

Efficient Implementation of an Optimal Interpolator for Large Spatial Data Sets

Nargess Memarsadeghi ^{*1,2} and David M. Mount²
Nargess.Memarsadeghi@nasa.gov, mount@cs.umd.edu

¹ NASA/GSFC, Code 588, Greenbelt, MD, 20771

² University of Maryland, College Park, MD, 20742

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]. 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 is *ordinary kriging* [3]. This is because it is a best unbiased estimator [4, 3, 5]. Unfortunately, this interpolant is computationally very expensive to compute exactly. 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 [3, 5]. 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 [5]. Local methods also suffer from the problem that the resulting interpolant is not continuous. Meyer showed that while kriging produces smooth continuous surfaces, it has zero order continuity along its borders [6]. 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* [7]. 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* [7]. Simple kriging is not an

* Corresponding author: NASA GSFC, Architectures and Automation Branch, Code 588, Greenbelt, MD, 20771, Tel: 301-286-2938.

unbiased estimator for stationary data whose mean value differs from zero, however. We generalize these results by showing how to apply covariance tapering to the more general problem of ordinary kriging.

Our implementations combine, 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.* suggested, we use an iterative approach [8]. In particular, we use the SYMMLQ method [9], for solving the large but sparse ordinary kriging systems that result from tapering.

The main technical issue that need to be overcome in our algorithmic solution is that the points' covariance matrix for kriging should be symmetric positive definite [3, 10]. The goal of tapering is to obtain a sparse approximate representation of the covariance matrix while maintaining its positive definiteness. Furrer *et al.* used tapering to obtain a sparse linear system of the form $Ax = b$, where A is the tapered symmetric positive definite covariance matrix [7]. Thus, Cholesky factorization [11] could be used to solve their linear systems. They implemented an efficient sparse Cholesky decomposition method. They also showed 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. Matrix A in the ordinary kriging system, while symmetric, is not positive definite. Thus, their approach is not applicable to the ordinary kriging system [10]. Therefore, we use tapering only to obtain a sparse linear system. Then, we use SYMMLQ to solve the ordinary kriging system [9].

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 also developed a more efficient variant of the sparse SYMMLQ method for large ordinary kriging systems. This approach adaptively finds the correct local neighborhood for each query point in the interpolation process.

We start with a brief review of the ordinary kriging in Section 2. In Section 3 the tapering properties are mentioned. We introduce our approaches for solving the ordinary kriging problem in Section 4. Section 5 describes data sets we used. Then, we describe our experiments and results in Section 6. Section 7 concludes the paper. Full version of our paper has details that were omitted due to the space limit [10].

2 Ordinary Kriging

Kriging is an interpolation method named after Danie Krige, a South African mining engineer, who pioneered in the field of geostatistics [5]. Kriging is also referred to as the *Gaussian process predictor* in the machine learning domain [12]. Kriging and its variants have been traditionally used in mining and geostatistics applications [4, 5, 3]. 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 [3, 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 [3]. Minimizing the variance of the estimation error forms the objective function of an optimization problem. Ensuring unbiasedness of the error imposes a constraint on this objective function. Formalizing this objective function with its constraint results in the following system [10, 3, ?].

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

where C is the matrix of points' pairwise covariances, L is a column vector of all ones and of size n , and w is the vector of weights w_1, \dots, 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. It is also important that matrix C be positive definite [10, 3]. 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.

Pairwise covariances are modeled as a function of points' separation. These functions should result in a positive definite covariance matrix. Christakos [13] showed necessary and sufficient conditions for such permissible covariance functions. Two of these valid covariance functions, are the Gaussian and Spherical covariance functions (C_g and C_s , respectively). Please see [13, 3, 5] for details of these and other permissible covariance functions.

3 Tapering Covariances

Tapering covariances for the kriging interpolation problem, as described in [7], 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. The sparse representation via tapering is obtained through the Schur product of the original positive definite covariance matrix by another such matrix.

$$C_{tap}(h) = C(h) \times C_\theta(h). \quad (2)$$

The tapered covariance matrix, C_{tap} , is zero for points that are more than a certain distance apart from each other. It is also positive definite since it is the Schur product of two positive definite matrices. 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 [7] mention few valid tapering functions. We used *Spherical*, *Wendland₁*, *Wendland₂*, and *TopHat* tapers [7]. These tapers are valid for \mathbb{R}^3 and lower dimensions [7]. Tapers need to be as smooth as the original covariance function at origin to guarantee convergence to the optimal estimator [7]. Thus, for a Gaussian covariance function, which is infinitely differentiable, no taper

exists that satisfies this smoothness requirement. However, since tapers proposed in [7] still maintain positive definiteness of the covariance matrices, we examined using these tapers for Gaussian covariance functions as well. 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.

4 Our Approaches

We implemented both local and global methods for the ordinary kriging problem.

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 [14] 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. (2), where $C(h)$ is the points' pairwise covariance function, and $C_\theta(h)$ is a tapering function. We then solve the linear system using the SYMMLQ approach as mentioned in [9]. 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 [15]. Since ordinary kriging systems are symmetric and not positive definite, we used SYMMLQ. We implemented a sparse SYMMLQ method, similar to the sparse conjugate gradient method in [16]. In [16]'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.

Global Tapered and Projected Methods: This implementation is motivated by numerous empirical results in geostatistics indicating that interpolation weights associated with points that are very far from the query point tend to be 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 [17]. Stein showed conditioned under which the screening effect occurs for gridded data [17]. While the screening effect has been the basis for using local methods, there is no proof of this empirically supported idea for scattered data points [7]. 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 covari-

ance between the query point and the i^{th} 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*. This method is effectively the same as the fixed radius neighborhood size, except that the local neighborhood is found adaptively. 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

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 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. We generated our synthetic data sets using the Sgems [18] software. We generated values on a (1000×1000) grid, using the Sequential Gaussian Simulation (sgsim) algorithm of the Sgems software [19, 18]. Points were simulated through ordinary kriging with a Gaussian covariance function of range equal to 12, using a maximum of 400 neighboring points within a 24 unit radius area. Then, we created 5 *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. For each data set, 5% of the sampled points were from 10 randomly selected Gaussian distributions. The rest of the points were drawn from the uniform distribution. Details of the real data tests and results are in our full paper [10].

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++, using the GsTL and ANN libraries [19, 20]. GsTL is used to build and solving the linear systems. ANN is used for finding nearest neighbors for local approaches.

For each input data we examined various ordinary kriging methods on 200 query points. Half of these query points were sampled uniformly from the original grids. The other 100 query points were sampled from the Gaussian distributions.

We tested both local and global methods. Local methods used Gaussian elimination for solving the linear systems while global tapered methods used sparse SYMMMLQ. Running times are averaged over 5 runs.

We examined methods mentioned in Section 4. Global approaches require selection of a tapering function. For synthetic data, we examined all tapers mentioned in Section 3. 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 tapers performed similar to each other with respect to the estimation error, and so did *Wendland₁* and *Wendland₂* tapers. Wendland tapers give the lowest overall estimation errors. Among Wendland tapers, *Wendland₁* has lower CPU running times for solving the systems. Figure 1 shows the absolute errors and CPU running times, when *Wendland₁* taper is used.

For local approaches, using fixed radius neighborhoods resulted in lower errors for query points from the Gaussian distribution. Using fixed number of neighbors seems more appropriate for uniformly sampled query points. 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 an order of magnitude for query points from clusters, while giving better or very close by estimations compared to the fixed number of neighbors approach (Tables 1 and 2).

Tapering, used with sparse implementations for solving the linear systems, results in significant memory savings. Table 3 reports these memory savings for synthetic data to be a factor of 392 to 437.

Table 1. 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

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-

Table 2. 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 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

variance tapering [7] and iterative methods [14, 16]. Furrer *et al.* [7] had utilized covariance tapering along with sparse Cholesky decomposition to solve simple kriging systems. Their approach is not applicable to the general 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 global system on to an appropriate smaller system. Global tapered methods resulted in memory savings ranging from a factor of 4.54 to 437.27. Global tapered iterative methods gave better estimation errors compared to the local approaches. The estimation results of the global tapered method were very close to the global tapered and projected method. The global tapered and projected method solves the linear systems within order(s) of magnitude faster than the global tapered method.

8 Acknowledgements

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

References

1. Amidror, I.: Scattered data interpolation methods for electronic imaging systems: a survey. *J. of Electronic Imaging* **11**(2) (Apr. 2002) 157–176
2. Alfeld, P.: Scattered data interpolation in three or more variables. *Mathematical methods in computer aided geometric design* (1989) 1–33
3. Isaaks, E.H., Srivastava, R.M.: *An Introduction to Applied Geostatistics*. Oxford University Press, New York, Oxford (1989)
4. Journel, A., Huijbregts, C.J.: *Mining Geostatistics*. Academic Press Inc, New York (1978)
5. Goovaerts, P.: *Geostatistics for Natural Resources Evaluation*. Oxford University Press, Oxford (1997)
6. Meyer, T.H.: The discontinuous nature of kriging interpolation for digital terrain modeling. *Cartography and Geographic Information Science*, **31**(4) (2004) 209–216

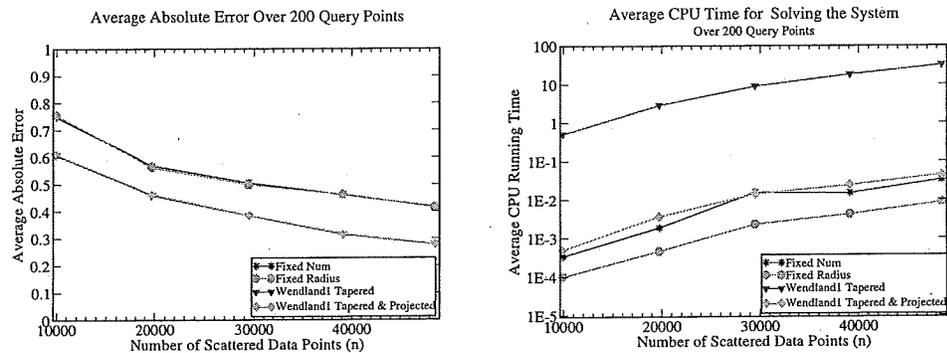


Fig. 1. Right: Average Absolute Errors. Left: Average CPU Running Times

7. Furrer, R., Genton, M.G., Nychka, D.: Covariance tapering for interpolation of large spatial datasets. *J. of Computational and Graphical Statistics* 15(3) (Sep. 2006) 502–523
8. Billings, S.D., Beatson, R.K., Newsam, G.N.: Interpolation of geophysical data using continuous global surfaces. *Geophysics* 67(6) (Nov-Dec 2002) 1810–1822
9. Paige, C.C., Saunderszi, M.A.: Solution of sparse indefinite systems of linear equations. *SIAM J. on Numerical Analysis* 12(4) (Sep. 1975) 617–629
10. Memarsadeghi, N., Mount, D.M.: Efficient implementation of an optimal interpolator for large spatial data sets. Technical Report CS-TR-4856, Computer Science Department, University of Maryland, College Park, MD, 20742 (Jan. 2007)
11. Loan, C.F.V.: *Intro. to Scientific Computing*. 2nd edn. Prentice-Hall, Upper Saddle River, NJ 07458 (2000)
12. Rasmussen, C.E., Williams, C.K.I.: *Gaussian Processes for Machine Learning*. MIT Press (2006)
13. Christakos, G.: On the problem of permissible covariance and variogram models. *Water Resources Research* 20(2) (Feb. 1984) 251–265
14. Nash, S.G., Sofer, A.: *Linear and Nonlinear Programming*. McGraw-Hill Companies (1996)
15. Shewchuk, J.R.: An intro. to the conjugate gradient method without the agonizing pain. CMU-CS-94-125, School of Computer Science, Carnegie Mellon University (1994)
16. Press, W.H., Teukolsky, S.A., Vetterling, W.T., Flannery, B.P.: *Numerical Recipes in C++*, The Art of Scientific Computing. 2nd edn. Cambridge University Press, The Edinburgh Building, Cambridge CB2 2RU, UK (2002)
17. Stein, M.L.: The screening effect in kriging. *Annals of Statistics* 1(30) (2002) 298–323
18. Remy, N.: *The Stanford Geostatistical Modeling Software (S-GeMS)*. SCRC Lab, Stanford University. (May 2004)
19. Remy, N.: *GsTL: The Geostatistical Template Library in C++*. Master's thesis, Department of Petroleum Engineering of Stanford University (Mar. 2001)
20. Mount, D.M., Arya, S.: ANN: A library for approximate nearest neighbor searching. <http://www.cs.umd.edu/mount/ANN/> (May 2005)