Eigenvalue stability of radial basis function discretizations for time-dependent problems*

Rodrigo B. Platte† and Tobin A. Driscoll†

Abstract

Differentiation matrices obtained with infinitely smooth radial basis function (RBF) collocation methods have, under many conditions, eigenvalues with positive real part, preventing the use of such methods for time-dependent problems. We explore this difficulty at theoretical and practical levels. Theoretically, we prove that differentiation matrices for conditionally positive definite RBFs are stable for periodic domains. We also show that for Gaussian RBFs, special node distributions can achieve stability in 1-D and tensor-product nonperiodic domains. As a more practical approach for bounded domains, we consider differentiation matrices based on least-squares RBF approximations and show that such schemes can lead to stable methods on less regular nodes. By separating centers and nodes, least-squares techniques open the possibility of the separation of accuracy and stability characteristics.

Keywords. radial basis functions, RBF, method of lines, numerical stability, least squares

1 Introduction

RBFs are increasingly being used in the numerical solution of partial differential equations [1, 2, 3, 4, 5], and are a viable alternative to more traditional methods, such as finite differences, finite elements, and spectral methods. RBF-based methods have several attractive features, most notably fast convergence (exponential for some cases) and the flexibility in the choice of node location. In the presence of rounding errors, however, it is often difficult to obtain highly accurate results – see, e.g. [5, 6, 7]. For time-dependent problems, in particular, differentiation matrices often have unstable eigenvalues requiring severe dissipation in time. In this article we are concerned with finding effective ways to solve time-dependent problems using RBFs.

Given a set of *centers* $x_0^c, \ldots, x_N^c$ in $\mathcal{R}^d$, an RBF approximation takes the form

$$F(x) = \sum_{k=0}^{N} \lambda_k \, \phi\big(\|x - x_k^c\|\big), \tag{1}$$

where $\|\cdot\|$ denotes the Euclidean distance between two points and $\phi(r)$ is a function defined for $r \geq 0$. The coefficients $\lambda_1, \ldots, \lambda_N$ may be chosen by interpolation or other conditions at a set of *nodes* that typically coincide with the centers. In this article, however, we may allow node and center locations to differ. Common choices for ϕ fall into two main categories: infinitely smooth and containing a free parameter, such as multiquadrics ($\phi(r) = \sqrt{r^2 + c^2}$), inverse quadratics ($1/(c^2 + r^2)$) and Gaussians ($\phi(r) = e^{-(r/c)^2}$); and piecewise smooth and parameter-free, such as cubics ($\phi(r) = r^3$) and thin plate splines ($\phi(r) = r^2 \ln r$).

Although several authors have investigated RBF-methods for time-dependent problems [2, 3, 8], these methods remain underdeveloped compared to those for elliptic problems. In this article we are particularly interested in using the method of lines. In periodic regions, like the unit circle and the unit sphere, we prove in section 2.2 that RBF methods are time-stable for all conditionally positive

*Supported by NSF DMS-0104229.

†Department of Mathematical Sciences, University of Delaware, Newark, DE, 19716. E-mail: platte@math.udel.edu and driscoll@math.udel.edu.

definite RBFs and node distributions. However, in nonperiodic domains experience suggests that RBFs will produce discretizations that are unstable in time unless highly dissipative time stepping is used.

In [9] we exploited a connection between Gaussians RBFs (GRBFs) in 1-D and polynomials. Using standard tools of potential theory, we found that GRBFs are susceptible to a Runge phenomenon. Moreover, we found that the use of GRBFs with arbitrary nodes may lead to very large Lebesgue constants, making it difficult to obtain very accurate approximations. Using potential theory, however, one can obtain stable nodes that prevent the Runge phenomenon and allow stable approximations. One way to stabilize RBF-approximations for time-dependent problems is to use these special nodes to generate differentiation matrices, as we show in section 3.

Stable nodes, however, are not known for general regions in high dimensions and are not suitable for adaptive resolution. A viable alternative for stabilizing RBFs in time-dependent problems is the use of least-squares techniques. In section 4 we explore a discrete least-squares method that has the simplicity of collocation for nonlinearities and the like, yet allows stable explicit time integration. Section 5 contains our final remarks.

2 RBFs and the method of lines

The method of lines refers to the idea of semidiscretizing in space and using standard methods for the resulting system of ordinary differential equations in time. A rule of thumb is that the method of lines is stable if the eigenvalues of the spatial discretized operator, scaled by the time-step Δt, lie in the stability region of the time-discretization operator, although in some cases the details of stability are more technical and restrictive [10].

2.1 Unstable eigenvalues: A case study

Consider as a test problem the transport equation,

$$u_t = u_x, \quad -1 \le x \le 1, \ t > 0 \tag{2}$$

$$u(t,1) = 0, \quad u(0,x) = u_0(x). \tag{3}$$

A differentiation matrix for this problem can be easily obtained by noting that $\mathbf{u} = A\boldsymbol{\lambda}$ and $\mathbf{u_x} = B\boldsymbol{\lambda}$, where A and B are matrices with elements $A_{i,j} = \phi(\|x_i - x_j^c\|)$ and $B_{i,j} = \dfrac{d}{dx}\phi(\|x - x_j^c\|)\Big|_{x=x_i}$, x_j are $N+1$ collocation nodes, $\mathbf{u}$ and $\mathbf{u_x}$ are vectors containing the RBF approximations of the function u and u_x at the collocation nodes, and $\boldsymbol{\lambda}$ is the vector of the coefficients λ_j. The differentiation matrix is then given by $\tilde{D} = BA^{-1}$. In order to enforce the boundary condition, assuming that $x_N = 1$, we delete the last row and column of $\tilde{D}$ to produce a matrix we now call D. This leads to the coupled system of ordinary differential equations

$$\mathbf{u}_t = D\mathbf{u}. \tag{4}$$

The difficulty of using the method of lines with RBFs for (2)-(3) with arbitrary nodes is that some eigenvalues of the differentiation matrix may have positive real parts. This is illustrated in Figure 1. This figure was obtained with GRBFs using coincident, equally spaced centers and nodes in $[-1,1]$. In Figure 1(a) the shape parameter is fixed, $c = 1$. Notice that for $N = 5$ all eigenvalues have negative real part, but as N is increased eigenvalues move to the right half-plane making it difficult to use explicit finite-difference methods for time integration. Similarly, in Figure 1(b) we observe that, for fixed N, eigenvalues move to the right half-plane as c is increased. It is well known that the limit $c \to \infty$ is equivalent to polynomial interpolation so that the Runge phenomenon causes instability. While experiments indicate that for a given N, the shape parameter c can be chosen small enough that all eigenvalues will lie in the left half-plane, this requirement is rather restrictive for large values of N – to the extent that spectral convergence seems to be compromised.

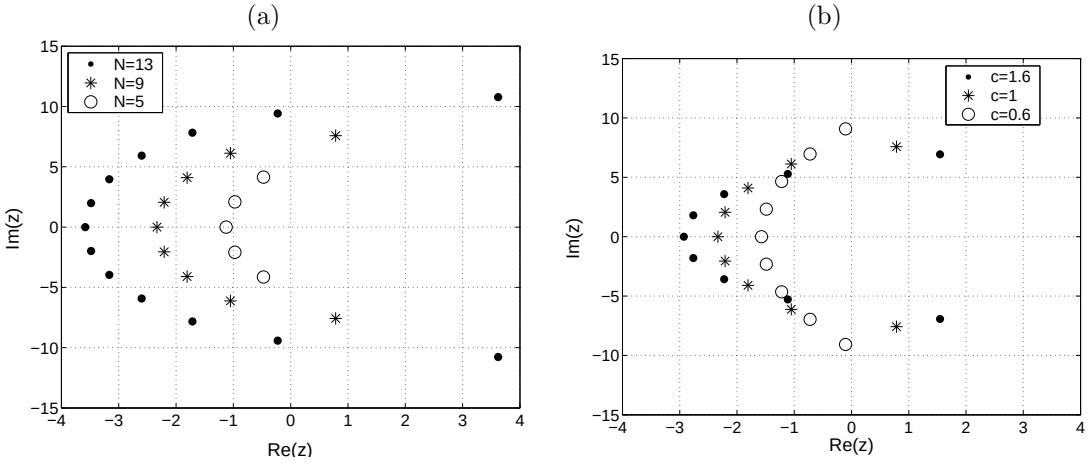

Figure 1: Eigenvalues of D for GRBFs with equally spaced nodes in $[-1, 1]$: (a) $c = 1$; (b) $N = 9$.

2.2 Spectra in the absence of boundaries

In polynomial approximation, boundaries play a major role in stability. Similar observations have been made experimentally in RBF approximation [11]. There is reason to think, then, that in the absence of boundaries (e.g. the differentiation matrix $\tilde{D}$), eigenvalue stability is possible. In this section we show that this is indeed the case.

We shall need the concept of conditionally positive definite functions. A radial function $\phi : \mathcal{R} \to \mathcal{C}$ is called conditionally positive definite of order m if for any set of distinct nodes $x_0, x_1, ..., x_N$, and for all $\lambda \in \mathcal{C}^{N+1} \setminus \{0\}$ satisfying

$$\sum_{j=0}^{N} \lambda_j p(x_j) = 0, \tag{5}$$

for all polynomials p of degree less than m, the quadratic form

$$\sum_{i=0}^{N} \sum_{j=0}^{N} \bar{\lambda_i} \lambda_j \phi(\|x_i - x_j\|) \tag{6}$$

is positive [12]. In this case, it is common to augment expansion (1) with a polynomial of degree at most $m - 1$ in order to impose (5). For the rest of this section we shall assume that interpolation nodes and centers coincide, i.e., $x_j = x_j^c$. The augmented RBF expansion at these nodes takes the form,

$$F(x_i) = \sum_{j=0}^{N} \lambda_j \phi(\|x_i - x_j\|) + \sum_{k=0}^{m-1} \alpha_k p_k(x_i), \tag{7}$$

where $\{p_0, p_1, ..., p_{m-1}\}$ is a basis for the space of d-variate polynomials of degree at most $m-1$. At this point we are interested in the spectrum of the finite-dimensional RBF operators that represent a differential operator with constant coefficients $\mathcal{L}$, like the Laplacian or a convection operator.

We can write (5) and (7) in matrix form,

$$\begin{bmatrix} A & P \\ P^T & 0 \end{bmatrix} \begin{bmatrix} \boldsymbol{\lambda} \\ \boldsymbol{\alpha} \end{bmatrix} = \begin{bmatrix} \mathbf{F} \\ 0 \end{bmatrix}, \tag{8}$$

where the elements of A are $A_{i,j} = \phi(\|x_i - x_j\|)$, the elements of P are $P_{i,j} = p_j(x_i)$, and $\mathbf{F}_i = F(x_i)$. An RBF discretization of the operator $\mathcal{L}$ can then be written as

$$L = \begin{bmatrix} A^L & P^L \end{bmatrix} \begin{bmatrix} A & P \\ P^T & 0 \end{bmatrix}^{-1} \begin{bmatrix} I_{N+1} \\ 0 \end{bmatrix}, \tag{9}$$

where $A^L_{i,j} = \mathcal{L}\phi(\|x - x_j\|)|_{x=x_i}$, $P^L_{i,j} = \mathcal{L}p_j(x)|_{x=x_i}$, and I_{N+1} is the identity matrix of order $N+1$. We shall next show that the eigenvalues of L are purely imaginary if A^L is antisymmetric, and real if A^L is symmetric. We point out that for positive definite RBFs, the restriction that $\mathcal{L}$ must have constant coefficients can be dropped, and only linearity is needed.

Suppose that ν is an eigenvalue of L with eigenvector $\mathbf{u}$. Then we have that

$$L\mathbf{u} = \nu\mathbf{u} \iff A^L\boldsymbol{\lambda} + P^L\alpha = \nu(A\boldsymbol{\lambda} + P\boldsymbol{\alpha}) \text{ and } P^T\boldsymbol{\lambda} = 0.$$

Notice that $\boldsymbol{\lambda}^* P = \boldsymbol{\lambda}^* P^L = 0$ and (6) gives $\boldsymbol{\lambda}^* A\boldsymbol{\lambda} > 0$, where $*$ denotes the complex conjugate transpose. Therefore, we obtain

$$\nu = \frac{\boldsymbol{\lambda}^* A^L \boldsymbol{\lambda}}{\boldsymbol{\lambda}^* A\boldsymbol{\lambda}},$$

and since A is symmetric,

$$\bar{\nu} = \frac{\boldsymbol{\lambda}^* (A^L)^T \boldsymbol{\lambda}}{\boldsymbol{\lambda}^* A\boldsymbol{\lambda}}.$$

Thus if A^L is symmetric, we have that $\nu = \bar{\nu}$, and if it is antisymmetric, $\nu = -\bar{\nu}$.

Gaussians and inverse quadratics are positive definite RBFs and multiquadrics are conditionally positive definite of order 1 [12]. Moreover, the matrix of elements $\left.\frac{d}{dx}\phi(\|x - x_j\|)\right|_{x=x_i}$ is antisymmetric. Hence, $\tilde{D}$ has only imaginary eigenvalues.

For conditionally positive definite RBFs, therefore, deviations of the spectrum from the imaginary axis occur when boundary conditions are enforced in $\tilde{D}$ to generate D. RBFs methods for differential equations on the unit circle or unit sphere, on the other hand, are boundary-condition free, making RBF methods suitable for time-dependent problems on such regions.

Differential equations with periodic boundary conditions on a interval of the real line can be naturally mapped to a boundary condition free problem on the unit circle. For instance, solving $u_t = u_x$ with periodic boundary conditions in $[0, 2\pi]$ is equivalent to solving $u_t = u_\theta$ on the unit circle, where θ is the polar angle. Considering the norm $\|x_i - x_j\| = \sqrt{2 - 2\cos(\theta_i - \theta_j)}$, one can easily show, in light of the observations above, that the RBF differentiation matrix in this case has only imaginary eigenvalues

Similarly, problems on the unit sphere are boundary-free. As an example, consider the convective test problem presented in [13],

$$u_t + (\cos\alpha - \tan\theta\sin\varphi\sin\alpha)u_\varphi - (\cos\varphi\sin\alpha)u_\theta = 0, \tag{10}$$

where α is constant and the spherical coordinates are defined by $x = \cos\theta\cos\varphi$, $y = \cos\theta\sin\varphi$, and $z = \sin\varphi$. We again consider the Euclidean metric, so for nodes x_j on the sphere, $\|x_i - x_j\|^2 = 2 - 2(\cos\theta_i\cos\theta_j\cos(\varphi_i - \varphi_j) + \sin\theta_i\sin\theta_j)$. To demonstrate that explicit time integrators with a positive definite RBF can be stably used for this problem, all is needed is to show that A^L for the operator $\mathcal{L}(\cdot) = -(\cos\alpha - \tan\theta\sin\varphi\sin\alpha)\partial_\varphi(\cdot) + (\cos\varphi\sin\alpha)\partial_\theta(\cdot)$ is antisymmetric. Notice since this is not a constant coefficient operator, so imaginary spectrum can only be proved for positive definite RBFs. Straightforward calculations show that

$$A^L_{i,j} = -2\left.\frac{d\phi(r)}{dr^2}\right|_{r^2=\|x_i-x_j\|^2} (\cos\alpha\cos\theta_i\cos\theta_j\sin(\varphi_i - \varphi_j) + \sin\alpha(\cos\theta_i\sin\theta_j\cos\varphi_i - \cos\theta_j\sin\theta_i\cos\varphi_j)),$$

which is indeed antisymmetric.

It is worth pointing out that the spectrum of $\tilde{D}$ in such conditions is sensitive to perturbations on the nodes, i.e., small deviations of the set of nodes from the set of centers is likely to generate unstable eigenvalues. In Figure 2 the spectra of two differentiation matrices are shown for the periodic transport problem on the unit circle. Both matrices were obtained with equally spaced centers and nodes, GRBFs, $N = 13$, and $c = 1$. In the first case we considered centers $\theta^c_j = 2\pi j/14$, $j = 0, ..., 13$, and nodes $\theta_j = \theta^c_j$. Imaginary spectrum is guaranteed in this case and the numerical results agree with this prediction. In the second case we used $\theta_j = \theta^c_j + 0.01$. The matrix obtained in this instance has real eigenvalues as shown in the figure.

In the remainder of this article, we shall explore ways to stabilize RBF methods for time-dependent problems with boundary conditions. We shall next study how the interpolation nodes can be used to avoid unstable eigenvalues.

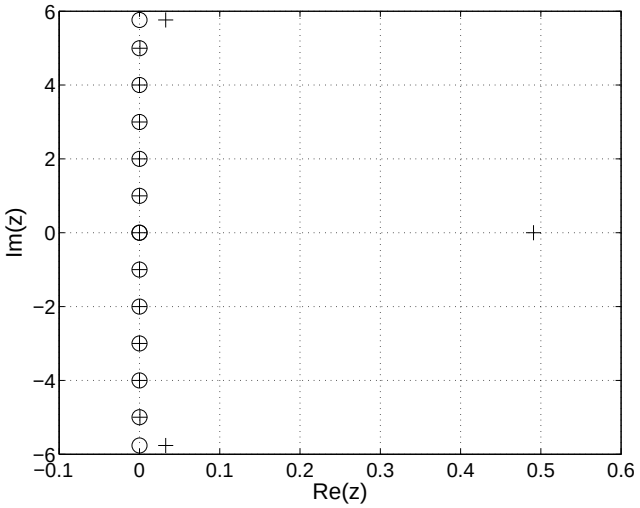

Figure 2: Eigenvalues of $\tilde{D}$ on the unit circle with equally spaced centers and nodes: set of centers and nodes coincide ($\circ$); nodes are shifted 0.01 units from centers ($+$).

3 Gaussian RBFs as polynomials

For GRBFs with equally spaced centers in $[-1,1]$, i.e., $x_k^c = -1 + 2k/N = -1 + kh$, $k = 0, \ldots, N$, we have that (1) becomes

$$F(x) = \sum_{k=0}^{N} \lambda_k e^{-(x+1-kh)^2/c^2} = e^{-(x+1)^2/c^2} \sum_{k=0}^{N} \lambda_k e^{(2kh-k^2h^2)/c^2} e^{2kxh/c^2}. \tag{11}$$

Following [9], we make the definition $\beta = 2h/c^2 = 4/(Nc^2)$ and use the transformation

$$s = e^{\beta x}, \qquad s \in [e^{-\beta}, e^{\beta}],$$

to find that

$$G(s) = F(\log(s)/\beta) = e^{-\frac{N}{4\beta}(\log s+\beta)^2} \sum_{k=0}^{N} \tilde{\lambda}_k s^k = \psi_\beta^N(s) \sum_{k=0}^{N} \tilde{\lambda}_k s^k, \tag{12}$$

where the $\tilde{\lambda}_k$ are independent of s. Throughout this section we assume that β is a fixed parameter.

Using the fact that G/ψ_β^N is a polynomial, in [9] we presented necessary conditions for uniform convergence of the GRBF interpolation process. Specifically, if we let μ be the limiting node density function [14] of nodes on $[-1,1]$ and define

$$u_\beta(z) = \frac{\beta}{4} \operatorname{Re}\left[(z+1)^2\right] - \int_{-1}^{1} \log(|e^{\beta z} - e^{\beta t}|)\mu(t)dt, \tag{13}$$

then the GRBF interpolant converges exponentially to the target function, provided that this function has an analytic extension in the largest region of the complex plane that includes all level curves of u_β that cross the interval $[-1,1]$. On the other hand, if the function being interpolated does not satisfy this requirement, approximations lead to spurious oscillations whose amplitude grows exponentially with N. This is analogous to the Runge phenomenon in polynomial interpolation.

In [9] we also showed that one can find node distributions for which GRBF interpolation converges whenever the target function is analytic on $[-1,1]$. In Figure 3 we present plots of these density functions for several values of β. Notice that for large β, the density functions approximately constant except near the ends of the interval. For small β, on the other hand,

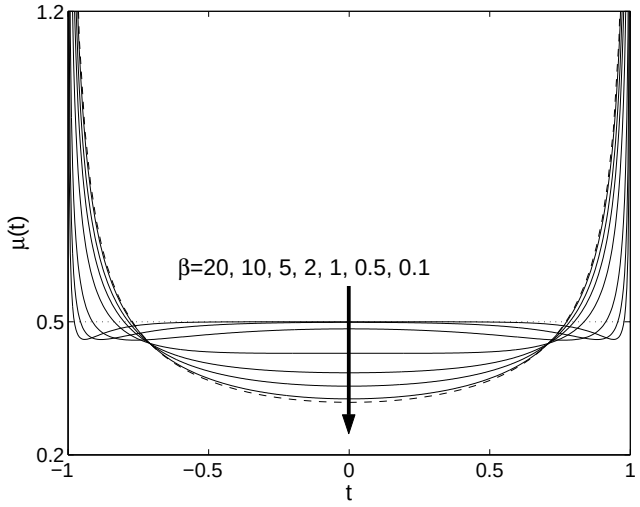

Figure 3: Numerical approximations of the optimal density functions for several values of β. Dashed line shows the Chebyshev density function.

density functions are close to the Chebyshev density function. In Figure 4 we show 21 nodes generated with such density functions. For large β the nodes are nearly equally spaced and for small β they are similar to Chebyshev extreme points [15].

As in polynomial interpolation, although convergence may be guaranteed for sufficiently smooth functions for a given set of interpolation nodes, approximations may not converge in the presence of rounding errors due to the rapid growth of the Lebesgue constant [16]. For GRBFs, we found that under most conditions these constants grow exponentially with N. If nodes are obtained with optimal density functions, however, the growth of these constants seems to be logarithmic [9]. In this article we shall see that these optimal nodes also lead to stable approximations for time dependent problems.

3.1 GRBF differentiation matrices

Although the differentiation matrix for RBFs can be generated using the guidelines presented in section 2, for GRBFs in 1-D with equally spaced nodes, we can derive an explicit formula for the entries of the differentiation matrix. This approach circumvents the difficulty of inverting the usually ill-conditioned interpolation matrix A.

Using the fact the GRBFs with equally spaced centers are polynomials in a transformed variables, we can find the entries of the differentiation matrix using the aid of Lagrange interpolation. In [17] Berrut and Trefethen argue that the barycentric form of the Lagrange interpolation should be the method of choice for polynomial interpolation. In order to differentiate GRBF interpolants, consider the barycentric formula for the GRBF interpolant presented in [9],

$$F(x) = v(x)\frac{\displaystyle\sum_{k=0}^{N}\frac{w_k}{\left(e^{\beta x}-e^{\beta x_k}\right)}f(x_k)}{\displaystyle\sum_{k=0}^{N}\frac{w_k}{\left(e^{\beta x}-e^{\beta x_k}\right)}v(x_k)}. \tag{14}$$

where the w_k are the barycentric weights defined by

$$w_k = \left(\prod_{\substack{j=0\\ j\neq k}}^{N} e^{-\frac{\beta}{4}(x_k+1)^2}\left(e^{\beta x_k}-e^{\beta x_j}\right)\right)^{-1} \tag{15}$$

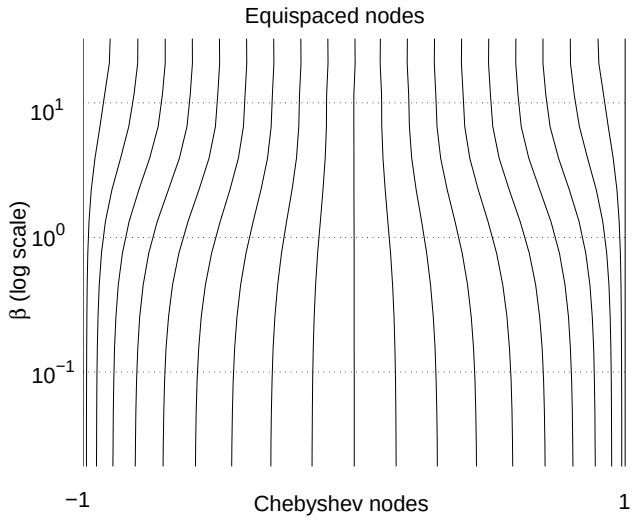

Figure 4: Node locations obtained using density function computed by solving an integral equation for $N = 20$ and several values of β.

and

$$v(x) = \frac{1}{N} \sum_{k=0}^{N} e^{-\frac{N\beta}{4}(x-x_k^c)^2}.$$

From (14) the following expression for the GRBF cardinal function can be derived:

$$L_j(x) = \frac{w_j v(x)}{e^{\beta x} - e^{\beta x_j}} \Bigg/ \sum_{k=0}^{N} \frac{w_k v(x_k)}{e^{\beta x} - e^{\beta x_k}}, \quad x \neq x_j,\ L_j(x_j) = 1. \tag{16}$$

We can rewrite (16) as

$$L_j(x) = \frac{w_j v(x)}{s_i(x)} \left(\frac{e^{\beta x} - e^{\beta x_i}}{e^{\beta x} - e^{\beta x_k}} \right), \tag{17}$$

where $s_i(x) = \sum_{k=0}^{N} w_k v(x_k)(e^{\beta x} - e^{\beta x_i})/(e^{\beta x} - e^{\beta x_k})$. Multiplying both sides of (17) by s_i and differentiating the resulting equation gives

$$L_j'(x)s_i(x) + L_j(x)s_i'(x) = w_j \left(v'(x) \left(\frac{e^{\beta x} - e^{\beta x_i}}{e^{\beta x} - e^{\beta x_k}} \right) + v(x) \left[\frac{e^{\beta x} - e^{\beta x_i}}{e^{\beta x} - e^{\beta x_k}} \right]' \right).$$

Since $s_i(x_i) = w_i$ and $L_j(x_i) = 0$ for $j \neq i$, we have

$$L_j'(x_i) = \frac{\beta v(x_i) e^{\beta x_i} w_j}{(e^{\beta x_i} - e^{\beta x_j}) w_i}. \tag{18}$$

In order to derive an expression for $L_j'(x_j)$, notice that $v(x) = \sum_{k=0}^{N} L_k(x) v(x_k)$. Differentiating v and solving the resulting equation for $L_j'(x_j)$ gives

$$L_j'(x_j) = \frac{v'(x_j)}{v(x_j)} - \sum_{\substack{k=0 \\ k \neq j}}^{N} L_k'(x_j) \frac{v(x_k)}{v(x_j)}. \tag{19}$$

Therefore, the entries of the first-order differentiation matrix are $\tilde{D}_{i,j} = L_j'(x_i)$. This formulation for $\tilde{D}$ is both far more robust than the one presented in the previous section and computationally more efficient. We point out, however, that in some cases it is necessary to rescale (15) to avoid overflow [9].

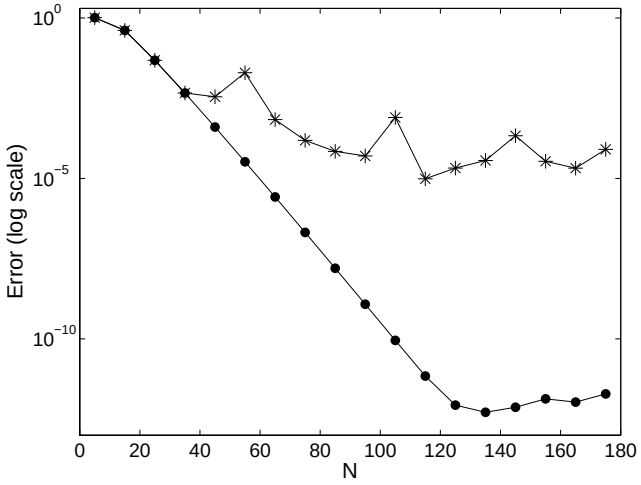

Figure 5: Error in the approximation of df/dx, where $f(x) = 1/(1 + 25x^2)$, using the barycentric formulation for D ($\bullet$) and the standard RBF algorithm ($*$).

In Figure 5 we present the error of the approximation of df/dx from values of $f(x) = 1/(1+25x^2)$ at nodes in $[-1, 1]$. We used $\beta = 2$ and the nodes were generated with approximate optimal density functions. We observe that the use of both stable nodes and the stable formulation for the differentiation matrix permits convergence to machine precision, while the use of differentiation matrices obtained with the standard algorithm presented in section 2 gives very poor results for large N. We point out that the need for stable nodes is due not only to the Runge phenomenon, but also to avoid the rapid growth of the Lebesgue constants. In the presence of rounding errors, if we were to approximate $\sin(\pi x)$ instead of f in the approximation problem above using equispaced nodes, spectral convergence would be lost for $N > 40$ for the standard algorithm and for $N > 65$ for the barycentric algorithm with an minimum error of about 10^{-6}, even though convergence would be guaranteed for this function by Theorem 2.2 in [9].

To illustrate how optimal nodes stabilize RBF approximations of time-dependent problems, in Figure 6 we show the behavior of the eigenvalues of the RBF differential matrix D with incorporated boundary conditions. The matrix D was derived with the barycentric algorithm and $\beta = 2$. Now all eigenvalues lie in left-half plane, so standard explicit time integration techniques can be used together with this spatial approximation.

The spectral radius of D is shown in Figure 7 as a function of N for β =0.1, 1, and 10. Notice that for large N, the spectral radius grows as $O(N^2)$. We observe that for $\beta = 10$, the growth was $O(N)$ for several values of N, but for large N the effects of the rapid growth of the density function near the boundary (see Figure 3) forces clustering of the nodes. The spacing between nodes for large N near the ends of the interval is approximately $O(1/N^2)$. The restriction on time-step sizes for GRBFs on stable nodes, is therefore, similar to the one for polynomial approximation.

The results in this section extend immediately to tensor-product regions of uniform center locations in higher dimensions. Although this type of region is usually of little interest to RBF users, they help us to illustrate the fact that the location of collocation nodes can be used to stabilize RBFs and improve accuracy. Figure 8 presents results for the convective test problem, $u_t = u_x + u_y$, for $(x, y) \in [-1, 1]\times[-1, 1]$ and $t > 0$, with initial condition $u(0, x, y) = \exp(-20((x-0.2)^2+(y-0.2)^2))$ and boundary conditions $u(t, 1, y) = u(t, x, 1) = 0$. We used GRBFs with $\beta = 2$, 27 nodes, and equally spaced centers. We can generate stable nodes in this square by taking the tensor product of stable nodes in $[-1, 1]$ (Figure 8(a)). In Figure 8(b) we show rescaled eigenvalues of the RBF convection matrix when $\Delta t = 0.04$. Notice that for stable nodes, they lie inside the fourth-order Runge-Kutta region of stability; for equispaced nodes, however, some lie outside. Figure 8(d) shows the computed solution with clustered nodes at $t = 0.7$ using fourth order Runge-Kutta.

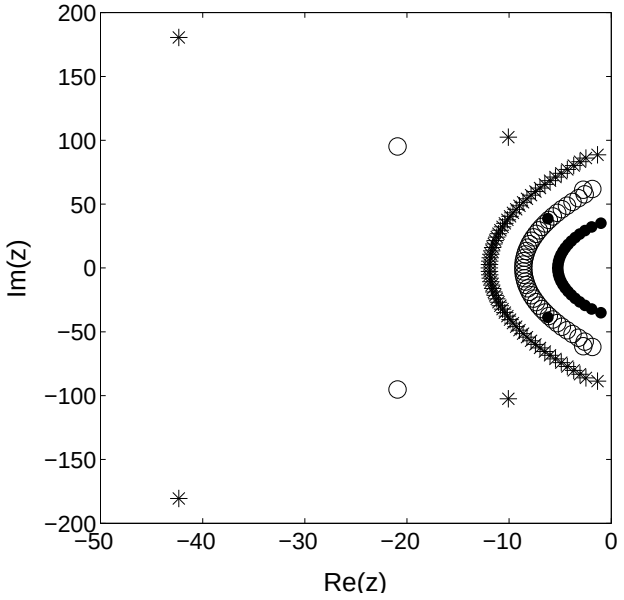

Figure 6: Eigenvalues of the GRBF differentiation matrix for $\beta = 2$ and stable nodes: N=30 ($\bullet$); N=50 ($\circ$); and N=70 ($*$).

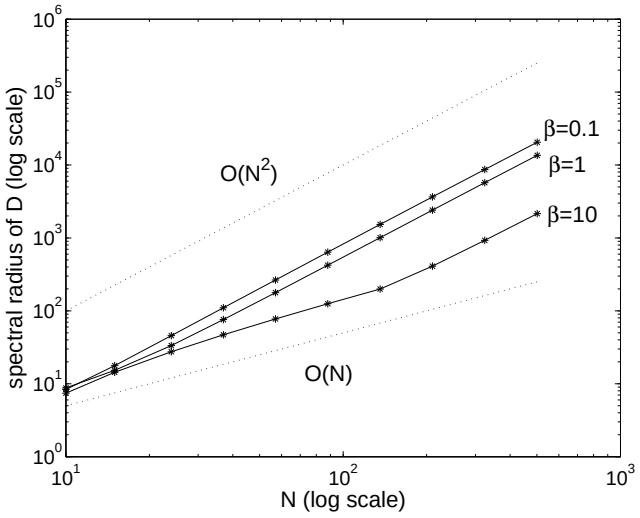

Figure 7: Spectral radius of the GRBF differentiation matrix as a function of N. For large values of N, the spectral radius grows as $O(N^2)$.

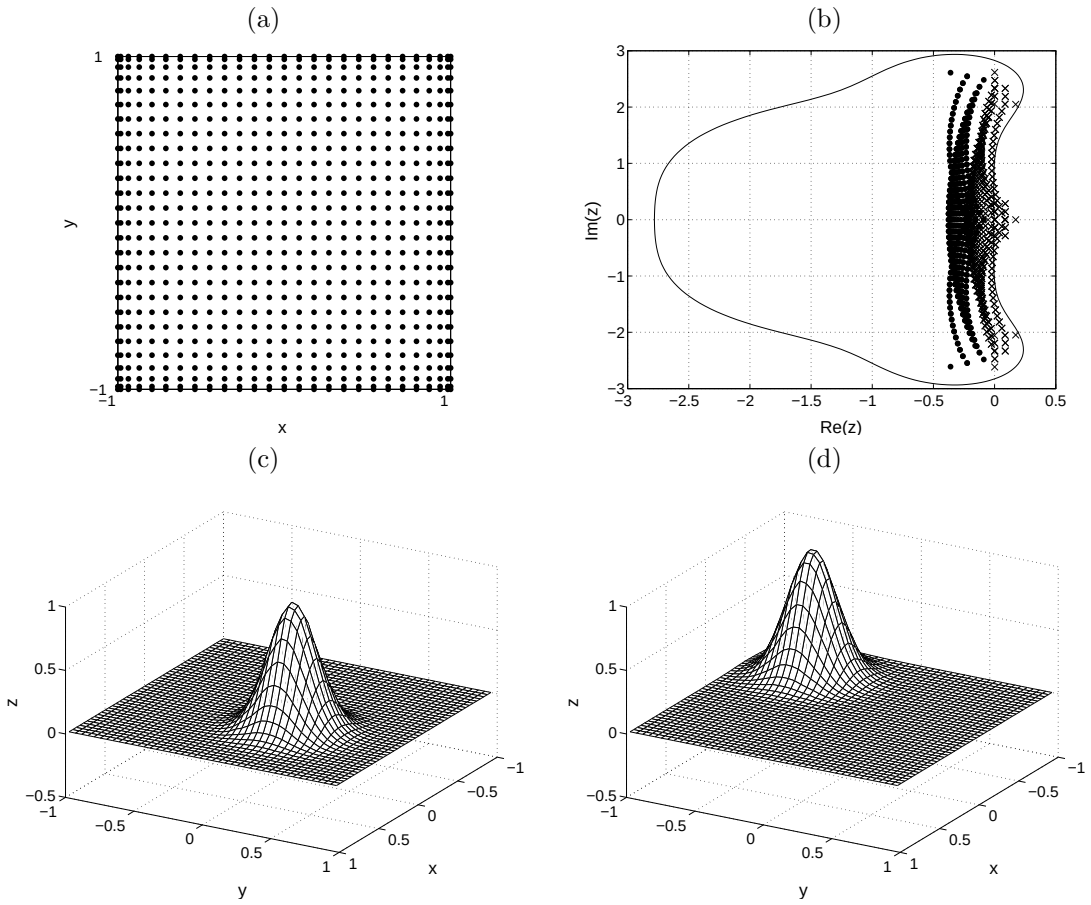

Figure 8: A 2-D test problem: (a) stable nodes; (b) Runge-Kutta stability region, scaled eigenvalues for stable nodes ($\bullet$) and for equispaced nodes ($\times$); (c) initial condition; (d) numerical solution at $t = 0.7$.

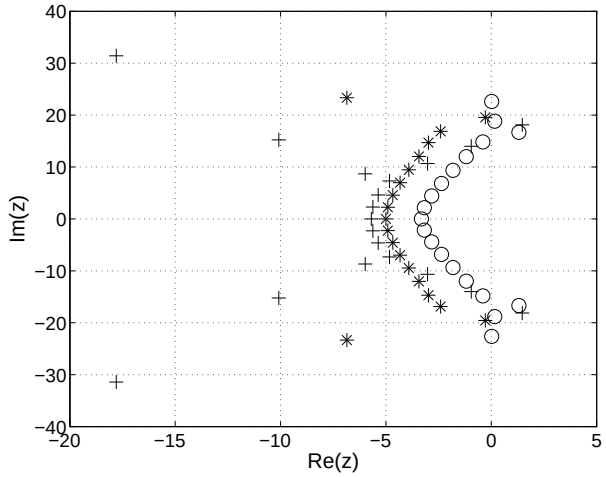

Figure 9: Spectra of differentiation matrices generated with multiquadrics and three sets of interpolation nodes: equally spaced ($\circ$); Chebyshev ($+$); and GRBF stable nodes ($*$).

Although asymptoticly stable nodes for other radial functions, like multiquadrics and inverse quadratics, are not known, Figure 9 indicates that clustering of nodes may also be used to stabilize discretizations obtained with these functions. This figure presents the spectrum of D obtained with multiquadrics, shape parameter $c = 1$, equally spaced centers, $N = 19$, and three sets of interpolation nodes. Notice that for Chebyshev and equispaced nodes the differentiation matrices present unstable eigenvalues, but for GRBF stable nodes for $\beta = 1$, D has only eigenvalues with nonpositive real part.

3.2 GRBFs and mapped polynomial methods

Since the early 1990s attention has been given to mapped polynomial methods, such as the one introduced by Kosloff and Tal-Ezer [18]. The interpolant for mapped polynomial methods takes the form

$$F(x) = \sum_{k=0}^{N} \lambda_k P_k(y),$$

where P_k form a polynomial basis and y is some function of x. The mapping proposed in [18] is

$$y = \frac{1}{\xi} \sin(x \sin^{-1} \xi), \qquad 0 < \xi < 1.$$

The stable interpolation nodes are thus given by Chebyshev nodes [15] in the variable y. The parameter ξ controls the degree to which the grid is stretched under the mapping.

Different strategies to choose the parameter ξ have been proposed in the literature (see [19] and references therein). The goal is to allow near-spectral convergence in space with a time-step restriction of $O(N^{-1})$. To achieve this objective one has to take ξ close to 1 as N is increased; effectively one makes an explicit tradeoff between accuracy and stability.

According to (12), GRBFs with equally spaced centers can be seen as a mapped polynomial method. We believe that in most cases, given β one can find ξ so that both methods present similar convergence and stability properties. Figure 10 shows limiting node density functions for the Kosloff-Tal-Ezer method. Comparing this with Figure 3, we see similar clustering behavior of the density functions near ± 1.

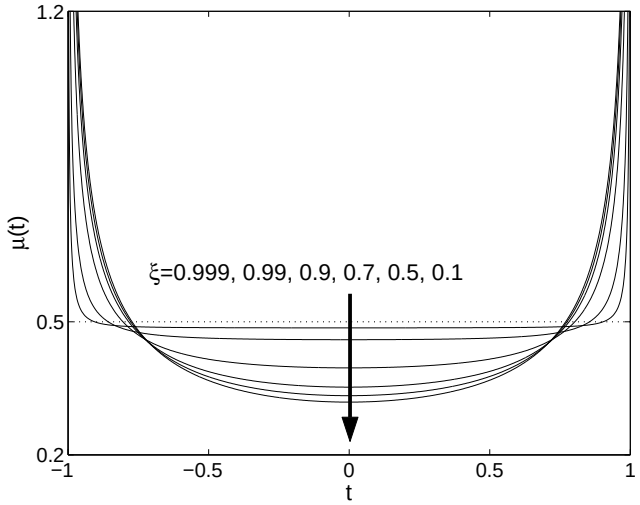

Figure 10: Density functions for several values of ξ for the Kosloff-Tal-Ezer modified Chebyshev method.

4 Least-squares approximations

Although the previous section presents an stable algorithm for Gaussian RBFs, stable nodes for other radial functions, like multiquadrics, are yet not known. Moreover, the task of finding stable nodes becomes more complex – perhaps impossible – in higher dimensions with complicated geometry. We propose using least-squares approximations to avoid this difficulty. In [20] Buhmann presents several benefits of using least-squares instead of interpolation.

We seek least-squares approximations in a discrete norm. With this approach, given $N+1$ centers, we select M nodes at which to compute the residual of the approximations; quadrature nodes are one obvious choice. The goal is to minimize the norm of the residual

$$R(\boldsymbol{\lambda}; x) = u(x) - \sum_{k=0}^{N} \lambda_k \phi(\|x - x_k^c\|)$$

in the interior of the domain and enforce the boundary condition at boundary nodes.

Boundary conditions can be enforced weakly or strongly. In the first approach, the coefficients λ_k minimize the residual in the interior and boundary; i.e., boundary conditions may not be satisfied exactly at boundary nodes. In this case, a weighted norm may be used to penalize errors at boundary nodes more heavily than at interior nodes [21]. We found that this technique may require very large weights at the boundary to stabilize explicit RBF-based methods for time-dependent problems, which in turn usually worsens the condition number of the matrices used in the approximations. We believe that a more efficient way to use discrete least-squares is to enforce boundary conditions strongly.

In order to derive a least-squares differentiation matrix, assume a general region in $\mathcal{R}^d$ and Dirichlet boundary conditions. Given M_I nodes in the interior of the domain, we can write the norm of the residual at these points in matrix form,

$$\mathbf{R}(\boldsymbol{\lambda}) = \|A_I \boldsymbol{\lambda} - \mathbf{u}_I\|. \tag{20}$$

Here $\|\cdot\|$ is the discrete 2-norm in $\mathcal{R}^{M_I}$ and A_I is the RBF evaluation matrix at interior nodes. If in addition we have M_b nodes on the boundary, we require

$$A_b \boldsymbol{\lambda} = \mathbf{u}_b, \tag{21}$$

where A_b is the RBF evaluation matrix at boundary nodes and $\mathbf{u} = [\mathbf{u}_I, \mathbf{u}_b]^T$ is a vector containing the values of the target function at the least-squares nodes.

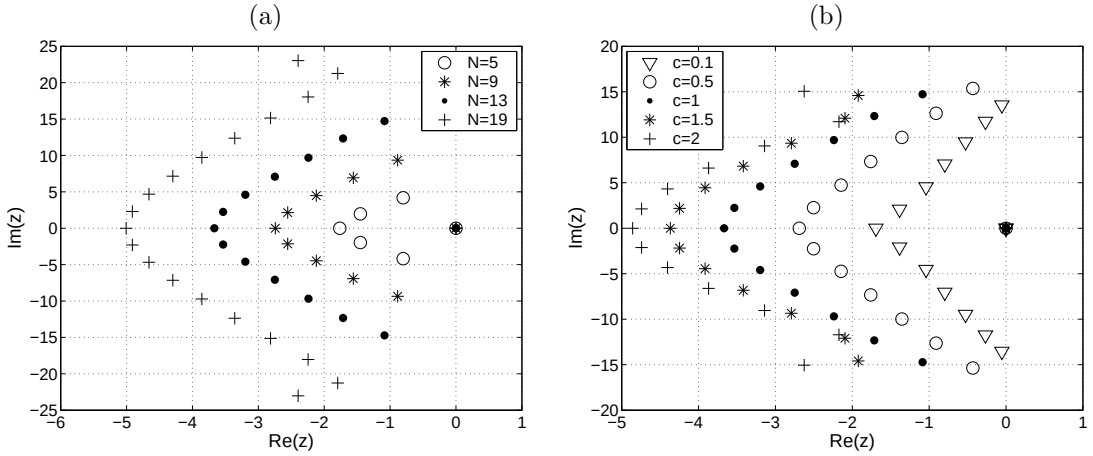

Figure 11: Eigenvalues of D for multiquadric RBFs with equally spaced centers: (a) $c = 1$; (b) $N = 13$.

To solve this constrained least-squares problem, we use the method of direct elimination. The method consists of reducing the number of unknowns in $\boldsymbol{\lambda}$ in order to satisfy the constraint equation (21), and solving the resulting unconstrained reduced system through a QR factorization. The details of the derivation that follows can be found in [22].

We start by computing the pivoted QR decomposition of A_b,

$$A_b \Pi_b = Q_b \begin{bmatrix} R_{b1} & R_{b2} \end{bmatrix},$$

where Π_b is a permutation matrix, $Q_b \in \Re^{M_b \times M_b}$ is orthogonal, and $R_{b1} \in \Re^{M_b \times M_b}$ is upper triangular and nonsingular. Moreover, let

$$A_I \Pi_b = \begin{bmatrix} A_{I1} & A_{I2} \end{bmatrix},$$

where $A_{I1} \in \Re^{M_I \times M_B}$, and define $\hat{A}_{I2} = A_{I2} - A_{I1} R_{b1}^{-1} R_{b2}$. The solution of the constrained least-squares problem is then given by [22]

$$\boldsymbol{\lambda} = \Pi_b \begin{bmatrix} R_{b1} & R_{b2} \\ 0 & R_A \end{bmatrix}^{-1} \begin{bmatrix} Q_b^T \mathbf{u}_b \\ Q_A^T \mathbf{u}_I \end{bmatrix},$$

where $Q_A R_A$ is the reduced QR decomposition of $\hat{A}_{I2}$, $Q_A \in \Re^{M_I \times (N+1-M_b)}$ and $R_A \in \Re^{(N+1-M_b)\times(N+1-M_b)}$.

If we now let B be defined as in section 2, $\mathbf{u}_\mathbf{x} = B\boldsymbol{\lambda}$, where here B can be rectangular, we have

$$\tilde{D} = B \; \Pi_b \begin{bmatrix} R_{b1} & R_{b2} \\ 0 & R_A \end{bmatrix}^{-1} \begin{bmatrix} 0 & Q_b^T \\ Q_A^T & 0 \end{bmatrix}. \tag{22}$$

Notice that $\tilde{D}$ is now an $M \times M$ matrix. Boundary conditions can then be enforced by modifying the matrix $\tilde{D}$ to reflect desired values of $\mathbf{u}_b$. For our test problem given by (2) and (3), we can enforce (3) by simply removing the last column and row of $\tilde{D}$, as in section 2, to obtain the matrix D. We point out that this method can also be used with other boundary conditions, like Neumann boundary conditions, by modifying the constraint equation accordingly and minimizing the residual at interior and boundary nodes.

Figure 11 presents the eigenvalues of the least-squares differentiation matrix for multiquadrics. To generate this data we used twice as many nodes as centers. All numerical results presented in this section were obtained with Chebyshev nodes ($x_j = \cos(\pi j/(2N)),\ j = 0, 1, ..., 2N$). This choice of nodes is not required for stability and in several numerical tests equally spaced nodes were successfully used. In Figure 11(a), we used shape parameter $c = 1$ and equally spaced centers. It

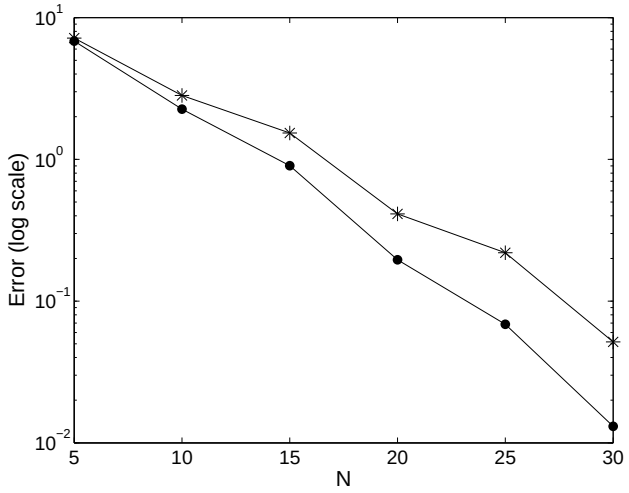

Figure 12: Error in the approximation of df/dx, where $f(x) = \tanh(10x) - \tanh(10)$, using multiquadric RBFs with equally spaced centers ($*$) and imperfectly adapted centers ($\bullet$).

can be observed in this figure that if N is increased, most the spectrum moves further to the left of the imaginary axis. In Figure 11(b) we fixed $N = 13$ and varied c. All eigenvalues presented in this plot allow stable explicit time integration. For instance, one could use fourth order Runge-Kutta in time with $\Delta t = 0.13$ if $N = 13$ and $c = 1$.

Two of the most important features of RBFs are their flexibility in the shape parameter and center locations, compared to a standard polynomial basis. The centers locations can be exploited to increase resolution in specific regions. For instance, the function $f(x) = \tanh(10x) - \tanh(10)$ varies sharply near $x = 0$ and is almost constant in other parts of the interval $[-1, 1]$. Therefore, clustering centers more densely in the middle of the interval, one would expect to get better accuracy. This is indeed the case, as shown in Figure 12. In this plot we compare derivative approximations with equally spaced centers and adapted centers given by $x_j^c = (2/\pi)\sin^{-1}(-1 + 2j/N)$. As expected, the error decays faster if adapted centers are used. No unstable eigenvalue was observed for all data presented.

The standard representations of smooth RBFs subspaces are in most circumstances ill-conditioned. This would make continuation of Figure 12 to, say, 10^{-10} virtually impossible in double precision. For GRBFs with equally spaced centers, however, well-conditioned representations can be computed. In Appendix A we present an algorithm based on an Arnoldi-like iteration to generate orthogonal basis for GRBFs with equally spaced centers. This approach also relies on a close connection between GRBFs and polynomials. In Figure 13 the error of the approximations of the derivative of $f(x) = \sin(\pi x)$ using GRBFs with $\beta = 2$ is presented. The least-squares differentiation matrix was used for these approximations. Notice that the Arnoldi iteration permits approximations close to machine precision, while convergence for the standard GRBF basis stops with error of $O(10^{-6})$.

Figure 14 shows the eigenvalues of the least-square GRBF differentiation matrix for $\beta = 2$ and $N =$30, 50, and 70. The Arnoldi iteration was used to compute an orthogonal basis.

In Figure 15 we consider the numerical solution of the transport equation with initial condition $u_0(x) = \exp(-(5x - 3.5)^{10})$. In Figure 15(a) the exact solution is presented for $t = 1$ together with two numerical solutions obtained with GRBFs and 20 centers. Basis functions were computed with the Arnoldi iteration with $\beta = 1$. It can be observed that the least-squares method gives slightly better results. For the least-squares approximations we used twice as many Chebyshev nodes as centers. Figure 15(b) shows the maximum error, $\max |u_{exact}(t, x) - u_{approx}(t, x)|$, $(t, x) \in [0, 1] \times [-1, 1]$, for several values of N. In this instance the least-squares method and the collocation method on GRBF nodes presented similar rates of convergence and errors. Fourth-order Runge-Kutta was used for time-integration with time-step $\Delta t = 10^{-3}$.

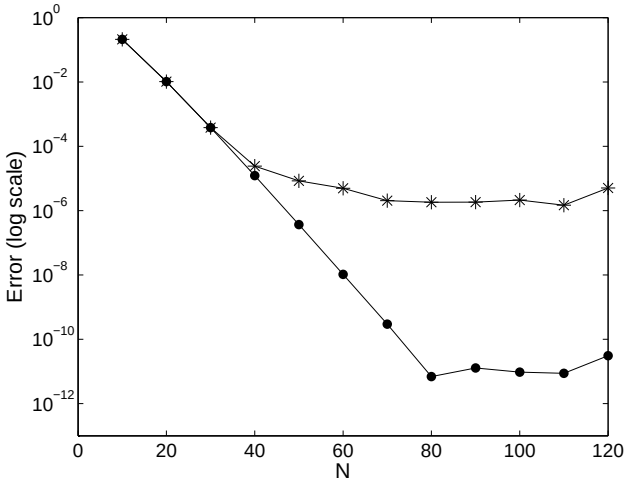

Figure 13: Error in the approximation df/dx, where $f(x) = \sin(\pi x)$, using an orthogonal basis generated with an Arnoldi iteration ($\bullet$) and the standard Gaussian radial basis ($*$).

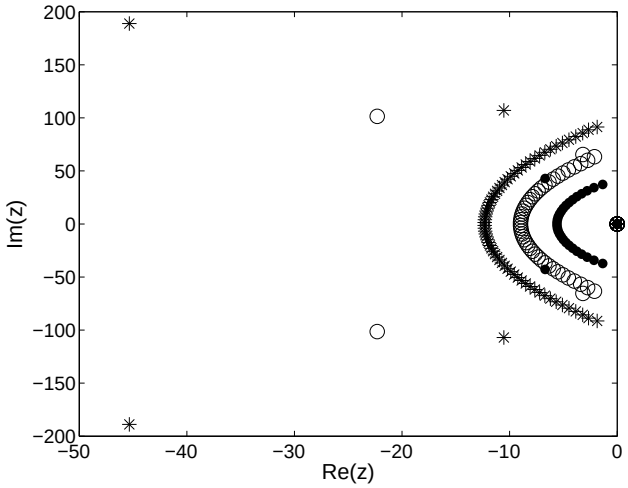

Figure 14: Eigenvalues of the least-square GRBF differentiation matrix for $\beta = 2$: N =30 ($\bullet$); N =50 ($\circ$); and N =70 ($*$).

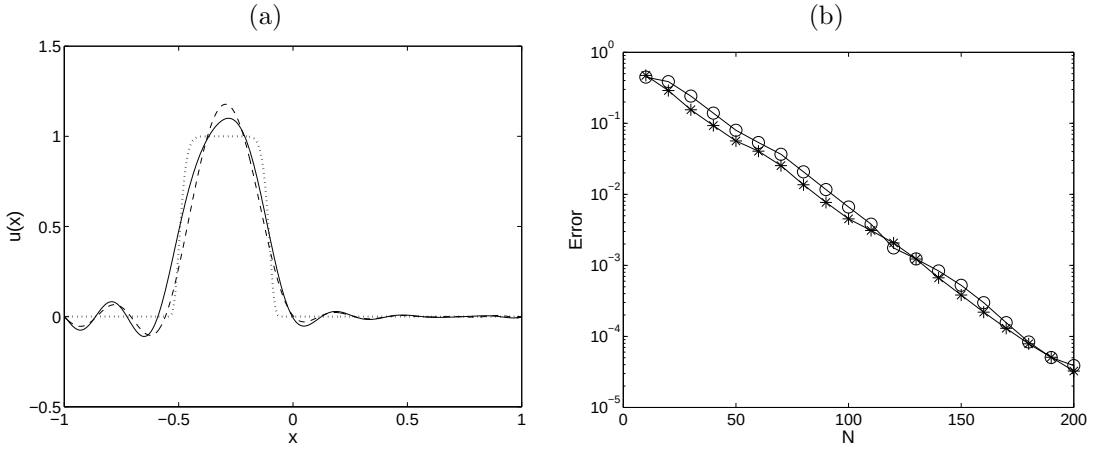

Figure 15: (a) Solution of the transport problem at $t = 1$ using $N = 20$ centers: exact solution (dotted line); least-squares method (solid line); and collocation method on GRBF nodes (dashed line). (b) maximum error for several values of N obtained with the collocation method on GRBF nodes ($\circ$) and the least-squares method ($*$).

To illustrate the least-squares scheme on 2-D regions, we solve the wave equation $u_{tt} = u_{xx}+u_{yy}$ with zero Dirichlet boundary conditions in a peanut-like region defined by parametric equations $x = \sqrt{\cos^2\theta + 4\sin^2\theta}\ \cos\theta$, $y = \sqrt{\cos^2\theta + 4\sin^2\theta}\ \sin\theta$, $0 \le \theta < 2\pi$. Figure 16 presents the eigenvalues of the Laplacian operator discretized with multiquadrics, shape parameter $c = 1$, and 200 uniformly spaced centers. Notice that if the collocation method is used with nodes that coincide with centers, we obtain a matrix that has complex eigenvalues (Figure 16(a)). Using the least-squares method with 432 uniformly spaced nodes, on the other hand, one obtains a matrix with almost purely real spectrum, as expected for the Laplacian, and smaller spectral radius.

Figure 17 presents the numerical solution of the wave equation with initial conditions $u(0, x, y) = \exp(-30(x^2 + (y-1)^2)) + \exp(-50(x^4 + (y+1)^4))$, and $u_t(0, x, y) = 0$. The least-squares method was used together with a leapfrog discretization in time with time step $\Delta t = 0.01$. Notice that this time-discretization scheme for the second order derivative requires a purely real spectrum for stability. A fine grid was used to plot the solution at t=0, 0.33, 0.66, 1.

5 Final remarks

Eigenvalue stability is a crucial factor in the usefulness of RBF discretizations for time-dependent problems. In section 2 we proved that under mild conditions, RBF methods are eigenvalue stable in the absence of boundaries, including methods on periodic domains. However, in the presence of boundaries, RBF collocation is quite likely to be unstable.

In section 3 we showed that Gaussian RBF collocation is stable when special node distributions are used in one dimension. As far as we know, this is the first conclusive demonstration that node locations can eliminate instability asymptotically. While in principle this result should extend to tensor product regions, there is probably little practical interest in implementing RBF methods in such cases, since polynomials are also available.

For problems in complicated geometries, finding universally stable nodes for RBF collocation seems daunting. In section 4 we proposed using least-squares approximations as the foundation of a differentiation matrix. This offers the possibility of separating the requirements of accuracy (governed mostly by the RBF centers) and stability (mandating clustering near boundaries). The added flexibility can be used to adapt centers to data, or to use an Arnoldi-like iteration for equispaced centers to circumvent RBF conditioning issues. Differentiation matrices based on the least-squares idea can incorporate boundary conditions strongly and remain as convenient as collo-

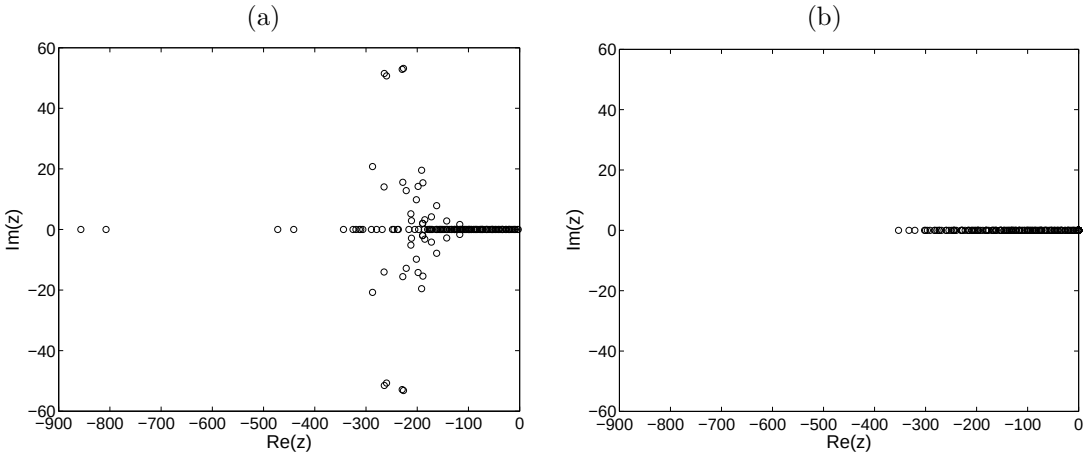

Figure 16: Eigenvalues of the Laplacian operator on the peanut region discretized with multiquadrics: (a) collocation method on uniformly spaced nodes; (b) the least-squares method.

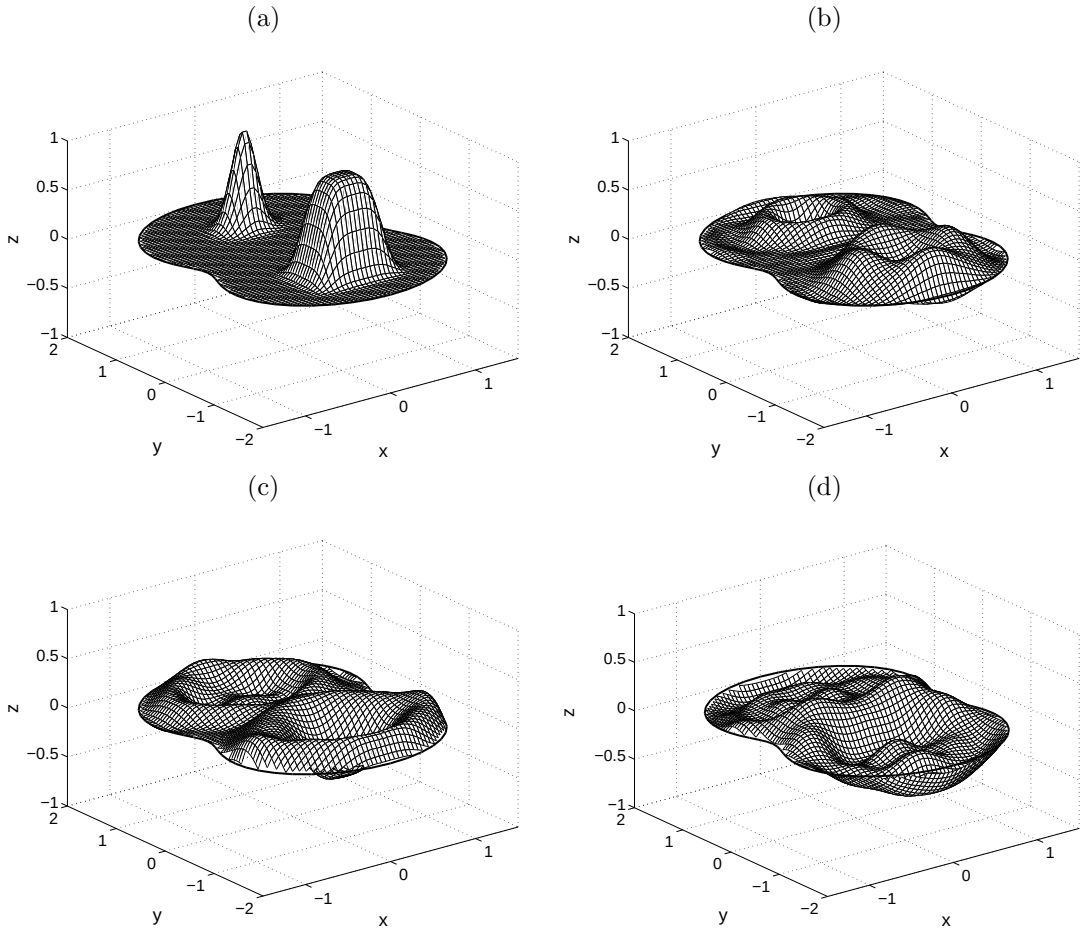

Figure 17: Numerical solution of a vibrating peanut-like membrane using multiquadrics: (a) $t = 0$; (b) $t = 0.33$; (c) $t = 0.66$; and (d) $t = 1$.

cation methods for variable coefficients and nonlinearity. We have demonstrated that they remain eigenvalue stable for widely different discretization parameters. A systematic exploration of their accuracy and stability will be undertaken in future work.

Acknowledgments

We would like to thank Drs. B. Fornberg and G. Wright for fruitful discussions on the stability of RBFs on the unit circle and sphere.

Appendix A

An Arnoldi-like iteration for GRBF approximations

The Arnoldi iteration has been widely used to construct orthonormal bases of the Krylov subspaces, $\mathrm{Span}(b, Ab, A^2b, ..., A^nb)$, where A is a given matrix and b is a vector. In general, the basis $\{b, Ab, A^2b, ..., A^nb\}$ is computationally unstable, however, the Arnoldi iteration allows stable computations. Notice that a new member of the GRBF basis can be constructed from an old one through pointwise multiplication by a function of x. This is the starting point for our Arnoldi-like iteration. The Arnoldi algorithm to produce an orthogonal GRBF basis on $[-1, 1]$, $\{q_0, q_1, \ldots, q_N\}$, is depicted below:

$q_0(x) = \exp(-N\beta(x+1)^2/4)$
$q_0'(x) = (-N\beta(x+1)/2)q_0(x)$
$q_0(x) = q_0(x)/\|q_0\|$
$q_0'(x) = q_0'(x)/\|q_0\|$
For $k = 1 : N$
  $v(x) = q_{k-1}(x)\exp(\beta x)$
  $v'(x) = q_{k-1}'(x)\exp(\beta x) + \beta q_{k-1}(x)\exp(\beta x)$
  For $j = 1 : k-1$
    $v(x) = v(x) - \langle q_j, v\rangle q_j(x)$
    $v'(x) = v'(x) - \langle q_j, v\rangle q_j'(x)$
  end
  $q_k(x) = v(x)/\|v\|$
  $q_k'(x) = v'(x)/\|v\|$
end

In our implementation, the inner product $\langle\cdot,\cdot\rangle$ is the usual discrete L_2 inner product. The inner most loop is the modified Gram-Schmidt orthogonalization. In some cases reorthogonalization may be needed due to rounding errors. In this algorithm we also included the steps to generate the derivatives q_j' of q_j.

If roundoff errors are not present, this algorithm reduces to a Lanczos-like iteration. For the orthogonal basis $\{q_0, \ldots, q_N\}$ generated with the algorithm described above, there exist constants a_k and b_k such that

$$sq_k = a_{k-1}q_{k-1} + b_kq_k + a_kq_{k+1},$$

where $s = e^{\beta x}$. Furthermore, if $\beta \ll 1$ then $a_k = O(\beta)$ and $b_k = 1 + O(\beta)$.

The three term formula can be easily verified since $sq_j \in \mathrm{Span}\{q_0, \ldots, q_{j+1}\}$, which implies that $\langle sq_k, q_j\rangle = \langle q_k, sq_j\rangle = 0$ if $j+1 < k$. Hence, $sq_k = c_kq_{k-1} + b_kq_k + a_kq_{k+1}$ for some constants c_k, b_k, and a_k. And $c_k = \langle sq_k, q_{k-1}\rangle = \langle q_k, sq_{k-1}\rangle = \langle q_k, c_{k-1}q_{k-2} + b_{k-1}q_{k-1} + a_{k-1}q_k\rangle$ gives $c_k = a_{k-1}$.

The bounds for b_k can be obtained from

$$b_k = \langle sq_k, q_k\rangle = \sum_{j=0}^{M} e^{\beta x_j} q_k^2(x_j)dx,$$

which gives $e^{-\beta} \le b_k \le e^{\beta}$ and $b_k = 1 + O(\beta)$ for $\beta \ll 1$. Similarly, we can show that $e^{-\beta} \le \|sq_k\| \le e^{\beta}$. Now using that this basis is orthonormal, we have $|a_{k-1}|^2 + |a_k|^2 = |b_k|^2 - \|sq_k\|^2$, and it follows that $a_k = O(\beta)$, for small β.

We point out that for $c > 0.7$, a simple modification of (12) gives a well-conditioned GRBF approximation,

$$F(x) = e^{-(x+1)^2/c^2} \sum_{k=0}^{N} \lambda_k T_k \left(\frac{e^{\beta x} - \cosh(\beta)}{\sinh(\beta)} \right) = e^{-\frac{\beta N}{4}(x+1)^2} \sum_{k=0}^{N} \lambda_k T_k \left(\frac{e^{\beta x} - \cosh(\beta)}{\sinh(\beta)} \right), \quad (23)$$

where T_k is the kth Chebyshev polynomial. For smaller values of the shape parameter, however, the exponential term in front of the sum becomes very close to zero for some values of x, to the extent that accuracy is compromised. This is usually the case when the parameter β is fixed. Note that in the limit $c \to \infty$, (23) becomes a sum of polynomials, in agreement with [11, 23].

References

[1] G.E. Fasshauer, Solving partial differential equations by collocation with radial basis functions, In *Surface fitting and multiresolution methods*, (Edited by A. LeMéhauté et al.), pp. 131-138, Proc. Chamonix, Vanderbilt University Press, (1997).

[2] Y.C. Hon and X.Z. Mao, An efficient numerical scheme for Burger's equation, *Appl. Math. Comput.* **95** (1), 37-50, (1998).

[3] E.J. Kansa, Multiquadrics - A scattered data approximation scheme with applications to computational fluid dynamics - II. Solutions to hyperbolic, parabolic, and elliptic partial differential equations, *Computers Math. Applic.*, **19** (8/9), 147-161, (1990).

[4] E. Larsson and B. Fornberg, A numerical study of some radial basis function based solution methods for elliptic PDEs, *Computers Math. Applic.*, **46** (5/6), 891-902, (2003).

[5] R.B. Platte and T.A. Driscoll, Computing eigenmodes of elliptic operators using radial basis functions, *Computers Math. Applic.*, **48** (3/4), 561-576, (2004).

[6] R. Schaback, Error estimates and condition numbers for radial basis function interpolation, *Adv. in Comput. Math.* **3** (3), 251-264, (1995).

[7] E.J. Kansa and Y. C. Hon, Circumventing the ill-conditioning problem with multiquadric radial basis functions: Applications to elliptic partial differential equations, *Computers Math. Applic.*, **39** (7/8), 123-137, (2000).

[8] E.J. Kansa, H. Power, G.E. Fasshauer, and L. Ling, A volumetric integral radial basis function method for time-dependent partial differential equations. I. Formulation, *Eng. Anal. Bound. Elem.*, **28** (10), 1191-1206, (2004).

[9] R.B. Platte and T.A. Driscoll, Polynomials and potential theory for Gaussian radial basis function interpolation, *SIAM J. Numer. Anal.*, to appear.

[10] S.C. Redy and L.N. Trefethen, Stability of the method of lines, *Numer. Math.*, **62** (2), 234-267, (1992).

[11] B. Fornberg, G. Wright, and E. Larsson, Some observations regarding interpolants in the limit of flat radial basis functions, *Computers Math. Applic.*, **47** (1), 37-55, (2004).

[12] R. Schaback and H. Wendland, Characterization and construction of radial basis functions, In *Multivariate approximation and applications*, (Edited by N. Dyn et al.), pp.1-24, Cambridge University Press, (2001).

[13] B. Fornberg and D. Merrill, Comparison of finite difference and pseudospectral methods for convective flow over a sphere, *Gheophys. Res. Lett.*, **24** (24), 3245-3248, (1997).

[14] B. Fornberg, *A Practical Guide to Pseudospectral Methods*. Cambridge University Press, New York, (1996).

[15] L.N. Trefethen, *Spectral Methods in* MATLAB. SIAM, Philadelphia, (2000).

[16] L.N. Trefethen and J.A.C. Weideman, Two results on polynomial interpolation in equally spaced points, *J. Approx. Theory*, **65** (3), 247-260, (1991).

[17] J.-P. Berrut and L. N. Trefethen, Barycentric Lagrange interpolation, *SIAM Rev.*, **46** (3), 501-517, (2004).

[18] D. Kosloff and H. Tal-Ezer, A modified Chebyshev pseudospectral method with an $O(N^{-1})$ time step restriction, *J. Comput. Phys.*, **104** (2), 457-469, (1993).

[19] J.L. Mead and R.A. Renault, Accuracy, resolution, and stability properties of a modified Chebyshev method, *SIAM J. Sci. Comput.*, **24** (1), 143-160, (2002).

[20] M.D. Buhmann, *Radial basis functions*, Cambridge University Press, Cambridge, UK, (2003).

[21] E.D. Eason, A review of least-squares methods for solving partial differential equations, *Int. J. Numer. Meth. Eng.*, **10** (5), 1021-1046, (1976).

[22] Å. Björck, *Numerical methods for least squares problems*, SIAM, Philadelphia, (1996).

[23] T.A. Driscoll and B. Fornberg, Interpolation in the limit of increasingly flat radial basis functions, *Computers Math. Applic.*, **43** (3/5), 413-422, (2002).