$h - \varepsilon$ Refinement of Finite Difference Formulas Generated by Radial Basis Functions

Eric Heisler, Grady B. Wright

November 22, 2010

Abstract

Finite difference methods generated by radial basis functions(RBF-FD) are often used to numerically solve PDEs. Here we present a novel way to choose a value for the shape parameter found in many RBF-FDs. An effective value is found by examining the Taylor expansion of the RBF-FD formula and attempting to reduce the error terms in the approximation. This method is then tested on a variety of PDEs. Finally, several ideas to further improve the method are introduced and tested.

1 Introduction

When solving differential equations numerically using a discrete set of data, there are many techniques available. One of the most common is to use a finite difference(FD) method. The most familiar FD methods are based on polynomial interpolation. Another choice which has been gaining popularity is to interpolate the data with basis functions which depend on the distance between points, known as radial basis functions(RBFs). One interesting feature of many of these RBF-FD methods is that they contain a parameter which affects the shape of the RBF. This gives the user an additional control over the method, but presents the problem of choosing an effective value.

There exist many different kinds of radial basis functions which can be used to generate finite difference methods for solving differential equations. Several of these have been derived and explained in detail by Wright et al. [1, 2], Shu et al. [3, 4], as well as in [5, 6]. These RBF-FD methods have proven useful in many situations, especially when using irregularly arranged data points, or for spherical domains [7, 8, 9]. Also many of these RBFs, such as Gaussian and generalized multiquadric RBFs shown below, depend on a shape parameter, ε. Many studies have shown that the quality of the solution can depend on the value of this parameter(see for example [2, 4, 10, 11, 12, 13, 14], and that a good choice of ε can result in reduced error. However, there exists no efficient algorithm to select an effective value.

$$\begin{array}{rl} \text{Gaussian} & \phi(r) = e^{-\varepsilon^2 r^2} \\ \text{Generalized Multiquadric} & \phi(r) = (1_{(}\varepsilon r)^2)^{\nu/2},\ \nu \neq 0,\ \nu \notin 2\mathbb{N} \end{array}$$

Here we will introduce a method for selecting ε based on the Taylor expansion of the RBF-FD formula. This method will be numerically tested on Poisson's equation in two and three dimensions as well as Helmholtz's equation in two dimensions. Then we will discuss and test some additional improvements to the method including combining the method with multigrid, limiting the values of ε and iteratively improving the approximation of ε.

2 Procedure

For the purpose of this study, we will be focusing only on the Gaussian RBF, $e^{-\varepsilon^2 r^2}$ where r is the distance between points and ε is a shape parameter to be chosen. We will also assume a uniform grid with spacing h. However, the procedures described can be easily extended to other RBFs, and although the Taylor expanded error formulas depend on a uniform grid, the ideas could potentially be extended to other grid configurations.

As a first example, we will use a two dimensional Poisson's equation(1) with a five-point RBF-FD formula generated by the Gaussian RBF.

$$\nabla^2 u = \frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} = f(x, y). \tag{1}$$

First, we assume the domain $[0, 1] \times [0, 1]$ is divided into n equally spaced cells of length h, and let $f_{i,j}$ denote the value of some function f at (x_i, y_j) for $i, j = 0, 1, \ldots, n$. Then we define the following central difference operators:

$$\delta_x^2 f_{i,j} = \frac{f_{i+1,j} - 2f_{i,j} + f_{i-1,j}}{h^2}, \qquad \delta_y^2 f_{i,j} = \frac{f_{i,j+1} - 2f_{i,j} + f_{i,j-1}}{h^2}, \tag{2}$$

Using this notation, the 5-point RBF-FD approximation to Poisson's equation becomes

$$approx(\nabla^2 u_{i,j}) = \left[\delta_x^2 + \delta_y^2\right] \gamma(h, \varepsilon) u_{i,j} \approx f_{i,j} \tag{3}$$

Note that this formula is very similar to the standard cubic polynomial based five-point FD formula, but is multiplied by the following function, $\gamma(\varepsilon, h)$.

$$\gamma(\varepsilon, h) = \frac{(\varepsilon h)^2 e^{(\varepsilon h)^2} \left[e^{(\varepsilon h)^2} + (\varepsilon h)^2 - 1\right] \operatorname{csch}^2\left(\frac{(\varepsilon h)^2}{2}\right)}{2\left[1 + 3\cosh\left((\varepsilon h)^2\right) + 2\sinh\left((\varepsilon h)^2\right)\right]} \tag{4}$$

One important feature of this function is that $\gamma = 1$ when $\varepsilon = 0$. This allows us to choose an ε that gives us the same result as the standard method. However, since we are free to choose the value of ε, there is the potential that a better approximation can be found by choosing a more effective value. **(maybe this topic should be explained in the introduction)**

To find an effective value of ε, we will look at the error formula for the RBF-FD method. Given a uniform grid, the RBF-FD formula can be expanded in a Taylor series(5).

$$\begin{aligned}\left[\delta_x^2 + \delta_y^2\right] \gamma(h, \varepsilon) u =& \nabla^2 u + \frac{h^2}{12}\left[\frac{\partial^4 u}{\partial x^4} + \frac{\partial^4 u}{\partial y^4} + 9\varepsilon^2 f\right] + \\ & \frac{h^4}{360}\left[\frac{\partial^6 u}{\partial x^6} + \frac{\partial^6 u}{\partial y^6} + \frac{45}{2}\varepsilon^2\left(\frac{\partial^4 u}{\partial x^4} + \frac{\partial^4 u}{\partial y^4}\right)\right] + O(h^6).\end{aligned} \tag{5}$$

This expansion shows that the error in the approximation is $O(h^2)$, and thus has second order accuracy. Notice that the $O(h^2)$ term is dependent on ε^2, suggesting that it may be possible to reduce the error by choosing a value of ε that minimizes this error term. In fact, if we choose this particular value (6), we can eliminate the $O(h^2)$ term completely. Since the next remaining term in the expansion is $O(h^4)$, this approximation should now have fourth order accuracy.

$$\varepsilon^2(x, y) = \frac{-\left(\dfrac{\partial^4 u}{\partial x^4} + \dfrac{\partial^4 u}{\partial y^4}\right)}{9f} \tag{6}$$

The process of computing this ε is not particularly straightforward. Since it depends on derivatives of the function, u, which is unknown, an approximation of u must be found first. This can be done by using the standard FD method, or in other words, using an initial value of $\varepsilon = 0$. Now the derivatives in (6) can be approximated.

In practice, these fourth derivatives do not actually need to be approximated. It is possible to rearrange the formula so that it only requires the use of second derivatives by substituting (7) into the formula for ε. This simplifies the calculation and potentially reduces the error in the approximation of ε.

$$\frac{\partial^4 u}{\partial x^4} + \frac{\partial^4 u}{\partial y^4} = \nabla^2 f - 2\frac{\partial^4 u}{\partial x^2 \partial y^2} \tag{7}$$

It is important to remember that this value of ε is dependent on an approximated solution, and therefore includes some error. This means that it will not completely eliminate the $O(h^2)$ term in the error formula. Hopefully, however, the remaining $O(h^2)$ error will be reduced enough to become insignificant. This is to be tested in the following numerical examples.

3 Numerical results

3.1 Poisson's equation in 2-D

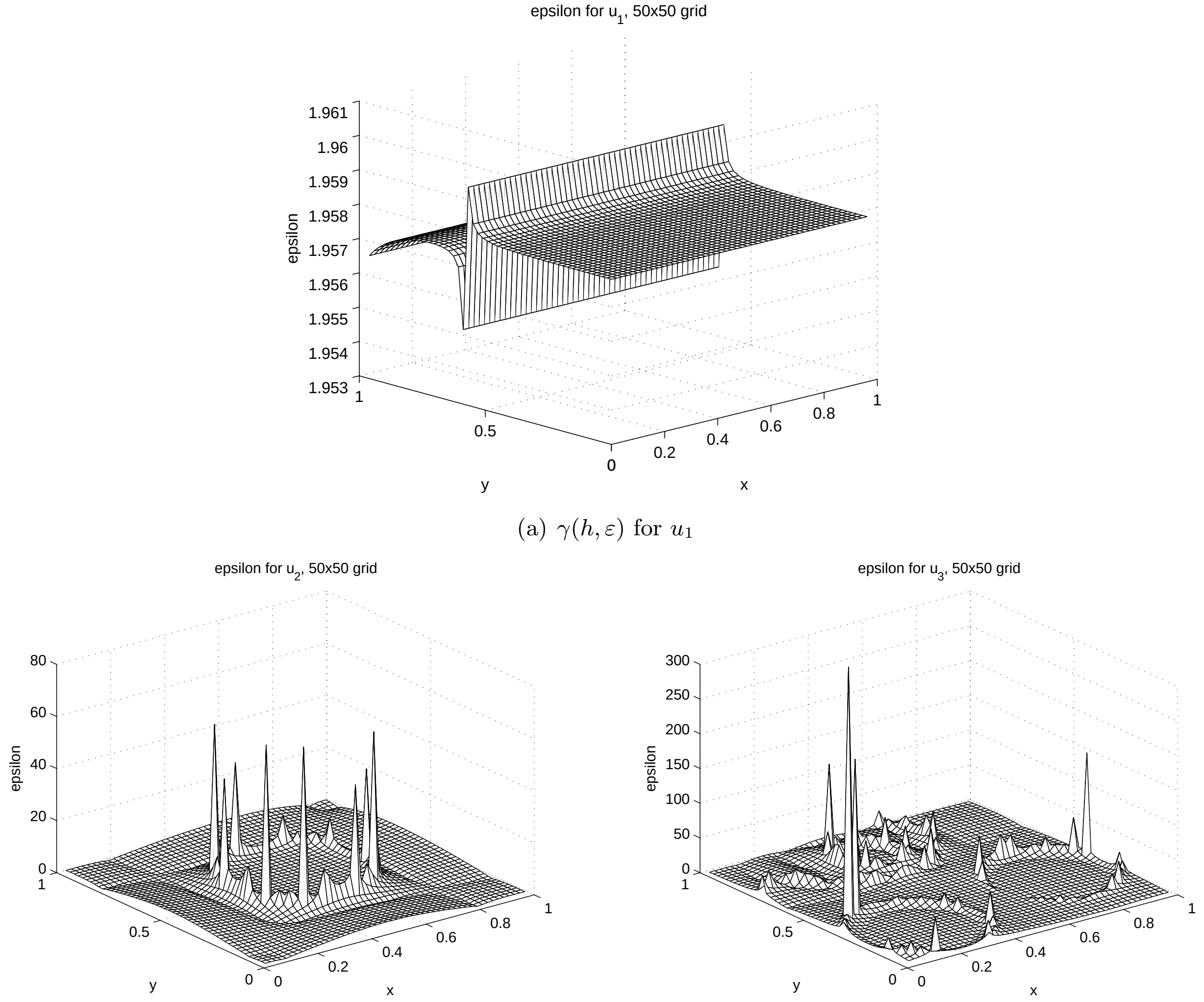
Here we will use the method described above to solve Poisson's equation in two dimensions(1). To test the effectiveness of this method, the following exact solutions for $u(x, y)$ will be used on the unit square $[0, 1]^2$.

$$\begin{aligned}
u_1(x,y) &= \sin(2\pi x)\sin(\frac{5}{3}\pi y) &(8)\\
u_2(x,y) &= \left(1-\frac{x}{2}\right)^6\left(1-\frac{y}{2}\right)^6 + 1000(1-x)^3x^3(1-y)^3y^3 + y^6\left(1-\frac{x}{2}\right)^6 + x^6\left(1-\frac{y}{2}\right)^6 &(9)\\
u_3(x,y) &= \frac{3}{4}\exp\left(-\frac{(9x-2)^2+(9y-2)^2}{4}\right) + \frac{3}{4}\exp\left(-\frac{(9x+1)^2}{49} - \frac{9y+1}{10}\right)\\
&\quad + \frac{1}{2}\exp\left(-\frac{(9x-7)^2+(9y-3)^2}{4}\right) - \frac{1}{5}\exp\left(-(9x-4)^2-(9y-7)^2\right) &(10)
\end{aligned}$$

u_1 was chosen for its simplicity, and u_2 and u_3 are examined by Ding et al. [15]. The corresponding f_1, f_2 and f_3 are determined by differentiating these functions, and the Dirichlet boundary conditions are determined by evaluating them on the boundary.

The function u_1 has the convenient property that the ε which eliminates the $O(h^2)$ term is constant over the whole domain. When these values for ε are approximated using the procedure described above, they appear to be roughly constant over the square with just a little sharp variation where the function f is close to zero. This could be expected because the formula for ε contains f in the denominator. This does not present such a significant problem because the values of f are known and such problematic regions can be handled specially. For the purpose of these tests, however, such regions will not receive special treatment. Plugging these ε values into γ gives the results seen in figure (1(a)). Similarly, ε was computed for u_2 and u_3. The resulting γ values are shown in figures (1(b)) and (1(c)). These plots also show regions where γ behaves erratically, but in these cases the function f is finite and appears smooth in these regions. Also, the values derived analytically from the u functions have a similar shape. This suggests that the irregular behavior is not simply the result of a poor approximation.

Now that we have approximated values of ε that should eliminate the $O(h^2)$ error term, we can test their effectiveness in solving Poisson's equation. To compare the results of the refined RBF-FD method



(a) $\gamma(h, \varepsilon)$ for u_1

(b) $\gamma(h, \varepsilon)$ for u_2

(c) $\gamma(h, \varepsilon)$ for u_3

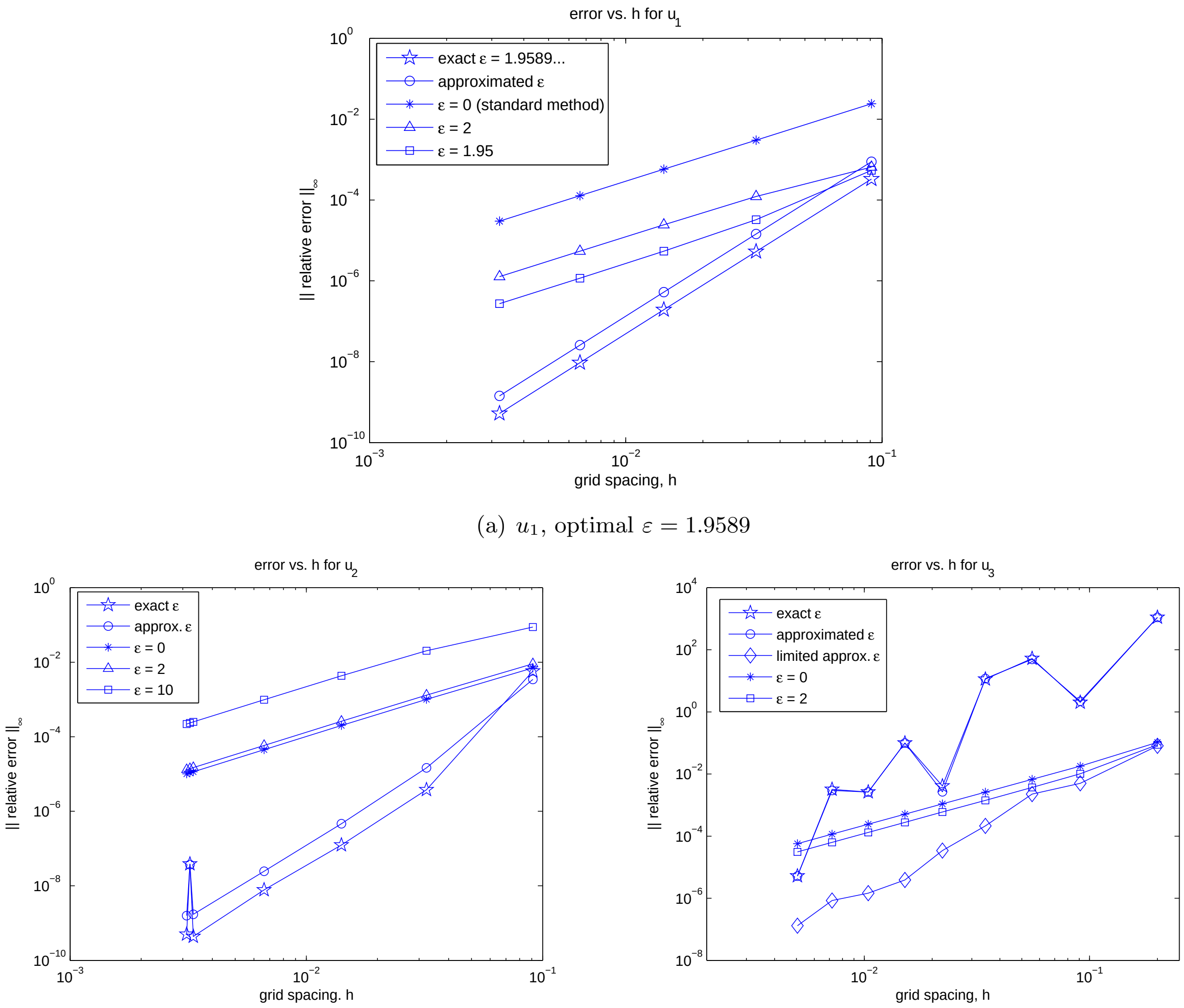
Figure 1: The approximated values of $\gamma(h, \varepsilon)$ for u_1, u_2 and u_3

with the traditional finite difference method, the solution was approximated for a range of grid spacings using the following values for ε:

- the value which eliminates the $O(h^2)$ error term derived analytically from the test function u
- the value which eliminates the $O(h^2)$ error term approximated by first solving with $\varepsilon = 0$
- $\varepsilon = 0$ (the standard FD method)
- some other constant values for ε

These results can be seen in figures (2.a,b,c). The shown error is the infinity-norm of the relative error. The two-norm was also calculated for each problem, but is very similar to the infinity-norm, so it is not displayed.

The results for u_1 and u_2 show both a significant decrease in the error and an increase in the order of accuracy. In both cases, the order of accuracy does increase from approximately $O(h^2)$ to approximately $O(h^4)$ for both the analytical and approximated values of ε. For u_1, the constant values 2 and 1.95 show



(a) u_1, optimal $\varepsilon = 1.9589$

(b) u_2

(c) u_3

Figure 2: error vs. h for u_1, u_2 and u_3. "'Exact ε"' refers to the analytic optimal value.

a decrease in error, but the accuracy is only $O(h^2)$. Since 1.95 differs by less than one percent from the optimal value, the choice of ε appears to be relatively sensitive.

The results for u_3 are a little less clear. The analytical and approximated values of ε give very erratic, but roughly $O(h^4)$ accuracy. The problem is that the magnitude of the error is incredibly high for coarse grids, and only matches the $\varepsilon = 0$ error for very fine grids. This makes the method very impractical for such problems. Fortunately, this can be remedied by limiting the values of ε as discussed in section 4. Figure (2(c)) shows that the limited values of ε give reduced error and higher order of accuracy.

3.2 A sixth order solution to Poisson's equation

We have shown that by choosing ε to eliminate the $O(h^2)$ term in the error formula for the finite difference method, we can successfully reduce the error and increase the order of accuracy for Poisson's equation. Looking again at the error formula(5), the $O(h^4)$ term also depends on ε^2. This means that we may be able to find a value which eliminates both the $O(h^2)$ and $O(h^4)$ terms. Since the next remaining term is $O(h^6)$, we would have a sixth order finite difference method. In fact, there may be a choice of ε

which eliminates even more terms in the error expansion. However, additional terms depend on higher order derivatives of the solution and require more computational effort. Most importantly, since we are only using an approximation of the solution to find ε, we must consider the error introduced by this approximation and can not expect to eliminate any of the error terms completely.

Here we have attempted to eliminate both the $O(h^2)$ and $O(h^4)$ error using a technique which reduces the dependency on high order derivatives of the solution, u, and tries to reduce error in the approximation of ε. It places more dependency on the known function, f, and makes use of exact derivatives of f. First, recall the central difference operators from (2) and the error expansion from (5). Using Poisson's equation, we can rewrite (5) as

$$\begin{aligned}\left[\delta_x^2+\delta_y^2\right]\gamma(h,\varepsilon)u_{i,j} =& f_{i,j}+\frac{h^2}{12}\left[\nabla^2 f-2\frac{\partial^4 u}{\partial x^2\partial y^2}+9\varepsilon^2 f\right]_{i,j}+\\ &\frac{h^4}{360}\left[\frac{\partial^6 u}{\partial x^6}+\frac{\partial^6 u}{\partial y^6}+\frac{45}{2}\varepsilon^2\left(\nabla^2 f-2\frac{\partial^4 u}{\partial x^2\partial y^2}\right)\right]_{i,j}+O(h^6).\end{aligned} \tag{11}$$

Plugging in the approximation

$$\frac{\partial^4 u}{\partial x^2\partial y^2}=\delta_x^2\delta_y^2 u_{i,j}-\frac{h^2}{12}\left[\frac{\partial^6 u}{\partial x^4\partial y^2}+\frac{\partial^6 u}{\partial x^2\partial y^4}\right]_{i,j}+O(h^4), \tag{12}$$

and using the relationships

$$\frac{\partial^6 u}{\partial x^6}+\frac{\partial^6 u}{\partial y^6}=\frac{\partial^4 f}{\partial x^4}+\frac{\partial^4 f}{\partial y^4}-\frac{\partial^6 u}{\partial x^4\partial y^2}-\frac{\partial^6 u}{\partial x^2\partial y^4}, \tag{13}$$

$$\frac{\partial^6 u}{\partial x^4\partial y^2}+\frac{\partial^6 u}{\partial x^2\partial y^4}=\frac{\partial^4 f}{\partial x^2\partial y^2}, \tag{14}$$

which are derived by differentiating Poisson's equation appropriately, we can rewrite (11) as

$$\begin{aligned}\left[\delta_x^2+\delta_y^2\right]\gamma(h,\varepsilon)u_{i,j} =& f_{i,j}+\frac{h^2}{12}\left[\nabla^2 f-2\delta_x^2\delta_y^2 u+9\varepsilon^2 f\right]_{i,j}+\\ &\frac{h^4}{360}\left[\nabla^4 f+2\delta_x^2\delta_y^2 f+\frac{45}{2}\varepsilon^2\left(\nabla^2 f-2\delta_x^2\delta_y^2 u\right)\right]_{i,j}+O(h^6).\end{aligned} \tag{15}$$

The $O(h^2)$ and $O(h^4)$ terms can thus be eliminated by selecting

$$\varepsilon_{i,j}^2=\frac{2}{45}\left.\frac{60\delta_x^2\delta_y^2 u-30\nabla^2 f-h^2[\nabla^4 f+2\delta_x^2\delta_y^2 f]}{12f+h^2\nabla^2 f-2h^2\delta_x^2\delta_y^2 u}\right|_{i,j}. \tag{16}$$

To get a truly sixth order accurate approximation, the $\nabla^2 f$ and $\nabla^4 f$ terms must be computed analytically. Otherwise this formula is only fourth order accurate. Moreover, the values of u must be close enough that the approximation in (12) does not contain any extra $O(h^2)$ error. Unfortunately, since u is unknown, the best we can do is to find a good initial approximation and hope that the additional error is small enough to be insignificant.

Using this method, the results in figure (1) were obtained. Though the sixth order method produced results with less error than the fourth order method for each case, the order of accuracy did not increase beyond fourth order. To improve the approximation of ε, better approximations for u were plugged into equation (16). However, the accuracy did not improve.

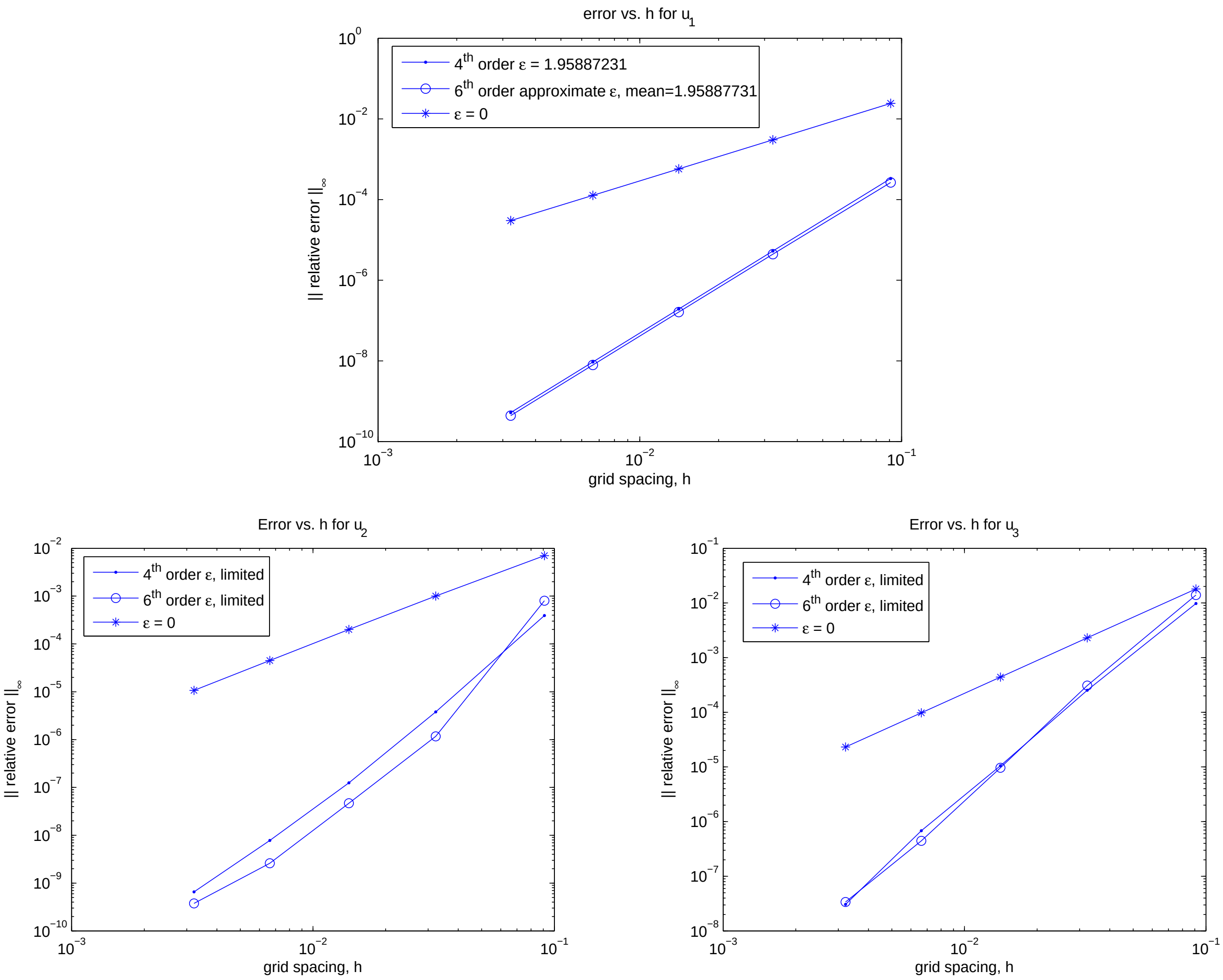


Figure 3: Error vs. h with the sixth order method for u_1, u_2 and u_3

3.3 Poisson's equation in 3-D

When solving problems in three or more dimensions, computations can very quickly become impractical due to the number of operations and memory requirements. Therefore it is important to use a method with a high order of accuracy. Similar to the two dimensional case, an RBF-generated finite difference scheme can be employed. The resulting seven-point approximation and corresponding $\gamma(\varepsilon h)$ are shown in (17) and (18). Again, we have the result $\gamma(\varepsilon, h) = 1$ when $\varepsilon = 0$, assuring us at least second order accuracy.

$$approx(\nabla^2 u_{i,j}) = \left[\delta_x^2 + \delta_y^2 + \delta_z^2\right]\gamma(h,\varepsilon)u_{i,j} \approx f_{i,j} \tag{17}$$

$$\gamma(\varepsilon, h) = \frac{2\left(\varepsilon h\right)^2\left[2\left(\varepsilon h\right)^2 e^{-(\varepsilon h)^2} - 3e^{-(\varepsilon h)^2} + 3\right]}{e^{-4(\varepsilon h)^2} + 4e^{-2(\varepsilon h)^2} - 12e^{-(\varepsilon h)^2} + 7} \tag{18}$$

It is also possible to find a value for the shape parameter, ε, which eliminates the $O(h^2)$ term in the error expansion (19). This value also depends on the solution to the equation, $u(x, y, z)$, and therefore requires the problem to be solved once before ε can be found.

$$\varepsilon^2 = \frac{-5\left(\dfrac{\partial^4 u}{\partial x^4} + \dfrac{\partial^4 u}{\partial y^4} + \dfrac{\partial^4 u}{\partial z^4}\right)}{42\nabla^2 u} \tag{19}$$

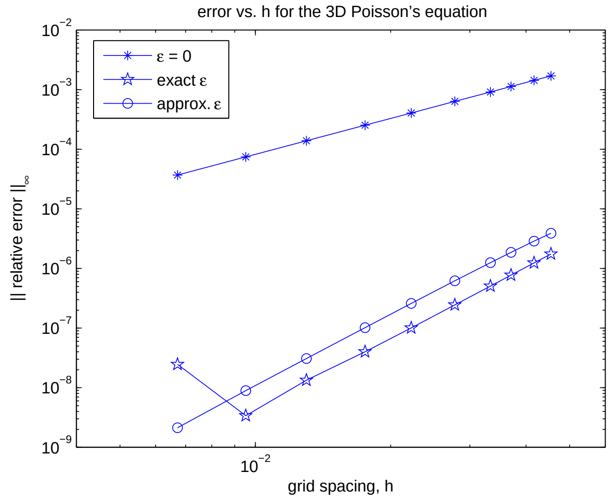


Figure 4: error vs. h for 3-D Poisson's equation

The method was tested on the following solution, $u(x,y,z)$ (20) [16]. This solution was chosen for its homogeneous boundary conditions, which allow the use of a fast, direct Fourier-based solver. Also, the value of ε from equation (19) is constant over the whole domain.

$$u(x,y,z) = \sin(\pi x)\sin(\pi y)\sin(\pi z) \tag{20}$$

The results in figure 4 show a dramatically reduced error and an order of accuracy increased from approximately $O(h^2)$ to approximately $O(h^4)$. One noticeable feature is that the computation of γ at very small grid spacings introduces considerable round-off error. This is the cause of the irregular behavior when using the analytically derived optimal ε, which is simply a constant. The error was reduced by rearranging the computation of γ, but was not eliminated. Interestingly, when using the approximated value, the error continues to decrease regularly, even surpassing the result of the analytical value.

3.4 Helmholtz's equation in 2-D

Another common problem involving the Laplacian operator is Helmholtz's equation (21).

$$\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} + \lambda u = f(x,y), \text{ where } \lambda \text{ is constant} \tag{21}$$

In fact, since RBF interpolation is only needed for the derivative term, the part of the RBF-FD formula which contains $\gamma(\varepsilon,h)$ is the same as that of Poisson's equation. Looking at the error expansion, the value of ε which eliminates the $O(h^2)$ term(22) is similar to the one from Poisson's equation, but the function f in the denominator becomes $f - \lambda u$.

$$\varepsilon^2 = \frac{-\left(\dfrac{\partial^4 u}{\partial x^4} + \dfrac{\partial^4 u}{\partial y^4}\right)}{9(f-\lambda u)} \tag{22}$$

The method was tested on the following problem(23) using the values $\lambda = 1.6, 6.4, 25.6$ [17]. The results, shown in figure 5, show consistent improvement for each value of λ. Again, the error is significantly reduced and the order of accuracy increases to approximately $O(h^4)$.

$$u = \sin(\pi x)\sin(2\pi y), \text{ (which gives } f = (\lambda - 5\pi^2)\sin(\pi x)\sin(2\pi y)) \tag{23}$$

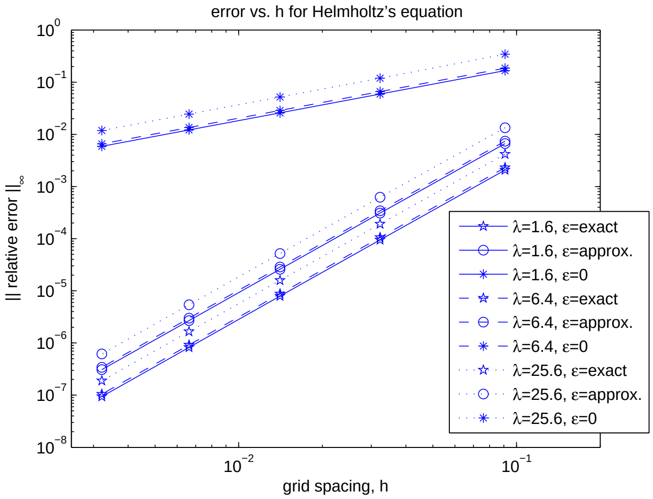


Figure 5: error vs. h for Helmholtz's equation with $\lambda = 1.6, 6.4, 25.6$

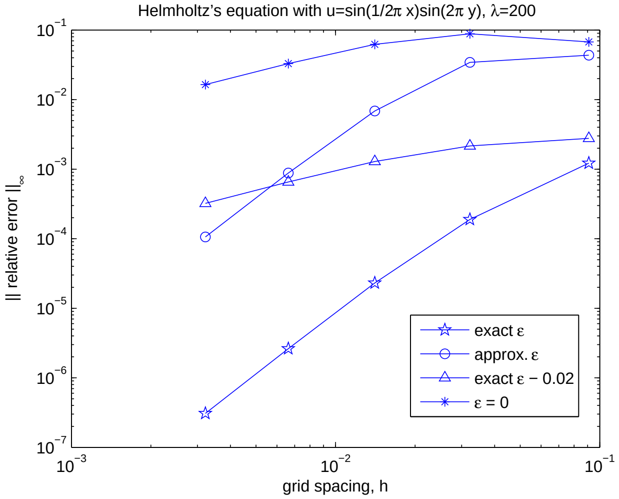


Figure 6: error vs. h for Helmholtz's equation with a particularly bad choice of u and λ

The method was also tested for several other $u(x, y)$ functions including polynomials and Gaussian peaks, and for a range of λ values. The results were consistently positive, meaning that we were able to achieve reduced error and a higher order of accuracy. Of particular note are some choices of functions and λ values which seem to converge very slowly and have relatively large error when using the traditional FD method. For example, when using the solution $u = \sin(\frac{1}{2}\pi x)\sin(2\pi y)$ with $\lambda = 200$ the error appears to decrease very slowly or not at all until h becomes very small. Even in this case, the ε-refined RBF method can achieve $O(h^4)$ accuracy and small error for more reasonable values of h as seen in figure 6. This represents a very significant improvement over the traditional method. Another feature that appears in this result is the large difference between the analytical and approximate ε. This suggests that it may be worthwhile to try improving the approximation of ε. This is explored further in section 4.

4 Further Improvements

4.1 Combining this method with multigrid

Multigrid has proven to be a very effective and efficient method for solving differential equations. We hope to improve it even further by combining it with this ε-refined RBF method. Here, we have implemented a single v-cycle multigrid method for solving Poisson's equation on a square. Since any multigrid method deals with a variety of different grid spacings, there are different ways to apply $\varepsilon - h$ refinement. At first, we tried computing values for ε for each discretization level of the multigrid cycle, but the results did not show the desired improvement in accuracy. We then tried computing ε for only the finest discretization, and then applying the corresponding γ to the right hand side of the equation, f. This modified right hand side was then used throughout the cycle.

Using the test functions from the section on Poisson's equation, u_1, u_2, u_3, the results showed smaller error and increased accuracy over the traditional method. However, it appeared to be very sensitive to the erratic behavior of ε for u_2 and u_3. In those cases, the error grows very large in the vicinity of the erratic behavior and increases with each iteration of the cycle. Fortunately it can be controlled by limiting the values of ε as discussed in the next section. Imposing these limits allows us to obtain results which show decreased error and greater accuracy. This can be seen in figures (7.a,b,c).

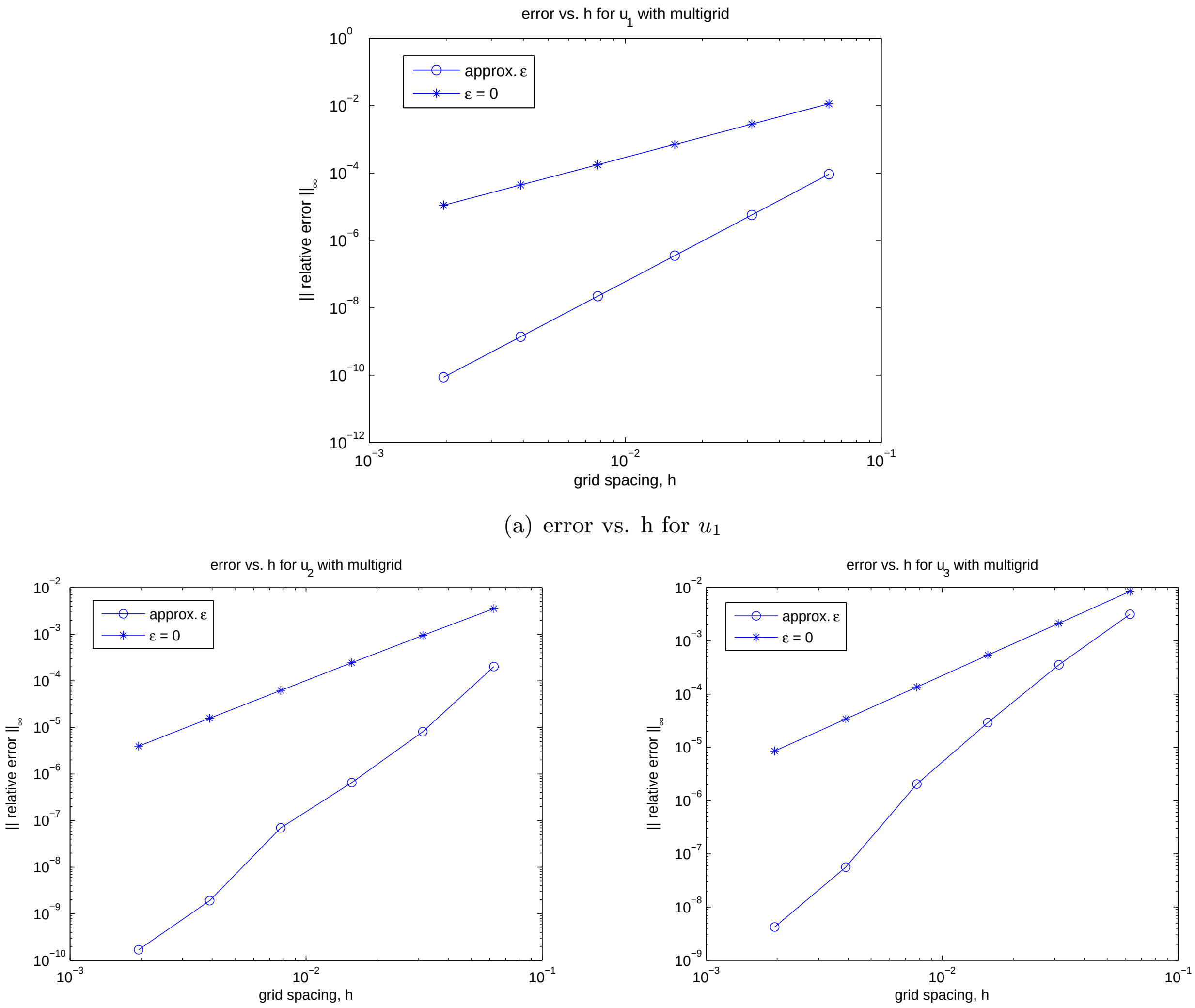
When using a multigrid method, the number of iterations for convergence is an important consideration. Also, recall that in order to use the RBF method, an approximate solution must be computed first. For these tests, the regular non-RBF method was used until the residual converged to a relatively large tolerance of 0.01, this solution was used as a starting point for the RBF method. It was then iterated until the residual was reduced to below 1e-8. Table (1) compares the number of iterations needed for the regular non-RBF method and our refined RBF method for a range of different grid spacings. For fine grid spacings the RBF and non-RBF methods require almost the same number of iterations, differing by only zero or one. For coarser grids the RBF method took a few more iterations, but still converged reasonably fast.

nodes, n	u_1		u_2		u_3	
$(h = \frac{1}{n})$	$\varepsilon = 0$	refined ε	$\varepsilon = 0$	refined ε	$\varepsilon = 0$	refined ε
2^4	8	11	8	12	10	14
2^5	10	11	11	12	11	12
2^6	11	12	11	12	12	14
2^7	12	12	12	14	12	13
2^8	13	13	13	13	13	13
2^9	14	15	14	14	14	15

Table 1: number of iterations required to converge to $\|residual\| < 1e-8$

4.2 Limiting the values of ε

As seen in some of the example problems, the choice of ε can have very erratic behavior. This is not a result of a poor approximation, but a property inherent to the form of ε. It was also seen that this led to large errors in the solution, particularly when using a multigrid method. For this reason, it seems appropriate to limit the values of ε. At first we tried simply constraining ε with $|\varepsilon| < Limit$. By tuning the limit, the irregularity in the solution's error could be smoothed out. However, each choice of limit only appeared to be effective for a particular grid spacing. Looking at γ in the approximation, ε is always multiplied by the spacing, h, suggesting that we should limit not only ε, but the product εh. Using



(a) error vs. h for u_1

(b) error vs. h for u_2

(c) error vs. h for u_3

Figure 7: error vs. h for Poisson's equation using multigrid with limited ε

the constraint: $|\varepsilon| < \frac{Limit}{h}$, the error induced by the erratic behavior was effectively smoothed out while maintaining the higher accuracy for different grid spacings. Figure (8) shows error vs. h using Poisson's equation with the solution u_3 for these four different situations: 1.)the unlimited, analytically derived ε, 2.)the unlimited, approximate ε, 3.)ε limited by $|\varepsilon| < Limit$, 4.)ε limited by $|\varepsilon| < \frac{Limit}{h}$.

4.3 Improving the approximation of ε

For all of the example problems, it was necessary to compute ε using an initial approximation of the solution. Since this approximation will certainly contain some error, we cannot expect to completely eliminate the $O(h^2)$ error. Even with this error, many of the cases shown resulted in an ε approximation which was close enough to achieve the desired result. However, there is a simple way to improve the approximation of ε. If the solution using ε has less error than the solution for $\varepsilon = 0$, this better solution can then be used to produce a better approximation for ε. This results in an iterative process to refine ε. This was tried with the test cases for Poisson's equation and Helmholtz's equation. For Poisson's equation, the error in the solution with the analytically derived ε and the approximated ε are very close. Because of this, when ε is iteratively improved, the error in the solution is reduced, but the improvement is not

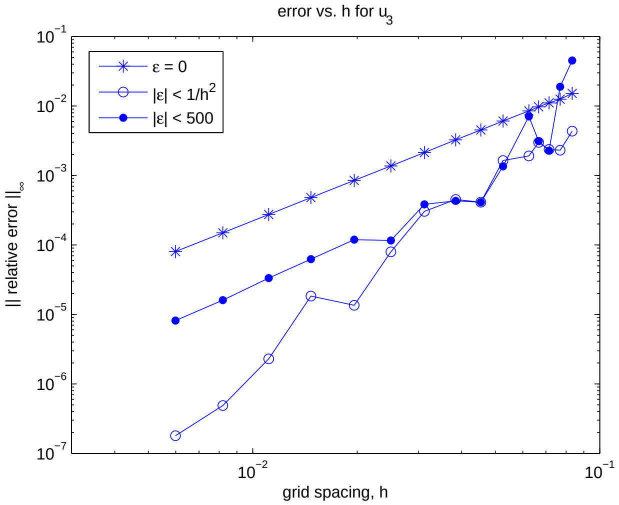


Figure 8: error vs. h for Poisson's equation with u_3 using different methods of limiting ε

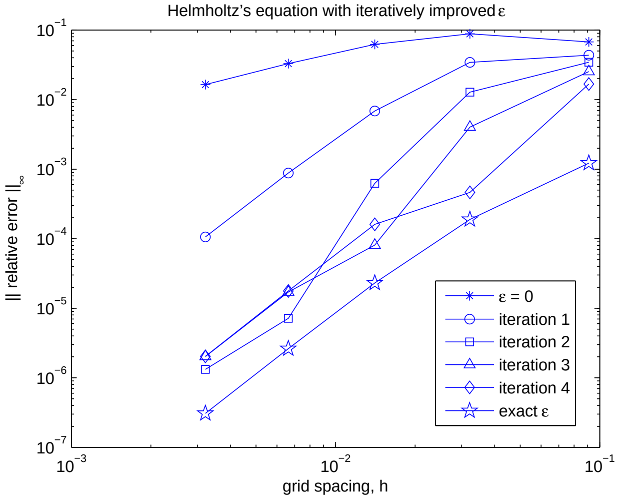


Figure 9: error vs. h for Helmholtz's equation with an iteratively improved ε

enough to justify the increase in computing effort. However, as was shown in the section on Helmholtz's equation, the difference can become relatively large. In this case, iteratively improving ε has a significant effect. This is shown in figure 9.

5 Conclusions

We have presented a method of selecting the shape parameter, ε, for Gaussian RBF-FD. The method is based on the error terms in the formula's Taylor expansion and gives values of ε which eliminate the leading terms. Thus, the order of accuracy can be increased. However, since the resulting formula for ε depends on the exact solution to the finite difference problem, in practice we can only approximate the desired ε. The numerical examples showed that this approximation was sufficient to increase the accuracy from the standard $O(h^2)$ to about $O(h^4)$ for Poisson's and Helmholtz's equations. A value of ε which eliminates all error larger than $O(h^6)$ was also found, but the numerical results showed only $O(h^4)$ accuracy.

We have also shown that the method is compatible with a multigrid method, and that in some cases the solution can be improved further by limiting or iteratively improving the approximation of ε.

References

[1] G. B. Wright. *Radial Basis Function Interpolation: Numerical and Analytical Developments.* PhD thesis, University of Colorado, Boulder, 2003.

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

[3] C. Shu, H. Ding, and K. S. Yeo. Local radial basis function-based differential quadrature method and its application to solve two-dimensional incompressible navier-stokes equations. *Comput. Methods Appl. Mech. Engrg.*, 192:941–954, 2003.

[4] C. Shu, H. Ding, and N. Zhao. Numerical comparison of least square-based finite-difference (lsfd) and radial basis function-based finite- difference (rbffd) methods. *Computers and Mathematics with Applications*, 51(8):1297 – 1310, 2006. Radial Basis Functions and Related Multivariate Meshfree Approximation Methods: Theory and Applications.

[5] A. I. Tolstykh, M. V. Lipavskii, and D. A. Shirobokov. High-accuracy discretization methods for solid mechanics. *Arch. Mech.*, 55:531–553, 2003.

[6] T. Cecil, J. Qian, and S. Osher. Numerical methods for high dimensional hamilton-jacobi equations using radial basis functions. *J. Comput. Phys.*, 196:327–347, 2004.

[7] B. Fornberg and C. Piret. A stable algorithm for flat radial basis functions on a sphere. *SIAM J. Sci. Comput.*, 200:178–192, 2007.

[8] N. Flyer and G. B. Wright. Transport schemes on a sphere using radial basis functions. *J. Comp. Phys.*, 226:1059–1084, 2007.

[9] N. Flyer and G.B. Wright. A radial basis function method for the shallow water equations on a sphere. *Proc. R. Soc. A*, 465:1949–1976, 2009.

[10] T. A. Driscoll and B. Fornberg. Interpolation in the limit of increasingly flat radial basis functions. *Comput. Math. Appl.*, 43:413–422, 2002.

[11] B. Fornberg, G. Wright, and E. Larsson. Some observations regarding interpolants in the limit of flat radial basis functions. *Comput. Math. Appl.*, 47:37–55, 2004.

[12] E. Larsson and B. Fornberg. Theoretical and computational aspects of multivariate interpolation with increasingly flat radial basis functions. *Comput. Math. Appl.*, 49:103–130, 2005.

[13] R. Schaback. Multivariate interpolation by polynomials and radial basis functions. *Constr. Approx.*, 21:293–317, 2005.

[14] G. Chandhini and Y. V. S. S. Sanyasiraju. Local RBF-FD solutions for steady convection-diffusion problems. *Int. J. Numer. Meth. Engng.*, 72:352–378, 2007.

[15] H. Ding, C. Shu, and D. B. Tang. Error estimates of local multiquadric-based differential quadrature (LMQDQ) method through numerical experiments. *Int. J. Numer. Meth. Engng*, 63:1513–1529, 2005.

[16] W. F. Spotz and G. F. Carey. A high-order compact formulation for the 3D Poisson equation. *Numer. Methods Partial Differential Equations*, 12:235–243, 1996.

[17] I. Singer and E. Turkel. High-order finite difference methods for the Helmholtz equation. *Computer Methods in Applied Mechanics and Engineering*, 163(1-4):343 – 358, 1998.