FLORIDA STATE UNIVERSITY, DEPARTMENT OF SCIENTIFIC COMPUTING

Solving PDEs with Radial Basis Functions

Isaac Lyngaas

December 10, 2014

1 INTRODUCTION

In this project, radial basis functions (RBFs) will be utilized in order to approximate elliptic partial differential equations (PDEs). In order to do this, a method that uses these RBFs through a collocation scheme will be implemented to find an interpolating function for a given set of data points. In addition to the implementation of a sample problem in one dimension, a two dimensional problem will be solved in order to show how RBFs can be used to solve PDEs in higher dimension. Along with a comparison to the finite element method, this method will also be analyzed to determine computational efficiency and error rates.

2 RBF INTERPOLATION

RBFs are real valued functions of the following form

$$\phi(r, \varepsilon)$$

where

$$r = \|\mathbf{x}\|_2 = \sqrt{x_1^2 + x_2^2 + \ldots + x_d^2}\,,\ \mathbf{x} \in \mathbb{R}^d$$

and ε is a shape parameter that affects the surface of the RBF. The utility of the shape parameter will be touched on in Section 4 when a sample problem has been implemented. Two examples of radial basis functions that will be tested are in Table 2.1.

Like using piecewise polynomials basis functions in the finite element method, the goal of this method is to take a linear combination of these RBFs to create an approximation of a

Table 2.1: RBFs implemented

Name	$\phi(r,\varepsilon)$
Multiquadric	$\sqrt{1+(r\varepsilon)^2}$
Gaussian	$\exp(-(r\varepsilon)^2)$

function which in this case will be named u. To do this, an interpolation function will be found for a given finite set of N data points or centers $\mathbf{x}_1,\ldots,\mathbf{x}_N \in \mathbb{R}^d$ for which we know the value of $f(\mathbf{x})$ at these centers. The RBF interpolant becomes

$$s(\mathbf{x}) = \sum_{j=1}^{N} w_j \phi(\|\mathbf{x}-\mathbf{x}_j\|,\varepsilon) \tag{2.1}$$

where the interpolation conditions are enforced by the equations

$$s(\mathbf{x}_i) = f(\mathbf{x}_i)\ for\ i = 1,\ldots,N.$$

This results in the following NxN linear sytem

$$H\mathbf{w} = \mathbf{f} \tag{2.2}$$

where $H_{ij} = \phi(\|\mathbf{x}_i-\mathbf{x}_j\|,\varepsilon)\ for\ i,j = 1,\ldots,N$ and $f_i = f(\mathbf{x}_i)\ for\ i = 1,\ldots,N$. This system can be solved to find the weights $w_i\ for\ i = 1,\ldots,N$ that makes up the approximation function

$$\hat{u}(\mathbf{x}) = \sum_{i=1}^{N} w_i \phi(\|\mathbf{x}-\mathbf{x}_i\|,\varepsilon) \tag{2.3}$$

to the problem $u = f$.

3 Solving PDEs

Now that it has been shown how an interpolation function can be found for $u = f$, this process needs to modified to solve the problem $u'' = f$. To modify the method, an approximation of the second derivative for the interpolating function needs to be found. In this case, the interpolating function(2.1) turns into

$$\frac{\partial^2}{\partial \mathbf{x}_i^2} s(\mathbf{x}) = \sum_{j=1}^{N} w_j \frac{\partial^2}{\partial (x_i^1)^2}\phi(\|\mathbf{x}-\mathbf{x}_j\|,\varepsilon) + \cdots + \sum_{j=1}^{N} w_j \frac{\partial^2}{\partial (x_i^d)^2}\phi(\|\mathbf{x}-\mathbf{x}_j\|,\varepsilon),\ \mathbf{x}\in\mathbb{R}^d. \tag{3.1}$$

The RBFs being used are sufficiently differentiable and can be written in the form $\phi(r(\mathbf{x}))$ where $r(\mathbf{x}) = \|\mathbf{x}\|$. Thus, the chain rule can be used to find that

$$\frac{\partial \phi}{\partial x_i} = \frac{d\phi}{dx_i}\frac{\partial r}{\partial x_i} \tag{3.2}$$

and

$$\frac{\partial^2 \phi}{\partial x_i^2} = \frac{d\phi}{dr}\frac{\partial^2 r}{\partial x_i^2} + \frac{d^2\phi}{dr^2}(\frac{\partial r}{\partial x_i})^2 \tag{3.3}$$

where

$$\frac{\partial r}{\partial x_i} = \frac{x_i}{r}, \tag{3.4}$$

$$\frac{\partial^2 r}{\partial x_i^2} = \frac{1-(\frac{dr}{dx_i})^2}{r} \tag{3.5}$$

and $\frac{d\phi}{dr}$ and $\frac{d^2\phi}{dr^2}$ can be determined based on the RBF used.

Now that it has been determined how to calculate the derivatives of our interpolating function, our system from (2.2) needs to be modified to handle the boundary conditions that will be imposed by an elliptic PDE. A nice way to think about how the boundary conditions could be introduced to the system is to order the N_i centers in the interior of the domain from $i = 1,\ldots,N_I$ and the N_B centers on the boundary of the domain from $i = N_I+1,\ldots,N_B$. Numbering the system this way allows the equations for the system to be formed into two blocks. The equations are of the following form for the N_i interior centers with $\mathbf{x} \in \mathbb{R}^d$

$$\sum_{j=1}^{N} w_j \left[\frac{\partial^2}{\partial (x_i^1)^2}\phi(\|\mathbf{x}_i - \mathbf{x}_j\|,\varepsilon) + \cdots + \frac{\partial^2}{\partial (x_i^d)^2}\phi(\|\mathbf{x}_i - \mathbf{x}_j\|,\varepsilon) \right] = f(\mathbf{x}_i) \; for \; i = 1,\ldots,N_I. \tag{3.6}$$

The equations are of the following form

$$\sum_{j=1}^{N} w_j \left[\frac{\partial}{\partial x_i^1}\phi(\|\mathbf{x}_i - \mathbf{x}_j\|,\varepsilon) + \cdots + \frac{\partial}{\partial x_i^d}\phi(\|\mathbf{x}_i - \mathbf{x}_j\|,\varepsilon) \right] = g(\mathbf{x}_i) \; for \; i = N_I+1,\ldots,N_B \tag{3.7}$$

for boundary centers with the Neumann boundary condition g or of the form

$$\sum_{j=1}^{N} w_j \phi(\|\mathbf{x}_i - \mathbf{x}_j\|,\varepsilon) = h(\mathbf{x}_i) \; for \; i = N_I+1,\ldots,N_B \tag{3.8}$$

for boundary centers with Dirichlet boundary condition h. Obviously the boundary centers could also be split and reordered in the case where the problem has both Neumann and Dirichlet boundary conditions. Again, an NxN linear system has been found that can be solved to find $w_i \; for \; i = 1,\ldots,N$ which forms the approximation

$$\hat{u}(\mathbf{x}) = \sum_{i=1}^{N} w_i \phi(\|\mathbf{x} - \mathbf{x}_i\|,\varepsilon) \tag{3.9}$$

for the problem $u'' = f$. Refer to [1] for more on the derivation of the equations from Section 2 and Section 3.

4 APPLICATION

Now that the method has been described in detail and shown that it is generalizable to many dimensions, it will now be tested with the RBFs from Table 2.1 in a one dimensional and two dimensional elliptic problem. Analysis will be done on the error results from these implementations and the resultant system from the method will be reviewed to get an idea of its computational efficiency. In addition, a comparison with an implementation of the finite element method will also be done.

4.1 ONE-DIMENSIONAL POISSON PROBLEM

As a numerical example, the RBF collocation method is implemented on the following one-dimensional Poisson problem

$$-u''(x) = \pi^2 \cos(\pi x),\ 0 < x < 1$$
$$u(0) = 1,\ u(1) = -1.$$

In order to solve this problem, the centers that will be used in the RBF method need to be determined. To do this, the N centers will be found by evenly spacing them on the interval $(0,1)$. Once the centers are determined, the equations for the system are of the form (3.6) where $d = 1$ or (3.8) based on whether the centers are on the boundary or in the interior of the domain. The system is then solved for the weights using Gaussian Elimination and the $L2$ error is found for the approximating function using the exact solution $u(x) = \sin(\pi x)$. Table 4.1 shows the $L2$ error, condition number for the system, and convergence rates for this problem with increasing number of centers (and increasing matrix size) for the RBFs in Table 2.1 with shape parameter $\varepsilon = 1$.

Table 4.1: RBF Collocation with Gaussian Elimination

Multiquadric RBF				Gaussian RBF			
Matrix Size	Condition	$L2$ Error	Convergence	Matrix Size	Condition	$L2$ Error	Convergenc
5	775.016	0.0100661	-	5	1205.11	0.00239383	-
9	1.18179e+07	0.000293511	5.09995	9	1.58361e+10	1.3503e-06	10.7918
17	7.02682e+15	5.34531e-07	9.10092	17	9.4672e+17	7.52846e-09	7.4867
33	2.47597e+18	2.33997e-06	-2.13014	33	2.5591e+19	8.90231e-09	-0.241824
65	8.7493e+18	1.00795e-05	-2.10686	65	7.75284e+18	3.31118e-08	-1.89509

From the results in Table 4.1, it can be seen that convergence rates for this method are very good when the condition of the system $\kappa(H) < 10^{16}$, however the approximation gets less accurate as $\kappa(H)$ continues to increase past 10^{16}. This makes sense if taken into account the heuristic from numerical analysis that the solution of a linear system loses $k - t$ digits of accuracy with the machine epsilon $\epsilon_M \approx 10^{-t}$ and $\kappa(H) \approx 10^k$, which in this case $\epsilon_M \approx 10^{-16}$. So as $\kappa(H)$ increases past 10^{16}, the solution for the weights of the approximating function get

less accurate causing higher error in the approximation.

The problem with the condition number of the systems being solved with this method is directly related to the RBFs. The RBFs that were used in this method have global support meaning that these functions are nonzero over the entire domain that they are evaluated over. This causes the system that is being solved to be densely structured thus giving it a higher condition. Although this will not be illustrated, one way in which the problem of high condition number can be combatted is to vary the shape parameter ε in order to lower the condition number refer to [1] for more on this.

In order to do a comparison with another method, the same problem is solved using the finite element method with piecewise linear elements. The $L2$ error and convergence rate from solving the problem with this method with increasing number of nodes(and increasing matrix size) are in Table 4.2.

Table 4.2: Finite Element Method with Piecewise Linear Elements

Matrix Size	$L2$ Error	Convergence
128	3.54958e-05	-
256	8.87405e-06	1.99998
512	2.21852e-06	2
1024	5.5463e-07	2
2048	1.38658e-07	2
4096	3.46644e-08	2
8192	8.6661e-09	2

A simple way in which a comparison can be done between the two methods is to compare the systems that need to be solved in order to find approximations with similar error. In this case, a comparison is done with the $17x17$ dense linear system that was solved to find an approximation that had an $L2$ error of $7.52846e-09$ from Table 4.1 and the $8192x8192$ tridiagonal linear system that was solved to find an approximation that had an $L2$ error of $8.6661e-09$ from Table 4.2. In this case, it takes $\mathcal{O}(17^3) = \mathcal{O}(4913)$ operations to find an approximation with the RBF method compared to $\mathcal{O}(8192)$ operations to find an approximation with the finite element method that have similar $L2$ errors. Thus, the RBF collocation method can be competitive for this problem. Clearly a more thorough analysis needs to be done in cases where other variations of the Finite Element method are considered, but at least in the case of piecewise linear elements a comparison like this can show that the RBF method can be a viable alternative for approximating elliptic PDEs.

4.2 Two-Dimensional Poisson Problem

The RBF collocation will now be implemented on the following two-dimensional Poisson problem

$$-\triangle u(x,y) = 2\pi^2 \sin(\pi x)\sin(\pi y),\ (x,y) \in (0,2)x(0,2),\ u(x,y) = 0\ on\ \Gamma$$

Like the one-dimensional problem, the centers need to be determined which will be done by evenly distributing N centers in the x direction by N centers in the y direction to have N^2 centers over $(0,2)x(0,2)$. The equations for the system are then formed based on the locations of these centers using equation (3.6) where $d = 2$ for the interior centers and (3.8) for the boundary centers. Table 4.3 shows the errors and convergence rates from solving this problem with an increasing number of centers (and increasing matrix size) for the RBFs in Table 2.1 with shape parameter $\varepsilon = 1$.

Table 4.3: Error for 2D RBF Collocation

Multiquadric RBF				Gaussian RBF			
Matrix Size	Condition	*L2* Error	Convergence	Matrix Size	*L2* Error	Condition	Convergen
81	1.17741e+08	0.00525634	-	81	3.91906e+13	0.00157866	-
289	1.63311e+15	2.28267e-05	7.84719	289	1.8334e+19	1.81188e-06	9.767
1089	5.24815e+20	6.00223e-06	1.92715	1089	6.17798e+19	1.52997e-05	-3.07795
4225	1.56265e+21	0.000160482	-4.74077	4225	3.28412e+21	0.000158126	-3.3695

As was the case for the one-dimensional problem, the two-dimensional problem has high convergence rates as long as the condition number is not too large.

5 Conclusion

After investigating solving PDEs with the RBF collocation method, there are several advantages and disadvantages to be noted. One advantage is that this method is meshfree meaning that it would work very well for problems with irregular geometries because it would involve no overhead time for creating a mesh. Another advantage is the ease in which this method can move to many dimensions. The disadvantages are that the systems created by the method are densely structured so they are computationally inefficient to calculate and they quickly become ill-conditioned as the systems grow.

References

[1] Scott A. Sarra and Edward J. Kansa. Multiquadric radial basis function approximation methods for the numerical solution of partial differential equations. *Advances in Computational Mechanics*, 2, 2009.