INTERNATIONAL JOURNAL OF
NUMERICAL ANALYSIS AND MODELING
Volume 15, Number 4-5, Pages 628–648

USING RBF-GENERATED QUADRATURE RULES TO SOLVE NONLOCAL ANOMALOUS DIFFUSION

ISAAC LYNGAAS AND JANET PETERSON

(Communicated by T. Iliescu)

Dedicated to Professor William J. Layton on the occasion of his 60^{th} *birthday*

Abstract. The goal of this work is to solve nonlocal diffusion and anomalous diffusion problems by approximating the nonlocal integral appearing in the integro-differential equation by novel quadrature rules. These quadrature rules are derived so that they are exact for a nonlocal integral evaluated at translations of a given radial basis function (RBF). We first illustrate how to derive RBF-generated quadrature rules in one dimension and demonstrate their accuracy for approximating a nonlocal integral. Once the quadrature rules are derived as a preprocessing step, we apply them to approximate the nonlocal integral in a nonlocal diffusion problem and when the temporal derivative is approximated by a standard difference approximation a system of difference equations are obtained. This approach is extended to two dimensions where both a circular and rectangular nonlocal neighborhood are considered. Numerical results are provided and we compare our results to published results solving nonlocal problems using standard finite element methods.

Key words. Nonlocal, anomalous diffusion, radial basis functions, RBF, quadrature

1. Introduction

In recent years, there has been an increased interest in nonlocal continuum models due to their ability to describe physical phenomena which are not well modeled by standard partial differential equation (PDE) models. Unlike standard PDE models, nonlocal models are free of spatial derivatives. Feature interactions are typically represented by an integral resulting in an integro-differential equation; these interactions are assumed to occur over a finite region governed by a horizon. Nonlocal models for anomalous diffusion are especially advantageous because the only difference in the model is the exponent in the kernel of the nonlocal integral. Thus methods for solving a nonlocal diffusion problem typically can be easily extended to model anomalous diffusion unlike PDE models. One complication in nonlocal models is that the region of integration for the nonlocal integral can extend past the physical domain. Thus so-called volume constraints are imposed in place of the standard boundary conditions in PDE models. For an overview of the analysis of nonlocal problems with volume constraints the reader is referred to [1].

In this paper we are interested in nonlocal diffusion and especially in the case of anomalous diffusion where the spatial spread of a diffusing quantity is not proportional to the square root of time as predicted by the heat equation. Several authors have investigated the numerical solution of such linear time dependent problems. For example, in [2, 3] the authors use standard finite element methods to approximate the nonlocal problem when the solution is continuous and discontinuous Galerkin for a discontinuous solution.

Received by the editors December 15, 2016 and, in revised form, June 15, 2017.

2000 *Mathematics Subject Classification.* 35R35, 49J40, 60G40.

The computational cost of solving a nonlocal problem is often higher than solving a local problem. One difficulty is that the bandwidth of the resulting matrix is typically larger for the nonlocal model due to the nonlocal interactions. Another difficulty arises when a Galerkin approach is used because the nonlocal integro-differential equation must be integrated over the spatial domain to obtain the weak formulation. This requires more complicated quadrature rules than when Galerkin methods are used to approximate standard PDE models, especially in the case of anomalous diffusion. In [4] the authors propose a coupled local-nonlocal model to help alleviate the computational costs of nonlocal problems for large-scale applications.

Radial basis functions (RBFs) are a class of functions which depend only on the distance to a fixed point so they are easily used on scattered grids and in higher dimensions. RBFs have their origins in techniques for performing function interpolation and were introduced in 1971 for topological interpolation using scattered data [5]. Since then they have become a powerful tool for multivariable interpolation problems [6]. Advantages of using RBFs for multivariable interpolation are their ease in using scattered data, their high rate of convergence and the fact that they are insensitive to the dimension of the space.

RBFs have been successfully used to solve PDEs from different standpoints. An RBF-based collocation method for elliptic problems was introduced in [7, 8] in 1990. Wendland [9] in 1999 used a Galerkin-RBF approach where the approximating functions and test functions were radial basis functions. In [10] the authors introduced an approach for using finite difference approximations on a scattered grid where RBFs were used to generate the finite difference stencils and in subsequent works [11, 12] use these RBF-generated stencils to solve the shallow water equations on the sphere.

In this work we use RBFs in a novel way to solve nonlocal diffusion problems. RBFs have been used to solve nonlocal problems when a Galerkin formulation is employed with RBFs as the approximating functions. See, for example [13, 14, 15]. Here we do not use a Galerkin method but instead approximate the time derivative by a standard backward difference formula (BDF) and approximate the nonlocal integral with an RBF-generated quadrature rule. This alleviates the difficulty previously described when using a Galerkin formulation. These quadrature rules are derived by extending the procedure of [10] for obtaining RBF-generated finite difference stencils on scattered grids. It is not feasible to use standard quadrature rules such as Newton-Cotes rules for approximating the nonlocal integral because of the singularity in the integrand; in addition, Gaussian quadrature rules are not practical when solving a nonlocal diffusion problem because each quadrature point is also a point where the time derivative must be approximated.

The RBF-generated quadrature rules are able to accurately approximate the singular nonlocal integral and as the number of quadrature points are increased near spectral accuracy can be obtained. However, conditioning of the matrix for deriving the quadrature rule becomes an issue when the grid spacing goes to zero; this conditioning issue is also present when RBFs are used for interpolation.

In § 2 we derive RBF-generated quadrature rules for a nonlocal integral in one dimension and present numerical results for approximating nonlocal integrals using these new quadrature rules. In § 3 we apply the RBF-generated quadrature rules to solve a nonlocal diffusion model in one dimension. In § 4 we extend the approach to two dimensions and indicate how it can be easily extended to three dimensions. Here we consider using both the tensor product of one dimensional rules as well as deriving new rules in two dimensions. These rules are applied to solving nonlocal diffusion problems in two dimensions. Numerical results for anomalous diffusion are presented for both one-dimensional and two-dimensional problems.

2. RBF-generated stencils

In [10] the authors derived RBF-generated finite difference stencils on scattered grids which allow the number of stencil nodes to remain small without the loss of accuracy usually associated with scattered node finite difference (FD) formulas. In this section we extend their approach to derive an RBF-generated quadrature rule for a nonlocal integral. We then apply the quadrature rules to nonlocal integrals and illustrate their numerical accuracy.

2.1. Choices of radial basis functions. A radial basis function (RBF) is radially symmetric about a given point so its value only depends on the distance from that point. If $\tilde{x}$ is the given point we denote the RBF as $\phi(r)$ where $r = \|x - \tilde{x}\|$; here we use the Euclidean distance which is typical. It is often useful to control the "flatness" of the RBF by including a so-called shape function ϵ so that to be precise we should write $\phi(r, \epsilon)$. However, for brevity, we omit the explicit dependence on the shape parameter unless we want to emphasize this dependence.

In generating the quadrature rules for the nonlocal integral one must select a specific RBF. There are a myriad of choices but probably the most common globally-supported RBF used in the literature is the multiquadric where $\phi(r,\epsilon) = \sqrt{1 + (\epsilon r)^2}$. Another popular RBF is the Gaussian where $\phi(r,\epsilon) = e^{-(\epsilon r)^2}$; we also use the inverse multiquadric where $\phi(r,\epsilon) = 1/\sqrt{1 + (\epsilon r)^2}$. All three are global-supported RBFs and C^∞. The Gaussian is a positive definite function so it will be used in most of the approximations but in § 2.3 we present simulations using all three RBFs and show that they give comparable results.

Once the specific RBF is chosen, one must choose a value for the shape parameter ϵ. As $\epsilon \to 0$ the flatness of the RBF increases. Lowering the value of the shape parameter typically increases the accuracy but only until ill-conditioning occurs [16]. Increasing the value of ϵ improves the conditioning of the linear systems but also increases the error. There has been some work towards choosing an optimal value for the shape parameter using greedy or genetic algorithms but we will not address this here. In situations where the linear system for determining the quadrature weights becomes ill-conditioned, we investigate increasing the shape parameter.

2.2. Derivation. Typically when a finite difference approximation to a derivative is generated on a uniform grid, we choose the points in the stencil and derive a formula which exploits the symmetry of the points. On a nonuniform grid typically many more quadrature points are needed to obtain the same degree of precision as on a uniform grid. Using the RBF-generated stencils derived in [10] alleviates this problem.

To derive an RBF-generated finite difference approximation using a specific RBF, we first choose the points in the stencil and define translations of the given RBF so that we have an RBF centered at each of the points in the stencil. Then we require the approximation to be exact for translations of the RBF in lieu of making it exact for monomials.

For example, assume we want to determine an RBF-generated three-point stencil for $u''(x_i)$ using the RBF $\phi(r)$ and the points x_{i-1}, x_i, x_{i+1}. Specifically let

$$u''(x_i) \approx c_{i,1}u(x_{i-1}) + c_{i,2}u(x_i) + c_{i,3}u(x_{i+1})\,,$$

where we seek the coefficients $c_{i,j}$, $j = 1, 2, 3$. Instead of making this formula exact for $1, x, x^2$ we make it exact for the RBFs $\phi(|x-x_{i-1}|)$, $\phi(|x-x_i|)$, and $\phi(|x-x_{i+1}|)$. Enforcing these conditions leads to the three equations

$$\phi''(|x_i - x_k|) = \sum_{j=i-1}^{i+1} c_{i,j}\phi(|x_j - x_k|), \quad k = i-1, i, i+1\,. \tag{1}$$

As in RBF interpolation, if $\phi(r)$ is not positive definite then the rule is typically required to also satisfy some monomials exactly; in the case when the multiquadratic RBF is chosen the rule is required to satisfy a constant exactly as well as the interpolation conditions given in (1). See [10] for details. The coefficient matrix for determining the RBF-generated finite difference approximation is a distance matrix and is identical to the RBF interpolation matrix. The matrix is symmetric and for strictly positive definite radial basis functions the matrix is positive definite [17]. The right-hand side requires differentiating the radial basis function which is easily done if we choose an RBF which is continuously differentiable. On a uniform grid, as the grid spacing goes to zero, the RBF-generated stencil for $u''(x)$ approaches the standard second centered difference stencil.

We now want to extend this approach to derive quadrature rules for a specific nonlocal integral in one spatial dimension using RBFs. In particular, we consider the integral

$$\mathcal{I}(u) = \int_{x-\delta}^{x+\delta} \frac{u(x) - u(z)}{|x-z|^{1+2s}}\,dz\,, \tag{2}$$

where x is a fixed point and δ is the given horizon. Note that the integrand of (2) is not defined at $z = x$ so we can not choose x as a quadrature point. For N quadrature points z_i and weights w_i we have an approximation to $\mathcal{I}(u)$ of the form

$$Q_N(u) = \sum_{i=1}^{N} w_i \frac{u(x) - u(z_i)}{|x-z_i|^{1+2s}}. \tag{3}$$

To determine the quadrature rule, we first choose quadrature points $z_i \in [x-\delta, x+\delta]\backslash\{x\}$, $i = 1, 2, \ldots, N$. Next we choose a radial basis function $\phi(r)$ and consider the translations $\phi(|z - z_i|)$, $i = 1, 2, \ldots, N$ centered at each quadrature point in the stencil. To derive the weights, we require the approximation $Q_N(u)$ to be exact when u is set to each translation of $\phi(r)$. If the multiquadric RBF is used for interpolation or generating RBF-FD stencils the approximation is also required to be exact for constants but we do not have to impose this here because it is automatically satisfied.

For example, suppose we have a mesh $\{x_j\}$, $j = 1, 2, \ldots, M$ with $x_j < x_{j+1}$ for all j and we set $x = x_i$ in the definition of $\mathcal{I}(u)$. Furthermore, assume we want to derive a two-point quadrature formula $Q_2(u)$ where $z_1 = x_{i-1}$ and $z_2 = x_{i+1}$. We require the quadrature rule to be exact when $u(x) = \phi(|x-z_1|)$ and $u(x) = \phi(|x-z_2|)$, i.e., $Q_2\big(\phi(|x-z_j|)\big) = \mathcal{I}\big(\phi(|x-z_j|)\big)$ for $j = 1, 2$. With $x = x_i$ these requirements yield two equations for the unknown weights w_1, w_2 where the linear system is given by

$$\begin{pmatrix} \dfrac{\phi(|x_i - x_{i-1}|) - \phi(|x_{i-1} - x_{i-1}|)}{|x_i - x_{i-1}|^{1+2s}} & \dfrac{\phi(|x_i - x_{i-1}|) - \phi(|x_{i+1} - x_{i-1}|)}{|x_i - x_{i+1}|^{1+2s}} \\ \dfrac{\phi(|x_i - x_{i+1}|) - \phi(|x_{i-1} - x_{i+1}|)}{|x_i - x_{i-1}|^{1+2s}} & \dfrac{\phi(|x_i - x_{i+1}|) - \phi(|x_{i+1} - x_{i+1}|)}{|x_i - x_{i+1}|^{1+2s}} \end{pmatrix} \begin{pmatrix} w_1 \\ w_2 \end{pmatrix}$$

$$= \begin{pmatrix} \displaystyle\int_{x_i-\delta}^{x_i+\delta} \frac{\phi(|x_i - x_{i-1}|) - \phi(|z - x_{i-1}|)}{|x_i - z|^{1+2s}}\, dz \\ \displaystyle\int_{x_i-\delta}^{x_i+\delta} \frac{\phi(|x_i - x_{i+1}|) - \phi(|z - x_{i+1}|)}{|x_i - z|^{1+2s}}\, dz \end{pmatrix}. \tag{4}$$

In general, the entries of the coefficient matrix A for finding the weights in one dimension using quadrature points $\{z_i\}$ are

$$A_{i,j} = \frac{\phi(|x - z_i|) - \phi(|z_j - z_i|)}{|x - z_j|^{1+2s}} \tag{5}$$

for fixed x. For the case of two uniformly spaced quadrature points the coefficient matrix is symmetric; however for more than two quadrature points, even on a uniform grid, the matrix is not symmetric. For nonuniformly spaced quadrature points A is not symmetric. This is in contrast to the RBF interpolation matrix, or equivalently for the coefficient matrix in deriving RBF-FD stencils, where the entries are $\phi(|x_i - x_j|)$ thus yielding a symmetric matrix whose entries are even functions. The entries of the coefficient matrix given in (5) are not even and thus the standard argument for showing invertibility of the RBF interpolation matrix is not applicable here. However, numerical computations indicate that the matrix is indeed invertible. When the quadrature points are uniformly spaced about the fixed point x, then the quadrature weights are symmetric but for scattered quadrature points they are not.

The entries in the right-hand side vector $\mathbf{f}$ for the system to compute the weights in one dimension are

$$f_i = \int_{x-\delta}^{x+\delta} \frac{\phi(|x - z_i|) - \phi(|z - z_i|)}{|x - z|^{1+2s}}\, dz\,. \tag{6}$$

When an RBF-generated finite difference stencil is derived, the right-hand side of the system involves the derivative of an RBF which is easily computed but in this case it involves evaluating a singular, nonlocal integral.

2.3. Numerical results. In this section we apply RBF-generated quadrature rules to evaluate the one-dimensional nonlocal integral (2). Initially we compare symmetric two-point, four-point and six-point quadrature rules to approximate the nonlocal integral when $s = 0$. When the horizon δ is chosen as a function of h, we demonstrate that the two-point rule is second order accurate, the four-point rule is fourth order accurate and the six-point rule is sixth order accurate as $h \to 0$. On the other hand, if the horizon is fixed so that the interval of integration is fixed, then as the grid spacing decreases we can include more quadrature points and the results demonstrate near spectral accuracy. A numerical simulation using nonuniformly spaced quadrature points is also provided. When $s > 0$ the problem becomes more difficult because the right-hand side of the system to be solved for the weights is harder to accurately approximate. This is discussed and results for several values of $s \in (0, 0.9]$ are presented. To maintain the same convergence rates as in the case $s = 0$, the shape parameter is adjusted.

To calculate the quadrature weights a linear system with the right-hand side vector given by (6) must be solved. In one dimension and $s = 0$ this can often be found by using a symbolic algebra package but for $s > 0$ and higher dimensions this is not always possible. For the numerical results reported herein the entries in the right-hand side are computed using an adaptive Gauss quadrature rule. However, for small values of s when the integration interval $[x - \delta, x + \delta]$ is sufficiently large, a composite Gauss quadrature rule is adequate. In addition, as the quadrature

Table 1. Approximations to the nonlocal integral (2) using the two quadrature points $x-\delta, x+\delta$ and the multiquadic RBF for two different choices of $u(x)$ with $\delta = h$.

		$u(x) = x^4 - x^2$		$u(x) = \sin(\pi x)$	
h	Weight	Rel. Error	Rate	Rel. Error	Rate
1/4	0.125646	6.4296 e-02		2.0620 e-02	
1/8	0.062610	1.7178 e-02	1.90	4.6682 e-03	2.14
1/16	0.031265	4.3691 e-03	1.98	1.1308 e-03	2.04
1/32	0.015627	1.0970 e-03	1.99	2.8033 e-04	2.01
1/64	0.007813	2.7456 e-04	2.00	6.9930 e-05	2.00

points fill the interval of integration the coefficient matrix for calculating the weights becomes ill-conditioned. We discuss this and demonstrate the effect of increasing the shape parameter to try to alleviate the ill-conditioning. In cases where ill-conditioning of the system affects the results, we provide the magnitude of $K(A)$, the condition number of the linear system using the infinity matrix norm. All of the quadrature weights obtained are positive and their sum, which is the condition number of the rule, is less than the length of the interval $[x-\delta, x+\delta] = 2\delta$. The shape parameter used in calculating the weights was chosen to be one, unless otherwise indicated.

When calculating the error in the quadrature method, the exact solution must be known. As with determining the right-hand side of the linear system to calculate the weights, this can sometimes be found exactly. However, in the results presented here all exact values for the integral were determined using an adaptive Gauss quadrature method. All errors reported are relative errors.

In the initial results reported here the multiquadric RBF is used because it is the most prevalent RBF in the literature. Then these results are compared with ones using the Gaussian RBF and the inverse multiquadric RBF. The results are comparable and in the sequel the Gaussian RBF will be used in all simulations because it is positive definite.

We want to use these quadrature rules to solve a nonlocal diffusion problem, so we think of setting up a grid and using a subset of these grid points as quadrature points. We discretize the interval $[0,1]$ and set the fixed point x to be the midpoint of the interval, i.e., $x = 0.5$. The horizon, δ, is either fixed, e.g., $\delta = 1/8$, or if the grid is uniform with spacing h, the horizon δ is sometimes set as a function of h, e.g., $\delta = 2h$.

2.3.1. The case $s = 0$. We first look at approximating the nonlocal integral (2) using a two-point quadrature rule on a uniform mesh with $\delta = h$. In this case the 2×2 symmetric dense system (4) must be solved once to obtain the weights. Note that this rule uses the endpoints of the interval $[x-\delta, x+\delta]$ as quadrature points just like the trapezoidal rule but, of course, with different weights. However, when the trapezoidal rule is applied to this nonlocal integral the approximations fail to converge due to the singularity. In Table 1 we tabulate the relative error and numerical rate of convergence for the two-point RBF-generated rule when the multiquadric RBF is used. Because the grid is uniform, the weights are symmetric so only one is reported. Table 1 provides the results for two choices of $u(x)$. As can be seen from the table, the numerical rate of convergence is second order as $h \to 0$.

Table 2. Approximations to the nonlocal integral (2) using the two quadrature points $x-\delta, x+\delta$ when $u(x) = x^4 - x^2$ and the Gaussian and inverse multiquadric RBF are used.

	Gaussian RBF			Inverse Multiquadric RBF		
h	Weight	Rel. Error	Rate	Weight	Rel. Error	Rate
1/4	0.1266914	7.3151 e-02		0.1265515	7.1966 e-02	
1/8	0.0627365	1.9228 e-02	1.93	0.0628197	5.3139 e-03	1.81
1/16	0.031280e	4.8639 e-03	1.98	0.0312943	5.3139 e-03	1.95
1/32	0.0156288	1.2195 e-03	2.00	0.0156307	1.3392 e-03	1.99
1/64	0.0078130	3.0510 e-04	2.00	0.0078132	3.3547 e-04	2.00

Table 3. Numerical approximations to (2) for $u(x) = x^4 - x^2$ using 4 and 6 uniformly spaced quadrature points. The Gaussian RBF is used.

	$\delta = 2h$			$\delta = 3h$		
h	$K(A)$	Rel. Error	Rate	$K(A)$	Rel. Error	Rate
1/8	10^3	2.2257 e-04		10^6	1.8940 e-05	
1/16	10^5	1.4906 e-05	3.90	10^9	3.4452 e-07	5.49
1/32	10^6	9.4810 e-07	3.98	10^{11}	3.9685 e-09	5.87
1/64	10^8	6.3806 e-08	3.89	10^{14}	2.7832 e-08	-

Table 2 provides results for $u(x) = x^4 - x^2$ analogous to those in Table 1 using the Gaussian RBF and the inverse multiquadric RBF. If one compares these results to those in Table 1 for the same $u(x)$ we see that the errors are of the same magnitude and the rates are all second order. Because the Gaussian RBF is positive definite, we will use this in all subsequent simulations.

Table 3 provides results when four and six uniformly spaced quadrature points are used to approximate the given nonlocal integral. When four uniformly spaced quadrature points are used we set $\delta = 2h$ with an interval length of $4h$ and when six are used $\delta = 3h$ for an interval length of $6h$. A dense 4×4 or 6×6 linear system is solved to obtain the weights which are symmetric. For brevity, only the results for $u(x) = x^4 - x^2$ are provided; similar results were obtained for other choices of $u(x)$. In addition, the magnitude of the condition number of the coefficient matrix used in finding the weights is provided. As can be seen from Table 3, the approximation is fourth order when four quadrature points are used and sixth order when $\delta = 3h$ until the system becomes too ill-conditioned. To alleviate this ill-conditioning, one may increase the shape parameter as if often done in RBF interpolation but this also degrades the error as can be seen in Table 4 where we provide the results when the shape parameter is set to two and to ten for the case of six uniformly spaced quadrature points.

When implementing these RBF-generated quadrature rules to solve the nonlocal problem we typically implement them in one of two ways. If δ, which governs the horizon, is a function of the grid spacing h then we fix the quadrature rule as $h \to 0$. Numerical results using this approach are presented in Tables 1–4. In other situations, δ is fixed and doesn't depend on h. In this case, the quadrature rule can vary as the grid spacing is decreased. For example, if $\delta = 1/8$ then for $h = 1/8$

Table 4. Numerical approximations to (2) for $u(x) = x^4 - x^2$ using 6 uniformly spaced quadrature points and two different choices of the shape parameter. The Gaussian RBF is used.

	$\epsilon = 2$			$\epsilon = 10$		
h	$K(A)$	Rel. Error	Rate	$K(A)$	Rel. Error	Rate
1/8	10^4	2.8841 e-04	-	10^0	6.9487 e-02	
1/16	10^7	6.3667 e-06	5.50	10^2	3.0140 e-02	1.21
1/32	10^9	1.0854 e-07	5.87	10^3	2.3437 e-04	7.01
1/64	10^{11}	1.2596 e-09	6.43	10^6	6.3763e-06	5.18

Table 5. Numerical approximations to (2) for $u(x) = x^4 - x^2$ when $\delta = 0.125$. The number of uniformly spaced quadrature points in the horizon is increased as h is decreased. The Gaussian RBF is used with a shape parameter of one.

h	No. Quad. Pts.	$K(A)$	Rel. Error	Rate
1/8	2	10^0	1.9228 e-02	
1/16	4	10^5	1.4906 e-05	10.33
1/24	6	10^{11}	3.1003 e-08	15.23
1/32	8	10^{17}	3.5437 e-07	-

Table 6. Numerical approximations to (2) for $u(x) = x^4 - x^2$ when two nonsymmetric quadrature points are used in the interval $(x - \delta, x + \delta)$ with $\delta = h$. The Gaussian RBF is used with a shape parameter of one.

h	Weights		Rel Error	Rate
1/6	0.1009728	0.1006130	3.44336 e-02	
1/12	0.0501257	0.0501021	8.62528 e-03	2.00
1/24	0.0250216	0.0250202	2.15296 e-03	2.00
1/48	0.0125060	0.0125059	5.07620 e-04	2.08

we use the two-point rule, for $h = 1/16$ we use the four-point rule, etc. On the other hand, if one fixes the rule, say the two-point rule, then the errors will not decrease with h because both the rule and the interval length is fixed. In Table 5 we provide results when δ is fixed and the order of the rule is increased. As can be seen from the table, spectral convergence is achieved until the quadrature points fill the interval to the point that the coefficient matrix becomes ill-conditioned.

Lastly, we consider the case of a nonuniform grid; for simplicity we consider a two-point rule where the interval of integration is $[x - \delta, x + \delta]$ for $\delta = h$ but the quadrature points are no longer the endpoints of the interval but two grid points interior to the interval of integration. Depending on the grid, the rule could have, e.g., one quadrature point to the left of x and two to the right but we consider only the case where we have one point to each side of x so we can compare with the results in Table 2. For the grid used here, a uniform grid on $[0, 1]$ is mapped using $\sin(\pi x/2)$. Because the grid is nonuniform the weights are not symmetric. As can be seen from Table 6, the results are second order accurate as $h \to 0$ which is the same as for the uniform grid case.

2.3.2. The case $s > 0$. When values of s are chosen which are greater than zero then a couple of issues arise. First, the nonlocal integral given in (6) for the right-hand side of the system for calculating the weights is more difficult to approximate accurately. Figure 1 illustrates the behavior of the integrand in (6) when the Gaussian RBF is used with $x = 0.5, z_i = 0.375$ for various choices of s. The second difficulty is that exact values of the nonlocal integral when $s > 0$, even for polynomial choices for $u(x)$, are typically not available so that one must resort to using approximations to compute the exact integral. As the value of s increases, the difficulty of evaluating a good approximation for the exact integral increases.

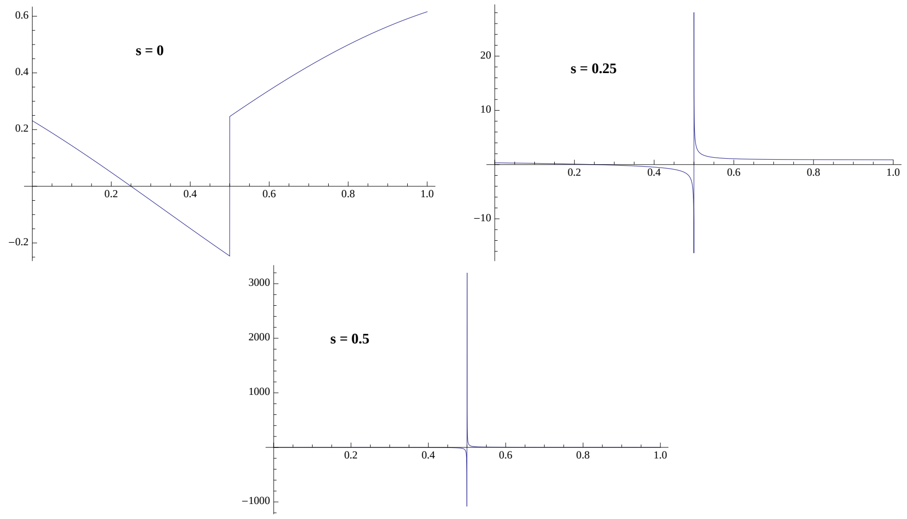

Figure 1. Comparison of the integrand in (6) for three values of s.

Table 7 demonstrates results for three values of s using a two-point RBF-generated quadrature rule on a uniform mesh. Comparing to Table 2 we see that we still get second order accuracy. Unlike the case when $s = 0$ an adaptive quadrature rule, rather than a composite rule, must be used to evaluate the right-hand side (6) to get accurate results. In these examples, the coefficient matrix for solving for the weights is well conditioned, as in the case for $s = 0$. For values of $s > 0.5$, we are not able to accurately compute the right-hand side of the system for the weights when $\epsilon = 1$. However, increasing the value of the shape parameter decreases the severity of the singularity in the integral in the right-hand side (6). The relative errors and rates for $s = 0.6, 0.7, 0.8$ and $s = 0.9$ using $\epsilon = 5$ are given in Table 8. To get second order accuracy, higher precision arithmetic was required. When more quadrature points are used so that they become closely packed, the conditioning of the matrix becomes an issue, as is the case for $s = 0$.

3. The One-dimensional Nonlocal Diffusion Problem

We consider the specific nonlocal diffusion problem

$$(7)\quad \frac{\partial u(x,t)}{\partial t} + \frac{2(1-s)}{\delta^{2-2s}} \int_{x-\delta}^{x+\delta} \frac{u(x,t) - u(z,t)}{|x-z|^{1+2s}}\, dz = f(x,t), \qquad \forall (x,t) \in \Omega \times (0,T]\,,$$

TABLE 7. Numerical approximations to (2) for $u(x) = x^4 - x^2$ using the two quadrature points $x - \delta$, $x + \delta$ when $\delta = h$. The Gaussian RBF is used with a shape parameter of one.

	$s = 0.2$		$s = 0.4$		$s = 0.5$	
h	Rel Error	Rate	Rel Error	Rate	Rel Error	Rate
1/8	1.9112 e-02		2.1557e-02		4.2314 e-02	
1/16	4.8567 e-03	1.98	5.4662 e-03	1.98	6.4938 e-03	2.70
1/32	1.2191 e-03	1.99	1.3728 e-03	1.99	1.6266 e-03	2.00
1/64	3.0508 e-04	2.00	3.4809 e-04	1.97	4.0684 e-04	2.00

TABLE 8. Numerical approximations to (2) for $u(x) = x^4 - x^2$ using the two quadrature points $x - \delta$, $x + \delta$ when $\delta = h$, $h = 1/8, 1/16, 1/32, 1/64$ for various values of $s > 0.5$. The Gaussian RBF is used with a shape parameter of five.

$s = 0.6$		$s = 0.7$		$s = 0.8$		$s = 0.9$	
Rel. Error	Rate	Rel Error	Rate	Rel Error	Rate	Rel Error	Rate
0.235 e-00		0.241 e-00		0.246 e-00		0.247 e-00	
3.296 e-02	2.56	3.553 e-02	2.76	3.854 e-02	2.67	4.208 e-02	2.55
9.706 e-03	2.02	1.046 e-02	1.76	1.133 e-02	1.77	1.236 e-02	1.77
2.504 e-03	1.95	2.697 e-03	1.96	2.922 e-03	1.96	3.185 e-03	1.96

where $\Omega = (a, b) \subset \mathbb{R}$ is a bounded domain, T a given time, $f(x, t)$ a given source function, and $s \in [0, 1)$. As before, $\delta > 0$ is called the horizon and governs the interval where interactions occur; specifically at any time t, the point x interacts with points z in the interval $[x - \delta, x + \delta]$. One major difference in nonlocal models and standard local models are boundaries. Because the domain of integration in the nonlocal integral may extend outside of Ω we must know the value of $u(x, t)$ there. To this end, we define the following regions

$$\Omega = (a, b), \qquad \Omega' = (a - \delta, b + \delta), \qquad \Gamma = \overline{\Omega'} \backslash \Omega = [a - \delta, a] \cup [b, b + \delta] \,. \tag{8}$$

Then, we append to (7) the volume constraint

$$u(x, t) = g(x, t) \quad \forall (x, t) \in \Gamma \times (0, T] \,, \tag{9}$$

where $g(x, t)$ is a given function. In addition, we impose the initial condition

$$u(x, 0) = u_0(x) \quad \forall x \in \Omega \,, \tag{10}$$

where $u_0(x)$ is a given function. In this section we apply the RBF-generated quadrature rules from § 2 to approximate the solution to this problem. Where available, we compare our results with those generated in [3] which uses a finite element (FE) method for approximating the nonlocal problem.

The advantage of using RBF-generated quadrature rules over higher order quadrature rules for the nonlocal integral is that we can set up a system of equations comparable in size to that obtained using standard methods such as FE. This would not be possible with higher order quadrature rules because in the nonlocal problem each quadrature point produces a node where the solution is unknown.

3.1. Approximation. We approximate this nonlocal problem defined by (7), (9), (10) using a BDF in time and an RBF-generated quadrature rule to approximate the nonlocal integral. Because an endpoint of the nonlocal integral may extend outside of Ω we partition the interval $[a-\delta, b+\delta]$ to get

$$a-\delta = x_{-p} < \cdots < x_{-1} < x_0 < x_1 < \cdots < x_J < x_{J+1} < \cdots < x_{J+q} = b+\delta \tag{11}$$

with $x_0 = a$, $x_{J+1} = b$, and J, p, q positive integers. We write a difference equation at each node x_i, $i = 1, \dots, J$ in Ω. For simplicity of exposition we approximate the time derivative using a first order BDF but it will be clear how to use a higher order BDF which we implement in the simulations. Assume we are using N points in the quadrature rule where the quadrature points are a subset of $\{x_{-p}, \dots, x_{J+q}\}$ and $\mathbb{J}_i$ denotes the set of N indices of these quadrature points for node x_i. Letting $U_i^n \approx u(x_i, t^n)$, we have the difference equation

$$\frac{U_i^n - U_i^{n-1}}{\Delta t} + \frac{2(1-s)}{\delta^{2-2s}} \sum_{q \in \mathbb{J}_i} w_q \frac{U_i^n - U_q^n}{|x_i - x_q|^{1+2s}} = f(x_i, t^n)\,, \tag{12}$$

at the grid point x_i; here w_q are the known RBF-generated quadrature weights. Once an RBF-quadrature rule is chosen, the weights can be determined at each node x_i, $i = 1, \dots, J$ as a preprocessing step; however if the grid is uniform, this only has to be done for a single node.

As an example, suppose we have a uniform grid and set $\delta = h$. In this case we use a two-point RBF-generated quadrature rule and at each x_i, $i = 1, \dots, J$ we use the quadrature points x_{i-1} and x_{i+1}. Assuming we have solved for the weights w_1, w_2, we have a linear system for the unknown vector $(U_1^n, U_2^n, \cdots, U_J^n)^T$ with a symmetric tridiagonal coefficient matrix A with entries

$$\begin{aligned} A_{i,i} &= \frac{1}{\Delta t} + \frac{2(1-s)}{\delta^{2-2s}} \left(\frac{w_1}{|x_i - x_{i-1}|^{1+2s}} + \frac{w_2}{|x_i - x_{i+1}|^{1+2s}} \right) \\ A_{i,i-1} = A_{i,i+1} &= \frac{-2(1-s)}{\delta^{2-2s}} \frac{w_1}{|x_i - x_{i-1}|^{1+2s}}\,. \end{aligned} \tag{13}$$

The ith component of the right-hand side is just $f(x_i, t^n) + U_i^{n-1}/\Delta t$ plus any terms that come from imposing the volume constraint. From our numerical results in § 2.3 we expect this scheme to be second order accurate in space.

If we set $\delta = 2h$ then we use a four-point quadrature rule with the quadrature points $x_{i-2}, x_{i-1}, x_{i+1}, x_{i+2}$. In this case we expect fourth order in space but the bandwidth of the matrix increases from three to five.

3.2. Numerical results. One difficulty in comparing results for nonlocal problems is the ability to manufacture exact solutions, i.e., once $u(x)$ is chosen we must determine the source term $f(x,t)$. If a symbolic algebra package is unable to evaluate the nonlocal integral then a technique described in [2] can be used where the term $u(x,t)$ in the nonlocal integral is written as $u(x+\xi, t)$ and is expanded in terms of a Taylor series. For choices of $u(x,t)$ which are polynomial in x, this method can produce exact solutions; in other cases sufficient terms in the Taylor series expansion must be used.

To compare our results with previously generated finite element approximations in [3], we consider the example where $\Omega = (0,1)$ and the exact solution $u(x,t) = x^2(1-x^2)\sin(t)$. Using the Taylor series approach, we have that when $s = 0$ the right-hand side $f(x,t)$ is given explicitly by

$$f(x,t) = x^2(1-x^2)\cos t + \sin t(-2 + 12x^2 + \delta^2)\,. \tag{14}$$

Table 9. A comparison of the FE and RBF approaches for solving the nonlocal diffusion problem when $s = 0$.

	RBF approach			FE approach	
h	No. Quad. Pts.	Rel.Error	Rate	Rel. L^2 Error	Rate
1/8	2	3.259014 e-03		3.3101 e-02	
1/16	4	5.959943 e-06	9.09	8.1641 e-03	2.02
1/32	8	1.338944 e-09	12.12	2.0353 e-03	2.00

In all cases the RBF error reported is a relative error measured in the standard Euclidean vector norm.

In the FE approximations reported in [3] the horizon δ is fixed at 0.125 and a uniform grid is used. The FE approximations are obtained using continuous piecewise linear polynomials on a uniform grid, a first order BDF in time and a 5-point Gauss Kronrod quadrature rule to approximate the nonlocal integral over z and a two-point Gauss quadrature rule to approximate the spatial integral in the weak form of the equations. Our calculations use a third order BDF due to the high accuracy of the quadrature rule.

The amount of total computations for the RBF quadrature approach and a FE approach for approximating the nonlocal diffusion problem can be analyzed from two different standpoints: the amount of computations needed to find the coefficient matrix formulation and the amount of computations it takes to solve the resultant system at each time step. The FE approach requires a quadrature routine for a nonlocal integral in each nonzero entry of its coefficient matrix and its right-hand side while the RBF quadrature approach is calculated as a preprocessing step and results in known coefficients just as when finite differences are used to solve a local problem. The quadrature rules used in the FE approach are minimized in terms of number of computations so that the desired convergence rates for the element being used is reached. In [3], it was shown that higher values of s require higher order quadrature rules to get optimal accuracy for the element used, for example with $s = .75$ a Gauss-Kronrod rule with 20 and 50 quadrature points is needed to near optimal convergence rates using linear elements.

The amount of computation needed at each time step can be determined by analyzing the coefficient matrix structure of each approach. In our numerical example, the horizon is fixed and not a function of h, so if we fix the quadrature rule, e.g., the two-point rule, then the error in approximating the nonlocal integral does not decrease with h. Instead, we employ a rule which incorporates all of the grid points in $[x_i-\delta, x_i+\delta]$. This does not increase the size of the matrix but it does increase its bandwidth. The coefficient matrix for the FE case is always tridiagonal; however, the entries are costly to compute because they involve not only approximating the spatial integrals but also the nonlocal integral compared with this approach where the entries in the coefficient matrix are computed using explicit formulas such as (13).

In Table 9 the FE results are compared with the RBF-generated quadrature approach for a fixed horizon of $\delta = 1/8$. As expected, the FE approximation is second order accurate using the L^2 norm because continuous piecewise linear polynomials are used and a backward Euler scheme with $\Delta t = \Delta x^2$. As can be seen from the table the RBF-generated quadrature approach yields spectral convergence as the number of quadrature points are doubled.

TABLE 10. A comparison of the FE and RBF approaches for solving the nonlocal diffusion problem when $s = 0.5$.

	RBF approach			FEM	
h	No. Quad. Pts.	Rel.Error	Rate	Rel. L^2 Error	Rate
1/8	2	5.1538 e-03		3.2785 e-02	1.91
1/16	4	4.9830 e-06	6.69	8.1128 e-03	2.01
1/32	8	1.3726 e-09	11.83	2.0275 e-03	2.00

Since we are interested in anamolous diffusion we compare the RBF approach with the FE results for $s = 0.5$ reported in [3]. To use the same exact solution as before we must generate a new source term; for $s = 0.5$ we have

$$f(x,t) = x^2(1-x^2)\cos t + \sin t\left(-2 + 12x^2 + \delta^2\frac{2}{3}\right).$$

Once again, the size of the coefficient matrix in each approach is the same and for two quadrature points both matrices are tridiagonal but the bandwidth of our coefficient matrix increases as the number of quadrature points increase. However, the entries in the matrix for our approach are much cheaper to compute.

The error results for the RBF approach in all cases are better than the FE approach because the FE results we compare against only use linear elements. This means that while the horizon stays fixed and the spatial discretization decreases the element size for the FE approach get smaller and the RBF-quadrature approach adds quadrature points. The convergence of the FE approach is determined by the type of the element and is second order for linear elements while the RBF approach decreases error at a spectral rate as quadrature points are added. We do not want to compare with FE results using higher order piecewise polynomials because they would increase the number of points in the formulation meaning a larger system needs to be solved. The matrices from the FE approach with piecewise linear elements are the same size as those of the RBF approach.

Oftentimes the horizon δ is set as a function of h rather than being fixed as in Tables 9 and 10. Simulations for the nonlocal diffusion model when the horizon is set as a function of h are as expected from the results of § 2 and are not presented here for brevity.

4. Solving the Nonlocal Diffusion Problem in Higher Dimensions

We now turn to solving the nonlocal diffusion problem in $\mathbb{R}^d$ for $u(\mathbf{x},t)$. For simplicity we take $d = 2$ but it should be obvious from the exposition how to extend the method to three dimensions. In particular, we seek $u(\mathbf{x},t)$ satisfying

$$(15)\quad \frac{\partial u(\mathbf{x},t)}{\partial t} + \frac{2(1-s)}{\delta^{2-2s}}\int_{H_{\mathbf{x},\delta}} \frac{u(\mathbf{x},t)-u(\mathbf{z},t)}{\|\mathbf{x}-\mathbf{z}\|^{2+2s}}\,d\mathbf{z} = f(\mathbf{x},t), \qquad \forall(\mathbf{x},t)\in\Omega\times(0,T],$$

where $\|\cdot\|$ denotes the Euclidean norm and $H_{\mathbf{x},\delta}$ denotes the neighborhood of $\mathbf{x}$ containing points which influence the solution at $\mathbf{x}$. As before, we define Ω' to be Ω appended with a layer of width δ around it and the region where the volume constraint $u(\mathbf{x},t) = g(\mathbf{x},t)$ is imposed is $\Gamma = \overline{\Omega}'\backslash\Omega$. As in the one-dimensional case, we first derive RBF-generated quadrature rules for the nonlocal integral. Then to solve the nonlocal diffusion problem we apply a BDF in time along with our quadrature rule for the nonlocal integral to get a set of difference equations.

Before we extend our RBF approach to higher dimensions, we must make a choice for the neighborhood $H_{\mathbf{x},\delta}$. In the literature this region is typically defined as the ball about $\mathbf{x}$ of radius δ, i.e., $H_{\mathbf{x},\delta} = \{\mathbf{z} \in \mathbb{R}^2 : \|\mathbf{x} - \mathbf{z}\| \leq \delta\}$ where $\| \cdot \|$ denotes the standard Euclidean norm. For discretization methods such as finite elements, this adds some computational difficulties because the nonlocal integral domain typically requires integration over partial elements. In the RBF-generated quadrature approach it only requires using a quadrature rule over a circle to evaluate the terms on the right-hand side of the linear system for the weights. A second choice for $H_{\mathbf{x},\delta}$ is the rectangular region $\{\mathbf{z} \in \mathbb{R}^2 : \|\mathbf{x}-\mathbf{z}\|_1 \leq \delta\}$. For this choice one may derive quadrature rules in the manner analogous to § 2 for one dimension or we may consider tensor products of one-dimensional rules. We consider quadrature rules using both choices for $H_{\mathbf{x},\delta}$ in § 4.1.

In § 4.2 we solve the two-dimensional nonlocal diffusion equation (15) using both circular and rectangular choices for $H_{\mathbf{x},\delta}$. In addition, for the case when $H_{\mathbf{x},\delta}$ is a circle we compare the results to some published finite element simulations.

4.1. RBF-generated quadature rules in two dimensions. In this section we derive quadrature rules to approximate the two-dimensional nonlocal integral

$$\int_{H_{\mathbf{x},\delta}} \frac{u(\mathbf{x}) - u(\mathbf{z})}{\|\mathbf{x} - \mathbf{z}\|^{2+2s}}\, d\mathbf{z} \tag{16}$$

when $H_{\mathbf{x},\delta}$ is either a circular region about $\mathbf{x}$ of radius δ or a square region of side 2δ centered at $\mathbf{x}$. To derive the quadrature rules directly as we did in one dimension, we choose an RBF $\phi(r)$, quadrature points $\mathbf{z}_i \in H_{\mathbf{x},\delta}$, $i = 1, \ldots, N$ and seek an approximation of the form

$$\int_{H_{\mathbf{x},\delta}} \frac{u(\mathbf{x}) - u(\mathbf{z})}{\|\mathbf{x} - \mathbf{z}\|^{2+2s}}\, d\mathbf{z} \approx \sum_{i=1}^{N} w_i \frac{u(\mathbf{x}) - u(\mathbf{z}_i)}{\|\mathbf{x} - \mathbf{z}_i\|^{2+2s}}\, .$$

To determine the weights we require the rule to be exact for translations of $\phi(r)$ to each quadrature point $\mathbf{z}_i$. Analogous to (5) the entries in the coefficient matrix for determining the weights are

$$A_{i,j} = \frac{\phi(\|\mathbf{x} - \mathbf{z}_i\|) - \phi\|\mathbf{z}_j - \mathbf{z}_i\|)}{\|\mathbf{x} - \mathbf{z}_j\|^{2+2s}} \tag{17}$$

and the terms for determining the right-hand side of the linear system for the weights are

$$f_i = \int_{H_{\mathbf{x},\delta}} \frac{\phi(\|\mathbf{x} - \mathbf{z}_i\|) - \phi(\|\mathbf{z} - \mathbf{z}_i\|}{\|\mathbf{x} - \mathbf{z}\|^{2+2s}}\, d\mathbf{z}\, . \tag{18}$$

The only difference in the approach for the two choices of the neighborhood $H_{\mathbf{x},\delta}$ is the quadrature rule used to approximate f_i in (18). As in the one-dimensional case, this matrix is dense and, for most cases, not symmetric. In higher dimensions, more quadrature points are used resulting in a larger matrix but these weights can be determined as a preprocessing step when solving the nonlocal problem. As in one-dimension, the errors in the quadrature weights can be degraded by (i) not calculating the terms f_i in (18) accurately enough, (ii) not obtaining an accurate enough approximation to the exact integral for comparison and (iii) an ill-conditioned coefficient matrix. These problems were illustrated in § 2 but can be more prominent in higher dimensions as more quadrature points and higher values of s are used.

Another option for computing the quadrature rule when $H_{\mathbf{x},\delta}$ is a rectangle of side 2δ centered at $\mathbf{x}$ is to use tensor products of one-dimensional rules. This option is discussed in the next section.

For the results reported for the accuracy of the RBF-generated quadrature rule, the exact solution $u(\mathbf{x}) = x_1^2 x_2$ is chosen with $\mathbf{x} = (x_1, x_2)$. All errors reported are relative errors. For the case when $H_{\mathbf{x},\delta}$ is a circular region we use Gauss cubature formulas to calculate the terms f_i in (18) and for the exact solution for comparison purposes. These rules are designed exclusively for integration domains of a circular shape; see [18] for details. For the case when $H_{\mathbf{x},\delta}$ is a rectangle we use an adaptive Gauss quadrature rule.

4.1.1. RBF-generated rules on a rectangle. We first define the neighborhood $H_{\mathbf{x},\delta}$ to be the square centered at $\mathbf{x}$ with side 2δ. When using a rectangular region for $H_{\mathbf{x},\delta}$ one may consider using either tensor products of one-dimensional rules or deriving the rules in a direct manner by making them exact for translations of the RBF. However, when tensor product rules are used, new quadrature rules in one dimension must be derived for the integral

$$\int_{x-\delta}^{x+\delta} \frac{u(x)-u(z)}{\|x-z\|^{2+2s}}\, dz\,; \tag{19}$$

i.e., where the kernel has been modified from (2) to match the exponent in the kernel of (16). The problem with this approach is that the integrals which must be calculated for the right-hand side of the linear system for computing the weights become much more difficult to approximate due to the new exponent in the kernel. To see this, note that when $s = 0$ the exponent $2 + 2s$ in the denominator corresponds to the $s = 0.5$ case with exponent $1 + 2s$ and when $s = 0.45$ the exponent is 2.9 which corresponds to $s = 0.9$ case for the results in § 2. So, in practice, we can only derive tensor product rules for the range $0 \leq s < 0.5$. In 2D, this difficulty does not occur because we have the Euclidean norm of the difference raised to the power $2 + 2s$ instead of the absolute value of the difference to that power. For small s the tensor product rules behave as expected; that is, if we use the same quadrature points as when we derive the rule directly, then the rates of convergence of the tensor product rules matches that of the rules derived in a direct manner. For this reason, we do not include the tensor product case here.

We first consider the case when $\delta = h$. If we use all of the possible quadrature points in the rectangle of side 2δ centered at $\mathbf{x}$, then there are eight quadrature points as illustrated in the right schematic in Figure 2. We also present results when we use the four quadrature points illustrated in the schematic on the left in the same figure. The results are tabulated for the $s = 0$ case for both choices of quadrature points in Table 11. As can be seen from the table, using four points in two dimensions yields second order convergence and using all eight quadrature points produces fourth order convergence. These results correspond to the one-dimensional results in Tables 2 and 3, respectively. We also present results for $s = 0.1$ and $s = 0.25$ for the four-point rule in Table 12 where second order accuracy is illustrated. As s increases, the ability to accurately calculate the terms on the right-hand side of the linear system for calculating the weights is degraded. For example, for $s = 0.1$ our adaptive Gauss quadrature scheme calculates the terms to an error of 10^{-12} but when $s = 0.25$ the scheme can only achieve an error of 10^{-8} before a reasonable minimum step size is reached. The magnitude of this error does not adversely affect the four-point rule but as more accurate quadrature rules are used, the effect is seen.

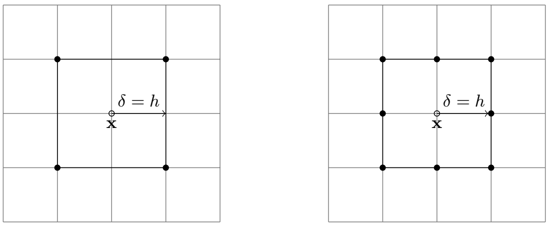

FIGURE 2. Two choices of quadrature points when $H_{\mathbf{x},\delta}$ is a square of side 2δ centered $\mathbf{x}$ and $\delta = h$.

TABLE 11. Comparison of the accuracy of the RBF-generated quadrature rules when $\delta = h$ and $s = 0$ for the two choices of sets of quadrature points illustrated in Figure 2.

	4 quadrature points		8 quadrature points	
h	Rel.Error	Rate	Rel. Error	Rate
1/8	5.6725 e-02		3.9960 e-03	
1/16	5.2226 e-02	0.12	1.1398 e-02	5.13
1/32	1.5526 e-02	1.75	6.2961 e-04	4.18
1/64	4.0229 e-03	1.95	3.8240 e-05	4.04

TABLE 12. Accuracy of a four-point RBF-generated quadrature rule to approximate the nonlocal integral when $\delta = h$ and $s = 0.1, 0.25$.

	$s = 0.1$		$s = 0.25$	
h	Rel.Error	Rate	Rel. Error	Rate
1/8	6.1082 e-02		6.8756 e-02	
1/16	5.3753 e-02	0.18	5.6319 e-02	0.29
1/32	1.5974 e-02	1.75	1.6722 e-02	1.75
1/64	4.1382 e-03	1.95	4.3228 e-03	1.95

When $\delta = 2h$ there are a total of 24 possible uniformly spaced quadrature points which are illustrated in the schematic on the right in Figure 3. However, the question is whether or not anything is gained by using all of these points. If we only use the four corner points $(x_1 \pm 2h, x_2 \pm 2h)$ then this mimics the case in Table 11 where second order convergence is achieved. However, from the results with $\delta = h$, we expect that fourth order convergence can be attained. We consider two optional choices as illustrated in Figure 3 with eight quadrature points; both of which achieve fourth order convergence. In Table 13 we only provide results for the choice of eight quadrature points illustrated in the middle schematic in Figure 3 and for all 24 points. Both choices of quadrature points produce results which are fourth order accurate but the case with 24 quadrature points has smaller errors

Table 13. Comparison of the accuracy of two RBF-generated quadrature rules for $s = 0$ when the quadrature points are chosen as in the middle and right schematics in Figure 3.

	8 quadrature points			24 quadrature points		
h	Rel.Error	Rate	$K(A)$	Rel. Error	Rate	$K(A)$
1/4	5.5401 e-03		10^1	4.9779 e-04		10^6
1/8	6.7329 e-04	3.04	10^2	3.9980 e-05	3.64	10^9
1/16	4.6808 e-05	3.85	10^3	2.6625 e-06	3.91	10^{13}
1/32	2.9989 e-06	3.96	10^5	2.1007 e-06	0.34	10^{15}

Table 14. Accuracy of an eight-point RBF-generated quadrature rule to approximate the nonlocal integral when $\delta = h$ and $s = 0.1, 0.25$.

	$s = 0.1$		$s = 0.25$	
h	Rel.Error	Rate	Rel. Error	Rate
1/8	3.7458 e-03		4.6831e-04	
1/16	5.6256 e-04	2.73	3.6890e-04	0.34
1/32	3.9797 e-05	3.82	1.4939e-05	4.63
1/64	2.6596 e-06	3.90	1.9204 e-03	-

than when only eight quadrature points are used. However, the condition number $K(A)$ of the coefficient matrix using 24 points for determining the weights grows much faster than the case with only eight points as can be seen from the table. Results for two values of $s > 0$ are illustrated in Table 14 using the eight-point rule used in Table 13. For the case $s = 0.25$ the error in approximating the right-hand side of the linear system is of the order 10^{-7} which contaminates the error when h is small enough.

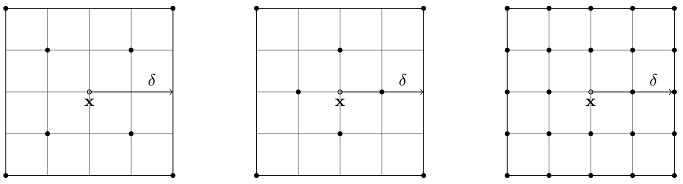

Figure 3. Three choices of quadrature point sets when $H_{\mathbf{x},\delta}$ is a square of side 2δ centered $\mathbf{x}$ and $\delta = 2h$.

4.1.2. RBF-generated rules on a circle. Now we consider the case when the nonlocal neighborhood $H_{\mathbf{x},\delta}$ is a ball of radius δ about $\mathbf{x}$. Possible quadrature points when $\delta = h$, $\delta = 2h$ and $\delta = 3h$ are illustrated in Figure 4. When $\delta = h$ we use four symmetrically placed quadrature points and for $\delta = 2h$ we use twelve. Table 15 provides results for $s = 0$ using both four and twelve quadrature points.

Table 15. Comparison of accuracy of four point ($\delta = h$) and twelve point ($\delta = 2h$) quadrature rules when the integration domain is a circle of radius δ centered at $\mathbf{x}$ and $s = 0$.

	$\delta = h$		$\delta = 2h$	
h	Rel.Error	Rate	Rel. Error	Rate
1/4	5.0694 e-01		5.2359 e-03	
1/8	5.6845 e-02	3.16	4.3977 e-04	3.57
1/16	2.2094 e-02	1.36	3.1605 e-05	3.80
1/32	5.9617 e-03	1.99	2.0457 e-06	3.95

Table 16. Accuracy of the RBF-generated quadrature rules when δ is fixed at 0.125 and $h \to 0$ for $s = 0$. $H_{\mathbf{x},\delta}$ is a ball of radius δ about $\mathbf{x}$.

h	No. Quad. Points	Rel.Error	Rate	$K(A)$
1/8	4	5.8158 e-02		10^0
1/16	12	5.1590 e-04	6.82	10^3
1/32	24	9.1757 e-07	9.14	10^{12}

The accuracy is second and fourth order for $\delta = h$ and $\delta = 2h$, respectively. Of course, one may investigate whether all twelve quadrature points are needed when $\delta = 2h$ as we did in the case when $H_{\mathbf{x},\delta}$ is a rectangle. If the horizon is fixed instead of a function of the grid spacing, then the number of quadrature points can be increased as the grid spacing decreases. This is illustrated in Table 16.

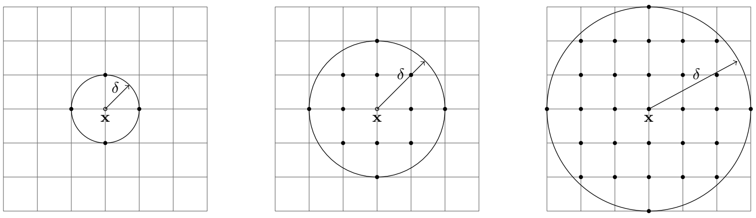

Figure 4. Quadrature points used when $H_{\mathbf{x},\delta}$ is a ball of radius δ about $\mathbf{x}$. The figure on the left is the case when $\delta = h$ and four points are used; the case on the right is when $\delta = 2h$ and 12 quadrature points are used.

4.2. Nonlocal diffusion problem in two dimensions. In this section we apply some of the quadrature rules derived in § 4.1 to solve the nonlocal diffusion equation given in (15) along with volume constraints and an initial condition. Analogous to the one-dimensional case, we obtain a difference equation by approximating the nonlocal integral by an RBF-generated quadrature rule and the time derivative by a high order BDF. The quadrature rule is chosen based upon whether the region $H_{\mathbf{x},\delta}$ is a circle or a rectangle and the order of accuracy required.

Table 17. A comparison of the FE and RBF approaches for solving the nonlocal diffusion problem when $s = 0$ and $u = x_1 x_2 t$.

	RBF approach			FE approach	
h	No. Quad. Pts.	Rel.Error	Rate	Rel. L^2 Error	Rate
1/8	4	5.4815 e-16		1.7244 e-02	
1/16	12	8.4850e-15		4.1271 e-03	2.06
1/32	48	1.0097 e-12		8.1020 e-04	2.35

Table 18. Accuracy of solution to the nonlocal problem when $u(\mathbf{x}, t) = x_1^2 x_2 t$, $\delta = 2h$ and $s = 0, 0.1, 0.25$. A 24 point quadrature rule is used to approximate the nonlocal integral with $\epsilon = 8$.

	$s = 0$		$s = 0.1$		$s = 0.25$	
h	Rel.Error	Rate	Rel. Error	Rate	Rel. Error	Rate
1/8	2.02419e-04	-	1.98842e-04	-	1.79318e-03	-
1/16	1.47768e-06	7.10	8.57702e-06	4.54	6.07776e-05	4.89
1/32	2.09035e-07	2.82	1.80505e-07	5.57	5.4471e-07	6.80

In the work [3] the authors present results solving this nonlocal diffusion problem when $u(\mathbf{x}, t) = x_1 x_2 t$ using continuous, linear triangular finite elements so we first compare our results with these. In this case the forcing term for the equation is just $f(\mathbf{x}, t) = x_1 x_2$ which can be found by the Taylor's series approach described in § 3. Here $H_{\mathbf{x},\delta}$ is chosen to be a circle of radius δ about $\mathbf{x}$ and $\delta = 0.125$ is fixed. Because the exact solution is linear in the independent variables, one might expect the FE approximation to be exact using linear elements but due to the approximation of the nonlocal integral it is not. However, our RBF-generated quadrature rule approach is exact. Because δ is fixed we increase the number of quadrature points as $h \to 0$. These results are illustrated in Table 17 for $s = 0$. For both approaches the size of the coefficient matrix is identical but for the RBF approach, as the number of quadrature points are increased entries in the bandwidth of the matrix are filled compared with the FE approach. However, for the FE approach entries in the matrix and right-hand side are computed using numerical quadrature; moreover, the nonlocal integral requires extra effort because one must determine which elements, or portions of elements, are included in the integration domain $H_{\mathbf{x},\delta}$ so this greatly complicates the computations.

When the exact solution to the nonlocal problem is chosen to be even slightly more complicated, such as $u(\mathbf{x}, t) = x_1^2 x_2 t$, then determining the forcing function $f(\mathbf{x}, t)$ becomes unwieldy using the Taylor's series approach described in § 3. For this reason, $f(\mathbf{x}, t)$ is usually found by calculating the temporal derivative of the exact solution and then approximating the nonlocal integral by using, e.g., Matlab. In Table 18 we present results for the eight-point quadrature rule on a rectangle used in Tables 13,14 for three values of s.

5. Concluding Remarks

In this work we show that using RBF-generated quadrature rules to approximate the nonlocal integral in the integro-differential equations arising in nonlocal models is a viable approach. In fact, nearly spectral accuracy can be obtained by increasing the number of quadrature points with minimal, if any, extra work as compared with standard approaches such as finite elements. Both nonlocal diffusion and anomalous diffusion problems can be solved using this approach. In two dimensions both circular and rectangular neighborhoods which define the region influencing a point $\mathbf{x}$ are considered and show similar results. The RBF-generated quadrature weights are determined as a preprocessing step by solving a small, dense system. Difficulties that arise in determining the weights are the ability to accurately approximate the integrals appearing on the right-hand side of the system and the possibility of ill-conditioning of the coefficient matrix as the spacing of the quadrature points goes to zero. This ill-conditioning is also seen when using RBF interpolation and is typically handled by increasing the flatness of the radial basis function by increasing the shape parameter. Accuracy of the right-hand side entries is essential when a high accuracy quadrature rule is being derived for small grid spacing.

Acknowledgement

This work was supported by the U.S. Department of Energy Office of Science grant DE-SC0009324.

References

[1] Q. Du, M. Gunzburger, R. B. Lehoucq, and K. Zhou, "Analysis and approximation of nonlocal diffusion problems with volume constraints," SIAM review, vol. 54, pp. 667–696, 2012.

[2] X. Chen and M. Gunzburger, "Continuous and discontinuous finite element methods for a peridynamics model of mechanics," Comp. Meth. in Appl. Mech. Engrg., vol. 200, pp. 1237–1250, 2011.

[3] D. Witman, M. Gunzburger, and J. Peterson, "Reduced-order modeling for nonlocal diffusion problems," Int. J. Num. Fluids, vol. 83, no. 3, pp. 307–327, 2017.

[4] M. DElia, M. Perego, P. Bochev, and D. Littlewood, "A coupling strategy for nonlocal and local diffusion models with mixed volume constraints and boundary conditions," Computers & Mathematics with Applications, vol. 71, pp. 2218–2230, 2016.

[5] R. L. Hardy, "Multiquadric equations of topography and other irregular surfaces," Journal of Geophysical Research, vol. 76, pp. 1905–1915, 1971.

[6] M. J. Powell, "Radial basis functions for multivariable interpolation: a review," in Algorithms for Approximation, pp. 143–167, Clarendon Press, 1987.

[7] E. J. Kansa, "Multiquadricsa scattered data approximation scheme with applications to computational fluid-dynamicsi surface approximations and partial derivative estimates," Computers & Mathematics with Applications, vol. 19, pp. 127–145, 1990.

[8] E. J. Kansa, "Multiquadricsa scattered data approximation scheme with applications to computational fluid-dynamicsii solutions to parabolic, hyperbolic and elliptic partial differential equations," Computers & Mathematics with Applications, vol. 19, pp. 147–161, 1990.

[9] H. Wendland, "Meshless galerkin methods using radial basis functions," Mathematics of Computation, vol. 68, no. 228, pp. 1521–1531, 1999.

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

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

[12] 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," Journal of Computational Physics, vol. 231, pp. 4078–4095, 2012.

[13] G. Ladányi and I. Jenei, "Analysis of plastic peridynamic material with rbf meshless method," Pollack Periodica, vol. 3, pp. 65–77, 2008.

[14] S. D. Bond, R. B. Lehoucq, and S. T. Rowe, "A galerkin radial basis function method for nonlocal diffusion," in Meshfree Methods for Partial Differential Equations VII, pp. 1–21, Springer, 2015.

[15] S. Kazem and J. Rad, "Radial basis functions method for solving of a non-local boundary value problem with neumanns boundary conditions," Applied Math. Modeling, vol. 36, pp. 2360–2369, 2012.

[16] T. A. Driscoll and B. Fornberg, “Interpolation in the limit of increasingly flat radial basis functions,” Computers & Mathematics with Applications, vol. 43, pp. 413–422, 2002.

[17] I. J. Schoenberg, “Metric spaces and positive definite functions,” Trans. of the AMS, vol. 44, pp. 522–536, 1938.

[18] A. H. Stroud, Approximate calculation of multiple integrals. Englewood Cliffs, New Jersey: Prentice-Hall, Inc, 1971.

Department of Scientific Computing, Florida State University, Tallahassee, FL 32306-4120
E-mail: `irlyngaas@gmail.com and jpeterson@fsu.edu`