1) permitting *r* to vary among the basis functions; (2) scaling and/or rotating the independent variables for some applications where the magnitudes of the variables differ widely; and (3) applying it in overlapping subdomains.

Thin-Plate Splines (TPS) - Thin-plate splines (or surface splines) provide a means to characterize an irregular surface by using functions that minimize an energy functional (Duchon, 1977). This methodology is very similar to the multiquadric-biharmonic method. The primary difference in these two methods is the function solved. A one-dimensional version of the function is

$$H(x) = \sum_{i=1}^{N} \alpha_i \left|x - x_i\right|^2 \log\left|x - x_i\right| \tag{5}$$

Here, the problem is approached from an engineering or physical representation of the surface. That is, for a 1-D problem, elementary cubic splines can be interpreted as equilibrium positions of a beam undergoing bending deformation. For a 2-D problem (such as a surface), these splines can be determined from the minimization of the bending energy (thus defining the equilibrium position) of a thin plate (which reduces to IPS). Since these types of splines are invariant with rotation and translation, they are very powerful tools for the interpolation of moving or flexible surfaces. This method is not limited to 2-D problems, but is extendable to 3-D problems.

Non-Uniform B-Splines (NUBS) - The Non-Uniform B-Spline method is based on the fact that splines in their most primitive form are used to represent curves in three-dimensional space. Therefore, a tensor product of two splines can be used to represent a surface in three-dimensional space. In order to do the surface blending needed in two- and three-dimensional computational applications, it is recommended (Boeing, 1995) that polynomial B-splines be used because rational splines have a tendency to generate poles and cause numerical problems. The resulting method therefore represents a surface by the tensor product of two B-splines:

$$S_{kt}(x,y) = \sum_{i=1}^{m-1} \sum_{j=1}^{n-1} P_{ij} B_{ik}(x) B_{jt}(y) \tag{6}$$

where S is the surface deflection at any point (x,y), P_{ij} are coefficients multiplying these splines in order to fit the data (control points), and B_{ik} and B_{jt} are the B-splines in the x and y directions, respectively. The NUBS method is implemented with the aid of a library of routines called DT_NURBS developed at the David Taylor Research Center (Boeing, 1995).

Inverse Isoparametric Mapping (IIM) - The Inverse Isoparametric Mapping method is based on finite element analysis where an isoparametric element uses the same shape functions to interpolate both the coordinate and the displacement vectors. This interpolation is a one-to-one mapping, termed isoparametric, from a local to a global Cartesian or displacement plane. On the other hand, if one has a given point in the global domain and wants the local coordinates corresponding to it, the inverse of this mapping involves a system of nonlinear equations, even for the linear strain element. The system of nonlinear equations can be solved numerically using an iterative approach.

The implementation here is one developed by Fithen (1995), but is currently unpublished. It uses a bilinear (4-node) element and no information from one mesh is taken into account to determine the cell that contains a given point from the other mesh. This method was only derived for two-dimensional applications and it is not capable of extrapolation.

SAMPLE TESTS

The test cases were designed to examine a number of different situations that could arise during use of the algorithms. Combinations of functions (e.g., constant, linear, and sinusoidal) were chosen to represent different types of data that might be encountered. In addition to the assessment of the methods, the purpose of these tests was to determine the limiting characteristics of each of the algorithms.

A series of five test sets were developed which included over 260 test cases, from constant to 7 × 7 cycle oscillation combinations. Results of the full set of test cases are provided in Smith et al. (1995). Quantitative results are presented using tables and plots of maximums and averages. Two examples that readily illustrate the qualitative assessment of the methods are also presented. These two test cases are shown in Figure 1. The first test case is a simple one-cycle sinusoid, while the second test case illustrates the impact of increasing the oscillations on the plate to three cycles. Two-dimensional cases are shown as they are representative of all of the methods evaluated.

RESULTS

The overall comparison of each method, including timing and memory information and based on the analytical test cases, is shown in Table 1. Average CPU Time is based on an average of all runs that were successfully completed. Average CPU Memory is based on the memory used to obtain all successful runs. The accuracy of each method is a qualitative assessment of the visual inspection and the quantitative results.

Infinite-Plate Splines - The overall results for the IPS method are mixed. The method showed good results in several test cases, but exhibited limiting behavior in many of the more rigorous test cases. IPS produced good results for all of the test cases in which the function was a constant or linear function, as seen by the relative errors in Figure 2. The relative errors were less than approximately 0.1%.

In contrast, IPS displayed difficulty in interpolating sinusoidal functions. These difficulties became increasingly more evident as the number of cycles in the sine function increased. There are significant errors in this case. The interpolated function appears to reproduce the number of cycles correctly with errors produced in the magnitude of the function. Changes in amplitude do not appear to affect the quality of the interpolation. It appears from Figure 2 that sinusoidal interpolations are very good. However, these data are based only on the few sinusoidal interpolations that the IPS was able to interpolate and thus does not include the errors for the cases where the method failed to resolve the function.

Extrapolation using this method ranges from good to very poor. What is not evident without additional plots is that the IPS method fails for higher sinusoidal functions. One of the limiting features of this method is a tendency to introduce oscillations where none originally existed, as illustrated in Figure 3 In addition, for data that change rapidly near the edge of the domain of the data points, an overprediction of the data curvature can be computed, leading to the "potato chip" effect of Figure 4.

Finite-Plate Splines - The overall accuracy of FPS is very good for the two-dimensional cases tested. The error varies across the range of data functions (Figure 5), typically vary less than 1% and for several functions << 1%. As in the IPS cases, the greatest error magnitudes were encountered for the sinusoidal test cases. In the majority of the test cases, there are substantial differences between the maximum and the average errors, indicating, and visually verified, that the method is sensitive to function variations on or near the edge of the mesh. For the two sinusoidal test cases selected for presentation (Figure 6), the FPS method produces a consistent, smooth interpolation.

The major disadvantage of this method is its very high CPU time and memory requirements compared to other methods (Table 1). FPS requires the definition of a virtual mesh that will cover the interface between the structural and the aerodynamic grids. The creation of this third surface and the number of points required for accurate interpolation can drive the CPU memory requirement well beyond a typical workstation's capacity.

Multiquadric-Biharmonic - The MQ method performs excellently for constant and most of the linear and sinusoidal functions (overall error for most cases << 1%). There is, however, a tendency toward higher errors for the sinusoidal functions (Figure 7). The method exhibited no sensitivity to any particular location or sensitivity to the grid spacing. Some sensitivity was noted to direction of the function when a large directional bias of the data is initially present (e.g., compare Figures 8a and 8b). A user input parameter, r, was introduced in the formulation in order to make the basis function infinitely differentiable. Larger values tend to better represent constant functions. Also, as pointed out by Kansa (1990), a varying r improves the conditioning number of the coefficient matrix of the linear system to be solved. For all the test cases run for the present study, the parameter was varied exponentially from 10^{-5} to 10^{-3}. Actually, no difference in the results was found for $r < 10^{-5}$; and for $r > 1$, even for some of the small test cases, the coefficient matrix becomes ill conditioned. Thus, for similar oscillatory patterns, the r parameter should remain in the range from 10^{-5} to 10^{-3}. The interpolated values for the sample sinusoidal functions are shown in Figure 8. The single cycle result shows only a slight clip at the corners, but the three-cycle sinusoid shows a higher-frequency oscillatory behavior in the direction orthogonal to the sinusoidal variation. Similar behavior was observed for extrapolations.

A subdomaining approach was also evaluated for the MQ method. Instead of solving a large single linear system problem, the original surface is subdivided into a prescribed number of subdomains. The function values within a subdomain are influenced only by the points within the same subdomain. There are some common overlapping regions where the quantities are blended using weighted averages, improving the continuity of the interpolated field. The implementation of the subdomain concept was done based on the maximum number of input (structural) points in each direction (*x, y*, and *z*) allowed in a given region. This approximately defines the size of the local linear system to be solved. More points enter in the region through predefined overlapping areas. Most of the cases were run using 20 as the maximum number of points in each direction and 10% overlapping. This gives a reasonable size of sub-problems to be solved and each subdomain samples a substantial portion of the original problem. Within each subdomain, the data were scaled to a unitary domain. Scaling the data may be essential in certain distorted grids, but for most of the cases tested it was not of any advantage over the direct use of the input data (Kansa, 1990). It is apparent that the main advantages of subdomaining are that the overall CPU time requirement decreases, that the dimensions of the arrays within the computer code can be reduced, thus reducing the overall memory requirements, and that the condition number of the coefficient matrix of the linear system increases (Kansa, 1990 and Smith et al., 1995) thus improving accuracy of the local solution.

Thin-Plate Splines - The TPS method is a hybrid of the Multiquadric-Biharmonic and the Infinite-Plate Spline Methods. Indeed, it is a local version of the latter, generalized to higher dimensionality, while its equations are identical to those of the former method except for the basis function used. The behavior of the TPS method (Figures 9 and 10) is a mix of the behaviors exhibited by the IPS and MQ methods, except that the problems experienced by these method appear to have been corrected in most instances. Some sensitivity exists with respect to grid clustering, but the maximum errors remain close to 1% or lower. The oscillations due to directional bias seen from the MQ and IPS methods are almost completed damped (Figure 10). Additionally, TPS does not require the input of the *r* parameter, as does the MQ method. The results from the overall set of test cases showed that TPS is the most robust and consistently accurate for both the two- and three-dimensional surfaces among all of the algorithms that were examined during this study.

Non-Uniform B-Splines - NUBS produced good results for most of the test cases. This method produced better or equivalent results compared with MQ and IPS for most test cases, with most errors less than 1% (Figure 11). Some sensitivity to grid clustering and surfaces in three-dimensions (shells) is noted by the larger relative errors, particularly for higher sinusoidal functions (>10%).

The current formulation of NUBS is somewhat inconsistent. Some of the simulations have some "oscillations" forming in the resulting contours for some test cases, similar to the MQ implementation, while other simulations are very exact, as depicted in Figure 12. The oscillations, when they occur, appear to be caused by minute differences when the linear bivariate interpolation scheme correlates the known function grid and the unknown function grid, causing the point to be incorrectly located with respect to the function. The requirement of this search algorithm is a negative when comparing this method with the others, but it does permit the user to access accurate CAD databases.

Inverse Isoparametric Mapping - This methodology is a two-dimensional application, so that the testing was constrained by the following: two-dimensional surfaces (plates), regular grids, no extrapolation, and no beam element implementation. The overall accuracy of the code for the test cases examined was very good. The maximum error encountered was approximately 5%. Most of the errors remained much lower than 1%, as seen in Figure 13. The method had no problems running any of the two-dimensional interpolation test cases, and showed no directional bias or amplitude sensitivity. The errors computed for the one- and three-cycle sinusoid cases are indistinguishable from the original functions, and are thus not shown.

CONCLUSIONS

A selected set of algorithms was tested using superposition of mathematical functions to evaluate their accuracy, smoothness and robustness when performing interpolation and extrapolation between non-contiguous meshes. Limited examples of the full set of 260 test cases were presented here, but an expanded examination of the test cases can be found in Smith et al. (1995). These limited examples demonstrate the primary features of the results determined from full set of test cases. Additional studies of most of these algorithms, as applied to typical aeroelastic aircraft applications, can be found in Smith et al. (2000).

The results of the full set of 260 test cases, comprised of interpolation and extrapolation of combinations of functions ranging from constant to multiple-superimposed sinusoidal functions, are summarized here for the convenience of the reader:

- The Infinite-Plate Spline (IPS) is the method applied in many codes where interpolations are necessary. IPS generates errors on the order of 10^{-5}% or smaller on meshes where the data does not rapidly vary or where there is not a strong directional bias in the data. For data with large gradients, the interpolation can vary up to 10% from the original data. It is not recommended for extrapolations, particularly where the data varies rapidly near the edges of the mesh. While the IPS method works well for some applications, it is not as robust and accurate as other methods which are readily available. If the data are rapidly varying, it can take up to 50 times the CPU time required for other methods, and is prone to failure.

- The Thin-Plate Spline (TPS) method is a generalization of IPS to multiple dimensions. It corrects most of the IPS problems, and yields results on the order of 10^{-5}% and lower except for highly varying sinusoidal data, where the relative errors remain at approximately 1%. It may introduce some small high frequency oscillations when interpolating or extrapolating some rapidly varying data. This method was the most general, handling all of the test cases, and the most robust of the methods tested.
- The Multiquadric-Biharmonic (MQ) method yielded results similar to the TPS method. Its computational requirements were approximately the same (low). MQ generated results with either 10^{-14} % or 1% relative errors. Larger errors were seen when the gradients of the data were large. High frequency oscillations were noticeable for rapidly varying data. The MQ method also requires a user-input parameter, r, which can affect the accuracy of the interpolations.
- Non-Uniform B-Spline (NUBS) algorithm requires approximately the same computational time as MQ and TPS if a simple bilinear search algorithm is implemented. The Non-Uniform B-Spline method requires additional development to develop an accurate search algorithm to apply the interpolation, and requires a larger computational memory than the Multiquadric-Biharmonic and Thin-Plate Spline methods. This method does not perform as well on three-dimensional configurations (errors of approximately 10%), as it does on two-dimensional configurations (errors typically on the order of 10^{-14}%).
- The Finite-Plate Spline (FPS) method provided generally excellent interpolations for the two-dimensional cases (plates) attempted. The differences between the maximum and average relative errors indicate that the method is sensitive to the location of the data gradients. In all but the sinusoidal test cases, the relative errors were less than 1%. The primary drawback of this method is its computational memory requirements, which - even with subdomaining - can rapidly outpace a workstation's memory.

- The Inverse Isoparametric Mapping (IIM) shows the best accuracy of all the methods, however it is currently only available for two-dimensional (plate) applications. It generally had relative errors much less than 1%, and only exceeded 1% for high frequency (5 - 9) oscillations. No directional bias was noted.

The following conclusions/recommendations for application of these methods are suggested based on the quantitative and qualitative results:

- The best overall, robust method is Thin Plate Splines. Note that some small high frequency oscillations do occur for rapidly fluctuating data, particularly in one direction.
- The most accurate of all methods is the Inverse Isoparametric Mapping, but it is limited to two-dimensional meshes with no extrapolation.
- The second most accurate method, with no oscillatory problems, is the Non-Uniform B-Spline method. However, a second-order or higher search algorithm is necessary to implement this method, which may result in a significant CPU time penalty.
- The Finite Plate Spline method is also recommended for accurate smoothness in the interpolated data, but it requires CPU and memory that may be beyond typical workstations.
- The Infinite Plate Spline method, if already implemented, is an excellent interpolation method for two-dimensional meshes where the data gradients are not rapidly changing. The Thin Plate Spline method, a derivation of the IPS method in multiple dimensions, is recommended should the reservations above not be compatible with the intended application.

ACKNOWLEDGMENTS

This research was funded by the Air Force Research Laboratory Aeronautical Sciences Division (AFRL/VAAC) via contract F33615-94-C-3010, Task 001. Captain Kenneth Moran was the Task 001 contract monitor; Dr. Donald Kinsey was the task order contract manager..

The authors would like to acknowledge the technical expertise provided to the authors on algorithms developed or coded by Dr. Richard Franke of Naval Postgraduate School, Dr. Kari Appa of Northrop Corporation, Dr. Robert Ames of David Taylor Research Center, and Dr. Robert Fithen of the University of New Orleans.

Appendix I. References

Appa, K. (1989). "Finite-Surface Spline," *Journal of Aircraft*, Vol. 26, No. 5, pp. 495-496.

Boeing Computer Services (1995). *DT_NURBS Spline Geometry Subprogram Library User's Manual*, Version 2.3, February, pp. 3-1 - 4-5.

Conte, S. D., and de Boor, C. (1980). *Elementary Numerical Analysis*, McGraw-Hill, Inc., New York, New York, 1980, pp. 289-293.

Done, G. T. S. (1965). "Interpolation of Mode Shapes: A Matrix Using Two-Way Spline Curves," Aeronautical Quarterly, Vol. XVI, Part 4, pp. 333-349.

Duchon, J. (1977). "Splines Minimizing Rotation-Invariant Semi-Norms in Sobolev Spaces," In Schempp, W. and Zeller, K., editors, *Constructive Theory of Functions of Several Variables, Oberwolfach 1976*, pp. 85-100, Springer-Verlag, Berlin.

Fithen, R. (1995). Private correspondence.

Franke, R. (1982). "Scattered Data Interpolation: Tests of Some Methods," *Mathematics of Computations*, Vol. 38, No. 157 (January), pp. 181-200.

Harder, R. L., and Desmarais, R. N. (1972). "Interpolation Using Surface Splines," AIAA Journal, Vol. 10, No. 2, pp. 189-191.

Hardy, R. L. (1971). "Multiquadric Equations of Topography and Other Irregular Surfaces," *Journal of Geophysical Research*, Vol. 76, No. 8, March, pp. 1905-1915.

Kansa, E. J. (1990). "Multiquadrics - A Scattered Data Approximation Scheme with Applications to Computational Fluid Dynamics, Parts I and II," *Computers and Mathematics with Applications*, Vol. 19, No. 8/9, pp. 127 - 161.

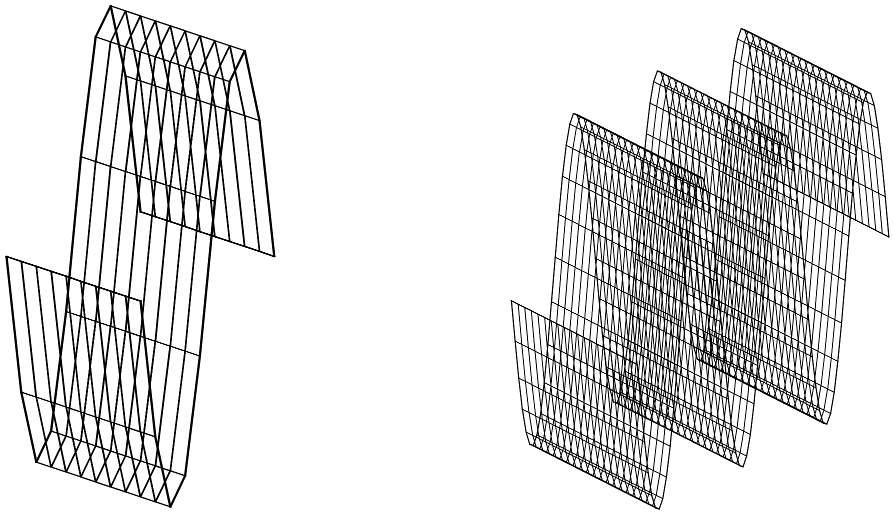
MacNeal-Schwindler Corp . (1987). “Handbook for Aeroelastic Analysis,” Vol. I and II,., November 1987.

Smith, M. J., Hodges, D. H., and Cesnik, C. E. S. (2000). “An Evaluation Of Computational Algorithms Suitable for Fluid-Structure Interaction,” to appear in the *AIAA Journal of Aircraft*, Vol. 37, No. 2.

Smith, M. J., Hodges, D. H., and Cesnik, C. E. S. (1995). “An Evaluation Of Computational Algorithms To Interface Between CFD And CSD Methodologies,” WL-TR-96-3055.

Table 1. Comparative Results for the Six Interface Methods for the Analytical Test Cases

Method	Avg. CPU Time (Sec)	Avg. CPU Mem. (MB)	Accuracy
IPS	100	160	Inconsistent
FPS (2D)	93	180	Good
MQ	2.4	33	Good
TPS	1.9	33	Excellent
NUBS	2.4	15	Good, Some Inconsistency
IIM (2D)	3.1	4.5	Excellent



a) One-Cycle Oscillation b) Three-Cycle Oscillations

Figure 1. Examples of Sinusoidal Functions with a Peak-to-Peak Amplitude of 2.

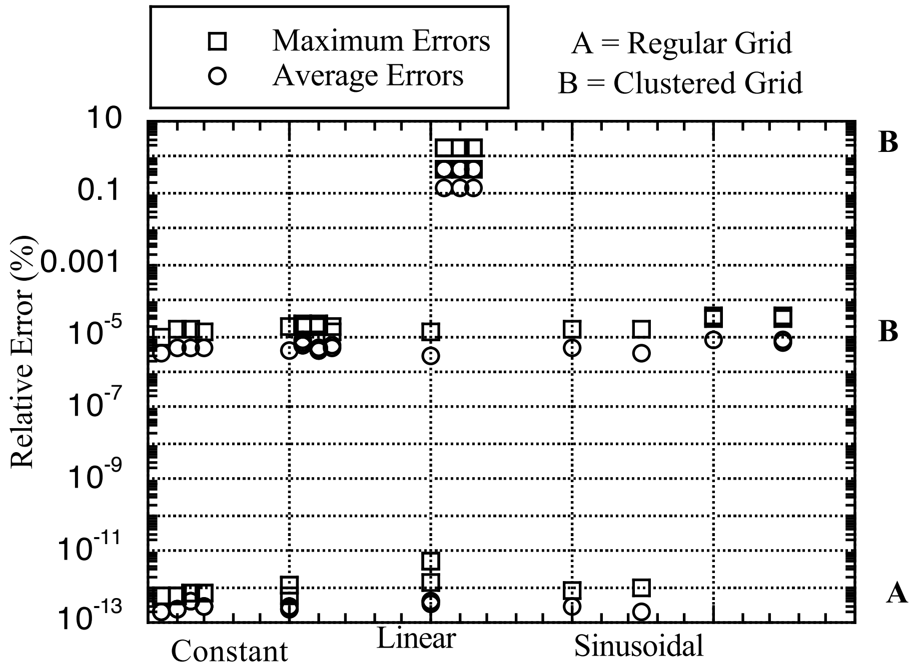
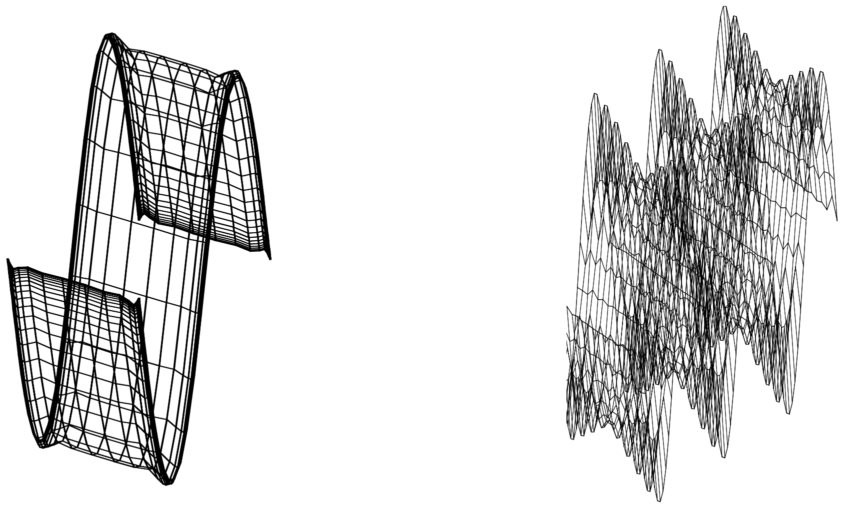


Figure 2. Variation of Error for Function Type Using the Infinite-Plate Spline Method



a) One-Cycle Sinusoid b) Three-Cycle Sinusoid

Figure 3. Example of Oscillations Induced by the Infinite-Plate Spline Method for Sinusoidal Functions at a Peak-to-Peak Amplitude of 2

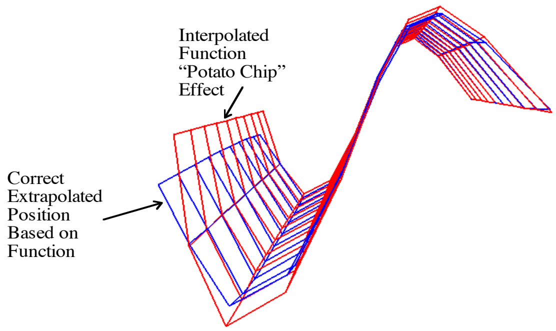


Figure 4. An Illustration of the “Potato Chip” Extrapolation Effect Encountered in the Infinite-Plate Spline Method

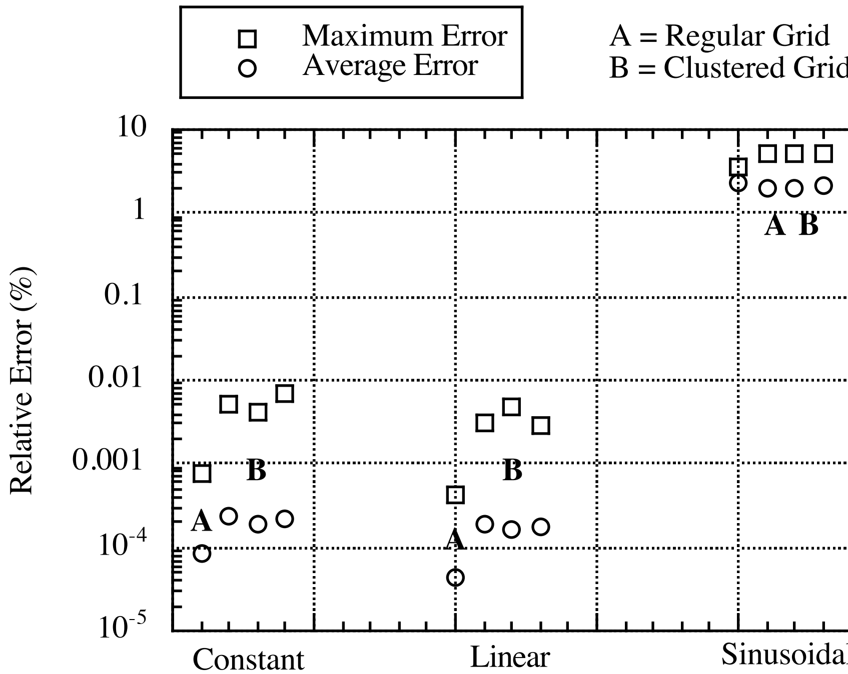
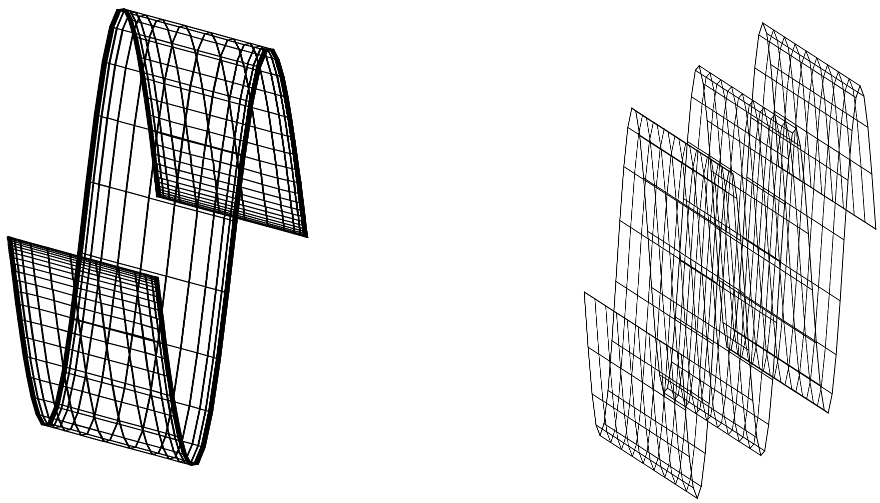


Figure 5. Variation of Error for Test Set One Based Upon Function Type Using Finite-Plate Splines



a) One-Cycle Sinusoid

b) Three-Cycle Sinusoid

Figure 6. Example of Interpolations Computed by the Finite-Plate Spline Method for Sinusoidal Functions at a Peak-to-Peak Amplitude of 2

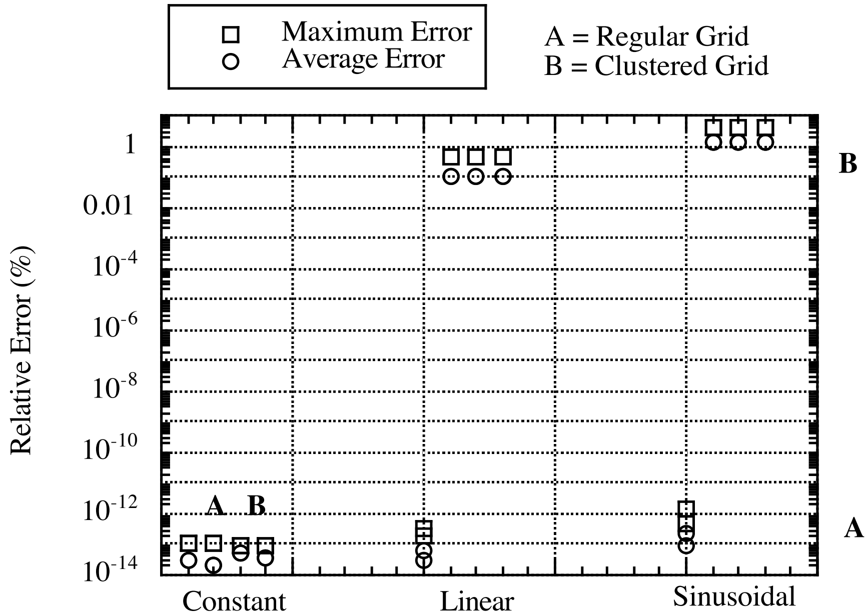
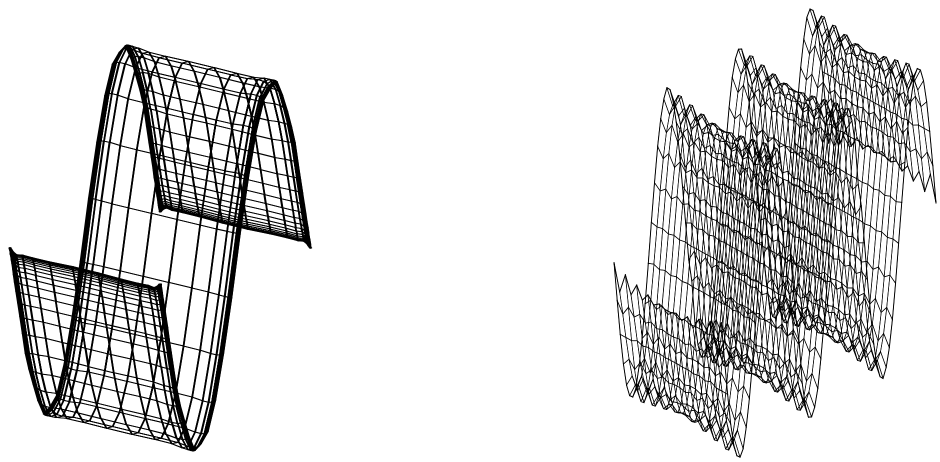


Figure 7. Variation of Error Based Upon Function Type Using Multiquadric-Biharmonic Method



a) One-Cycle Sinusoid b) Three-Cycle Sinusoid

Figure 8. Example of Interpolations Computed by the Multiquadric-Biharmonic Method for Sinusoidal Functions at a Peak-to-Peak Amplitude of 2

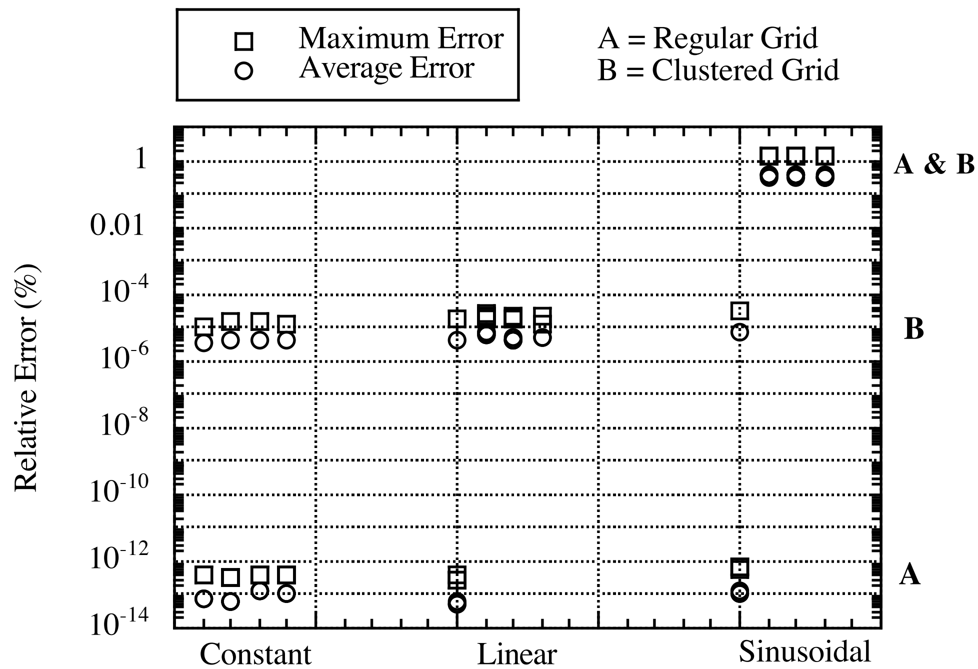
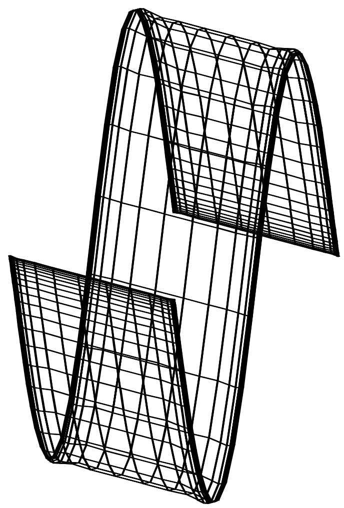
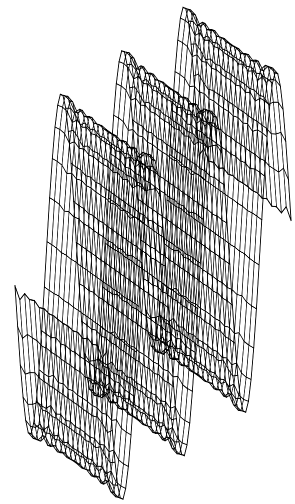


Figure 9. Variation of Error Based Upon Function Type Using Thin-Plate Splines



a) One-Cycle Sinusoid



b) Three-Cycle Sinusoid

Figure 10. Example of Interpolations Computed by the Thin-Plate Spline Method for Sinusoidal Functions at a Peak-to-Peak Amplitude of 2

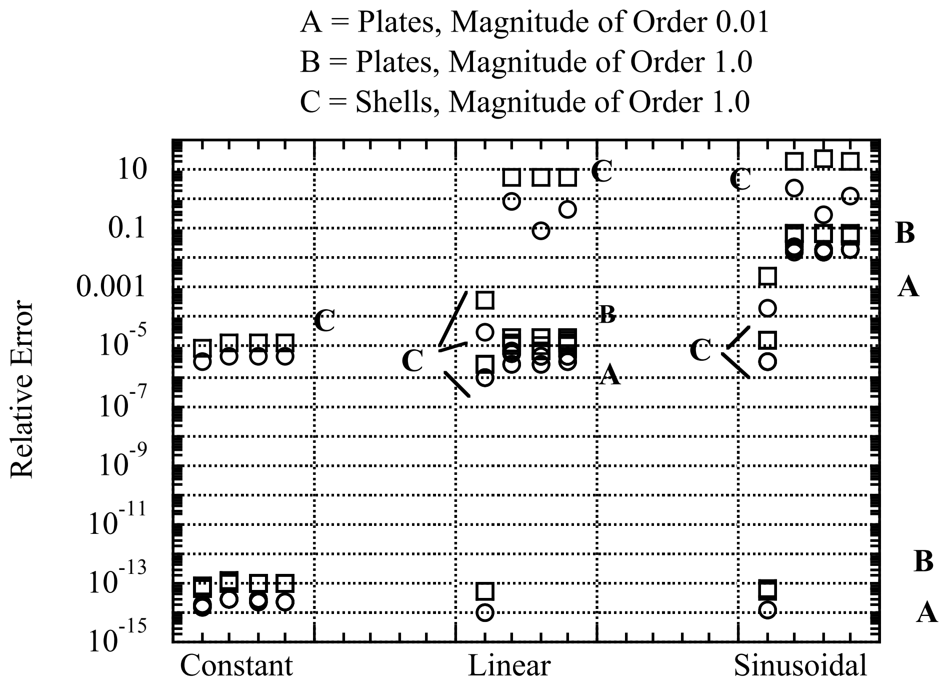
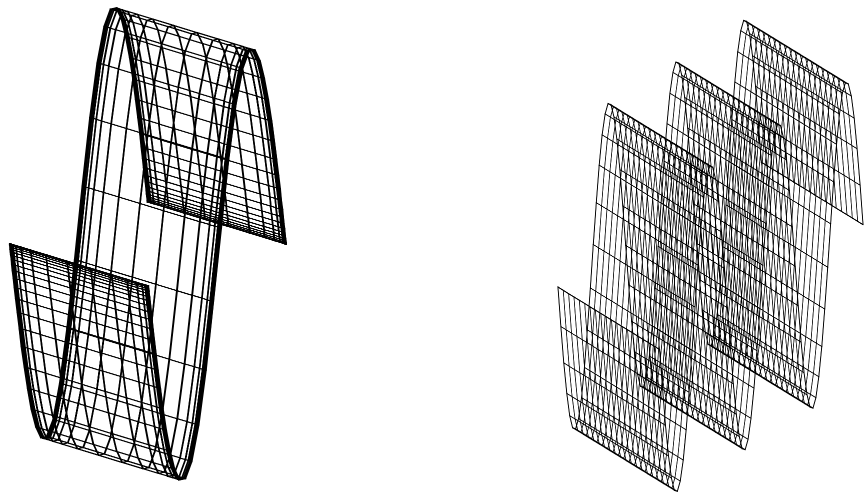


Figure 11. Variation of Error Based Upon Function Type for Plates Using the NUBS Method



a) One-Cycle Sinusoid

b) Three-Cycle Sinusoid

Figure 12. Example of Interpolations Computed by the Non-Uniform B-Spline Method for Sinusoidal Functions at a Peak-to-Peak Amplitude of 2

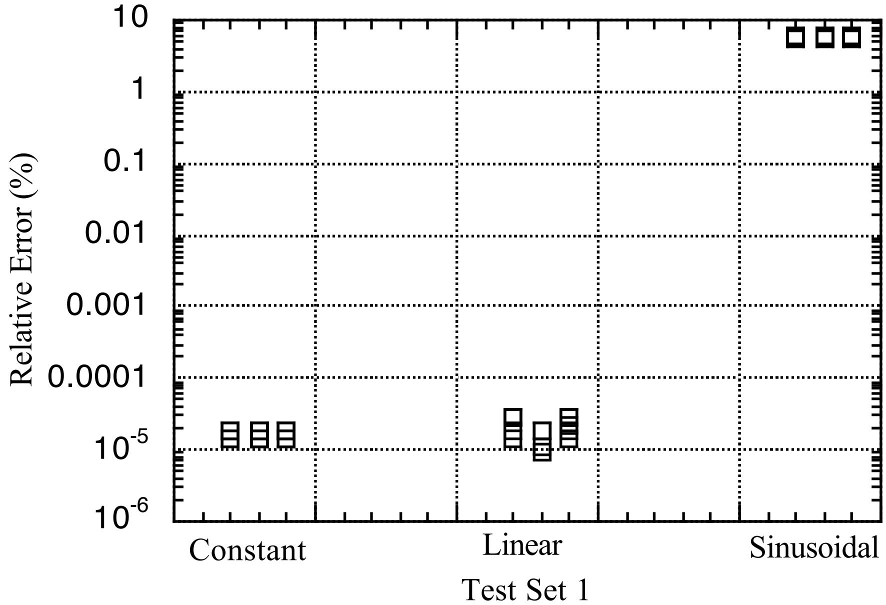


Figure 13. Variation of Error for Test Set One Based Upon Function Type Using the Inverse Isoparametric Method (Regular Grid Errors are all 0.0 and are not Plotted)

Figure 1. Examples of Sinusoidal Functions with a Peak-to-Peak Amplitude of 2.

Figure 2. Variation of Error for Function Type Using the Infinite-Plate Spline Method

Figure 3. Example of Oscillations Induced by the Infinite-Plate Spline Method for Sinusoidal Functions at a Peak-to-Peak Amplitude of 2

Figure 4. An Illustration of the "Potato Chip" Extrapolation Effect Encountered in the Infinite-Plate Spline Method

Figure 5. Variation of Error for Test Set One Based Upon Function Type Using Finite-Plate Splines

Figure 6. Example of Interpolations Computed by the Finite-Plate Spline Method for Sinusoidal Functions at a Peak-to-Peak Amplitude of 2

Figure 7. Variation of Error Based Upon Function Type Using Multiquadric-Biharmonic Method

Figure 8. Example of Interpolations Computed by the Multiquadric-Biharmonic Method for Sinusoidal Functions at a Peak-to-Peak Amplitude of 2

Figure 9. Variation of Error Based Upon Function Type Using Thin-Plate Splines

Figure 10. Example of Interpolations Computed by the Thin-Plate Spline Method for Sinusoidal Functions at a Peak-to-Peak Amplitude of 2

Figure 11. Variation of Error Based Upon Function Type for Plates Using the NUBS Method

Figure 12. Example of Interpolations Computed by the Non-Uniform B-Spline Method for Sinusoidal Functions at a Peak-to-Peak Amplitude of 2

Figure 13. Variation of Error for Test Set One Based Upon Function Type Using the Inverse Isoparametric Method (Regular Grid Errors are all 0.0 and are not Plotted)