A Numerical Study of the Accuracy and Stability of Symmetric and Asymmetric RBF Collocation methods for Hyperbolic PDEs

Scott A. Sarra
Marshall University

May 9, 2007

Abstract

Differentiation matrices associated with radial basis function (RBF) collocation methods often have eigenvalues with positive real parts of significant magnitude. This prevents the use of the methods for time-dependent problems, particulary if explicit time integration schemes are employed. In this work, accuracy and eigenvalue stability of symmetric and asymmetric RBF collocation methods are numerically explored for some model hyperbolic initial boundary value problems in one and two dimensions.

keywords: Radial Basis Functions, Numerical Partial Differential Equations, Hyperbolic Partial Differential Equations, Eigenvalue Stability, Collocation Methods.

1 Introduction

Over the last several decades radial basis functions (RBFs) have been found to be successful for the interpolation of smooth functions on scattered data sets. More recently, the RBF methods have emerged as an important type of method for the numerical solution of partial differential equations (PDEs) [1]. Most PDE results have concerned elliptic PDEs for which symmetric [2] and asymmetric collocation methods [1] are well developed. Recently, the connection between the RBF collocation methods and pseudospectral methods has been described [3].

To date, RBF collocation methods for time-dependent PDE problems have not become as accepted as RBF methods for elliptic PDEs. The

Name of RBF	Definition
Multiquadric (MQ)	$\phi(r,c) = \sqrt{1+c^2r^2}$
Inverse Quadratic (IQ)	$\phi(r,c) = 1/(1+c^2r^2)$
Gaussian (GA)	$\phi(r,c) = e^{-c^2r^2}$
Inverse Multiquadric (IMQ)	$\phi(r,c) = 1/\sqrt{1+c^2r^2}$

Table 1: Global, infinitely smooth RBFs.

major obstacle has been eigenvalue stability for explicit time integration schemes. The spectrum of differential operators discretized by RBF collocation methods often contain eigenvalues with large positive real parts which makes explicit time integration schemes explosively unstable. The problem is most severe with scattered, unstructured center locations [4]. To date, little progress has been made in understanding eigenvalue stability of RBF collocation methods. Only recently in [5] were stable center locations for the asymmetric collocation method identified for the Gaussian RBF using potential theory in the specialized cases of 1d and 2d tensor product grids. In this work we seek more generalized results which apply to scattered center locations in complex domains. We focus not on identifying stable center locations, but on forming differentiation matrices with stable spectra on any set of centers.

For time dependent PDEs we take a method of lines approach where the PDE is discretized in space with RBFs and the resulting system of ordinary differential equations (ODEs) is advanced in time with an explicit ODE method. As an ODE method, we use a low dispersion, low dissipation, fourth-order, six stage explicit Runge-Kutta (LDDRK46) method [6]. LDDRK46 is commonly used to accurately advance hyperbolic PDEs over long time periods in applications such as aeroacoustics. For the time integration to be stable, LDDRK46 requires that the discretized PDE has a spectrum for which all real parts of its eigenvalues are nonpositive. We call such a spectrum a *stable spectrum*. The stable spectrum can be scaled by an appropriate time step Δt so that the spectrum fits in the stability region of the RK method. The stability region of LDDRK46 is shown in figure 3.

Our choice of RBFs is restricted to global infinitely differentiable RBFs that have a shape parameter and that are positive definite, or in the case of the MQ conditionally positive definite of order one. Some popular choices are listed in table 1. The results also apply to other conditionally positive definite RBFs if some additional notational effort is taken.

Associated with RBF methods is an *uncertainty principle* [7] that states that the methods can not be accurate and well conditioned at the same time. The conditioning of the methods is measured by the matrix condition number

$$\kappa(B) = \|B\| \left\|B^{-1}\right\| = \sigma_{\max}/\sigma_{\min} \tag{1}$$

where σ are the singular values of the matrix B. For the RBFs in table 1, small shape parameters produce the most accurate results, but are also associated with a poorly conditioned system matrix. For fixed values of the shape parameter, the conditioning of the system matrix deteriorates as the separation distance between centers, defined as

$$q_{\Xi} = \frac{1}{2} \min_{i \neq j} \left\|\xi_i - \xi_j\right\|_2 , \tag{2}$$

decreases. Although RBF methods are applicable for arbitrary, randomly scattered sets of centers, the best results are usually obtained with sets of centers with a small *fill distance*

$$h_{\Xi} = \sup_{\xi \in \Xi} \min_{\xi_j \in \Xi} \left\|\xi - \xi_j\right\| . \tag{3}$$

This is since error estimates are often stated in terms of the decay of the fill distance [8].

2 RBF Collocation Methods

Let Ξ be a finite distinct set of points $x \in \mathbb{R}^d$, which are traditionally called *centers*. Let N_I represent the number of centers, x_I, at which the PDE is enforced and N_B the number of centers, x_B, at which boundary conditions are enforced. The total number of centers is $N = N_I + N_B$. Throughout, we order the centers $\Xi = [x_I; x_B]$.

Let L be a linear differential operator and consider the time dependent PDE

$$\frac{\partial u}{\partial t} = Lu, \qquad x \in x_I \tag{4}$$

with appropriate boundary conditions applied at $x \in x_B$. Spatial derivatives can be approximated by multiplying by the *differentiation matrix* as

$$Lu \approx Du$$

where D will be defined below. The differentiation matrix may discretize a single spatial derivative or an entire differential operator.

Two RBF collocation approaches exist, asymmetric and symmetric. Both can be formulated in term of a *system matrix* H_m which discretizes the differential operator and a *evaluation matrix* B_m consisting of the functions serving as the basis of the approximation space. The m subscript refers to the particular formulation, $m = a$ for the asymmetric method and $m = s$ for the symmetric method. The methods derive their names from the fact that the symmetric method has a symmetric system matrix while the asymmetric system matrix is in general not symmetric.

2.1 Asymmetric

The asymmetric method approximates the unknown PDE solution u by

$$u(x) = \sum_{j=1}^{N} \lambda_j \phi(\|x - x_j\|_2), \quad x \in \mathbb{R}^d. \tag{5}$$

For the PDE (4) we have the approximation

$$Lu(x_i) = \sum_{j=1}^{N} \lambda_j L\phi(\|x_i - x_j\|_2), \quad i = 1, \ldots, N_I, \tag{6}$$

and where boundary conditions are enforced we have

$$u(x_i) = \sum_{j=1}^{N} \lambda_j \phi(\|x_i - x_j\|_2), \quad i = N_I + 1, \ldots, N. \tag{7}$$

From equations (5), (6), and (7) it follows that the asymmetric differentiation matrix is

$$D_a = H_a B_a^{-1} = \begin{bmatrix} L\phi \\ \phi \end{bmatrix} \begin{bmatrix} \phi \end{bmatrix}^{-1} \tag{8}$$

where the two blocks of H_a are generated as

$$\begin{aligned} (L\phi)_{i,j} &= L\phi(\|x_i - x_j\|_2), \qquad i = 1, \ldots, N_I \quad j = 1, \ldots, N \\ (\phi)_{i,j} &= \phi(\|x_i - x_j\|_2), \qquad i = N_I + 1, \ldots, N \quad j = 1, \ldots, N \end{aligned}$$

and B_a is defined by

$$(\phi)_{i,j} = \phi(\|x_i - x_j\|_2), \qquad i, j = 1, \ldots, N.$$

For all positive definite RBFs, as well as the conditionally positive definite MQ, the *evaluation matrix* B_a is known to be invertible [9]. Thus we can always form D_a.

The asymmetric method discretizes steady problems

$$Lu = f \tag{9}$$

as $u = B_a H_a^{-1} f$ where the matrix H_a is referred to as a *Kansa matrix*. Counter examples have been given to show that a Kansa matrix may not be invertible [10]. However, extensive computational evidence indicates that the matrix is very rarely singular and the asymmetric method has become well-established for steady problems.

2.2 Symmetric

The symmetric method approximates the unknown PDE solution u by

$$u(x) = \sum_{j=1}^{N-N_B} \lambda_j L^* \phi(\|x - x_j\|_2) + \sum_{N-N_B+1}^{N} \lambda_j \phi(\|x - x_j\|_2), \quad x \in \mathbb{R}^d. \tag{10}$$

The operator L^* is the operator L applied to the second argument x_j. For any x and y, $L^*\phi(\|x - y\|_2)$ and $L\,\phi(\|x - y\|_2)$ are equal in absolute value. In particular, if L is an odd order differential operator $L = -L^*$ and if L is of even order $L = L^*$.

For the PDE (4), at non-boundary points the symmetric method approximation is

$$Lu(x_i) = \sum_{j=1}^{N-N_B} \lambda_j LL^* \phi(\|x_i - x_j\|_2) + \sum_{N-N_B+1}^{N} \lambda_j L\phi(\|x_i - x_j\|_2), \quad x \in \mathbb{R}^d, \tag{11}$$

and at points where boundary conditions are enforced

$$u(x_i) = \sum_{j=1}^{N-N_B} \lambda_j L^* \phi(\|x_i - x_j\|_2) + \sum_{N-N_B+1}^{N} \lambda_j \phi(\|x_i - x_j\|_2), \quad i = N_I + 1, \ldots, N. \tag{12}$$

It follows from equations (10), (11), and (12) that the asymmetric differentiation matrix is

$$D_s = H_s B_s^{-1} = \begin{bmatrix} LL^*\phi & L\phi \\ L^*\phi & \phi \end{bmatrix} \begin{bmatrix} L^*\phi & \phi \end{bmatrix}^{-1} \tag{13}$$

RBF	$\max(\kappa_s)$	$\max(\kappa_a)^\star$
MQ	1e14	1e10
IQ	1e13	1e9
IMQ	1e13	1e9
GA	1e11	1e7

Table 2: Stable condition number bounds for the symmetric and asymmetric methods for the 1d advection example for five $N = 100$ center distributions: equidistant, CPS (16), mapped CPS (17), RBF, and random. (* For randomly located centers, the asymmetric method does not have a stable spectrum for any condition number.)

where the blocks of H_s are

$$\begin{aligned}
(LL^*\phi)_{i,j} &= LL^*\phi(\|x_i - x_j\|_2), \qquad i = 1,\ldots,N_I \quad j = 1,\ldots,N_I \\
(L\phi)_{i,j} &= L\phi(\|x_i - x_j\|_2), \qquad i = 1,\ldots,N_I \quad j = N_I+1,\ldots,N \\
(L^*\phi)_{i,j} &= L^*\phi(\|x_i - x_j\|_2), \qquad i = N_I+1,\ldots,N \quad j = 1,\ldots,N_I \\
\phi_{i,j} &= \phi(\|x_i - x_j\|_2), \qquad i = N_I+1,\ldots,N \quad j = N_I+1,\ldots,N
\end{aligned}$$

and the blocks of B_s are

$$\begin{aligned}
(L^*\phi)_{i,j} &= L^*\phi(\|x_i - x_j\|_2), \qquad i = 1,\ldots,N \quad j = 1,\ldots,N_I \\
\phi_{i,j} &= \phi(\|x_i - x_j\|_2), \qquad i = 1,\ldots,N \quad j = N_I+1,\ldots,N.
\end{aligned}$$

Unlike the asymmetric differentiation matrix, it can not be shown that the symmetric differentiation matrix D_s is well defined as the evaluation matrix B_s is a Kansa matrix. However, like the asymmetric method for steady problems, extensive computational evidence indicates that B_s is very rarely singular when boundary conditions are properly applied.

The symmetric method discretizes the steady problem (9) as $u = B_s H_s^{-1} f$. Unlike the steady asymmetric method, the symmetric method can ensure the solution of steady problems as H_s is known to be invertible [2].

3 Numerical Examples

3.1 One Dimension

Consider the advection equation

$$\frac{\partial u}{\partial t} - \frac{\partial u}{\partial x} = 0 \qquad (14)$$

on the interval $\Omega = [-1, 1]$ with boundary condition $u(1, t) = 0$. For this example, $L = \frac{\partial}{\partial x}$, $L^* = -\frac{\partial}{\partial x}$, and $LL^* = -\frac{\partial^2}{\partial x^2}$. A Matlab function that produces the MQ differentiation matrices D_a and D_s is in figure 12. In order to enforce the zero Dirichlet boundary condition at $x = 1$ the elements in the last row and last column of the differentiation matrix are set to zero.

First, the spectra of the derivative matrices D_a and D_s are examined on five sets of $N = 100$ center locations. The center locations are:

1. equidistant
$$x_i = -1 + \frac{2i}{N}, \quad i = 0, 1, \ldots, N \tag{15}$$
2. the standard Chebyshev pseudospectral (CPS) grid
$$x_i = -\cos(i\pi/N), \quad i = 0, 1, \ldots, N \tag{16}$$
3. the mapped Chebyshev pseudospectral (mapped CPS) grid
$$x_i = \frac{\arcsin\left[-\gamma \cos(i\pi/N)\right]}{\arcsin \gamma}, \quad i = 0, 1, \ldots, N \text{ and } 0 < \gamma < 1 \tag{17}$$
4. a "RBF grid" consisting of 86 equidistant interior centers and 7 mildly clustered centers, as specified by (17), near each boundary, and
5. random center locations generated by the following Matlab script:

```
x = rand(100,1);   x = sort(x);
x(1) = 0;   x(end) = 1;
x = 2*x - 1;   % [0,1] -> [-1,1].
```

The CPS grid densely clusters centers around the endpoints of Ω while the mapped CPS grid clusters points around the boundary points, but less densely. The RBF grid clusters centers around the boundaries but maintains uniform interior resolutions. This is unlike the pseudospectral grids and illustrates the flexibility of the RBF methods. Random center locations are not recommended for computation, but we have included random grids as a severe test of the methods.

It is observed that the eigenvalue stability of the methods is related to the condition number of the evaluation matrix. Bounding $\kappa(B)$ above results in D with a stable spectrum. Table 2 lists the condition number bounds for the evaluation matrix B that result in differentiation matrices with stable spectra for $N = 100$ and various RBFs on the five types of center locations. One thousand random sets of centers were tests.

As dictated by the uncertainty principle, the condition number should be kept close to the upper bound to ensure maximum accuracy. The stable condition number bound is dependent on N, the RBF, and the type of collocation method. The dependence on N is illustrated in left image of figure 1 for the MQ RBF and equidistant centers. From an implementation point of view, it would be more valuable to know the value of the shape parameter or the minimum separation distance (2) at which instability arises rather than a condition number bound. However, we have only be able to observe such a relationship in the specialized case of equally spaced centers for which the condition number bound is equivalent to holding the product $c\,q_{\Xi}$ constant.

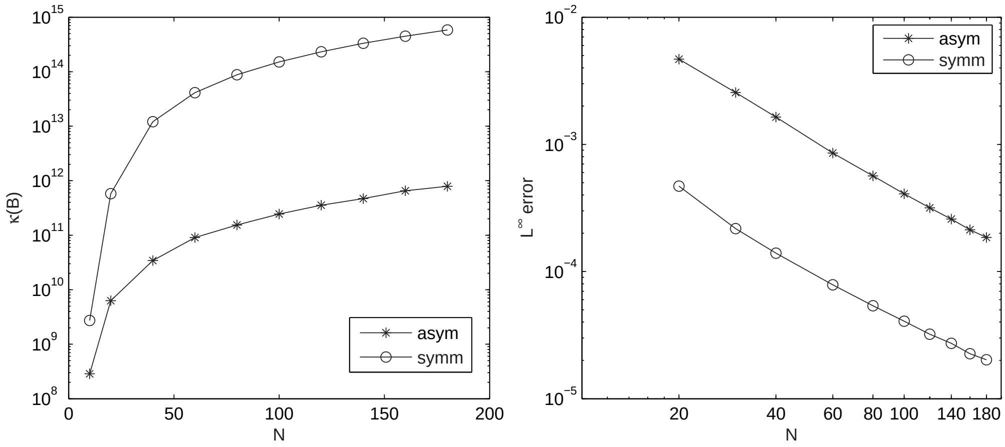

Figure 1: Left: Upper bound of the stable condition number range of the MQ B_a and B_s for equation (18) with increasing N equidistant center distributions. Right: Maximum error from equidistant center MQ approximation of equation (18) at $t = 5$ for increasing N.

Decreasing the condition number significantly below the stability bound still results in D with a stable spectrum, but the at the expense of accuracy in both methods. For a fixed set of centers, the size of the spectral radius versus the magnitude of $\kappa(B)$ differs in the two methods. As $\kappa(B_a)$ increases (c decreases) the spectral radius of D_a increases. However, for the symmetric method, as $\kappa(B_s)$ increases the spectral radius of D_s decreases. This is illustrated in figures 2 and 3 for $N = 100$ equally spaced centers. A similar result was observed for the asymmetric method for a discretized advection operator without boundary conditions in [11] where the spectral radius of D_a was shown to grow with decreasing c (which corresponds to increasing $\kappa(B_a)$).

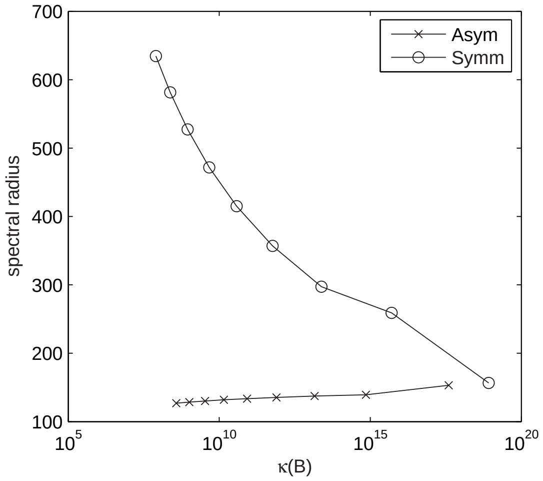

Figure 2: Spectral radius of D versus $\kappa(B)$ for the first order PDE 18.

To assess the accuracy of the methods, consider equation (18) on the interval $\Omega = [-1, 1]$. The initial condition and Dirichlet boundary condition at $x = 1$ are specified using the exact solution $u(x, t) = cos(\pi(x+t))$. In the right image of figure 1 the convergence in the max norm of the symmetric and asymmetric MQ methods is examined on equidistant centers. The condition number upper bounds from the left image of figure 1 are used. For this example, the symmetric method exhibits both better accuracy and stability properties. Figure 1 indicates an overall unimpressive algebraic convergence rate for both methods. It is well known that RBF methods converge at spectral rates [8], but this requires that the shape parameter c be held fixed while N increases. However, for this example this proves impossible if eigenvalue stability is to be maintained.

To examine the methods without the need to impose boundary conditions, consider the variable coefficient equation

$$\frac{\partial u}{\partial t} + x\frac{\partial u}{\partial x} = 0 \tag{18}$$

on the interval $[-1, 1]$. The initial condition is taken from the exact solution $u(x, t) = e^{-400(xe^{-t})^2}$. The differentiation matrix for this example is $D = \mathrm{diag}(-x)D_m$.

With $N_B = 0$, it is not difficult to find a set of centers for which B_s is singular. For example, $N = 99$ equidistant centers on $[-1, 1]$ produce a singular B_s. A condition number bound of $\kappa(B_a) \leq 5e15$ produces D on

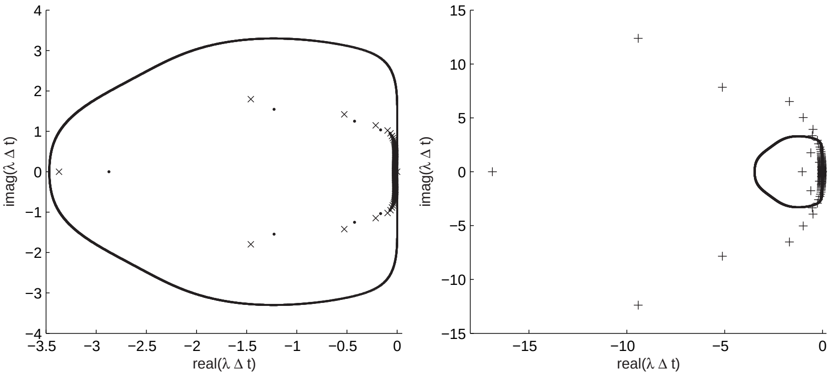

Figure 3: Stability region of LDDRK46 and spectra of the MQ D_s for equation 18 scaled by $\Delta t = 0.005$. Left: $c = 11$ and $\kappa(B_s) = 3.7e13$ (.), $c = 13$ and $\kappa(B_s) = 2.3e12$ (x). Right: $c = 100$ and $\kappa(B_s) = 2.7e6$ (+).

centers	$t = 1$	$t = 10$
equidistant	1.4e-7	2.3e-7
mapped CPS	2.3e-5	1.7e-3
RBF	1.1e-6	4.9e-4
random	6.4e-3	2.8e-1

Table 3: Max norm errors of the asymmetric MQ method for the 1d variable coefficient advection example with $N = 100$ equidistant, mapped CPS, RBF, and random centers. Time integration with LDDRK46 and $\Delta t = 0.01$.

$N = 100$ equidistant, mapped pseudospectral (17) and RBF center distributions using 86 equidistant interior centers and 7 mildly clustered centers, as specified by (17) near each boundary. As a severe test we form D on 100 random sets of centers using the same condition number range and 31 unstable spectra result. The accuracy results on the different center locations are in table 3. The most accurate results were obtained on equidistant centers.

3.2 Two Dimensions

As a 2d dimensional example consider the advection equation

$$\frac{\partial u}{\partial t} - \mu_x \frac{\partial u}{\partial x} - \mu_y \frac{\partial u}{\partial y} = 0 \tag{19}$$

	symmetric		asymmetric	
centers	$\kappa(B_s)$ bound	$er_{t=.4}$	$\kappa(B_a)$ bound	$er_{t=.4}$
uniform	1e14	2.4e-4	1e14	3.7e-5
optimal	1e14	8.6e-5	1e12	4.2e-5
random	1e14	7.4e-4	–	4.7e-2

Table 4: Stable condition number bounds and accuracy results for example for the 2d advection equation.

on the unit square $[0,1] \times [0,1]$ with $\mu_x = \mu_y = 1$. Dirichlet boundary conditions $u(1,y,t) = 0$ and $u(x,1,t) = 0$ are applied and an initial condition $u(x,y,0) = e^{-100[(x-0.7)^2+(y-0.7)^2]}$ is used. For this example, $L = \mu_x \frac{\partial}{\partial x} + \mu_y \frac{\partial}{\partial y}$, $L^* = -\mu_x \frac{\partial}{\partial x} - \mu_y \frac{\partial}{\partial y}$, and $LL^* = -\mu_x^2 \frac{\partial}{\partial x} - 2\mu_x \mu_y \frac{\partial^2}{\partial x \partial y} - \mu_y^2 \frac{\partial}{\partial y}$.

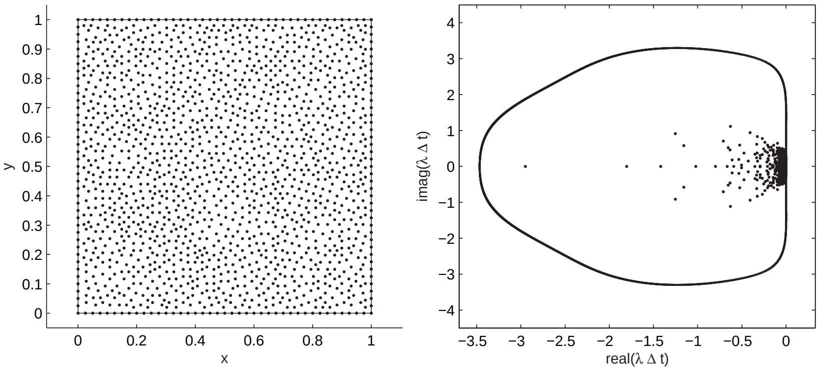

Figure 4: Left: $N = 1600$ center locations for the 2d advection example. Right: Stability region of LDDRK46 and eigenvalues of the MQ D_s scaled by $\Delta t = 0.005$.

Using the MQ RBF, the example is solved on three sets of $N = 1600$ centers: equidistant tensor product, random, and near-optimal. The near-optimal centers are located according to the geometric greedy algorithm of [12] which produces well-distributed near-optimal center sets in the sense that the center sets have small fill distances (3) and large separation distances (2). The center sets are produced by adding a new center to fill the currently largest hole in the existing center set by placing the new center in the center of the hole. The $N = 1600$ near-optimal centers are pictured in the left image of figure 4. For the random center locations we first obtain a uniform coverage of the boundary since we are interested in applying

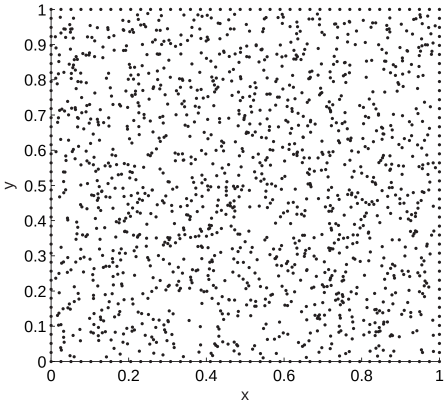

Figure 5: 116 boundary centers and 1484 randomly located interior centers.

boundary conditions and then randomly distribute centers in the interior of the domain. We have used 116 boundary centers and 1484 randomly located interior centers. An example random center distribution is in figure 5.

Listed in table 4 are the condition number bounds of the methods on the three center distributions that produce differentiation matrices with stable spectra. Additionally, the max norm errors of the methods are recorded at $t = 0.4$ before the hump has left the computational domain. Time integration was with LDDRK46 with $\Delta t = 0.005$. For the symmetric method, all center locations were stable for $\kappa(B_s) \leq 1e14$. As was the case in the 1d examples, the asymmetric method with random center locations did not have a stable spectrum.

A 2d example that does not require that boundary conditions be imposed is the variable coefficient advection equation

$$\frac{\partial u}{\partial t} - 2y\frac{\partial u}{\partial x} + 2x\frac{\partial u}{\partial y} = 0 \tag{20}$$

on the unit circle centered at the origin with initial condition

$$u(x, y) = e^{-200((x-0.3)^2+(y-0.3)^2)}.$$

The initial condition is in the right image of figure 6. The solution at time $t = N\pi$ agrees with the initial data for any integer N since the flow has then made N complete rotations.

Three sets of $N = 1600$ center locations were used. The centers are shown in figure 7. The first set is uniformly distributed, the second set

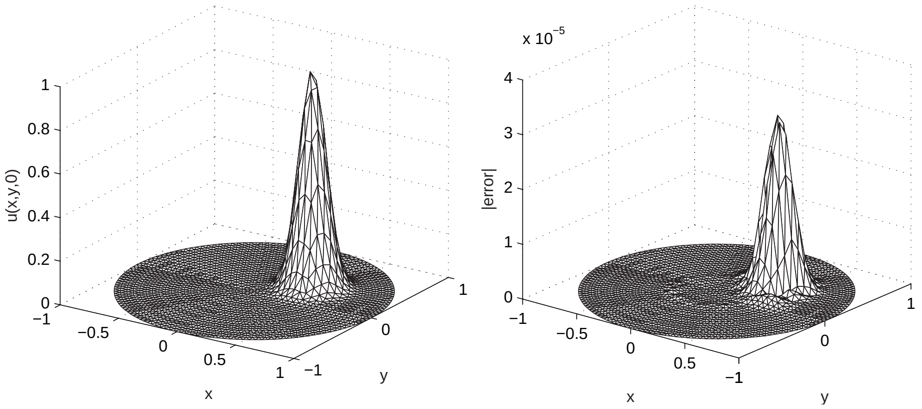

Figure 6: Left: Initial condition for the 2d variable advection example. Right: Error after 10 rotations on the uniformly distributed centers of the left image of figure 7.

center distribution	$er_{t=\pi}$	$er_{t=5\pi}$	$er_{t=10\pi}$
uniform	3.1e-6	1.6e-5	3.1e-5
near-optimal (n-o)	2.3e-5	1.3e-4	2.4e-4
n-o w/uniform boundary	8.9e-6	4.6e-5	8.8e-5

Table 5: Accuracy results for the 2d variable coefficient advection example.

is near-optimal, and the third set uniformly distributes 160 centers on the boundary and then distributes the interior centers with the near-optimal algorithm. A Matlab script to produce the near-optimal centers with uniform boundary centers is in figure 11.

With the asymmetric method, the example is discretized as

$$\mathrm{D}_a = \text{ diag } (-2\mathrm{y})\ \mathrm{DX}_a + \text{ diag } (2\mathrm{x})\ \mathrm{DY}_a \tag{21}$$

where DX_a is the asymmetric differentiation matrix with respect to x and DY_a is the asymmetric differentiation matrix with respect to y. On all three sets of centers a shape parameter of $c = 4.5$ is used which results in $\kappa(B_a) \approx 3.0e12$ and a stable spectrum. The example is advanced in time with LDDRK46 and a time step of $\Delta t = \pi/400$. The results are summarized in table 5. Even though boundary conditions are not enforced, uniformly distributing centers on the boundary results in smaller boundary region errors. As was the case in the 1d example that did not require that boundary conditions be enforced, it is easy to find center distributions for which B_s is numerically singular.

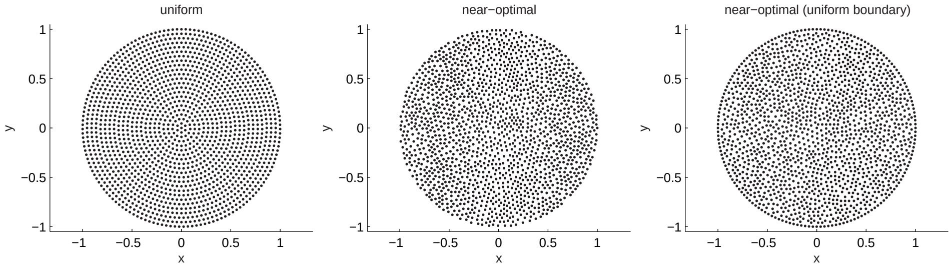

Figure 7: Center locations for the 2d variable coefficient advection example. Left: uniformly distributed. Middle: near-optimal. Right: near-optimal with uniform boundary.

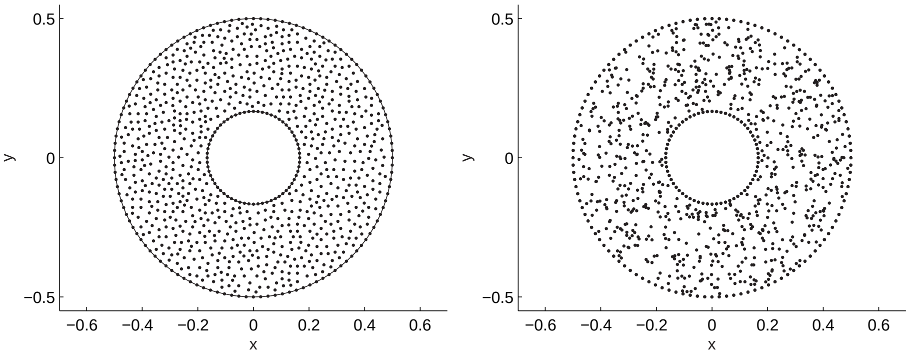

Figure 8: Left: Domain and $N = 1000$ near optimal centers for equation (23). Right: Randomly located centers.

Our final example is Maxwell's equations in two space dimensions [13] for the transverse-magnetic mode (TM) field,

$$\begin{aligned} \frac{\partial E}{\partial t} &= \frac{\partial H_y}{\partial x} - \frac{\partial H_x}{\partial y} \\ \frac{\partial H_x}{\partial t} &= -\frac{\partial E}{\partial y} \\ \frac{\partial H_x}{\partial t} &= \frac{\partial E}{\partial x} \end{aligned} \tag{22}$$

where H_x and H_y are the x and y components of the magnetic vector H and E is the electric field in the z direction. The domain consists of the space between two concentric circles $\frac{1}{6} < r = \sqrt{x^2 + y^2} < \frac{1}{2}$. The case of perfectly conducting boundaries is considered, thus a zero Dirichlet boundary condition $E = 0$ is used. From the system (22) a wave equation can be extracted for the electric field E,

$$\frac{\partial^2 E}{\partial t^2} = \frac{\partial^2 E}{\partial x^2} + \frac{\partial^2 E}{\partial y^2}. \tag{23}$$

The initial conditions are taken from the exact solution

$$E(r, \theta, t) = \cos(\omega t + \theta)\left[J_1(\omega r) + \alpha Y_1(\omega r)\right]$$

where $\theta = \arctan(y/x)$, J and Y are respectively first-order Bessel functions of the first and second kind, $\alpha = 1.76368380110927$ and $\omega = 9.813695999428405$. For equation (23) we have $L = L^* = \frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial y^2}$ and $LL^* = \frac{\partial^4}{\partial x^4} + 2\frac{\partial^4}{\partial x^2 \partial y^2} + \frac{\partial^4}{\partial y^4}$. After discretizing equation (23) in space with the RBF differentiation matrix D we have the semi-discrete equation

$$E_{tt} = DE. \tag{24}$$

A typical approach to advance equation (24) in time is to rewrite it as a first order system. Introducing new variables $u = E$ and $v = E_t$, or $U = [u\ v]^T$, equation (24) can be written as the equivalent first order system

$$\begin{bmatrix} u \\ v \end{bmatrix}_t = \begin{bmatrix} 0 & \mathrm{I} \\ D & 0 \end{bmatrix} \begin{bmatrix} u \\ v \end{bmatrix} = \overline{D}\,U. \tag{25}$$

On 100 sets of random center locations, such as shown in the right image of figure 8, both D_s and D_a have eigenvalues with negative real parts for $\kappa(B)$ as large as $1e18$. However, the eigenvalues also have small imaginary parts that grow with $\kappa(B)$. A typical spectrum is shown in the left image of figure 9. The eigenvalues of D with non-zero imaginary parts cause the matrix $\overline{D}$ to have eigenvalues with positive real parts and prevent stable time integration of the first order system by LDDRK46.

Another approach is to work directly with the second order in time PDE (23). Störmer's method for $u_{tt} = F(u)$ is

$$u^{n+1} = 2u^n - u^{n-1} + h^2 F(u^n) \tag{26}$$

where h is the size of the time step. Methods for second order equations typically require the discretized operator to have a purely real spectrum for stability. In the $h^2\lambda$ plane, the stability interval of Störmer's method is the interval $[-4, 0]$. The method has second order accuracy. The stability interval of the method can be increased to the region shown in the center image of figure 9 by applying a symmetric filter [14] to time level u^{n-1} of method (26). The filtered method remains second order accurate and can be written as the novel linear multistep method

$$u^{n+1} = \frac{1}{16}\left[30u^n - 11u^{n-1} - 4u^{n-2} + u^{n-3} + 17h^2F(u^n)\right]. \qquad (27)$$

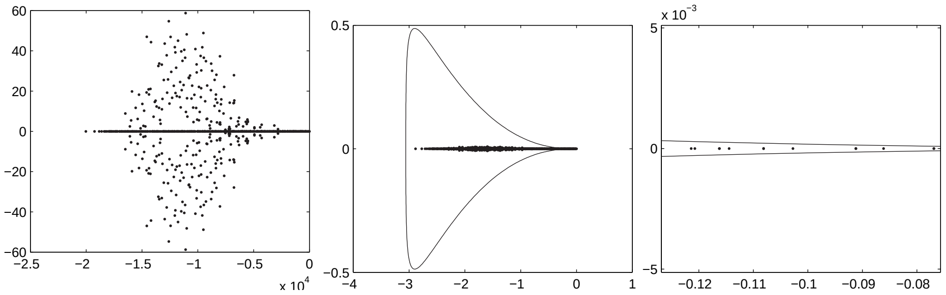

Figure 9: Left: Spectra of D_s on near-optimal centers shown in the left image of figure 8 for equation (23) with the MQ RBF and $c = 2$, $\kappa(B_s) = 6e12$. Middle: spectra scaled by h^2 and stability region of method (27). Right: middle image zoomed in near origin.

As an example we discretize equation (23) in space with the symmetric MQ method with $c = 2$ on $N = 1000$ near optimal centers. The problem is advanced in time with method (27) with $h = 0.012$. The spectrum of D_s is in the left image of figure 9 and the scaled spectrum within the stability region of method (27) is shown in the center image of figure 9. The right image of figure 9 zooms in on the region around the origin in the center image. The numerical solution at times $t = 10$ and $t = 50$ is shown in figure (10).

4 Summary

The only previous study of eigenvalue stability of RBF collocation methods for time dependent PDEs concentrated on identifying stable center locations. Only the asymmetric collocation method was considered in the specialized case of simple domains and tensor product grids. In this work, we have considered both the asymmetric and symmetric methods in the more general setting of complexly shaped domains and scattered centers for some model

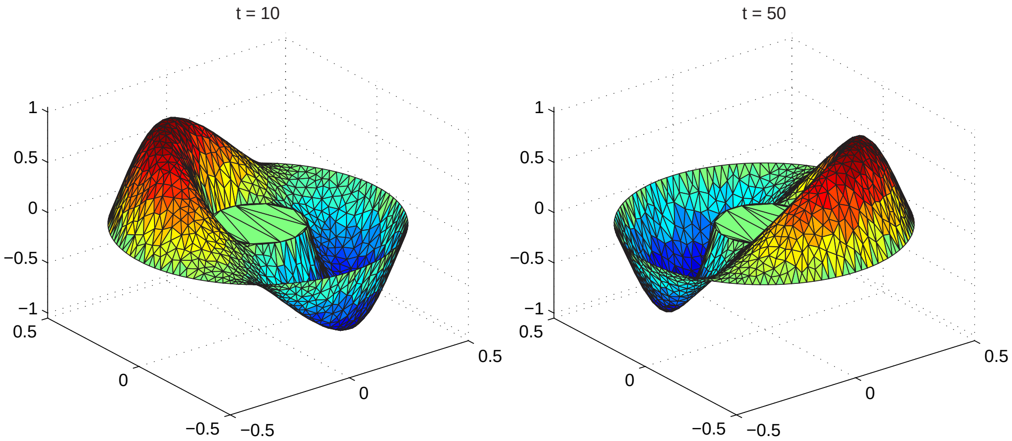

Figure 10: Numerical solution of equation (23). Left: $t = 10$. Right: t=50.

hyperbolic problems. The results of the numerical study give insight into the applicability and the stability properties of the asymmetric and symmetric methods for various types of hyperbolic problems.

The numerical results for the model problems show that eigenvalue stability with the symmetric method can be achieved for first order initial boundary values problems in the very general setting of complexly shaped domains and scattered centers by bounding the condition number of the evaluation matrix B_s. However, in the same setting, eigenvalue stability for the asymmetric method was only possible for structured center locations. In examples that did not require that boundary conditions be enforced, i.e., $N_B = 0$, the symmetric method was not applicable as center locations were easily found so that B_s was singular. In these examples, enforcing condition number bounds on B_a resulted in eigenvalue stability on center locations with structure, but not with randomly located centers.

For an initial boundary value problem with only second-order derivatives, the derivative matrices for both methods have eigenvalues with small imaginary parts which make impossible stable time stepping by standard explicit methods. However, we have presented a new method with a stability region suitable for the example problem.

For first-order problems, the spectral radius of D_a grew with $\kappa(B_a)$ while the spectral radius of D_s declined for increasing $\kappa(B_s)$. However, for second-order problems the spectral radius of D for both methods declined with increasing $\kappa(B)$, but the size of the imaginary part of the eigenvalues increased.

Neither method provided a clear advantage in accuracy, although the symmetric method did demonstrate superior accuracy on one of the model problems. It was found that good center locations for applications in complexly shaped domains are equidistant boundary centers and near-optimal interior centers.

The asymmetric differentiation matrix D_a can always be formed while it is known that certain sets of center locations exist for which the symmetric differentiation matrix D_s can not be formed. However, as in the case of the asymmetric method for steady problems, extensive computational evidence indicates that this is rarely the case in practice. Due to favorable stability properties for time dependent problems, the symmetric method should be the method of choice for first order hyperbolic initial boundary value problems.

References

[1] E. Kansa. Multiquadrics - a scattered data approximation scheme with applications to computational fluid dynamics II: Solutions to parabolic, hyperbolic, and elliptic partial differential equations. *Computers and Mathematics with Applications*, 19(8/9):147–161, 1990. 1

[2] G. E. Fasshauer. Solving partial differential equations by collocation with radial basis functions. *Proceedings of Chamonix*, pages 1–8, 1996. 1, 2.2

[3] G. E. Fasshauer. RBF collocation methods and pseudospectral methods. *Preprint. Illinois Institute of Technology*, 2004. 1

[4] B. Fornberg, T. Dirscol, G. Wright, and R. Charles. Observations on the behavior of radial basis function approximations near boundaries. *Computers and Mathematics with Applications*, (43):473–490, 2003. 1

[5] R. Platte and T. Driscoll. Eigenvalue stability of radial basis functions discretizations for time-dependent problems. *To Appear. Computers and Mathematics with Applications*, 2006. 1

[6] F. Q. Hu, M. Y. Hussaini, and J. L. Manthey. Low-dissipation and low-dispersion Runge-Kutta schemes for computational acoustics. *Journal of Computational Physics*, 124:177–191, 1996. 1

[7] R. Schaback. Error estimates and condition numbers for radial basis function interpolation. *Advances in Computational Mathematics*, 3:251–264, 1995. 1

[8] M. D. Buhmann. *Radial Basis Functions*. Cambridge University Press, 2003. 1, 3.1

[9] C. Micchelli. Interpolation of scattered data: Distance matrices and conditionally positive definite functions. *Constructive Approximation*, 2:11–22, 1986. 2.1

[10] Y. Hon and R. Schaback. On unsymmetric collocation by radial basis function. *Applied Mathematics and Computations*, 119:177–186, 2001. 2.1

[11] N. Flyer and G. Wright. Transport schemes on a sphere using radial basis functions. *Submitted to the Journal of Computational Physics*, 2006. 3.1

[12] S. De Marchi, R. Schaback, and H. Wendland. Near-optimal data-independent point locations for radial basis function interpolation. *Advances in Computational Mathematics*, pages 1–14, 2004. 3.2

[13] A. Taflove and S. Hagness. *Computational Electrodynamics: The Finite-Difference Time-Domain Method*. Artech House Publishers, second edition, 2000. 3.2

[14] S. A. Sarra and A. Aluthge. Computational mode control, expanded stability, and second order accuracy with leap-frog time differencing via symmetric filtering. *Under review, Journal Of Computational And Applied Mathematics*, 2007. 3.2

```
 function Xn = unitCircleNearOptimalCenters(nbPts,N,M,niPts)
 % INPUTS:
 %   nbPts  the number of equidistant boundary points on the circle
 %    N, M  number of points in the x (y) direction of the overlying mesh
 %   niPts  number of points added in the interior of the domain
 % OUTPUTS:
 %      Xn  near-optimal center locations, Xn(:,1) x coordinate, Xn(:,2) y
 % EXAMPLE USAGE: Xn = unitCircleNearOptimalCenters(160,200,200,1440,true);
 %     would end up with 160 + 1440 = 1600 points

% -- add all boundary points to Xn --------------------
   Xn = zeros(nbPts + niPts,2);  dt = (2*pi)/nbPts;
   theta = (0:dt:2*pi - dt)'; rho = abs(cos(theta)+i*sin(theta));
   [xc,yc] = pol2cart(theta,rho);  Xn(1:nbPts,1) = xc;  Xn(1:nbPts,2) = yc;

%--- identify possible interior center -----------------
   [X,Y] = meshgrid(linspace(-1,1,N),linspace(-1,1,M)); omegaNt = zeros(N*M,2);
   xL = reshape(X,N*M,1); yL = reshape(Y,N*M,1); [th,p] = cart2pol(xL,yL); I = 1;
     for k=1:length(th)
       t = th(k);  r = p(k);  ro = abs(cos(t)+i*sin(t));
       if (r<ro & r>0)                 % if pt is in region add to omegaN
        if (xL(k)~=0 & yL(k)~=0)
           omegaNt(I,:) = [xL(k) yL(k)]; I = I + 1;
        end
       end
    end
    omegaN = omegaNt(1:I,:);  clear omegaNt

%---- select near-optimal centers ----------------------------
    N = length(omegaN(:,1));      % number of candidates to be added to Xn
    bpi = 1:nbPts;                % index of boundary points in the  Xn(i,:)
    n = nbPts;                    % the number of points currently in Xn
    minDstToXn = zeros(N,1);   % for x in omegaN \ Xn compute the distance
    for k=1:N                  %      to the nearest point in Xn
       minDst = 100;
       for j=1:n
         dst = norm(omegaN(k,:)-Xn(j,:),2);
         if     dst<eps, minDst = 0;     % => omegaN(k,:) already is in Xn
         elseif dst<minDst, minDst = dst; end
       end
       minDstToXn(k) = minDst; % smallest distance of omegaN(k,:) to any Xn
     end
    for j = 1:niPts         % find the omegaN(k) in the laregest hole in Xn
          n = n + 1;
          [notUsed,indexAddToXn] = max(minDstToXn);
          Xn(n,:) = omegaN(indexAddToXn,:);  minDstToXn(indexAddToXn) = 0;
          for k=1:N                       % update distances for added point
            if minDstToXn(k) >= eps
               dst = norm(omegaN(k,:)-Xn(n,:),2);
               if dst<minDstToXn(k), minDstToXn(k) = dst; end
            end
          end
    end
```

Figure 11: Matlab function for finding near-optimal center locations on the unit circle with uniform boundary spacing.

```
function dm = adv1d_dms_mq(dmCh,shape,x)
   % produces differentiation matrix for
   %   Ut - Ux = 0   on [-1,1] with U(1,t) = 0
   % INPUTS:
   % dmCh = 0 symmetric, 1 asymmetric
   %        x   centers, for example x = linspace(-1,1,100);
   %    shape   shape parameter

    N = length(x); NB = 1;  % boundary points
    NI = N - NB;            % interior points
    H = zeros(N,N);  Hx = zeros(N,N); rx = zeros(N,N); r = zeros(N,N);
    for i=1:N, for j=1:N
      rx(i,j) = x(i) - x(j);   r(i,j) = abs( rx(i,j) );
    end, end

    if dmCh==0     % symmetric
         L = dPhi( r(1:NI,NI+1:N),rx(1:NI,NI+1:N),shape);     % block 2 of H
         LS = L';                                             % block 3 of H
         Hx(  1:NI,1:NI  )   = -dPhi2( r(1:NI,1:NI),rx(1:NI,1:NI),shape);  % block 1 of H
         Hx(  1:NI,NI+1:N)   = L;        % block 2
         Hx(NI+1:N,1:NI  )   = LS;       % block 3
         Hx(NI+1:N,NI+1:N  ) = phi(r(NI+1:N,NI+1:N),shape);    % block 4 of H

         H(  1:N,1:NI  ) =  -dPhi(r(1:N,1:NI),rx(1:N,1:NI),shape);  % block 1 of B
         H(1:N,NI+1:N  ) =   phi(r(1:N,NI+1:N),shape);
    elseif dmCh==1     % asymmetric
        Hx(1:N-1,:) = dPhi(r(1:N-1,:),rx(1:N-1,:),shape);    % interior points
        Hx(N,:) = phi(r(N,:),shape);                         % Dirichlet BC at x=1
        H = phi(r,shape);
    end

    cond(H)
    dm = Hx/H;  dm(end,:)=0;  dm(:,end)=0;  % zero Dirichlet BCs at x=1

    function mq = phi(r,c), mq = sqrt(1 + (r.*c).^2); end
    function mqD = dPhi(r,rx,c), mqD  = (rx.*c.^2)./sqrt(1 + (r.*c).^2); end
    function mqD2 = dPhi2(r,rx,c), mqD2  = (c.^2)./sqrt(1 + (r.*c).^2).^3; end
   end
```

Figure 12: Matlab function for calculating the MQ symmetric and asymmetric differentiation matrices for the 1d advection example.