https://ntrs.nasa.gov/search.jsp?R=20070030905 2018-07-24T00:55:22+00:00Z

Source of Acquisition
NASA Goddard Space Flight Center

Efficient Implementation of an Optimal Interpolator for Large Spatial Data Sets

Nargess Memarsadeghi*[1,2] and David M. Mount[2]
(CS-TR-4856)

[1] NASA/GSFC, Code 588, Greenbelt, MD, 20771
[2] University of Maryland, College Park, MD, 20742

Abstract. Interpolating scattered data points is a problem of wide ranging interest. A number of approaches for interpolation have been proposed both from theoretical domains such as computational geometry and in applications' fields such as geostatistics. Our motivation arises from geological and mining applications. In many instances data can be costly to compute and are available only at nonuniformly scattered positions. Because of the high cost of collecting measurements, high accuracy is required in the interpolants. One of the most popular interpolation methods in this field is called *ordinary kriging*. It is popular because it is a best linear unbiased estimator. The price for its statistical optimality is that the estimator is computationally very expensive. This is because the value of each interpolant is given by the solution of a large dense linear system. In practice, kriging problems have been solved approximately by restricting the domain to a small local neighborhood of points that lie near the query point. Determining the proper size for this neighborhood is a solved by *ad hoc* methods, and it has been shown that this approach leads to undesirable discontinuities in the interpolant.
Recently a more principled approach to approximating kriging has been proposed based on a technique called *covariance tapering*. This process achieves its efficiency by replacing the large dense kriging system with a much sparser linear system. This technique has been applied to a restriction of our problem, called *simple kriging*, which is not unbiased for general data sets. In this paper we generalize these results by showing how to apply covariance tapering to the more general problem of ordinary kriging. Through experimentation we demonstrate the space and time efficiency and accuracy of approximating ordinary kriging through the use of covariance tapering combined with iterative methods for solving large sparse systems. We demonstrate our approach on large data sizes arising both from synthetic sources and from real applications.

1 Introduction

Scattered data interpolation is a problem of interest in numerous areas such as electronic imaging, smooth surface modeling, and computational geometry [1,2].

* Corresponding author: NASA GSFC, Architectures and Automation Branch, Code 588, Greenbelt, MD, 20771, Tel: 301-286-2938, and Department of Computer Science, University of Maryland, College Park, MD, 20742. Email: nargess@cs.umd.edu.

Our motivation arises from applications in geology and mining, which often involve large scattered data sets and a demand for high accuracy. The method of choice (sometimes called the "gold standard" in this area [10]) is *ordinary kriging*. This is because it is a best unbiased estimator [6–8]. Unfortunately, this interpolant is computationally very expensive to compute exactly. The reason is that for n scattered data points, computing the value of a single interpolant involves solving a dense linear system of size roughly $n \times n$. This is infeasible for large n. In practice, kriging is solved approximately by local approaches that are based on considering only a relatively small number of points that lie close to the query point [6, 7]. There are many problems with this local approach, however. The first is that determining the proper neighborhood size is tricky, and is usually solved by *ad hoc* methods such as selecting a fixed number of nearest neighbors or all the points lying within a fixed radius. Such fixed neighborhood sizes may not work well for all query points, depending on local density of the point distribution [6]. Local methods also suffer from the problem that the resulting interpolant is not continuous. Meyer showed that while kriging produces smooth continues surfaces, it has zero order continuity along its borders [10]. Thus, at interface boundaries where the neighborhood changes, the interpolant behaves discontinuously. Therefore, it is important to consider and solve the global system for each interpolant. However, solving such large dense systems for each query point is impractical.

Recently a more principled approach to approximating kriging has been proposed based on a technique called *covariance tapering* [5]. We will discuss these issues in greater detail below, but the problems arise from the fact that the covariance functions that are used in kriging have global support. In tapering these functions are approximated by functions that have only local support, and that possess certain necessary mathematical properties. This achieves greater efficiency by replacing large dense kriging systems with much sparser linear systems. Covariance tapering has been successfully applied to a restriction of our problem, called *simple kriging* [5]. Simple kriging is not an unbiased estimator for stationary data whose mean value differs from zero, however. In this paper we generalize these results by showing how to apply covariance tapering to the more general problem of ordinary kriging. Ordinary kriging ensures unbiasedness for stationary data, whose mean can have any value.

Our implementations combine and utilize and enhance a number of different approaches that have been introduced in literature for solving large linear systems for interpolation of scattered data points. For very large systems, exact methods such as Gaussian elimination are impractical since they require $O(n^3)$ time and $O(n^2)$ storage. As Billings *et al.* [3] suggested, we use an iterative approach. In particular we use the SYMMLQ method [14], for solving the large but sparse ordinary kriging systems that result from tapering.

There are a number of technical issues that need to be overcome in our algorithmic solution. We will see in Section 2 that the points' covariance matrix for kriging should be symmetric positive definite. The goal of tapering is to obtain a sparse approximate representation of the covariance matrix while maintain-

ing its positive definiteness. Furrer *et al.* [5] used tapering to obtain a sparse linear system of the form $Ax = b$, where A is the tapered symmetric positive definite covariance matrix. Thus, Cholesky factorization [9] could be used solve the linear system for their application. They further utilized the sparseness of A to implement an efficient sparse Cholesky decomposition method for solving the linear system. They also applied the Fast Fourier Transform in conjunction with their sparse implementation to obtain better efficiency in solving the system. In addition, they show if these tapers are used for a limited class of covariance models, the solution of the system converges to the solution of the original system. While their results show significant improvements over dense Cholesky factorization, their approach is not applicable to the ordinary kriging problem. This is mainly due to the fact that matrix A in the ordinary kriging linear system, while symmetric, is not positive definite. We will discuss details of the ordinary kriging system in Section 2.

In ordinary kriging, additional constraints are imposed on the interpolation coefficients to ensure the unbiasedness of the estimator. These constraints result in one or more additional rows and columns in matrix A. As we will see in Section 2, these constraints result in a matrix that fails to be positive-definite. Thus, efficient implementations of Cholesky factorization (which require a positive definite matrix) are not applicable to the ordinary kriging problem. Therefore, we use tapering only to obtain a sparse linear system, and then use a sparse iterative method to solve our linear systems. In particular, we use SYMMLQ method which is an iterative method for solving symmetric but not positive definite systems [14].

We have also developed a more efficient variant of the SYMMLQ method to solve large ordinary kriging systems. This variant is obtained by projecting our global system to an appropriate lower dimensional system. This approach can be viewed as adaptively finding the correct local neighborhood for each query point in the ordinary kriging interpolation process. We compare both quality of our results and running times with those obtained using traditional approaches based on neighborhood sizes for solving large ordinary kriging systems. We show that solving large kriging systems becomes practical via tapering and iterative methods, and results in lower estimation errors compared to traditional local approaches, and significant memory savings compared to the original global system. We achieve further significant speed-ups by introducing a variant of the global tapered system.

The remainder of the paper is organized as follows. We start in the next section with a review of the ordinary kriging. In Section 3 we describe the tapering properties as mentioned in [5] and the tapering functions used for our experiments. We proceed by introducing our approaches for solving the ordinary kriging problem in Section 4. Section 5 describes data sets used in this paper. Then, we describe our experiments and results in Section 6. We conclude the paper in Section 7.

2 Ordinary Kriging

Kriging is an interpolation method named after Danie Krige, a South African mining engineer, who pioneered in the field of geostatistics [6]. Kriging is also referred to as the *Gaussian process predictor* in the machine learning domain [16]. Kriging and its variants have been traditionally used in mining and geostatistics applications [6–8]. The most commonly used variant is called *ordinary kriging*, which is often referred to as a BLUE method, that is, a Best Linear Unbiased Estimator [5, 7]. Ordinary kriging is considered to be *best* because it minimizes the variance of the estimation error. It is *linear* because estimates are weighted linear combination of available data, and is *unbiased* since it aims to have the mean error equal to zero [7]. Minimizing the variance of the error forms the objective function of an optimization problem. Ensuring unbiasedness of the error imposes a constraint on this objective function. We proceed by explaining how this optimization problem is formalized, subject to the mentioned constraint.

The estimate of a random function U at location x_0, $\hat{u}_0$, using known variable values at n nearby locations has the form $\hat{u}_0 = \sum_{i=1}^{n} a_i u_i$ ([7]). In this linear combination, $u_1, u_2, \ldots, u_n$ are data values at n nearby locations. For simplicity, assume that locations of these points are presented as $x_1, \ldots, x_n$. In kriging interpolation methods, for any point where one performs the estimation, a *stationary* random function model with multiple random variables (one at each location) is assumed. That is $u_1, u_2, \ldots, u_n$ are viewed as values of n different random variables $U_1(x_1), U_2(x_2), \ldots, U_n(x_n)$, where $u_1 = U_1(x_1), \ldots, u_n = U_n(x_n)$. Stationarity of the random function model, implies that all these variables have the same mean, $E(U)$. The estimation error, R, is calculated as the difference between values of the actual random variable value $\widehat{U}(x_0)$ and the estimated random variable $U(x_0)$.

$$R(x_0) = \widehat{U_0}(x_0) - U_0(x_0) = \hat{u}_0 - u_0 = w^t Z, \tag{1}$$

where $w^t = (w_1, \ldots, w_n, -1)$, $Z^t = (u_1, \ldots, u_n, u_0)$, and $w_1 \ldots w_n$ are weights used in estimating $\widehat{U_0}(x_0)$. It is also known that the variance of a random variable created as a linear combination of other random variables, $V_1 \ldots V_n$, is estimated as follows (see [7], p. 216):

$$Var\left(\sum_{i=1}^{n} w_i' V_i\right) = \sum_{i=1}^{n}\sum_{j=1}^{n} w_i' w_j' \, Cov\left(V_i V_j\right). \tag{2}$$

where $V_1 \ldots V_n$ are random variables at given locations, and $w_1' \ldots w_n'$ are the weights associated with them. Equations 1 and 2 imply the following objective function for minimizing the variance of the estimation error.

$$Var(R(x_0)) = w^t C_Z w \;=\; \sum_{i}^{n}\sum_{j}^{n} w_i w_j C_{ij} \;-\; 2\sum_{i}^{n} w_i C_{i0} + C_0, \tag{3}$$

where $C_{ij} = Cov(U_i, U_j)$, $C_0 = Cov(U_0, U_0) = Var\,(U_0)$, and $w_1 \ldots w_n$ are unknown values which need to be found such that the above objective function

is minimized. Unbiasedness of estimates is ensured by adding additional appropriate constraints to the mentioned objective function. The following is proven in [7].

Lemma 1 (Isaaks and Srivastava [7]). *A kriging estimate of an stationary variable is unbiased iff sum of its kriging weights is 1.*

Adding this constraint to Eq. (3) introduces a Lagrange parameter, μ, to our system [13].

$$Var(R(x_0)) = \sum_i^n \sum_j^n w_i w_j C_{ij} \; - \; 2\sum_i^n w_i C_{i0} \; + \; C_0 \; + \; 2\mu(\sum_{i=1}^n w_i - 1). \quad (4)$$

Next, partial derivatives of the objective function with respect to $w_1, \ldots, w_n$ are taken and are set to zero. This results in $(n+1)$ equations, which are equivalent to the following system.

$$\begin{pmatrix} C & L \\ L^t & 0 \end{pmatrix} \begin{pmatrix} w \\ \mu \end{pmatrix} = \begin{pmatrix} C_0 \\ 1 \end{pmatrix}, \quad (5)$$

where C is the matrix of pairwise covariances for $U_1, \ldots, U_n$, L is a column vector of all ones and of size n, and w is the vector of weights $w_i, \ldots, w_n$. Therefore, the minimization problem for n points reduces to solving a linear system of size $(n+1)^2$, which is impractical for very large data sets via direct approaches. Note that the coefficient matrix in the above linear system is a symmetric matrix which is *not* positive definite since it has a zero entry on its diagonal.

Lemma 2 (Myers [12]). *The ordinary kriging system described in Eq. (5) has a solution if C is positive definite.*

Lemma 3 (Isaaks and Srivastava [7]). *The variance of the ordinary kriging estimation error is positive if C is positive definite. (This can be observed via Eq. (2).)*

Pairwise covariances are modeled as a function of points' separation. These functions should result in a positive definite covariance matrix. Christakos [4] showed necessary and sufficient conditions for such permissible covariance functions. Two of these valid covariance functions, used in this paper, are the Gaussian and Spherical covariance functions (C_g and C_s respectively).

$$C_g(h) = \exp\left(-\frac{3h^2}{a^2}\right), \quad (6)$$

$$C_s(h) = \begin{cases} 1 & \text{if} \quad h = 0; \\ 1 - 1.5\frac{h}{a} + 0.5(\frac{h}{a})^3 & \text{if} \quad 0 < h \leq a, \\ 0 & \text{otherwise} \end{cases} \quad (7)$$

where a is the *range* for the covariance values, and h is the Euclidean distance of a pair of points. The range is the distance after which the covariance values remain constant at their lowest possible value. Please see [4, 6, 7] for other examples of permissible covariance functions.

3 Tapering Covariances

Tapering covariances for the kriging interpolation problem, as described in [5], is the process of obtaining a sparse representation of the points' pairwise covariances so that positive definiteness of the covariance matrix as well as the smoothness property of the covariance function be preserved. One may first think of assigning zero to pairwise covariances of points that are further than a threshold distance, θ, from each other (truncating by assigning zeros to very small covariance values). However, the modified covariance matrix obtained in this manner may not necessarily be positive definite. It can be shown that the covariance matrix obtained in this manner is positive definite if the norm of the total change to the covariance matrix is smaller than the smallest eigenvalue of the covariance matrix. However, estimating or calculating the smallest eigenvalue of a large matrix reduces to solving a large linear system of the same size as the original problem. Instead, the sparse representation via tapering is obtained through the Schur product of the original positive definite covariance matrix by another positive definite covariance matrix.

$$C_{tap}(h) = C(h) \times C_\theta(h). \tag{8}$$

The resulting tapered covariance matrix, C_{tap}, has zero values for points that are more than a certain distance apart from each other. It is also positive definite since the Schur product of two positive definite matrices are positive definite. A taper is considered valid for a covariance model if it perseveres its positive-definiteness property and makes the approximate system's solution converge to the original system's solution.

The authors of [5] mention few valid tapering functions. They also showed that tapers need to be as smooth as the original covariance function to ensure convergence to the original system's solution. In this paper, we used the following tapers.

$$Spherical = \left(1 - \frac{h}{\theta}\right)_+^2 \left(1 + \frac{h}{2\theta}\right), \tag{9}$$

$$Wendland_1 = \left(1 - \frac{h}{\theta}\right)_+^4 \left(1 + \frac{4h}{\theta}\right), \tag{10}$$

$$Wendland_2 = \left(1 - \frac{h}{\theta}\right)_+^6 \left(1 + \frac{6h}{\theta} + \frac{35h^2}{3\theta^2}\right), \tag{11}$$

$$TopHat = \left(1 - \frac{h}{\theta}\right)_+, \tag{12}$$

where $x_+ = \max\{0, x\}$ and θ is chosen so that pairwise covariances can be supported in $[0, \theta)$. Note that the above tapers result in positive definite covariance functions in $\mathbb{R}^3$ and lower dimensions [5]. However, considering convergence to the optimal estimator, these tapers are not valid for all covariance functions. Tapers need to be as smooth as the original covariance function at origin to

guarantee convergence to the optimal estimator [5]. Thus, for a Gaussian covariance function, which is infinitely differentiable, no taper exists that satisfies this smoothness requirement. However, since tapers proposed in [5] still maintain positive definiteness of the covariance matrices, we examined using these tapers for Gaussian covariance functions as well. For this paper, we are using these tapers mainly to build a sparse approximate system to our original global system even though these tapers do not guarantee convergence to the optimal solution of the original global dense system theoretically. Of the above mentioned tapering functions, the top hat taper is closest to the truncation idea while guaranteeing positive definiteness of the covariance matrix.

4 Our Approaches

We implemented and examined both local and global interpolation methods for the ordinary kriging interpolation problem as follows.

Local Methods: This is the traditional and the most common way of solving kriging systems. That is, instead of considering all known values in the interpolation process, points within a neighborhood of the query point are considered. Neighborhood sizes are defined either by a **fixed number** of points closest to the query point or by points within a **fixed radius** from the query point. Therefore, the problem is solved locally. We experimented our interpolations using both of these local approaches. We defined the fixed radius to be the distance beyond which correlation values are less than 10^{-6} of the maximum correlation. Similarly, for the fixed number approach, we used maximum connectivity degree of points' pairwise covariances, when covariance values are larger than 10^{-6} of the maximum covariance value. Gaussian elimination [13] was used for solving the local linear systems in both cases.

Global Tapered Methods: In global tapered methods we first redefine our points' covariance function to be the tapered covariance function obtained through Eq. (8), where $C(h)$ is the covariance function which was used (Eq. (6) or (7)), and $C_\theta(h)$ is one of the tapering functions defined in Section 3. We then solve the linear system using the SYMMLQ approach as mentioned in [14]. Note that, while one can use conjugate gradient method for solving symmetric systems, the method is guaranteed to converge only when the coefficient matrix is both symmetric and positive definite [19]. Since ordinary kriging systems are symmetric and not positive definite, we used SYMMLQ. We modified the original SYMMLQ implementation to take advantage of the sparseness of the matrix A, similar to the sparse conjugate gradient implementation mentioned in [15]. Note that in [15]'s implementation, matrix elements that are less than or equal to a *threshold* value are ignored. Since we obtain sparseness through tapering, this *threshold* value for our application is zero. One appealing approach would be to obtain a sparse system by having a small nonzero *threshold* value, instead of obtaining sparseness through tapering. However, as mentioned before, this approach does not necessarily result in a positive definite covariance matrix, and

that is the main reason why tapering functions are of great value for kriging applications [5].

Global Tapered and Projected Methods: This implementation is motivated by numerous empirical results in geostatistics which show that interpolation weights associated with points that are very far from the query point tend to be very close to zero. That is, very far points do not seem to contribute much to the interpolation weights. This phenomenon is called the *screening effect* in the geostatistical literature [20]. Stein showed conditioned under which the screening effect occurs for gridded data [20]. While the existence of the screening effect has been the basis for using local methods in the past, there is no proof of this empirically supported idea for scattered data points [5]. We use this conjecture for solving the global ordinary kriging system $Ax = b$ and observing that many elements of b are zero after tapering. This indicates that for each zero element b_i , representing the covariance between the query point and the ith data point, we have $C_{i0} = 0$. Thus, we expect their associated interpolation weight, w_i, to be very close to zero. We assign zero to such w_i's, and consider solving a smaller system $A'x' = b'$, where b' consists of nonzero entries of b. We store indices of nonzero rows in b in a vector called *indices*. A' contains only those elements of A whose row and column indices both appear in *indices*. Then, we solve the projected system $A'x' = b'$. This method is effectively the same as the fixed radius neighborhood size. The difference is that the local neighborhood is found adaptively by looking at covariance values in the global system for each query point. There are several differences between this approach and the local methods. One is that we build the global matrix A once, and use relevant parts of it, contributing to nonzero weights, for each query point. Second, for each query, the local neighborhood is found adaptively by looking at covariance values in the global system. Third, the covariance values are modified through tapering.

5 Data Sets

We need large scattered data sets to test and evaluate performance of various approaches mentioned in Section 4. As mentioned before, we cannot solve the original global systems exactly for very large data sets, and thus cannot compare our solutions with respect to the original global systems. Therefore, we would need ground truth values for our data sets. Also, since performance of local approaches can depend on data points' density around the query point, we would like our data sets to be scattered non-uniformly. Therefore, we create our scattered data sets by sampling points of a large dense grid from both uniform and Gaussian distributions. Values of the dense grid are either synthetically generated or are real measurements.

We generated our synthetic data sets using the Sgems [18] software. We generated values on a (1000×1000) grid. Values were generated using the Sequential Gaussian Simulation (sgsim) algorithm of the Sgems software (please see [17,18] for more details). Points were simulated through ordinary kriging with a Gaussian covariance function (Eq. (6)) of range equal to 12. Each point was simulated

using a maximum of 400 neighboring points within a 24 unit radius area. Then, we created five *sparse* data sets by sampling 0.01% to 5% of the original simulated grid's points. This procedure resulted in sparse data sets of sizes ranging from over 9K to over 48K. The sampling was done so that the concentration of points in different locations vary. That is, for each data set, 5% of the sampled points were selected from ten randomly selected Gaussian distributions. The rest of the points were drawn from the uniform distribution. We then removed duplicates that were resulted from sampling in these two different manners.

We also used the exhaustive Walker Lake data set, which is described in [7]. This data set was originally derived from a digital elevation model from the Walker Lake area in Nevada, U.S. There are two variables measured at 78000 points on a 260×300 grid. These two variables are continuous and their values range from zero to several thousands. These variables, which we will refer to as U and V, are related to topographic features. Authors in [7] try to keep their book generic by mentioning that these variables can represent various features such as thickness of a geographic horizon, rainfall measurements, soil strength, etc. From the dense grid, we created two scattered data sets (one for U, and one for V). In each case we sampled less than 5% of the points from the grid. About 95% of the sampled points were from the uniform distribution while the rest were sampled from 5 Gaussian clusters.

6 Experiments

All experiments were run on a Sun Fire V20z running Red Hat Enterprise release 3, using the g++ compiler version 3.2.3. Our software is implemented in C++ and uses the Geostatistical Template Library (GsTL) [17] and Approximate Nearest Neighbor library (ANN) [11]. GsTL is used for building and solving the ordinary kriging systems, and ANN is used for finding nearest neighbors for local approaches.

For each input data we examined various ordinary kriging interpolation methods on 200 query points which are missing in our sparse data sets. One hundred of these query points were sampled uniformly from the original grids. The other 100 query points were sampled from the same Gaussian distributions that were used in the generation of a small percentage of the sparse data. We used two classes of interpolation techniques: local and global methods. Local methods used Gaussian elimination for finding the solution of the linear system while global methods used a sparse SYMMLQ with $threshold = 0$ (see Section 4). All experiments' running times are averaged over 5 runs. We examined methods mentioned in Section 4 for each query point. Global approaches require selection of a tapering function. Note that the covariance model for synthetic data is Gaussian, which is infinitely differentiable. Therefore, there is no function which is as smooth as the covariance model to guarantee convergence to the optimal solution. For synthetic data, we examined all tapers mentioned in Section 3 to introduce sparsity while maintaining positive-definiteness of the covariance matrix. For real data, we used the spherical tapering function since the underlying covariance model was spherical as well, and thus we have a valid taper. The

value for θ was chosen as the distance beyond which our data's covariance function, is less than 10^{-6}. After performing tapering and storing the global sparse covariance matrix, we examined two approaches for solving the linear system. One approach solves the tapered global system using sparse SYMMLQ, and the other approach solves the tapered and projected global system as described in Section 4. Next, we analyze the quality of results, time spent solving the linear systems, and memory savings associated with our global approaches.

Table 1. Average Absolute Errors over 200 Randomly Selected Query Points.

n	Local		Tapered Global							
	Fixed Num	Fixed Radius	Top Hat	Top Hat Projected	Spherical	Spherical Projected	W_1	W_1 Projected	W_2	W_2 Projected
48513	0.416	0.414	0.333	0.334	0.336	0.337	0.278	0.279	0.276	0.284
39109	0.461	0.462	0.346	0.345	0.343	0.342	0.314	0.316	0.313	0.322
29487	0.504	0.498	0.429	0.430	0.430	0.430	0.384	0.384	0.372	0.382
19757	0.569	0.562	0.473	0.474	0.471	0.471	0.460	0.463	0.459	0.470
9951	0.749	0.756	0.604	0.605	0.602	0.603	0.608	0.610	0.619	0.637

Table 2. Average CPU Times for Solving the System over 200 Random Query Points.

n	Local		Tapered Global							
	Fixed Num	Fixed Radius	Top Hat	Top Hat Projected	Spherical	Spherical Projected	W_1	W_1 Projected	W_2	W_2 Projected
48513	0.03278	0.00862	8.456	0.01519	7.006	0.01393	31.757	0.0444	57.199	0.04515
39109	0.01473	0.00414	4.991	0.00936	4.150	0.00827	17.859	0.0235	31.558	0.02370
29487	0.01527	0.00224	2.563	0.00604	2.103	0.00528	08.732	0.0139	15.171	0.01391
19757	0.00185	0.00046	0.954	0.00226	0.798	0.00193	02.851	0.0036	05.158	0.00396
9951	0.00034	0.00010	0.206	0.00045	0.169	0.00037	00.509	0.0005	00.726	0.00064

6.1 Synthetic Data

Table 1 gives the overall average absolute estimation errors over the 200 query points compared to the ground truth values generated on the original grid. Table 2 reports the corresponding average CPU running times for solving the linear systems involved. Even though there is no taper which is as smooth as the Gaussian model to guarantee convergence to the optimal estimates, in almost all cases, we obtained lower estimation errors when using global tapered approaches. As expected, smoother functions result in lower estimation errors. Also, results from *tapered and projected* cases are comparable to their corresponding *tapered global* approaches. In other words, projecting the global tapered system did not significantly affect the quality of results compared to the global tapered approach in our experiments. In most cases, Top Hat and Spherical

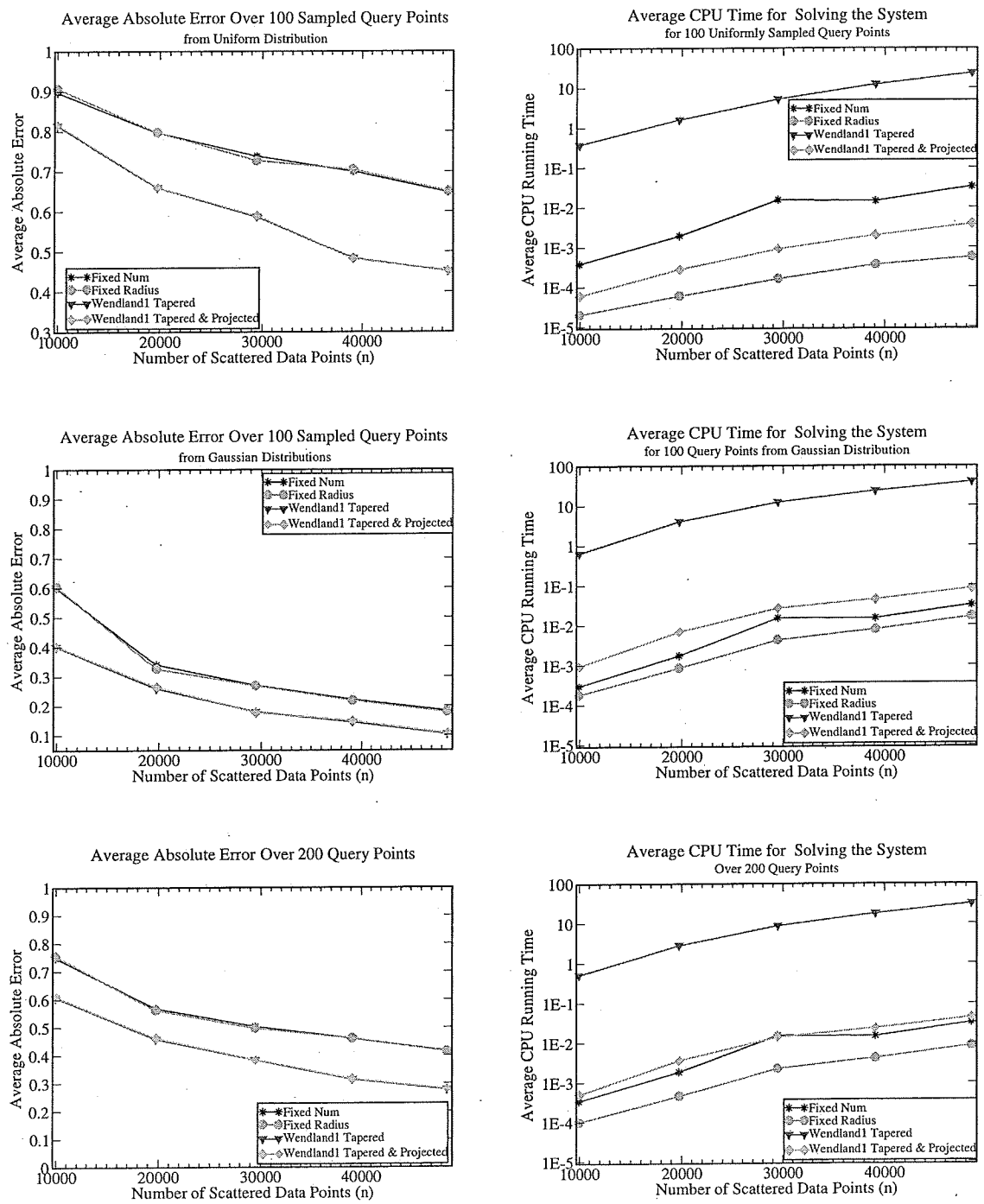

Fig. 1. Right: Average Absolute Errors. Left: Average CPU Running Times

Table 3. Memory Savings in the Global Tapered Coefficient Matrix

n	$(n+1)^2$ (Total Elements)	Stored Elements	% Stored	Savings Factor
48513	2,353,608,196	5,382,536	0.229	437.267
39109	1,529,592,100	3,516,756	0.230	434.944
29487	869,542,144	2,040,072	0.235	426.231
19757	39,0378,564	934,468	0.239	417.755
9951	99,042,304	252,526	0.255	392.206

tapers performed similar to each other with respect to the estimation error, and so did $Wendland_1$ and $Wendland_2$ tapers. Wendland tapers give the lowest overall estimation errors since they are smoother functions. Among Wendland tapers, $Wendland_1$ has lower CPU running times for solving the systems (Table 2). Thus, we plot the absolute errors and CPU running times for local approaches and global cases where $Wendland_1$ taper is being used (Figure 1). As seen in Table 2, global tapered and projected approaches are a factor of 2-3 orders of magnitude faster than the global tapered approaches, and are comparable to running times of the local approaches. Right column of Figure 1 displays these running times. The absolute estimation errors for global approaches, as seen on Table 1 and the left column of Figure 1, are lower than the local approaches.

For local approaches, using fixed radius neighborhoods resulted in lower overall errors for query points from the Gaussian distribution. Using fixed number of neighbors seems more appropriate for query points from the uniform distribution, were not enough points may be within a fixed radius. Considering maximum degree of points' covariance connectivity as number of neighbors to use in the local approach requires extra work and longer running times compared to the fixed radius approach. The fixed radius local approach is faster than the fixed neighborhood approach by 1-2 orders of magnitude for the uniform query points, and is faster within a constant factor to one order of magnitude for query points from clusters, while giving better or very close by estimations compared to the results obtained when using fixed number of neighbors.

Tapering covariances, when used with sparse implementations for solving the linear systems, results in significant memory savings. Ordinary kriging of n data points involves a coefficient matrix of size $(n+1)^2$ (see Section 2). Table 3 reports memory savings due to tapering. Memory needed after tapering is a factor 392 to 437 less than the original coefficient matrix's size.

6.2 Real Data

As explained in Section 5, we have two real data sets, each representing a different measurement, called U and V. Since we know that the underlying covariance model for these data sets are Spherical model (Eq. (7)), we only applied the Spherical taper (Eq. (9)). Table 4 reflects the overall average normalized absolute error. As before, global approaches give better estimation errors than the local approaches, even though the difference in errors is subtler compared to the synthetic data.

Similarly, Table 5 reflect CPU running times for solving the ordinary kriging systems. The running times, in contrast to the estimation errors, show significant improvements, even when using the global tapered system without projection. This is due to two reasons. First, the data sets are denser than the synthetic data. For real data,

we have sampled almost 5% of the original grid, while for synthetic data this ratio ranged from 0.01% to 5%. This makes the maximum number of neighbors to consider in local approaches much larger than number of neighbors that were considered in local approaches for synthetic data. Second, the original covariance model, Spherical model (Eq. (7)), is a tapered function itself (unlike the Gaussian model), even before more sparsity is introduced via tapering. This sparseness is not being taken advantage of in local approaches that use Gaussian elimination to solve the interpolation systems, and where the largest safest neighborhood is being used. Even though the tapered global systems are solved quite fast compared to the dense local systems, we still can improve their running times by an order of magnitude using the tapered and projected approach.

Table 6 indicates that global tapered approaches use a factor of 4-22 less memory compared the original global systems. Again, these factors are smaller compared to the synthetic data sets, since the sampled points are denser (higher percentage of the original grid).

Table 4. Average Normalized Absolute Errors over 200 Query Points.

n	Variable	Local		Tapered Global	
		Fixed Num	Fixed Radius	Spherical	Spherical Projected
3720	u	0.380	0.379	0.364	0.360
3675	v	0.346	0.346	0.342	0.341

Table 5. Average CPU Times over 200 Query Points.

n	Variable	Local		Tapered Global	
		Fixed Num	Fixed Radius	Spherical	Spherical Projected
3720	u	4.649	4.023	0.729	0.045
3675	v	4.649	4.650	1.642	0.119

Table 6. Memory Savings in the Global Tapered Coefficient Matrix

n	$(n+1)^2$ (Total Elements)	Stored Elements	% Stored	Savings Factor
3720	13,845,841	3,013,823	21.77	4.59
3675	13,512,976	2,973,046	22.00	4.54

7. Conclusion

Solving very large ordinary kriging systems via direct approaches is infeasible for large data sets. We implemented efficient ordinary kriging algorithms through utilizing co-

variance tapering [5] and iterative methods [13, 15]. Furrer *et al.* [5] had utilized covariance tapering along with sparse Cholesky decomposition to solve simple kriging systems. We explain in Section 1 why Cholesky decomposition is not applicable to the ordinary kriging problem. We used tapering with sparse SYMMLQ method to solve large ordinary kriging systems. We also implemented a variant of the global tapered method through projecting the large global system on to an appropriate smaller system. Global tapered methods resulted in saving factors ranging from 4.54 to 437.27 for the storage of the coefficient matrix of the ordinary kriging system compared to the original global system. Global tapered iterative methods gave better estimation errors compared to the local approaches. In all cases, the estimation results of the global tapered method were very close to the global tapered and projected method. This is while global tapered and projected method solves the linear systems within order(s) of magnitude faster than the global tapered method. This method can be viewed as a way of adaptively finding the correct neighbors to consider for the interpolation problem. Results of traditional local approaches depend on the underlying points' distribution, and whether or not enough points are included in the specified neighborhood.

8 Acknowledgements

We would like to thank Galen Balcom for his contributions to the C++ implementation of the SYMMLQ algorithm.

References

1. P. Alfeld. Scattered data interpolation in three or more variables. *Mathematical methods in computer aided geometric design*, pages 1–33, 1989.
2. I. Amidror. Scattered data interpolation methods for electronic imaging systems: a survey. *J. of Electronic Imaging*, 11(2):157–176, Apr. 2002.
3. S. D. Billings, R. K. Beatson, and G. N. Newsam. Interpolation of geophysical data using continuous global surfaces. *Geophysics*, 67(6):1810–1822, Nov-Dec 2002.
4. G. Christakos. On the problem of permissible covariance and variogram models. *Water Resources Research*, 20(2):251–265, Feb. 1984.
5. R. Furrer, M. G. Genton, and D. Nychka. Covariance tapering for interpolation of large spatial datasets. *J. of Computational and Graphical Statistics*, 15(3):502–523, Sep. 2006.
6. P. Goovaerts. *Geostatistics for Natural Resources Evaluation.* Oxford University Press, Oxford, 1997.
7. E. H. Isaaks and R. M. Srivastava. *An Introduction to Applied Geostatistics.* Oxford University Press, New York, Oxford, 1989.
8. A.G. Journel and C. J. Huijbregts. *Mining Geostatistics.* Academic Press Inc, New York, 1978.
9. C. F. Van Loan. *Intro. to Scientific Computing.* Prince-Hall, Upper Saddle River, NJ 07458, 2nd edition, 2000.
10. T. H. Meyer. The discontinuous nature of kriging interpolation for digital terrain modeling. *Cartography and Geographic Information Science,*, 31(4):209–216, 2004.
11. D. M. Mount and S. Arya. ANN: A library for approximate nearest neighbor searching. http://www.cs.umd.edu/ mount/ANN/, May 2005.

12. D. E. Myers. Pseudo-cross variograms, positive-definiteness, and cokriging. *Mathematical Geology*, 23(6):805–816, 1991.
13. S. G. Nash and A. Sofer. *Linear and Nonlinear Programming.* McGraw-Hill Companies, 1996.
14. C. C. Paige and M. A. Saunderszi. Solution of sparse indefinite systems of linear equations. *SIAM J. on Numerical Analysis*, 12(4):617–629, Sep. 1975.
15. W. H. Press, S. A. Teukolsky, W. T. Vetterling, and B. P. Flannery. *Numerical Recipes in C++, The Art of Scientific Computing.* Cambridge University Press, The Edinburgh Building, Cambridge CB2 2RU, UK, 2nd edition, 2002.
16. C. E. Rasmussen and C. K. I. Williams. *Gaussian Processes for Machine Learning.* MIT Press, 2006.
17. N. Remy. GsTL: The Geostatistical Template Library in C++. Master's thesis, Department of Petroleum Engineering of Stanford University, Mar. 2001.
18. N. Remy. *The Stanford Geostatistical Modeling Software (S-GeMS).* SCRC Lab, Stanford University, May 2004.
19. J. R. Shewchuk. An intro. to the conjugate gradient method without the agonizing pain. CMU-CS-94-125, School of Computer Science, Carnegie Mellon University, 1994.
20. M. L. Stein. The screening effect in kriging. *Annals of Statistics*, 1(30):298–323, 2002.