A UNIFIED FRAMEWORK FOR KERNEL AND POLYNOMIAL APPROXIMATION*

MILENA BELIANOVICH†, AKIL NARAYAN‡, GREGORY FASSHAUER§, AND VARUN SHANKAR¶

Abstract.

Key words. Radial basis function; polynomial; pseudospectral; high-order method.

AMS subject classifications. 68Q25, 68U05

1. Introduction.

Radial Basis Functions (RBFs) are a powerful and flexible tool for generating numerical methods for the solution of partial differential equations (PDEs). RBF collocation methods are easily applied to solving PDEs on irregular domains using scattered node layouts [3, 4, 21]. RBF-based methods also generalize naturally to the solution of PDEs on manifolds $\mathbb{M} \subset \mathbb{R}^3$ using only the Euclidean distance measure in the embedding space and Cartesian coordinates; see for example [1, 8–15, 17–19].

The remainder of this paper is organized as follows.

2. Review.

3. Unified interpolation with sparse kernels and polynomials.

We will now present a unified framework for kernel and polynomial approximation. Consider the hybrid *interpolant* given by:

$$s(\boldsymbol{x}) = \sum_{k=1}^{N} c_k \phi\left(\varepsilon \|\boldsymbol{x} - \boldsymbol{x}_k\|\right) + \sum_{j=1}^{\binom{\ell+d}{d}} d_j p_j(\boldsymbol{x}), \tag{1}$$

where ϕ is a radial kernel (a radial basis function) and ε its shape parameter, and the p_j functions constitute a basis for the space of total degree polynomials of degree ℓ in d dimensions. If the goal is to interpolate samples of a function $f(\boldsymbol{x}) : \mathbb{R} \to \mathbb{R}$ at some set of locations $X = \{\boldsymbol{x}_k\}_{k=1}^{N}$, the interpolation coefficients c_k and d_j are found by enforcing the following constraints:

$$s(\boldsymbol{x}_k) = f(\boldsymbol{x}_k), k = 1, \ldots, N, \tag{2}$$

$$\sum_{k=1}^{N} c_k p_j(\boldsymbol{x}_k) = 0, j = 1, \ldots, \binom{\ell+d}{d}, \tag{3}$$

where the first constraint enforces interpolation while the second constraint enforces polynomial reproduction [5]. In the context of such unified interpolants, the most commonly-used choice for ϕ is the polyharmonic spline (PHS) RBF [2, 6, 7], though other choices have been made in the literature [**?**, **?**]. In this work, we focus on the case where ϕ is a *compactly-supported* and *positive-definite* RBF. More specifically, we focus on the popular class of RBFs known as Wendland functions [20] given by:

$$\phi_{m,n}(r) = \begin{cases} \frac{1}{\Gamma(n)2^{n-1}} \int\limits_{r}^{1} s(1-s)^m (s^2 - r^2)^{n-1} ds & \text{for } 0 \le r \le 1, \\ 0 & \text{for } r > 1. \end{cases} \tag{4}$$

For these compactly-supported Wendland functions, the shape parameter ε serves as the reciprocal of the radius of support r.For the Wendland functions, the shape parameter ε serves as the reciprocal of the

*Submitted to the editors 08/01/2019.

†Kahlert School of Computing, University of Utah, Salt Lake City, UT USA (shankar@cs.utah.edu)

‡Department of Mathematics, and Scientific Computing and Imaging (SCI) Institute, University of Utah, Salt Lake City, UT USA (akil@sci.utah.edu).

§Department of Applied Mathematics and Statistics, Colorado School of Mines, Golden CO USA (fasshauer@mines.edu).

¶Kahlert School of Computing, University of Utah, Salt Lake City, UT USA (shankar@cs.utah.edu). Corresponding author.

radius of support r. Unfortunately, these RBFs have historically traded off increased sparsity (and improved conditioning) in the interpolation matrix for decreased accuracy and convergence rates [5]. As a side note, globally-supported RBFs also exhibit decreased accuracy and convergence rates upon spatial refinement, primarily due to ill-conditioning in the Gramian [5, 20].

Interestingly, recent work has shown [2,6,7,16,18] that at least for piecewise-smooth globally-supported RBFs (such as the polyharmonic splines) used as *local* interpolants, this decrease in accuracy can be overcome by augmenting the RBF interpolant with polynomials. Our goal in this work is to both demonstrate that this technique works in the context of compactly-supported RBFs used as global approximants, and to demonstrate that the use of compactly-supported RBFs creates a highly general and flexible unified framework for kernel and polynomial approximation.

To that end, we shall now present and discuss several special cases of the above interpolation scheme. We will also discuss efficient solution of the resulting linear systems for each of the limiting cases. In the following discussion, let $X = \{\boldsymbol{x}_k\}_{k=1}^N$ be the set of data sites for interpolation. Further, let q be the smallest pairwise distance between of nodes in X, and w be the largest pairwise distance. Finally, let $r = 1/\varepsilon$ be the support of the Wendland RBF. We are interested in two regimes: (1) $r < q$, where the polynomials dominate the approximation, and (2) $q < r < w$, where both RBFs and polynomials contribute to the approximation, but the RBFs still produce sparse matrices. There are two other cases that we do not consider in this work. The first is the case where $r = w$ and the compactly-supported RBFs produce dense matrices; this regime is undesirable due to ill-conditioning and a lack of computational efficiency. The second is the case where the polynomial is not even used in the approximation; this regime is simply the traditional use case of compactly-supported RBFs. Both these cases have been covered extensively elsewhere [5].

3.1. The polynomial limit ($r < q$).

3.1.1. Interpolation. When the support r of any Wendland RBF is made smaller than q, the Wendland RBFs each take on the value of 1 at the nodes and 0 elsewhere, resembling smooth bump functions. Thus, the interpolation constraints (2)–(3) enforced at the set of nodes X lead to the following linear system:

$$\begin{bmatrix} I & P \\ P^T & 0 \end{bmatrix} \begin{bmatrix} \boldsymbol{c} \\ \mathbf{d} \end{bmatrix} = \begin{bmatrix} \boldsymbol{f} \\ \mathbf{0} \end{bmatrix}, \tag{5}$$

where I is the $N \times N$ identity matrix and $P_{ij} = p_j(\boldsymbol{x}_i), i = 1, \dots, N, j = 1, \dots, \binom{\ell+d}{d}$ is the Vandermonde-like matrix of evaluations of the polynomial basis at X. To better understand this linear system, we use block Gaussian elimination to rewrite this as two equations:

$$P^T P \mathbf{d} = P^T \boldsymbol{f}, \tag{6}$$

$$\boldsymbol{c} = \boldsymbol{f} - P\mathbf{d}. \tag{7}$$

(6) is immediately recognizable as the system of normal equations arising from the least squares problem of minimizing $\|P\mathbf{d} - \boldsymbol{f}\|_2^2$. Thus, in this context, despite the presence of RBFs in the approximation, the polynomial coefficients $\mathbf{d}$ are identical to the coefficients obtained from the least-squares problem. In addition, (7) clearly shows that the RBF coefficient vector $\boldsymbol{c}$ is simply the residual vector from the polynomial least-squares problem.

Once the coefficients $\boldsymbol{c}$ and $\mathbf{d}$ are computed, the interpolant (1) can be evaluated anywhere. Let $X_e = \{\boldsymbol{x}_i^e\}_{i=1}^{N_e}$ be a set of *evaluation points*. Then, the interpolant can be evaluated at X_e as:

$$s(\boldsymbol{x})|_{X_e} = \begin{bmatrix} A_e & P_e \end{bmatrix} \begin{bmatrix} \boldsymbol{c} \\ \mathbf{d} \end{bmatrix} = A_e \boldsymbol{c} + P_e \mathbf{d}, \tag{8}$$

where $(A_e)_{ij} = \phi\left(\|\boldsymbol{x}_i^e - \boldsymbol{x}_j\|\right), i = 1, \dots, N_e, j = 1, \dots, N$ and $(P_e)_{ij} = p_j\left(\boldsymbol{x}_i^e\right), i = 1, \dots, N_e, j = 1, \dots \binom{\ell+d}{d}$. In general, A_e is a sparse and rectangular matrix, and its structure is determined by the smoothness of the Wendland RBF and its support r.

An interesting implication of this discussion is that the residual vector $\boldsymbol{f} - P\mathbf{d}$ from a polynomial least squares problem can be treated as a set of RBF coefficients. These RBF coefficients can then be evaluated against *any* compactly-supported RBF via (8) provided the support $r < q$. This turns any polynomial least squares problem into one of interpolation with the unified interpolant in (1).

3.1.2. Linear Algebra.
In the polynomial limit, the unified interpolant (1) can be computed efficiently using the QR decomposition as follows:

1. Compute $P = QR$, the (reduced) QR decomposition of the polynomial least-squares matrix P.
2. Solve the Schur complement system for the polynomial coefficients $\mathbf{d}$ as $\mathbf{d} = R^{-1}Q^T\boldsymbol{f}$.
3. Compute the RBF coefficients $\boldsymbol{c} = \boldsymbol{f} - P\mathbf{d}$.

Alternatively, the SVD could be used instead of the QR decomposition to find $\mathbf{d}$. When solved in this fashion, the block system (5) does not need to be explicitly computed or stored, nor do we need to compute the matrix P^T. This approach can be applied with a small modification if P is rank-deficient also: column pivoting can be used for the QR decomposition, and truncation of the singular values can be used for the SVD.

3.1.3. Error Estimates.

3.2. Hybrid approximation ($q < r$) .

3.2.1. Interpolation.
If $r > q$ (or conversely ϵ is sufficiently small), the constraints (2)–(3) can no longer be represented by (5). Instead, they now generate a block linear system with I replaced by a sparse matrix A with entries $A_{ij} = \phi\left(\|\boldsymbol{x}_i - \boldsymbol{x}_j\|\right), i, j = 1, \ldots, N$:

$$\begin{bmatrix} A & P \\ P^T & 0 \end{bmatrix} \begin{bmatrix} \boldsymbol{c} \\ \mathbf{d} \end{bmatrix} = \begin{bmatrix} \boldsymbol{f} \\ \mathbf{0} \end{bmatrix}. \tag{9}$$

Once again, we may use block Gaussian elimination to rewrite this as

$$P^T A^{-1} P\mathbf{d} = P^T A^{-1}\boldsymbol{f}, \tag{10}$$

$$A\boldsymbol{c} = \boldsymbol{f} - P\mathbf{d}. \tag{11}$$

Finding $\boldsymbol{c}$ now requires the inversion of the sparse matrix A. The exact properties of this approximation depend on the value of r, the smoothness of the Wendland kernel, and the polynomial degree ℓ.

3.2.2. Linear Algebra.
Since $A \neq I$, it is not possible to solve the above pair of equations for $\mathbf{d}$ and $\boldsymbol{c}$ using a single QR decomposition of the matrix P. Consequently, we are faced with two choices: (1) form and invert the entire matrix in (9), or (2) solve the pair of equations (10)–(11) efficiently. We choose the latter approach as we observed that it resulted in improved numerical stability and consequently greater accuracy.

Unfortunately, inverting the matrix $S = P^T A^{-1} P$ appears to be a non-trivial task. The condition number of A can range from $O(1)$ in the case of a small support r to arbitrarily large for large r (as A becomes more dense). Further, we have observed that the matrix $P^T P$ is typically very ill-conditioned since it is formed by evaluations of polynomials at a possibly arbitrary set of collocation points. Consequently, though the matrix S is symmetric positive-definite in exact arithmetic (since A^{-1} is symmetric positive-definite), we have observed that the product $S = P^T A^{-1} P$ loses symmetry due to rounding errors even when A has a condition number of $O(1)$; this loss in symmetry appears to significantly degrade the accuracy of the approximation. Fortunately, the symmetry of S can be maintained using standard numerical linear algebra. Since A is symmetric positive-definite, it can be written in terms of its Cholesky decomposition as

$$A = LL^T, \tag{12}$$

where L is a lower-triangular matrix. If A is sparse, it is also possible to maintain sparsity in L using a sparse Cholesky decomposition. Using this decomposition, we can rewrite (10) as:

$$P^T \left(LL^T\right)^{-1} P\mathbf{d} = P^T \left(LL^T\right)^{-1} \boldsymbol{f}, \tag{13}$$

$$\implies P^T L^{-T} \underbrace{L^{-1}P}_{B}\mathbf{d} = P^T L^{-T} \underbrace{L^{-1}\boldsymbol{f}}_{\mathbf{g}}, \tag{14}$$

$$\implies B^T B\mathbf{d} = B^T\mathbf{g}, \tag{15}$$

where $B = L^{-1}P$. This is in fact a system of normal equations for $\mathbf{d}$ that is attempting to find the vector $\mathbf{d}$ that minimizes $\|B\mathbf{d} - \mathbf{g}\|_2$. Using this approach, the coefficients $\boldsymbol{c}$ and $\mathbf{d}$ in the unified interpolant (1) can be computed as follows:

1. Compute $A = LL^T$, the Cholesky decomposition of the RBF matrix A.
2. Compute the matrix $B = L^{-1}P$ and the vector $\mathbf{g} = L^{-1}\boldsymbol{f}$.
3. Compute $B = \tilde{Q}\tilde{R}$, the QR decomposition of B.
4. Solve for the polynomial coefficients $\mathbf{d}$ as $\mathbf{d} = \tilde{R}^{-1}\tilde{Q}^T\mathbf{g}$.
5. Solve for the RBF coefficients $\boldsymbol{c}$ as $\boldsymbol{c} = L^{-T}L^{-1}(\boldsymbol{f} - P\mathbf{d})$.

It is important to note that if P is rank-deficient, B is also. In such a case, one can replace the QR decomposition with either its column-pivoted counterpart or with a truncated SVD. Regardless, once the coefficients are computed, the unified interpolant can be evaluated using (8).

4. Unified Gaussian Process (UGPs) with Kernels and Polynomials.

5. Results.

6. Summary and Future Work.

Acknowledgments. VS was supported by

REFERENCES

[1] K. A. Aiton, *A Radial Basis Function Partition of Unity Method for Transport on the Sphere*, master's thesis, Boise State University, USA, 2014.

[2] V. Bayona, N. Flyer, B. Fornberg, and G. A. Barnett, *On the role of polynomials in rbf-fd approximations: Ii. numerical solution of elliptic pdes*, Journal of Computational Physics, 332 (2017), pp. 257 – 273.

[3] V. Bayona, M. Moscoso, M. Carretero, and M. Kindelan, *RBF-FD formulas and convergence properties*, J. Comput. Phys., 229 (2010), pp. 8281–8295.

[4] O. Davydov and D. T. Oanh, *Adaptive meshless centres and RBF stencils for poisson equation*, J. Comput. Phys., 230 (2011), pp. 287–304.

[5] G. E. Fasshauer, *Meshfree Approximation Methods with MATLAB*, Interdisciplinary Mathematical Sciences - Vol. 6, World Scientific Publishers, Singapore, 2007.

[6] N. Flyer, G. A. Barnett, and L. J. Wicker, *Enhancing finite differences with radial basis functions: Experiments on the Navier-Stokes equations*, J. Comput. Phys., 316 (2016), pp. 39–62.

[7] N. Flyer, B. Fornberg, V. Bayona, and G. A. Barnett, *On the role of polynomials in RBF-FD approximations: I. Interpolation and accuracy*, J. Comput. Phys., 321 (2016), pp. 21–38.

[8] N. Flyer, E. Lehto, S. Blaise, G. B. Wright, and A. St-Cyr, *A guide to RBF-generated finite differences for nonlinear transport: shallow water simulations on a sphere*, J. Comput. Phys., 231 (2012), pp. 4078–4095.

[9] N. Flyer and G. B. Wright, *Transport schemes on a sphere using radial basis functions*, J. Comput. Phys., 226 (2007), pp. 1059–1084.

[10] N. Flyer and G. B. Wright, *A radial basis function method for the shallow water equations on a sphere*, Proc. Roy. Soc. A, 465 (2009), pp. 1949–1976.

[11] B. Fornberg and E. Lehto, *Stabilization of RBF-generated finite difference methods for convective PDEs*, J. Comput. Phys., 230 (2011), pp. 2270–2285.

[12] E. J. Fuselier and G. B. Wright, *A high-order kernel method for diffusion and reaction-diffusion equations on surfaces*, J. Sci. Comput., 56 (2013), pp. 535–565.

[13] E. Lehto, V. Shankar, and G. B. Wright, *A radial basis function (rbf) compact finite difference (fd) scheme for reaction-diffusion equations on surfaces*, SIAM Journal on Scientific Computing, 39 (2017), pp. A2129–A2151.

[14] C. Piret, *The orthogonal gradients method: A radial basis functions method for solving partial differential equations on arbitrary surfaces*, J. Comput. Phys., 231 (2012), pp. 4662–4675.

[15] C. Piret and J. Dunn, *Fast rbf ogr for solving pdes on arbitrary surfaces*, AIP Conference Proceedings, 1776 (2016).

[16] V. Shankar and A. L. Fogelson, *Hyperviscosity-based stabilization for radial basis function-finite difference (RBF-FD) discretizations of advection-diffusion equations*, J. Comput. Physics, 372 (2018), pp. 616–639.

[17] V. Shankar, A. Narayan, and R. M. Kirby, *RBF-LOI: augmenting radial basis functions (rbfs) with least orthogonal interpolation (LOI) for solving pdes on surfaces*, J. Comput. Physics, 373 (2018), pp. 722–735.

[18] V. Shankar and G. B. Wright, *Mesh-free semi-lagrangian methods for transport on a sphere using radial basis functions*, J. Comput. Phys., 366 (2018), pp. 170–190.

[19] V. Shankar, G. B. Wright, R. M. Kirby, and A. L. Fogelson, *A radial basis function (RBF)-finite difference (FD) method for diffusion and reaction–diffusion equations on surfaces*, J. Sci. Comput., 63 (2014), pp. 745–768.

[20] H. Wendland, *Scattered data approximation*, vol. 17 of Cambridge Monogr. Appl. Comput. Math., Cambridge University Press, Cambridge, 2005.

[21] G. B. Wright and B. Fornberg, *Scattered node compact finite difference-type formulas generated from radial basis functions*, J. Comput. Phys., 212 (2006), pp. 99–123.