# Using Gaussian eigenfunctions to solve boundary value problems

Michael McCourt

January 18, 2013

**Abstract**

Kernel-based methods are popular in computer graphics, machine learning, and statistics, among other fields; because they do not require meshing of the domain under consideration, higher dimensions and complicated domains can be managed with reasonable effort. Traditionally, the high order of accuracy associated with these methods has been tempered by ill-conditioning, which arises when highly smooth kernels are used to conduct the approximation. Recent advances in representing Gaussians using eigenfunctions have proven successful at avoiding this destabilization in scattered data approximation problems. This paper will extend these techniques to the solution of boundary value problems using collocation. The method of particular solutions will also be considered for elliptic problems, using Gaussian eigenfunctions to stably produce an approximate particular solution.

*This is dedicated to Graeme Fairweather, whose guidance and patience has instilled an everlasting love of mathematics in myself and countless others.*

## 1 Introduction

Kernel-based meshfree approximation methods have gained popularity in several fields, including scattered data interpolation [52], finance [25], statistics [49], machine learning [43] and others. One of the great benefits of using these methods is that no discretization of the relevant domain is required; basis functions are centered at various points throughout the domain, allowing for kernel-based methods to circumvent some of the barriers associated with higher dimensional problems. Additionally, a variety of kernels exist, providing users in each application the ability to tailor the solution basis to fit that application's specific opportunities and constraints.

Techniques for solving boundary value problems (BVPs) with radial basis functions (RBFs) have advanced significantly in the past two decades. The original method for solving elliptic partial differential equations (PDEs) with RBFs came in 1990 [31] and involved an unsymmetric collocation of basis functions at points chosen throughout the domain. Since that initial work, further analysis has been done on the convergence of this collocation method [46], which has encouraged its use despite its theoretic potential for failure [26]. A symmetric collocation technique was also developed [9] which ensured invertibility of the collocation system by using a modified set of basis functions.

Another popular method for solving BVP with radial basis functions is the method of fundamental solutions [8]. Essentially, this method replaces the BVP with an interpolation problem on the boundary using functions which satisfy the PDE. The mathematical formulation of this method is well-developed, but it is only applicable for homogeneous problems where the fundamental solution is known. The method of particular solutions [5] is an adaptation for inhomogeneous problems involving two approximation systems: one to satisfy the inhomogeneity in the interior, and another to satisfy the boundary conditions, assuming a now homogeneous problem. The use of radial basis functions to approximate particular solutions was discussed in [21, 27].

One of the great shortcomings of radial basis functions is that, for some parameterizations, the resulting linear system may be irrevocably ill-conditioned [10]. Even more troublesome is the fact that the most accurate parameterizations may lie in the ill-conditioned regime [19]. This ill-conditioning is especially significant for kernels with a great deal of smoothness, which often tempers the optimism of researchers hoping to exploit their spectral accuracy. In [11], this problem was addressed for Gaussians in $\mathbb{R}^d$ by

using a truncated eigenfunction expansion of the Gaussian. Here, we will extend the approximation via eigenfunctions to the solution of boundary value problems.

Many more methods for solving boundary value problems with kernels exist beyond what will be discussed in this paper. Multilevel methods [36, 30] have been presented, including for higher order problems [1], to attempt to mitigate the cost associated with solving dense systems generated by globally supported RBFs. Finite difference schemes based on RBFs [13, 14] have proven to be an effective meshfree solver for geological and climate based problems. Partition of unity methods [34] are being developed now to incorporate RBF collocation with other solution schemes for applications including crack propagation. Petrov-Galerkin techniques [2] have been developed to allow the weak form solution of PDEs, while recent work [47] has provided analytic support for this approach. Some work has been done incorporating RBFs into discontinuous Galerkin schemes [44]. Kernel based PDE solvers on manifolds [20] are beginning to mature as well.

To narrow our focus from all possible BVP solvers using kernels, we will discuss only collocation and the method of particular solutions. In Section 2 we consider the solution of boundary value problems by collocation with traditional Gaussian RBFs, and demonstrate the benefit of instead using the eigenfunction expansion. We will also consider the use of differentiation matrices [50] to solve problems. In Section 3 the eigenfunction expansion is applied to approximate particular solutions and solve BVPs with the method of particular solutions. We extend this particular solution approach in Section 3.3 to incorporate boundary data and produce a more accurate solution at less cost.

# 2 Collocation using Gaussian eigenfunctions

The original RBF collocation technique in [31] involved multiquadrics supplemented by linear polynomials. These basis functions are subject to severe ill-conditioning depending on the flatness of the multiquadrics. This ill-conditioning is the result of extremely flat basis functions looking too much alike, causing the representative columns in the collocation matrix to become indistinguishable and making the system appear to be low rank.

This problem is not unique to multiquadrics or to collocation techniques; indeed any application requiring the inversion of matrices generated by very smooth RBFs will fall victim to this as the RBFs approach their flat limit. It has been discussed for interpolation problems that, despite this perceived impasse, the problem itself is not necessarily ill-conditioned [35, 7, 42]. Rather, it is the solution approach (i.e., forming a linear system using the RBF basis) which deals the damage [33], and if an alternate method could handle the ill-conditioning the true solution could be found [19].

One such approach to solving this problem is to find a series expansion for the kernel which allows for the removal of the ill-conditioned terms analytically. This solution technique is called RBF-QR [18], and it has been used successfully on the circle/sphere for both interpolation [16] and PDEs [17]. In these papers, the authors discussed the possibility that the most accurate kernel parameterizations were also too ill-conditioned to treat directly, necessitating the series expansion approach.

In [11], a series expansion was developed to allow for stable approximation with Gaussians in $\mathbb{R}^d$; this expansion was based on the eigenfunctions of the associated Hilbert-Schmidt operator. Because the Gaussian kernel in higher dimensions is formed through tensor products, the higher dimensional series expansion is also formed with a tensor product, trivially allowing the move to $\mathbb{R}^d$. The approximation of derivatives using this series expansion was discussed in [39]. Here we would like to use these derivatives to solve boundary value problems with collocation.

## 2.1 Ill-conditioning in Gaussian basis collocation

Linear BVPs, without dependence on time, can generally be phrased in the form

$$\begin{aligned} \mathcal{L}u &= f, \qquad \text{on the interior } \Omega, \\ \mathcal{B}u &= g, \qquad \text{on the boundary } \partial\Omega, \end{aligned}$$

where $\mathcal{L}$ is the linear PDE operator, and $\mathcal{B}$ is the linear boundary condition operator. $\Omega \in \mathbb{R}^d$ is a bounded domain with Lipshitz boundary. In unsymmetric kernel collocation, we assume that the solution takes the

form

$$u(\boldsymbol{x}) = \sum_{k=1}^{N} a_k K(\boldsymbol{x}, \boldsymbol{x}_k) + \sum_{\ell=1}^{q} a_{N+\ell} p_\ell(\boldsymbol{x}) \tag{1}$$

where $\boldsymbol{x} \in \partial\Omega \cup \Omega$ is a $d$-dimensional vector for a problem in $\mathbb{R}^d$, $\{\boldsymbol{x}_k\}_{k=1}^N$ are the kernel centers, $N$ is the number of kernels used, $K$ is the kernel, $\{p_\ell\}_{\ell=1}^q$ are polynomial terms, and $q$ is the number of polynomial terms. For the time being, we will assume that no polynomial terms are necessary; later we will briefly discuss the effect this may have on the accuracy of the solution and optimal choice of $K$.

Choosing $q = 0$ will remove the polynomial terms and leave the pure kernel series

$$u(\boldsymbol{x}) = \sum_{k=1}^{N} a_k K(\boldsymbol{x}, \boldsymbol{x}_k). \tag{2}$$

Assuming that we have chosen $N_{\mathcal{L}}$ collocation points on the interior and $N_{\mathcal{B}}$ collocation points on the boundary, we can now apply the BVP operators to (2); note that the PDE operators act on the first kernel argument, as the second kernel argument defines the center of the kernel, not where the kernel is being evaluated. This will leave us with the continuous collocation equations

$$\begin{aligned} \sum_{k=1}^{N} a_k \mathcal{L} K(\boldsymbol{x}, \boldsymbol{x}_k) &= f(\boldsymbol{x}), \qquad \boldsymbol{x} \in \Omega, \\ \sum_{k=1}^{N} a_k \mathcal{B} K(\boldsymbol{x}, \boldsymbol{x}_k) &= g(\boldsymbol{x}), \qquad \boldsymbol{x} \in \partial\Omega. \end{aligned}$$

We must now choose a finite number of points, $N_{\mathcal{L}}$ on the interior and $N_{\mathcal{B}}$ on the boundary, at which to enforce these equations. If the $\{\boldsymbol{x}_k\}_{k=1}^{N_{\mathcal{L}}}$ interior points are ordered before the $\{\boldsymbol{x}_k\}_{k=1+N_{\mathcal{L}}}^{N_{\mathcal{B}}+N_{\mathcal{L}}}$ boundary points, this system of linear equations has the matrix form

$$\begin{pmatrix} \mathcal{L}K(\boldsymbol{x}_1, \boldsymbol{x}_1) & \cdots & \mathcal{L}K(\boldsymbol{x}_1, \boldsymbol{x}_N) \\ & \vdots & \\ \mathcal{L}K(\boldsymbol{x}_{N_{\mathcal{L}}}, \boldsymbol{x}_1) & \cdots & \mathcal{L}K(\boldsymbol{x}_{N_{\mathcal{L}}}, \boldsymbol{x}_N) \\ \mathcal{B}K(\boldsymbol{x}_{N_{\mathcal{L}}+1}, \boldsymbol{x}_1) & \cdots & \mathcal{B}K(\boldsymbol{x}_{N_{\mathcal{L}}+1}, \boldsymbol{x}_N) \\ & \vdots & \\ \mathcal{B}K(\boldsymbol{x}_{N_{\mathcal{L}}+N_{\mathcal{B}}}, \boldsymbol{x}_1) & \cdots & \mathcal{B}K(\boldsymbol{x}_{N_{\mathcal{L}}+N_{\mathcal{B}}}, \boldsymbol{x}_N) \end{pmatrix} \begin{pmatrix} a_1 \\ \\ \vdots \\ \\ a_N \end{pmatrix} = \begin{pmatrix} f(\boldsymbol{x}_1) \\ \vdots \\ f(\boldsymbol{x}_{N_{\mathcal{L}}}) \\ g(\boldsymbol{x}_{N_{\mathcal{L}}+1}) \\ \vdots \\ g(\boldsymbol{x}_{N_{\mathcal{L}}+N_{\mathcal{B}}}) \end{pmatrix} \tag{3}$$

By choosing $N_{\mathcal{L}} + N_{\mathcal{B}} = N$, the system (3) is square, and if it is nonsingular [46] it has a unique solution.

Theoretically, there is nothing requiring the kernel centers to be the same as the collocation points. We will consider no such instances here, although such material is presented for interpolation in [15] and PDEs [12, 48] suggesting that this may improve the error near the boundary. By choosing the kernel centers to match the collocation points, we trivially satisfy $N_{\mathcal{L}} + N_{\mathcal{B}} = N$ and must solve a square linear system to find $a_1, \ldots, a_N$.

To demonstrate their notoriously ill-conditioned behavior, we will consider Gaussian kernels

$$K(\boldsymbol{x}, \boldsymbol{z}) = \exp(-\varepsilon^2 \|\boldsymbol{x} - \boldsymbol{z}\|^2) \tag{4}$$

for the collocation solution. The value $\varepsilon$ is the *shape parameter*, so-called because for large $\varepsilon$ the Gaussians become very peaked, and for small $\varepsilon$ the Gaussians become very flat. A well-chosen $\varepsilon$ can allow for very accurate solutions (even more accurate than polynomials in some cases) whereas a poorly chosen $\varepsilon$ may provide little or no accuracy. See Figure 1 for a demonstration of the effect $\varepsilon$ can have on accuracy.

Figure 1 was generated by solving the boundary value problem

$$u_{xx}(x) = \frac{-\sinh(x)}{(1+\cosh(x))^2}, \quad x \in (-1, 1), \tag{5a}$$

$$u(x) = \frac{\sinh(x)}{1+\cosh(x)}, \qquad x \in \{-1, 1\}, \tag{5b}$$

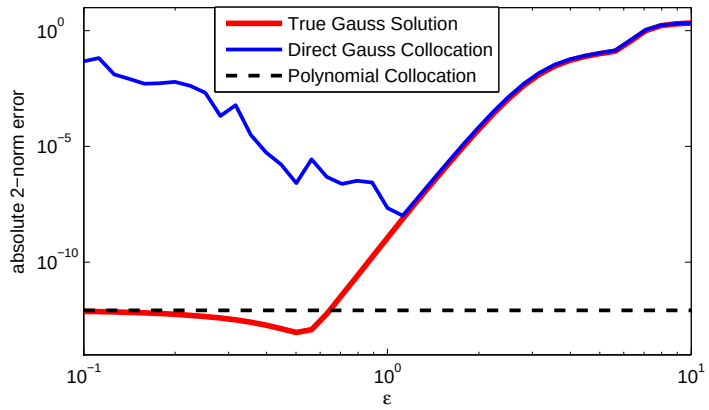



Figure 1: Solving (3) produces a good solution until ill-conditioning overwhelms the accuracy, preventing the solution from reaching its polynomial limit. If we could stably solve the system, we should find the "True Gauss Solution" curve. Error is computed at 200 evenly spaced points in the domain.

with $N = 16$ collocation points located at the Chebyshev nodes

$$x_k = \cos\left(\pi \frac{(k-1)}{N-1}\right), \qquad 1 \le k \le N.$$

Phrased in terms of the general BVP language from earlier, this problem has components

$$\mathcal{L} = \frac{d^2}{dx^2}, \qquad f(x) = \frac{-\sinh(x)}{(1+\cosh(x))^2},$$
$$\mathcal{B} = \mathcal{I}, \qquad g(x) = \frac{\sinh(x)}{1+\cosh(x)},$$

where $\mathcal{I}$ is the identity operator, $\mathcal{I}u = u$. The solution named "Direct Gauss Collocation" was computed by solving the system (3) using (4). The poor behavior as $\varepsilon \to 0$ is a result of the ill-conditioning in the collocation matrix: for $\varepsilon = 1$ the condition number is $\mathcal{O}(10^{13})$, even though the matrix is only size $N = 16$.

It has been proven [16] that this ill-conditioning is a symptom only of the choice of basis, and not fundamental to the approximation problem. For interpolation problems we have seen that the limit of Gaussian as $\varepsilon \to 0$ is well-defined, and is in fact equal to the polynomial interpolant [11]. We therefore expect that, in the absence of ill-conditioning, the Gaussian collocation solution would approach the "Polynomial Collocation" result; this polynomial solution was computed using the differentiation matrix approach from [50]. The "True Gauss Solution" displayed above shows this desired behavior; we will now explain how this solution is computed without the ill-conditioning inherent in solving (3).

## 2.2 Collocation using the stable basis

We need to replace the kernel $K(x,z) = \mathrm{e}^{-\varepsilon^2|x-z|^2}$ with its truncated eigenfunction expansion

$$\mathrm{e}^{-\varepsilon^2|x-z|^2} = \sum_{k=1}^{M} \lambda_k \varphi_k(x) \varphi_k(z),$$

where $\lambda_k$ and $\varphi_k$ are

$$\lambda_k = \sqrt{\frac{\alpha^2}{\alpha^2+\delta^2+\varepsilon^2}} \left(\frac{\varepsilon^2}{\alpha^2+\delta^2+\varepsilon^2}\right)^{k-1}, \tag{6a}$$
$$\varphi_k(x) = \gamma_k \mathrm{e}^{-\delta^2 x^2} H_{k-1}(\beta\alpha x), \tag{6b}$$

with $H_{k-1}$ the degree $k-1$ Hermite polynomial. The value $\alpha$ is the global scale parameter as defined in [11], and the auxiliary parameters

$$\beta = \left(1 + \left(\frac{2\varepsilon}{\alpha}\right)^2\right)^{\frac{1}{4}}, \quad \gamma_k = \sqrt{\frac{\beta}{2^{k-1}\Gamma(k)}}, \quad \delta^2 = \frac{\alpha^2}{2}\left(\beta^2 - 1\right),$$

are defined in terms of $\varepsilon$ and $\alpha$. The truncation value is assumed to satisfy $M > N$, although this assumption will be reconsidered later. The value $M$ is chosen large enough to satisfy a bound on the ratio $\lambda_M/\lambda_N$; this choice is described in [11], and will not be discussed here. Regardless of the value of $M$, the eigenfunction series will be the optimal $M$-term approximation to the Gaussian in the $L_2(\mathbb{R}, \rho)$ sense, where

$$\rho(x) = \frac{\alpha}{\sqrt{\pi}}\, \mathrm{e}^{-\alpha^2 x^2}$$

is a weight function which localizes the $L_2$ inner product [43].

In matrix form, this $M$-term series expansion can be written as

$$\mathrm{e}^{-\varepsilon^2|x-z|^2} = \begin{pmatrix}\varphi_1(x) & \dots & \varphi_M(x)\end{pmatrix} \begin{pmatrix}\lambda_1 & & \\ & \ddots & \\ & & \lambda_M\end{pmatrix} \begin{pmatrix}\varphi_1(z) \\ \vdots \\ \varphi_M(z)\end{pmatrix}.$$

Substituting this into the matrix from (3), and noting that the operators $\mathcal{L}$ and $\mathcal{B}$ apply to the first kernel argument, converts that matrix to

$$\begin{pmatrix} \mathcal{L}\varphi_1(x_1) & \cdots & \mathcal{L}\varphi_M(x_1) \\ & \vdots & \\ \mathcal{L}\varphi_1(x_{N_\mathcal{L}}) & \cdots & \mathcal{L}\varphi_M(x_{N_\mathcal{L}}) \\ \mathcal{B}\varphi_1(x_{N_\mathcal{L}+1}) & \cdots & \mathcal{B}\varphi_M(x_{N_\mathcal{L}+1}) \\ & \vdots & \\ \mathcal{B}\varphi_1(x_{N_\mathcal{L}+N_\mathcal{B}}) & \cdots & \mathcal{B}\varphi_M(x_{N_\mathcal{L}+N_\mathcal{B}}) \end{pmatrix} \begin{pmatrix}\lambda_1 & & \\ & \ddots & \\ & & \lambda_M\end{pmatrix} \begin{pmatrix} \varphi_1(x_1) & \cdots & \varphi_1(x_N) \\ & \vdots & \\ \varphi_M(x_1) & \cdots & \varphi_M(x_N) \end{pmatrix}. \tag{7}$$

This allows (3) to be written in block form as

$$\begin{pmatrix}\mathcal{L}\Phi_{\mathcal{L},1} & \mathcal{L}\Phi_{\mathcal{L},2} \\ \mathcal{B}\Phi_{\mathcal{B},1} & \mathcal{B}\Phi_{\mathcal{B},2}\end{pmatrix} \begin{pmatrix}\Lambda_1 & \\ & \Lambda_2\end{pmatrix} \begin{pmatrix}\Phi_{\mathcal{L},1}^T & \Phi_{\mathcal{B},1}^T \\ \Phi_{\mathcal{L},2}^T & \Phi_{\mathcal{B},2}^T\end{pmatrix} \boldsymbol{a} = \begin{pmatrix}\boldsymbol{f}_\mathcal{L} \\ \boldsymbol{g}_\mathcal{B}\end{pmatrix}, \tag{8}$$

where

$$\begin{aligned}
(\Phi_{\mathcal{L},1})_{j,k} &= \varphi_k(x_j) && \text{for} && 1 \le k \le N,\ x_j \in \Omega, \\
(\Phi_{\mathcal{L},2})_{j,k} &= \varphi_k(x_j) && \text{for} && N+1 \le k \le M,\ x_j \in \Omega, \\
(\Phi_{\mathcal{B},1})_{j,k} &= \varphi_k(x_j) && \text{for} && 1 \le k \le N,\ x_j \in \partial\Omega, \\
(\Phi_{\mathcal{B},2})_{j,k} &= \varphi_k(x_j) && \text{for} && N+1 \le k \le M,\ x_j \in \partial\Omega, \\
(\Lambda_1)_{k,k} &= \lambda_k && \text{for} && 1 \le k \le N, \\
(\Lambda_2)_{k,k} &= \lambda_{k+N} && \text{for} && 1 \le k \le M-N, \\
(\boldsymbol{f}_\mathcal{L})_j &= f(x_j) && \text{for} && x_j \in \Omega, \\
(\boldsymbol{g}_\mathcal{B})_j &= g(x_j) && \text{for} && x_j \in \partial\Omega.
\end{aligned}$$

For terms such as $\mathcal{L}\Phi_{\mathcal{L},1}$ which appear in (8), the operator passes through naturally using the matrix definitions above: $(\mathcal{L}\Phi_{\mathcal{L},1})_{j,k} = \mathcal{L}\varphi_k(x_j)$.

As discussed in [18], the ill-conditioning in this system exists primarily in the diagonal matrix containing $\Lambda_1$ and $\Lambda_2$. The RBF-QR approach to alleviating this ill-conditioning is described in [39] and converts the symmetric positive definite system (8) to the unsymmetric (but still nonsingular) system

$$\begin{pmatrix}\mathcal{L}\Phi_{\mathcal{L},1} & \mathcal{L}\Phi_{\mathcal{L},2} \\ \mathcal{B}\Phi_{\mathcal{B},1} & \mathcal{B}\Phi_{\mathcal{B},2}\end{pmatrix} \begin{pmatrix}\mathsf{I}_N \\ \Lambda_2(\Phi_{\mathcal{L},2}^T \ \ \Phi_{\mathcal{B},2}^T)(\Phi_{\mathcal{L},1}^T \ \ \Phi_{\mathcal{B},1}^T)^{-1}\Lambda_1^{-1}\end{pmatrix} \hat{\boldsymbol{a}} = \begin{pmatrix}\boldsymbol{f}_\mathcal{L} \\ \boldsymbol{g}_\mathcal{B}\end{pmatrix}. \tag{9}$$

The $\Lambda_2$ and $\Lambda_1^{-1}$ terms can be applied simultaneously, preventing overflow or underflow issues. Because the $\Lambda_2$ terms are exponentially smaller than the $\Lambda_1$ terms (refer to (6a)) there are no fears about this new formulation undergoing dangerous growth. The term $(\Phi_{\mathcal{L},2}^T \;\; \Phi_{\mathcal{B},2}^T)(\Phi_{\mathcal{L},1}^T \;\; \Phi_{\mathcal{B},1}^T)^{-1}$ is generally computed using the QR factorization (thus the name RBF-QR) to avoid mixing different orders of the eigenfunctions during the decomposition. We will refer to this eigenfunction approach, joint with RBF-QR, as **GaussQR**.

The new coefficients $\hat{\boldsymbol{a}}$ can be related to the standard Gaussian basis coefficients $\boldsymbol{a}$ by

$$\Lambda_1(\Phi_{\mathcal{L},1}^T \;\; \Phi_{\mathcal{B},1}^T)\boldsymbol{a} = \hat{\boldsymbol{a}},$$

but computing $\boldsymbol{a}$ is not recommended; the $\Lambda_1$ matrix is severely ill-conditioned because of the exponentially decreasing eigenvalues. Because of this, we solve for and evaluate the interpolant only in terms of the stable basis $\{\psi_k\}_{k=1}^N$:

$$\begin{aligned} u(x) &= \boldsymbol{\psi}(x)^T \hat{\boldsymbol{a}} \\ &= (\psi_1(x) \quad \cdots \quad \psi_N(x))\hat{\boldsymbol{a}} \\ &= (\varphi_1(x) \quad \cdots \quad \varphi_M(x)) \begin{pmatrix} \mathsf{I}_N \\ \Lambda_2(\Phi_{\mathcal{L},2}^T \;\; \Phi_{\mathcal{B},2}^T)(\Phi_{\mathcal{L},1}^T \;\; \Phi_{\mathcal{B},1}^T)^{-1}\Lambda_1^{-1} \end{pmatrix} \hat{\boldsymbol{a}}. \end{aligned} \tag{10}$$

By applying the specific BVP operators and functions described above, solving the system (9), and evaluating the solution with (10), we can generate the "True Gauss Solution" curve presented in Figure 1. That solution matches the standard basis solution for larger values of $\varepsilon$, and achieves the expected polynomial limit as $\varepsilon \to 0$. The global scale parameter $\alpha$ was set to 1 for these experiments.

## 2.3 Low-rank series approximate collocation

In order to produce the stable collocation solution in Section 2.2, the eigenfunction series must be chosen with $M > N$. As discussed in [11], it may be possible to choose $M < N$ when $N$ is large or for $\varepsilon \ll 1$. This is especially important in higher dimensions, where satisfying $\lambda_M/\lambda_N < \epsilon_{\text{mach}}$ for $\epsilon_{\text{mach}} \approx 10^{-16}$ requires more eigenfunctions depending on the dimension of the problem.

This shift to an early truncation point $M < N$ has a significant change on the collocation problem, because it converts the full-rank system (8) into a rank $M$ system. The transition follows the same pattern as before, except using a low-rank approximation to the Gaussian. Starting from (7), and imposing the restriction $M < N$ produces the rank-$M$ collocation system

$$\begin{pmatrix} \mathcal{L}\Phi_{\mathcal{L}} \\ \mathcal{B}\Phi_{\mathcal{B}} \end{pmatrix} \Lambda \begin{pmatrix} \Phi_{\mathcal{L}}^T & \Phi_{\mathcal{B}}^T \end{pmatrix} \boldsymbol{a} = \begin{pmatrix} \boldsymbol{f}_{\mathcal{L}} \\ \boldsymbol{g}_{\mathcal{B}} \end{pmatrix}.$$

We use similar block definitions as before, with

$$\begin{aligned} (\Phi_{\mathcal{L}})_{j,k} &= \varphi_k(x_j) && \text{for} && 1 \le k \le M,\ x_j \in \Omega, \\ (\Phi_{\mathcal{B}})_{j,k} &= \varphi_k(x_j) && \text{for} && 1 \le k \le M,\ x_j \in \partial\Omega, \\ (\Lambda)_{k,k} &= \lambda_k && \text{for} && 1 \le k \le M, \\ (\boldsymbol{f}_{\mathcal{L}})_j &= f(x_j) && \text{for} && x_j \in \Omega, \\ (\boldsymbol{g}_{\mathcal{B}})_j &= g(x_j) && \text{for} && x_j \in \partial\Omega. \end{aligned}$$

This system is still as ill-conditioned as the $\Lambda$ matrix, so we redefine the system as

$$\begin{pmatrix} \mathcal{L}\Phi_{\mathcal{L}} \\ \mathcal{B}\Phi_{\mathcal{B}} \end{pmatrix} \tilde{\boldsymbol{a}} = \begin{pmatrix} \boldsymbol{f}_{\mathcal{L}} \\ \boldsymbol{g}_{\mathcal{B}} \end{pmatrix}, \tag{11}$$

with

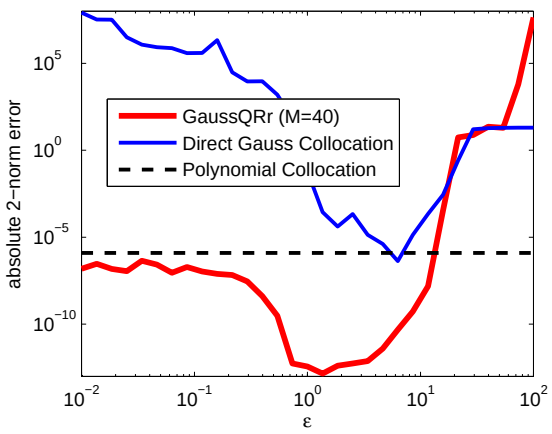
$$\Lambda \begin{pmatrix} \Phi_{\mathcal{L}}^T & \Phi_{\mathcal{B}}^T \end{pmatrix} \boldsymbol{a} = \tilde{\boldsymbol{a}}.$$

This allows us to avoid inverting $\Lambda$, as long as we work in the new basis $\{\varphi_k\}_{k=1}^M$, which is just the first $M$ eigenfunctions.

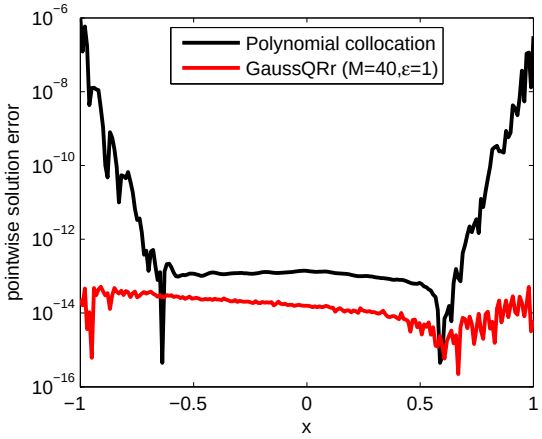
Because (11) is a system of $N$ equations in $M < N$ unknowns, there is likely no consistent solution. Instead, $\tilde{\boldsymbol{a}}$ must be determined in a least squares sense. We have named this low-rank solution method **GaussQRr** because a regression system is solved instead of a square system. This method is tested on the BVP

$$u_{xx}(x) = -9\pi^2 \sin(3\pi x) - \pi^2 \cos(\pi x), \quad x \in (-1, 1), \tag{12a}$$
$$u(x) = \sin(3\pi x) + \cos(\pi x) + 1, \qquad x \in \{-1, 1\}, \tag{12b}$$

using $N = 80$ collocation points. See Figure 2 to compare this method to the other methods "Polynomial Collocation" and "Direct Gauss Collocation" which we have previously used.




(a) Using GaussQRr, for some $\varepsilon$ values, we can achieve many orders of magnitude more accuracy than with polynomial collocation (of degree $N$).




(b) As is often the case, the polynomial solution error is concentrated at the boundaries. This contrasts with the evenly spread error for GaussQRr.

Figure 2: The GaussQRr method is an effective approach to solving the BVP (12a) for small $\varepsilon$. Parameter values $\alpha = 1$ and $M = .5N = 40$ were used for these experiments. Error is computed at 200 evenly spaced points in the domain.

In Figure 2a, we can see again that the Gaussian collocation solution computed in the Gaussian basis becomes ill-conditioned very quickly, preventing it from reaching its optimal accuracy. The GaussQRr method, performed here with $M = 40$, can find solutions with many orders of magnitude more accuracy than any directly computed solution. The "Polynomial Collocation" solution is displayed only for reference; because $M < N$, we no longer expect the limit of the GaussQRr solution to match the degree $N$ polynomial result. Additionally, we cannot trust solutions of GaussQRr for large values of $\varepsilon$ because the eigenvalues (6b) decay less quickly and our truncation assumption becomes less valid.

One of the positive outcomes of the GaussQRr solution is that the error is more evenly distributed throughout the domain. Figure 2 shows that the $\varepsilon = 1$ GaussQRr pointwise error at all $x \in [-1, 1]$ is roughly $\mathcal{O}(10^{-14})$, in contrast to the "Polynomial collocation" pointwise error which is significantly greater near the boundaries. The effect of point distribution will not be discussed here as that is a much too complicated topic; studies on this include [15] for interpolation and [38, 45] for PDEs. We note only that the points chosen here tend to be clustered near the boundary, as suggested in [50] for the polynomial collocation technique.

## 2.4 A nonlinear time stepping example

Thus far we have presented only linear examples, but the Gaussian eigenfunction expansion can also be exploited for nonlinear problems. When choosing $M > N$, this yields a nonlinear system of $N$ equations, and when $M < N$, this yields a nonlinear least squares problem in $M$ unknowns. We will consider an example using GaussQRr in this section.

The linear critical gradient equation [29] is a simplified model of the transport process within a magnetic confinement reactor. It can be written in 1D as

$$u_t - (\kappa(u_x)u_x)_x = f, \qquad x \in (-1,1),\ t > 0 \tag{13a}$$
$$u = g, \qquad x \in \{-1,1\} \tag{13b}$$
$$u = u_0, \qquad t = 0 \tag{13c}$$

where the diffusivity $\kappa$ is a function of the derivative $u_x$

$$\kappa(u_x) = \frac{\mu}{2\tau}\log(\cosh(2\tau u_x) + \cosh(2\tau C)) - \mu C + \frac{\mu - 2}{2\tau}\log(2) + \kappa_0 - B.$$

The parameters appearing in the diffusivity $\kappa$ determine the nonlinearity in the problem:

- $\mu$ - The steepness of the nonlinearity,
- $\tau$ - The severity of the change between constant and nonlinear diffusivity,
- $C$ - For $|u_x| \ll zC$ the diffusivity is basically constant,
- $\kappa_0$ - The minimum diffusivity, and
- $B$ - An integration constant to assure $\kappa(0) = \kappa_0$.

All experiments here will use the parameter values

$$\mu = 10, \qquad \tau = 1, \qquad C = .5, \qquad \kappa_0 = 1.$$

In a plasma physics setting, the source term $f(x) = \mathrm{e}^{-x}$ would be used to cause a pedestal to form at the magnetic separatrix. While this problem is useful for modeling magnetic confinement fusion, it is less useful for studying the accuracy of the numerical scheme because there is no analytic solution for that source. Instead, we will a solution which has a pedestal-like shape,

$$u(x,t) = \mathrm{erf}(4(1 - \mathrm{e}^{-t})x) + 1,$$

which also defines the functions

$$g(x,t) = \mathrm{erf}(4(1 - \mathrm{e}^{-t})x) + 1,$$
$$u_0(x) = 1.$$

For the GaussQRr approximation, we will require our solution to take the form

$$\hat{u}(x,t) = \sum_{k=1}^{M} a_k(t)\varphi_k(x) = (\varphi_1(x)\ \cdots\ \varphi_M(x)) \begin{pmatrix} a_1(t) \\ \vdots \\ a_M(t) \end{pmatrix} = \boldsymbol{\phi}(x)^T \boldsymbol{a}(t).$$

We choose $N-2$ collocation points are on the interior, and require $x_{N-1} = -1$ and $x_N = 1$ to satisfy the boundary conditions. This can now be substituted back into (13a) to yield the system of nonlinear ODEs

$$\boldsymbol{\phi}(x_j)^T \boldsymbol{a}_t(t) - \boldsymbol{\phi}_{xx}(x_j)^T \boldsymbol{a}(t) \left[\kappa'\left(\boldsymbol{\phi}_x(x_j)^T \boldsymbol{a}(t)\right) \boldsymbol{\phi}_x(x_j)^T \boldsymbol{a}(t) + \kappa\left(\boldsymbol{\phi}_x(x_j)^T \boldsymbol{a}(t)\right)\right] = f(x_j,t),$$

for $1 \le j \le N-2$. At this point, we are no longer writing the problem in its conservative form; this is hardly a problem though, since by using Gaussians we have already assumed that the solution is in the Gaussian native space, and thus has enough smoothness to justify the second derivative. Adding in the 2 equations from the boundary conditions (13c),

$$\boldsymbol{\phi}(x_j)^T \boldsymbol{a}(t) - g(x_j,t) = 0,$$

for $j = N-1, N$ gives a system of $N$ differential algebraic equations.

We choose here to discretize in time using the backwards Euler formula, although this choice is made more for simplicity than for any computational benefit. This leaves us with the nonlinear system of equations

$$\boldsymbol{\phi}(x_j)^T \left[\frac{\boldsymbol{a}^n - \boldsymbol{a}^{n-1}}{\Delta t}\right] - \boldsymbol{\phi}_{xx}(x_j)^T \boldsymbol{a}^n \left[\kappa'\left(\boldsymbol{\phi}_x(x_j)^T \boldsymbol{a}^n\right) \boldsymbol{\phi}_x(x_j)^T \boldsymbol{a}^n + \kappa\left(\boldsymbol{\phi}_x(x_j)^T \boldsymbol{a}^n\right)\right] - f(x_j, t_n) = 0$$

$$\boldsymbol{\phi}(x_j)^T \boldsymbol{a}^n - g(x_j, t) = 0 \qquad (14)$$

where, at each time step $t_n$, the solution is $\boldsymbol{a}^n$. The initial condition $\boldsymbol{a}^0$ is computed by solving the GaussQRr approximation problem

$$\begin{pmatrix} \boldsymbol{\phi}(x_1)^T \\ \vdots \\ \boldsymbol{\phi}(x_N)^T \end{pmatrix} \boldsymbol{a}^0 = \begin{pmatrix} u_0(x_1, 0) \\ \vdots \\ u_0(x_N, 0) \end{pmatrix}.$$

At each time step $t_k$ we need to solve a nonlinear least squares problem with $N$ equations and $M$ unknowns, the $\boldsymbol{a}(t_k)$. For the initial guess at each time step, we solve the system (14) with $\kappa \equiv 1$, which reduces the problem to a linear least squares system. Error results are displayed in Figure 3.

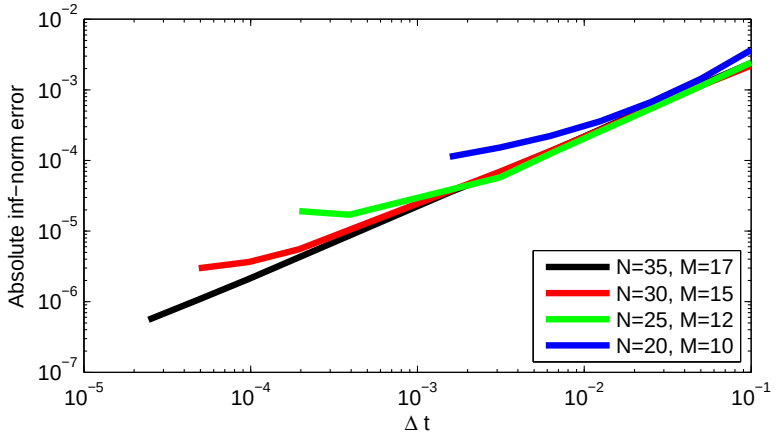



Figure 3: The error in the time stepping is bounded either by the $\mathcal{O}(\Delta t)$ error of the Euler discretization, or the GaussQRr accuracy. When the solution levels off the collocation error has become the dominant term. For all experiments, GaussQRr used the parameters $M = .5N$, $\varepsilon = 10^{-2}$, $\alpha = 1$. Collocation points are evenly spaced in the domain, and the error is computed at the collocation points at $t = .5$.

These experiments confirm that, at least for this example, the separation of spatial and temporal discretizations is appropriate. This so-called method of lines approach has not affected the accuracy of the backward Euler method, which converges with its standard order $\mathcal{O}(\Delta t)$. The convergence terminates when the error introduced by the spatial discretization dominates, which occurs for increasingly accurate solutions as $N$ is increased. Moreover, the GaussQRr solver appears to maintain its spectral convergence, subject to the accuracy bound imposed by the time stepping. Obviously, we have only tested it here for relatively small $N$, so further study will be needed for more complicated time dependent problems. It will also be useful to consider problems involving $M > N$, where the GaussQR collocation technique results in square nonlinear systems at each time step.

## 2.5 Solving problems with a differentiation matrix

The examples up until now have only solved problems in one spatial dimension, but with only minor notational corrections these techniques are valid in arbitrary dimensions. Various technical considerations for moving to higher dimensions are discussed in [11]. In this paper, the only significant change is the change in the definition of eigenfunctions from their 1D form to their tensor product form. This means that the kernel

in $\mathbb{R}^d$ would now take the form

$$e^{-\varepsilon^2\|\boldsymbol{x}-\boldsymbol{z}\|^2} = \sum_{k=1}^{M} \lambda_{\boldsymbol{m}_k} \varphi_{\boldsymbol{m}_k}(\boldsymbol{x}) \varphi_{\boldsymbol{m}_k}(\boldsymbol{z})$$

where $\boldsymbol{x}, \boldsymbol{z}$ are $d$ dimensional vectors and $\boldsymbol{m}_k$ is a $d$-term multiindex stating the order of the eigenfunctions in each dimension. The $\mathbb{R}^d$ eigenfunctions and eigenvalues are defined as

$$\varphi_{\boldsymbol{m}_k}(\boldsymbol{x}) = \prod_{j=1}^{d} \varphi_{(\boldsymbol{m}_k)_j}((\boldsymbol{x})_j).$$

Given this small change in notation, all the previous definitions carry over naturally to higher dimensions; examples using this solution approach will be discussed in Section 3. This flexibility in higher dimensions is one of the great benefits of working with meshfree kernel-based methods, but it does not necessarily mean that this is the optimal way of solving BVP in multiple spatial dimensions using Gaussian eigenfunctions. When presented with a suitably simple domain, it may be computationally efficient to choose points on a structured grid. This will allow for 1D differentiation matrices to be combined to approximate higher dimensional differentiation matrices.

This idea was discussed in [50] for polynomial collocation, where it is especially useful because polynomial interpolation in 1D is better defined than in higher dimensions. In [10] this approach was extended to RBF-based collocation methods. The use of differentiation matrices for GaussQR approximation was developed in [39].

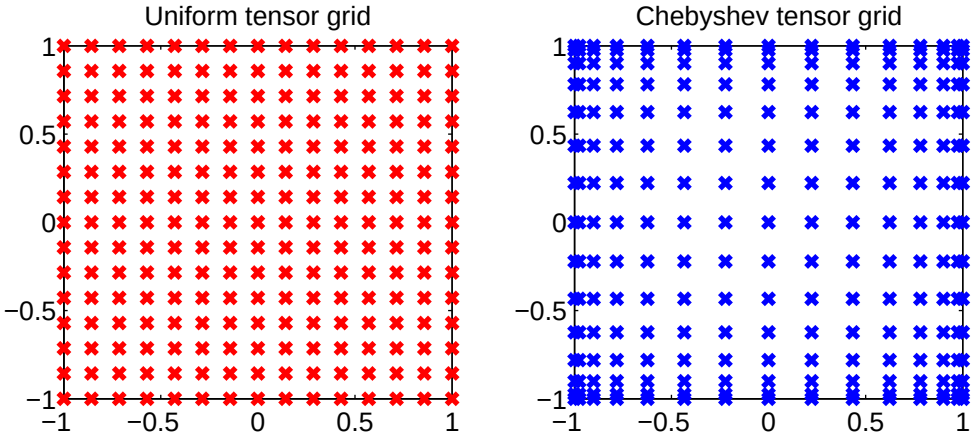



Figure 4: These 2D grids are actually structured copies of 1D grids. For any fixed $x$ (or $y$) the distribution of $y$ (or $x$) points is identical.

Two representative structured grids in 2D are displayed in Figure 4, although for this paper we will consider only the Chebyshev tensor product grid in $(x, y)$. These grids use $N^2$ points, with each "strip" of points containing $N$ points. We can take advantage of the structure of these grids by noting that each vertical strip of points contains the same ordering of $y$ values with the $x$ value constant; this allows the same differentiation matrix to apply on each vertical strip. A similar statement can be made for each horizontal strip of points.

Assume that we have a differentiation matrix $\mathsf{D}$ which applies the differential operator $\mathcal{D}$ to a vector of values evaluated at $x_1, \ldots, x_N$. By ordering the function values $u(x, y)$ in the vector

$$\boldsymbol{u}^T = (\ u(x_1, y_1)\ \cdots\ u(x_1, y_N)\ u(x_2, y_1)\ \cdots\ u(x_2, y_N)\ \cdots\ \cdots u(x_N, y_1)\ \cdots\ u(x_N, y_N)\ ),$$

the differentiation matrix $\mathsf{D}$ can be applied in the $x$ direction on the 2D grid with the matrix vector product

$$(\mathsf{I}_N \otimes \mathsf{D})\boldsymbol{u}.$$

Here $\otimes$ represents the Kronecker tensor product [51]. We can obtain a similar result in the $y$ direction with the product

$$(\mathsf{D} \otimes \mathsf{I}_N)\boldsymbol{u}.$$

If we were to construct a second derivative operator $\mathsf{D}$ on $N$ 1D Chebyshev nodes, the Laplacian on the $N^2$ 2D Chebyshev tensor grid would take the form

$$\mathsf{I}_N \otimes \mathsf{D} + \mathsf{D} \otimes \mathsf{I}_N.$$

By replacing rows associated with boundary values of $(x, y)$ with the associated boundary operator, we may solve boundary value problems with this differentiation matrix approach. As an example, we solve the Helmholtz problem

$$\nabla^2 u(x,y) + \nu^2 u(x,y) = f(x,y), \qquad -1 < x < 1, -1 < y < 1 \tag{15a}$$

$$u(x,y) = g(x,y), \qquad |x| = 1 \cup |y| = 1 \tag{15b}$$

using $\nu = 7$. The true solution is chosen to be $u(x,y) = J_0(6\sqrt{x^2+y^2})$, which necessitates that $f(x,y) = 13J_0(6\sqrt{x^2+y^2})$. Results are compared with $N = 20$ between tensor grid differentiation matrix solutions computed using polynomials (labeled "Trefethen"), the standard Gaussian basis (labeled "Fasshauer") and the stable basis (labeled "GaussQR"). The error is plotted as a function of the shape parameter in Figure 5.

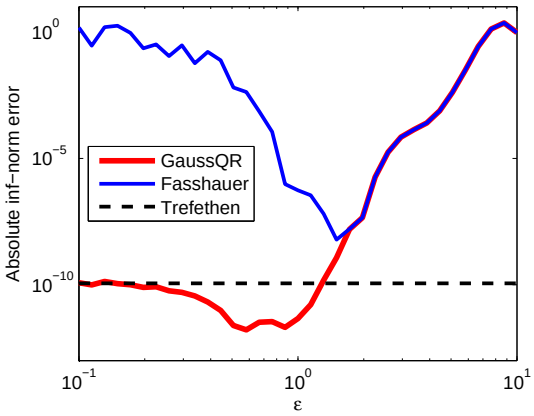



Figure 5: Polynomial (Trefethen), direct Gaussian (Fasshauer) and GaussQR differentiation matrices are tested for solving (15a). As was true for the 1D problems, the standard Gaussian collocation method fails in 2D for small $\varepsilon$, while the GaussQR method allows the solution to reach its $\varepsilon \to 0$ polynomial limit. $N^2 = 400$ collocation points are placed in the domain. For GaussQR, $\alpha = 1$ was used. The error is computed at the collocation points.

We can see here that, by using the Kronecker product on tensor style grids in multiple space dimensions, we can effectively implement the GaussQR method for BVP without being required to use GaussQRr, as was necessary for the interpolation examples in [11]. The ill-conditioning which would otherwise prevent this solution technique from its optimal accuracy is no longer a problem, and the computational cost is comparable to the polynomial collocation method. Because the differentiation matrix is only of size $N$, but the BVP linear system is of size $N^2$, there is significantly less cost in using RBF-QR to compute the differentiation matrix than in solving the full system.

# 3 The method of particular solutions using Gaussian eigenfunctions

When solving boundary value problems, it is often advantageous to transfer the problem to the boundary; the boundary is of lower dimension and requires less work to discretize, and irregularly shaped domains are

less of a problem. The actual mechanism by which this is done can take multiple forms. Boundary element methods [23] (also called boundary integral methods [37, 3]) involve solving a related integral equation on the boundary, rather than a PDE on the domain.

Another approach, called the method of particular solutions (MPS), finds a function which satisfies the interior condition and then solves a simpler approximation problem only on the boundary. The solution on the interior is often called a particular solution, and it can be used in conjunction with the boundary element method to form the dual reciprocity method [41]. This section will consider the applicability of the Gaussian eigenfunction expansion, and their associated stability for small $\varepsilon$, in finding particular solutions to boundary value problems.

## 3.1 The method of fundamental solutions

The method of fundamental solutions (MFS) is a powerful technique for solving homogeneous problems (i.e., with $f(\boldsymbol{x}) = 0$) with a linear operator $\mathcal{L}$ whose fundamental solution $G(x, z)$ is known. Its development is detailed in [8, 22]. We will briefly cover some of that material here.

Essentially, MFS converts a boundary value problem to an interpolation problem. We assume that the problem of interest fits the form

$$\mathcal{L}u(x) = 0, \qquad x \in \Omega, \tag{16a}$$
$$\mathcal{B}u(x) = g, \qquad x \in \partial\Omega. \tag{16b}$$

The fundamental solution is a kernel which satisfies

$$\mathcal{L}G(x, z) = \delta(x, z),$$

where $\delta(x, z)$ is the Dirac delta function. We know that $\mathcal{L}G(x, z) = 0$ for $x \in \Omega$ if $z \notin \Omega$, because $\delta(x, z) = 0$ for $x \neq z$. The assumption is therefore made that the solution $u$ is of the form

$$u(x) = \sum_{k=1}^{N} a_k G(x, z_k) \tag{17}$$

where the $N$ kernel centers $\{z_k\}_{k=1}^N$ are placed outside $\Omega \cup \partial\Omega$.

Automatically, the condition (16a) is satisfied, meaning the coefficients $\{a_k\}_{k=1}^N$ must be determined by satisfying (16b). This is often accomplished by choosing $N$ collocation points $\{x_k\}_{k=1}^N$ on the boundary, and then solving the linear system

$$\begin{pmatrix} \mathcal{B}G(x_1, z_1) & \cdots & \mathcal{B}G(x_1, z_N) \\ & \vdots & \\ \mathcal{B}G(x_N, z_1) & \cdots & \mathcal{B}G(x_N, z_N) \end{pmatrix} \begin{pmatrix} a_1 \\ \vdots \\ a_N \end{pmatrix} = \begin{pmatrix} g(x_1) \\ \vdots \\ g(x_N) \end{pmatrix}.$$

It should be noted that the choice of $N$ source terms is not required; often it is preferable to choose many fewer source terms than collocation points and solve an overdetermined system. Furthermore, the actual choice of source locations is sometimes also considered a variable in the problem. For simplicity, we will only study problems with a fixed set of $N$ sources.

In the simplest case, when $\mathcal{B} = \mathcal{I}$ (the Dirichlet boundary condition case), this is a kernel-based interpolation problem, using the basis $\{G(\cdot, z_k)\}_{k=1}^N$. More complicated boundary conditions are handled just as easily, and greater accuracy is expected than with a collocation method because of the absence of $\mathcal{L}$. Since $\mathcal{L}$ is a differential operator of higher degree than $\mathcal{B}$, more accuracy is lost when approximating it [52], making any solution involving both operators lower order than a solution involving only $\mathcal{B}$.

To demonstrate the impressive potential of the MFS, we apply it to the BVP

$$\nabla^2 u(x, y) = 0, \qquad 0 \leq x \leq 1,\ 0 \leq y \leq \pi/2,$$
$$u(x, y) = \mathrm{e}^x \cos(y), \qquad x = 0 \cup x = 1 \cup y = 0 \cup y = \pi/2.$$

For comparison, we also solve this problem with the GaussQRr technique derived in Section 2.3, and a fourth order finite difference (FD) scheme [28]. The $N$ MFS collocation points were chosen equally spaced on the

boundary, and the source centers were equally spaced on the circle with radius 2 and center $x = .5$, $y = \pi/4$. The GaussQRr solution used parameters $M = .8N$, $\varepsilon = 10^{-8}$ and $\alpha = 1$, and placed half its collocation points on the 2D tensor product Chebyshev nodes and half on the Halton points [24]. This choice of points allows scattered data throughout the interior of the domain, and well-structured points on the boundary. The results are displayed in Figure 6.

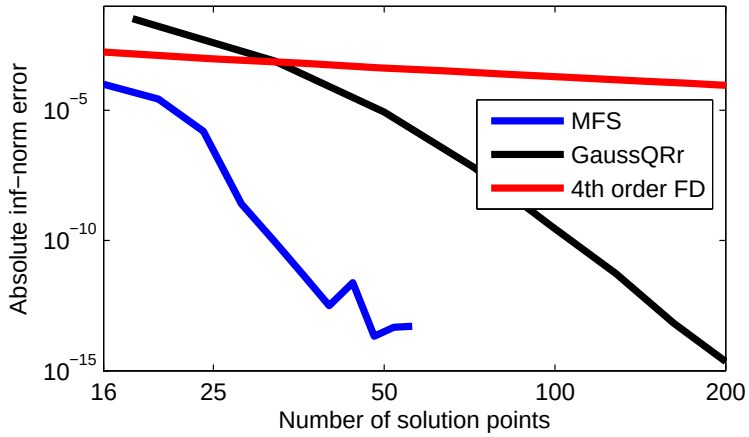



Figure 6: For the Laplace BVP, with Dirichlet boundary conditions, the MFS is vastly superior to finite differences, and even outperforms GaussQRr significantly.

It is clear that both the MFS and GaussQRr solutions are converging exponentially quickly, in contrast to the FD solution which is converging at its expected algebraic order. The MFS solution is much more accurate than the GaussQRr solution for fewer points. Part of this is the fact that GaussQRr places points on the interior and the boundary, and MFS only places points on the boundary because (16a) is satisfied analytically. Another factor contributing to the slightly worse behavior of GaussQRr is the presence of $\mathcal{L}$ in the system, requiring higher order derivatives which are approximated with less accuracy. These factors combined suggest that for sufficiently simple problems, the method of fundamental solutions is still the king.

## 3.2 Finding particular solutions with GaussQRr

It is unsurprising that the method of fundamental solutions is more efficient than GaussQRr collocation, because it has the advantage of considering a solution only on the boundary. Unfortunately, the method of fundamental solutions is only applicable on homogeneous problems. To counteract this shortcoming, the method of particular solutions (MPS) was developed to allow for an inhomogeneous differential equation [40]. Recently, the method of particular solutions has been reconsidered and improved for solving eigenvalue problems on polygonal domains [4]; here we will only consider MPS for boundary value problems.

In the MPS setting, the general BVP will take the general form

$$\begin{aligned} \mathcal{L}u(\boldsymbol{x}) &= f(\boldsymbol{x}), \qquad \boldsymbol{x} \in \Omega, \\ \mathcal{B}u(\boldsymbol{x}) &= g(\boldsymbol{x}), \qquad \boldsymbol{x} \in \partial\Omega, \end{aligned}$$

as was the case in Section 2; we will assume, as we did in Section 3.1 that the operator $\mathcal{L}$ has the Green's function $G(x, z)$. For the MFS setting, $f \equiv 0$, meaning that the solution could be built with the basis $\{G(x, z_k)\}_{k=1}^{N_F}$, but now that $f \neq 0$, we assume the solution takes the form

$$u(\boldsymbol{x}) = u_F(\boldsymbol{x}) + u_P(\boldsymbol{x}).$$

The two components now solve different problems:

- $u_P(\boldsymbol{x})$ solves the ill-posed BVP $\mathcal{L}u_P(\boldsymbol{x}) = f(\boldsymbol{x})$. If collocation with the basis $\{K(x, z_k)\}_{k=1}^{N_P}$ is used to solve this problem, this can be thought of as an approximation problem on the interior, using the basis $\{\mathcal{L}K(x, z_k)\}_{k=1}^{N_P}$.

- $u_F(\boldsymbol{x})$ requires the particular solution, and solves the BVP

$$\begin{aligned} \mathcal{L}u_F(\boldsymbol{x}) &= 0 && \boldsymbol{x} \in \Omega \\ \mathcal{B}u_F(\boldsymbol{x}) &= g(\boldsymbol{x}) - \mathcal{B}u_P(\boldsymbol{x}) && \boldsymbol{x} \in \partial\Omega \end{aligned}$$

  using MFS. This too is an approximation problem, only on the boundary, using the basis $\{G(x, z_k)\}_{k=1}^{N_F}$.

Because of the generally exceptional performance of the method of fundamental solutions, the main source of error for MPS is the approximation of the particular solution. This is a problem which may be remedied somewhat by the use of GaussQRr to find a particular solution, because one major source of error (the ill-conditioning for many values of $\varepsilon$) can be countered effectively. If we approximate the particular solution with Gaussian eigenfunctions,

$$u_P(\boldsymbol{x}) = \sum_{k=1}^{M} b_k \varphi_{\boldsymbol{m}_k}(\boldsymbol{x})$$

we can find the coefficients $\{b_k\}_{k=1}^{M}$ by choosing $N_P$ points $\{\boldsymbol{x}_k\}_{k=1}^{N_P} \in \Omega$ and solving the approximation problem

$$\begin{pmatrix} \mathcal{L}\varphi_{\boldsymbol{m}_1}(\boldsymbol{x}_1) & \cdots & \mathcal{L}\varphi_{\boldsymbol{m}_M}(\boldsymbol{x}_1) \\ & \vdots & \\ \mathcal{L}\varphi_{\boldsymbol{m}_1}(\boldsymbol{x}_{N_P}) & \cdots & \mathcal{L}\varphi_{\boldsymbol{m}_M}(\boldsymbol{x}_{N_P}) \end{pmatrix} \begin{pmatrix} b_1 \\ \vdots \\ b_M \end{pmatrix} = \begin{pmatrix} f(\boldsymbol{x}_1) \\ \vdots \\ f(\boldsymbol{x}_{N_P}) \end{pmatrix}. \tag{18}$$

We can then determine the fundamental solution (of the form (17)) by choosing $N_F$ collocation points $\{\hat{\boldsymbol{x}}_k\}_{k=1}^{N_F} \in \partial\Omega$, $N_F$ source points $\{\boldsymbol{z}_k\}_{k=1}^{N_F} \notin \Omega \cup \partial\Omega$, and solving the linear system

$$\begin{pmatrix} \mathcal{B}G(\hat{\boldsymbol{x}}_1, \boldsymbol{z}_1) & \cdots & \mathcal{B}G(\hat{\boldsymbol{x}}_1, \boldsymbol{z}_{N_F}) \\ & \vdots & \\ \mathcal{B}G(\hat{\boldsymbol{x}}_{N_F}, \boldsymbol{z}_1) & \cdots & \mathcal{B}G(\hat{\boldsymbol{x}}_{N_F}, \boldsymbol{z}_{N_F}) \end{pmatrix} \begin{pmatrix} a_1 \\ \vdots \\ a_{N_F} \end{pmatrix} = \begin{pmatrix} g(\hat{\boldsymbol{x}}_1) \\ \vdots \\ g(\hat{\boldsymbol{x}}_{N_F}) \end{pmatrix} - \begin{pmatrix} \varphi_{\boldsymbol{m}_1}(\hat{\boldsymbol{x}}_1) & \cdots & \varphi_{\boldsymbol{m}_M}(\hat{\boldsymbol{x}}_1) \\ & \vdots & \\ \varphi_{\boldsymbol{m}_1}(\hat{\boldsymbol{x}}_{N_F}) & \cdots & \varphi_{\boldsymbol{m}_M}(\hat{\boldsymbol{x}}_{N_F}) \end{pmatrix} \begin{pmatrix} b_1 \\ \vdots \\ b_M \end{pmatrix} \tag{19}$$

given the previously determined $\{b_k\}_{k=1}^{M}$.

To demonstrate the viability of this method, we will demonstrate it on the modified Helmholtz problem

$$\nabla^2 u(x,y) - \nu^2 u(x,y) = f(x,y), \qquad -1 < x < 1,\ -1 < y < 1 \tag{20a}$$

$$u(x,y) = g(x,y), \qquad |x| = 1 \cup |y| = 1 \tag{20b}$$

using $\nu = 3$ and true solution $u(x,y) = \mathrm{e}^{x+y}$. The fundamental solution for the operator $\mathcal{L} = \nabla^2 - \nu^2\mathcal{I}$ in $\mathbb{R}^2$ is

$$G(\boldsymbol{x}, \boldsymbol{z}) = \frac{1}{2\pi} K_0(\nu \|\boldsymbol{x} - \boldsymbol{z}\|),$$

where $K_0$ is the modified Bessel function of the second kind of order 0. For this example, we will use $\nu = 3$, and compare the solution using GaussQRr approximate collocation to MPS using a GaussQRr generated particular solution.

The MPS solution will use $N_F$ uniformly distributed points on the boundary for the MFS component, and $N_P \approx N_F$ Halton points on the interior for the GaussQRr particular solution. Source points will be placed quasi-uniformly at a distance $\sim 1/\nu^2$ orthogonally away from the boundary. The GaussQRr collocation solution will use the same $N_P$ points on the interior, and $N_B \approx .25N_F$ points uniformly on the boundary. GaussQRr, for both the particular solution approximation and the collocation, will use the parameters $M = .5N_P$, $\varepsilon = 10^{-5}$ and $\alpha = 1$. For both methods, the error will be computed at $35^2$ points uniformly distributed throughout the domain. The results are displayed in Figure 7.

As we can see here, for $N_B < 140$ MPS is at least 10 times more accurate than GaussQRr, although the collocation technique does catch up soon after. Because $N_B \approx .25N_F$, $N_P \approx N_F$, and $M = .5N_P$, both methods have about the same cost:

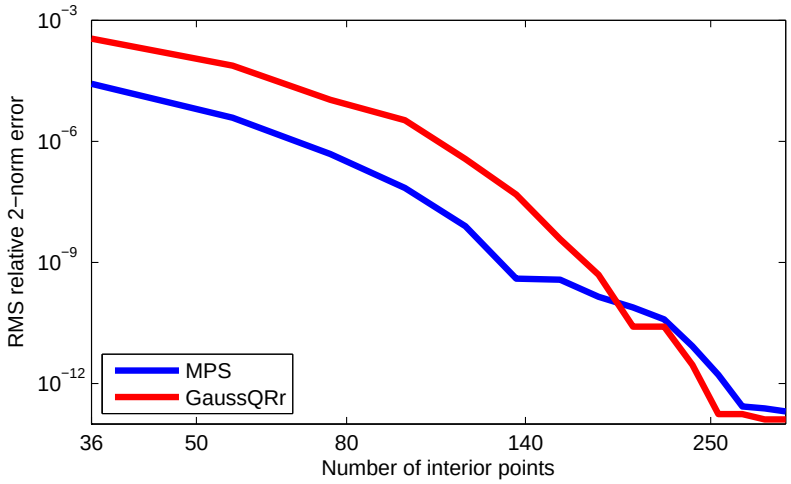



Figure 7: For the problem (20a), MPS using GaussQRr particular solution can be more effective than GaussQRr collocation. The x-axis is meant to represent the cost of the solve, because the cost in both settings is dominated by the interior solution.

- MPS has two costs: $\mathcal{O}(4/3N_P(.5N_P)^2)$ for the least squares solve of the particular solution, and $\mathcal{O}(1/3N_F^3)$. This total cost is roughly $\mathcal{O}(1/3(N_P^3 + N_F^3))$, or $\mathcal{O}(2/3N_P^3)$.
- GaussQRr collocation requires a least squares solve of a system with $N_P + N_B$ rows and $M$ columns. The cost of this is $\mathcal{O}(4/3(N_P + .25N_F)(.5N_P)^2)$ which is roughly $\mathcal{O}(5/12N_P^3)$.

This suggests that Gaussian eigenfunctions can be used to effectively approximate particular solutions, at least for problem as relatively simple as the one we have considered.

### 3.3 Incorporating collocation into the method of particular solutions

It was discussed in [52] that the accuracy of derivatives computed with an RBF interpolant are of a lower order than the interpolant itself; roughly one order of accuracy is lost per derivative taken. This was observed for GaussQRr approximations in [39], and suggests that approximations generated with the basis $\{\mathcal{L}\varphi_{\boldsymbol{m}_k}\}_{k=1}^M$ will be less accurate that those generated with the eigenfunction basis. Because of this, more complicated problems which require more accurate particular solutions may find MPS ineffective.

Collocation remains a viable option here, but it would be shameful to ignore the existence of the Green's functions given the excellent behavior of the method of fundamental solutions on homogeneous problems. Fortunately, it is not necessary to discard the MPS framework, because we can compute particular solutions using collocation. By incorporating boundary conditions into our particular solution, terms involving $\mathcal{B}\varphi_{\boldsymbol{m}_k}$ are included in the linear system, which benefits the accuracy because $\mathcal{B}$ is of lower order than $\mathcal{L}$.

This method will differ slightly from the MPS described in Section 3.2.

- $u_P(\boldsymbol{x})$ solves the BVP

$$\begin{aligned} \mathcal{L}u_P(\boldsymbol{x}) &= f(\boldsymbol{x}), & \boldsymbol{x} &\in \Omega, \\ \mathcal{B}u_P(\boldsymbol{x}) &= g(\boldsymbol{x}), & \boldsymbol{x} &\in \partial\Omega, \end{aligned}$$

  using $\{\boldsymbol{x}_k\}_{k=1}^{N_P} \in \Omega$ to handle the PDE and $\{\hat{\boldsymbol{x}_k}\}_{k=1}^{N_B} \in \partial\Omega$ to handle the BC.

- $u_F(\boldsymbol{x})$ requires the particular solution, and solves the BVP

$$\begin{aligned} \mathcal{L}u_F(\boldsymbol{x}) &= 0, & \boldsymbol{x} &\in \Omega, \\ \mathcal{B}u_F(\boldsymbol{x}) &= g(\boldsymbol{x}) - \mathcal{B}u_P(\boldsymbol{x}), & \boldsymbol{x} &\in \partial\Omega, \end{aligned}$$

using MFS. This is still an approximation problem on the boundary using the basis $\{G(\boldsymbol{x}, \boldsymbol{z}_k)\}_{k=1}^{N_F}$ and the collocation points $\{\tilde{\boldsymbol{x}}_k{}_{k=1}^{N_F}\} \in \partial\Omega$.

The difference with the earlier MPS is that the points $\tilde{\boldsymbol{x}}_k$ must be chosen differently than the points $\hat{\boldsymbol{x}}_k$, i.e., $\tilde{\boldsymbol{x}}_k \neq \hat{\boldsymbol{x}}_j$ for $1 \leq k \leq N_F$ and $1 \leq j \leq N_B$. If the boundary points were chosen the same for both the collocation and MFS, then the MFS would be tricked into believing $g(\boldsymbol{x}) - u_P(\boldsymbol{x}) = 0$ everywhere because the collocation would have already satisfied $g(\hat{\boldsymbol{x}}) = u_P(\hat{\boldsymbol{x}})$.

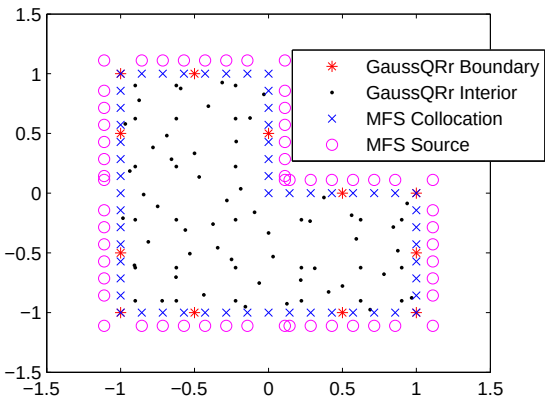
To test this method, we'll consider a more difficult problem than our previous MPS test. The BVP will now have mixed boundary conditions

$$\nabla^2 u(x,y) - \nu^2 u(x,y) = f(x,y), \qquad (x,y) \in \Omega, \tag{21a}$$
$$u(x,y) = g_D(x,y), \qquad (x,y) \in \Gamma_D, \tag{21b}$$
$$\frac{\partial}{\partial \boldsymbol{n}} u(x,y) = g_N(x,y), \qquad (x,y) \in \Gamma_N, \tag{21c}$$

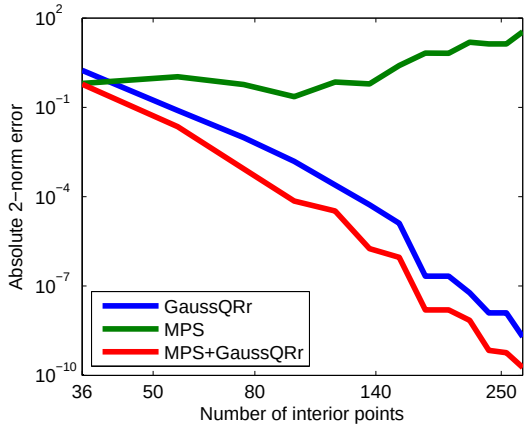
on the L-shaped geometry

$$\Omega = \{x \in (-1,1), y \in (-1,1) \mid x < 0 \cup y < 0\},$$
$$\Gamma_D = \{x \in [-1,1], y \in [-1,1] \mid x = -1 \cup (x = 0 \cap y > 0) \cup (x > 0 \cap y = 0)\},$$
$$\Gamma_N = \{x \in [-1,1], y \in [-1,1] \mid y = -1\}.$$

The setup of the problem, and the solution results are found in Figure 8.




(a) GaussQRr uses $N_P$ Halton points on the interior, and a roughly uniform set of $N_B \approx .2N_P$ points on the boundary. MFS uses $N_F \approx .7N_P$ points uniformly on the boundary and $N_F$ points at a distance $\sim 1/\nu^2$ from the boundary. This sample point distribution was created with $N_P = 76$.




(b) The MPS using the interpolation particular solution falters almost immediately, whereas the GaussQRr solution converges similarly to the earlier example. The introduction of a small number of boundary terms to the particular solution allows for the "MPS+GaussQRr" solution to converge much better than the "MPS", and even better than "GaussQRr".

Figure 8: We have chosen the true solution $u(x,y) = \sin(x^2 + y)$, and modified Helmholtz parameter $\nu = 3$. For this example, the refinement step of performing MFS on the GaussQRr collocation solution provides as much as an extra order of accuracy. GaussQRr techniques used the parameters $M = .5N_P$, $\varepsilon = 10^{-6}$ and $\alpha = 1$.

Figure 8a explains the distribution of collocation and source points chosen for the various solution methods. Three solution techniques are compared in Figure 8b: "MPS" uses GaussQRr interpolation on the interior to generate particular solutions and MFS to enforce the boundary; "GaussQRr" uses GaussQRr collocation from Section 2.3 to solve the full boundary value problem; "MPS+GaussQRr" uses the GaussQRr collocation solution as the particular solution and the MFS to enforce for the boundary terms. The "MPS+GaussQRr" solution is the most effective, and perhaps most noteworthy is that the quality of the

particular solution is so much more accurate after incorporating only a small number of boundary terms. It can safely be assumed that the improvement comes in the particular solution because that is the only difference between the "MPS" and "MPS+GaussQRr" curves.

In some sense, by computing the particular solutions with collocation, we have now shifted the burden of the solution from primarily on the boundary to primarily on the interior. For the traditional MPS, the particular solution is not unique, and the actual solution itself is governed by the MFS component. In this slightly different setting, the solution is first computed with collocation, and then MFS is used as a refinement technique to more effectively incorporate the boundary. Research is needed to determine if the MFS refinement could have a detrimental effect on the final solution, but in this one example, it only helps.

The choice of boundary points seems especially relevant for this setup. Because the GaussQRr method is performing approximate collocation (because $M < N_P + N_B$), it is unlikely that $g(\hat{\boldsymbol{x}}_k) = u_P(\hat{\boldsymbol{x}}_k)$ and therefore even more unlikely that $g(\tilde{\boldsymbol{x}}_k) = u_P(\tilde{\boldsymbol{x}}_k)$. Even so, if the MFS is tricked into thinking that the particular solution is doing a very good job, when in fact it is only doing a good job on a select set of points, then the MFS will not be improving the solution as much as it could. No specific actions were taken here to ensure that the GaussQRr collocation and MFS shared no boundary points, although Figure 8a suggests that at least some of the points did not overlap. In the future, it may be possible to fix the source points $\{\boldsymbol{z}_k\}_{k=1}^{N_S}$ and adaptively choose the MFS points $\{\tilde{\boldsymbol{x}}_k\}_{k=1}^{N_F}$ to account for the locations which collocation least accurately solved by solving an overdetermined system.

# 4 Conclusions and future work

We have presented methods, based on the GaussQR interpolation scheme, for solving boundary value problems. Collocation techniques, drawn from standard kernel-based collocation, proved useful for overcoming the traditional ill-conditioning associated with the flat RBF limit. The GaussQRr interpolation technique was also considered as a method for generating particular solutions within the Method of Particular Solutions. GaussQRr collocation proved even more useful for generating particular solutions, allowing for an accurate solution with a reasonable amount of work.

Looking ahead, we are interested in determining, for the collocation setting, the effect of adding a polynomial basis on the optimal $\varepsilon$ value for the solution. In Section 2.1 we introduced the idea, but dropped it to focus on the GaussQR replacement of direct Gaussian collocation. Given that the $\varepsilon \to 0$ limit produces polynomials, it is not necessary to include a polynomial term in the approximation to reproduce a truly polynomial solution. Even so, if a polynomial term were present, it might change the optimal $\varepsilon$ curve, and potentially also the optimal error that can be achieved.

The same uncertainties which stymie the GaussQR technique in the interpolation setting are present in the solution of boundary value problems. Specifically, the free parameters $\varepsilon$, $\alpha$ and $M$ need to be chosen correctly to take advantage of the potentially optimal accuracy available to kernel methods. Thus far, this work serves only as a proof of concept, and significant research will need to be done to provide good parameter values for general applications. Possible avenues for making informed parameter choices include extending existing statistical methods for determining $\varepsilon$ (such as cross-validation and maximum likelihood estimation) to include $\alpha$ and $M$. It may also be possible to study the parameter choices as $N$ increases, and to run many experiments for smaller $N$ to make a smarter decision for larger $N$.

Computational cost is also of great significance to any practical application, and much work needs to be done to make these methods useful in a high performance environment. The presence of dense matrices, as is often the case in kernel-based methods, is magnified by the need to perform a QR factorization for both GaussQR and GaussQRr. This is mitigated somewhat in the tensor grid setting discussed in Section 2.5, but for those sparse systems, appropriate iterative solvers [6] and preconditioning schemes need to be developed. Work has been done for general RBFs to incorporate tree-code [32] and FMM methods to allow for faster kernel evaluations, and it is likely that applying these methods to the GaussQR framework will improve the computational prospects.

## Acknowledgments

The author would like to thank Gregory Fasshauer for his comments on RBF boundary value problem solvers, C. S. Chen for his help in understanding key aspects of the method of particular solutions, Lois Curfman McInnes and Hong Zhang for their support at Argonne National Laboratory, and Charles Van Loan for his input on solving linear systems. The author would also like to thank Graeme Fairweather for his contributions to the Method of Fundamental Solutions which motivated this work.

## References

[1] H. Adibi and J. Es'haghi. Numerical solution for biharmonic equation using multilevel radial basis functions and domain decomposition methods. *Applied Mathematics and Computation*, 186(1):246 – 255, 2007.

[2] S. N. Atluri and T. Zhu. A new meshless local Petrov-Galerkin (MLPG) approach in computational mechanics. *Computational Mechanics*, 22(2):117–127, August 1998.

[3] J. P. Bardhan, R. S. Eisenberg, and D. Gillespie. Discretization of the induced-charge boundary integral equation. *Phys. Rev. E*, 80:011906, Jul 2009.

[4] T. Betcke and L. N. Trefethen. Reviving the method of particular solutions. *SIAM Review*, 47(3):469 – 491, 2005.

[5] C. S. Chen, S. Lee, and C.-S. Huang. Derivation of particular solutions using Chebyshev polynomial based functions. *Int. J. of Comp. Meth.*, 4(1):15–32, 2007.

[6] S.-C. T. Choi, C. C. Paige, and M. A. Saunders. MINRES-QLP: A Krylov subspace method for indefinite or singular symmetric systems. *SIAM J. Sci. Comput.*, 33(4):1810–1836, August 2011.

[7] T. A. Driscoll and B. Fornberg. Interpolation in the limit of increasingly flat radial basis functions. *Comput. Math. Appl.*, 43(3–5):413–422, 2002.

[8] G. Fairweather and A. Karageorghis. The method of fundamental solutions for elliptic boundary value problems. *Advances in Computational Mathematics*, 9:69–95, 1998.

[9] G. E. Fasshauer. Solving partial differential equations by collocation with radial basis functions. In C. Rabut A. Le M'ehaut'e and L. L. Schumaker, editors, *Surface Fitting and Multiresolution Methods*, pages 131–138. University Press, 1997.

[10] G. E. Fasshauer. *Meshfree Approximation Methods with* MATLAB. World Scientific Publishing Co., Inc., River Edge, NJ, USA, 2007.

[11] G. E. Fasshauer and M. McCourt. Stable evaluation of Gaussian RBF interpolants. *SIAM J. Sci. Comput.*, 34(2):A737–A762, 2012.

[12] A. I. Fedoseyev, M. J. Friedman, and E. J. Kansa. Improved multiquadric method for elliptic partial differential equations via PDE collocation on the boundary. *Computers & Mathematics with Applications*, 43(35):439 – 455, 2002.

[13] N. Flyer and B. Fornberg. Radial basis functions: Developments and applications to planetary scale flows. *Computers & Fluids*, 46(1):23 – 32, 2011. 10th ICFD Conference Series on Numerical Methods for Fluid Dynamics (ICFD 2010).

[14] N. Flyer, E. Lehto, S. Blaise, G. B. Wright, and A. St-Cyr. A guide to RBF-generated finite differences for nonlinear transport: Shallow water simulations on a sphere. *Journal of Computational Physics*, 231(11):4078 – 4095, 2012.

[15] B. Fornberg, T. A. Driscoll, G. Wright, and R. Charles. Observations on the behavior of radial basis function approximations near boundaries. *Computers & Mathematics with Applications*, 43(35):473 – 490, 2002.

[16] B. Fornberg, E. Larsson, and N. Flyer. Stable computations with Gaussian radial basis functions. *SIAM J. Sci. Comput.*, 33(2):869–892, 2011.

[17] B. Fornberg and C. Piret. On choosing a radial basis function and a shape parameter when solving a convective PDE on a sphere. *Journal of Computational Physics*, 227(5):2758 – 2780, 2008.

[18] B. Fornberg and C. Piret. A stable algorithm for flat radial basis functions on a sphere. *SIAM Journal on Scientific Computing*, 30(1):60 – 80, 2008.

[19] B. Fornberg and G. Wright. Stable computation of multiquadric interpolants for all values of the shape parameter. *Computers & Mathematics with Applications*, 48(56):853 – 867, 2004.

[20] E. J. Fuselier and G. B. Wright. A High-Order Kernel Method for Diffusion and Reaction-Diffusion Equations on Surfaces. *eprint arXiv:1206.0047*, May 2012.

[21] M. A. Golberg and C. S. Chen. *Discrete Projection Methods for Integral Equations*. Computational Mechanics Publications, 1997.

[22] M. A. Golberg and C. S. Chen. *The method of fundamental solutions for potential, Helmholtz and diffusion problems*. Computational engineering. Computational Mechanics Publications, WIT Press, 1998.

[23] W. S. Hall. *The Boundary Element Method*. Solid Mechanics and its Applications. Kluwer Academic Publishers, 1994.

[24] J. H. Halton. On the efficiency of certain quasi-random sequences of points in evaluating multi-dimensional integrals. *Numerische Mathematik*, 2:84–90, 1960. 10.1007/BF01386213.

[25] Y. C. Hon. A quasi-radial basis functions method for American options pricing. *Computers & Mathematics with Applications*, 43(35):513 – 524, 2002.

[26] Y. C. Hon and R. Schaback. On unsymmetric collocation by radial basis functions. *Applied Mathematics and Computation*, 119(23):177 – 186, 2001.

[27] M. S. Ingber and N. Phan-Thien. A boundary element approach for parabolic differential equations using a class of particular solutions. *Applied Mathematical Modelling*, 16(3):124 – 132, 1992.

[28] A. Iserles. *A first course in the numerical analysis of differential equations*. Cambridge Texts in Applied Mathematics. Cambridge University Press, 2009.

[29] S. C. Jardin, G. Bateman, G. W. Hammett, and L. P. Ku. On 1D diffusion problems with a gradient-dependent diffusion coefficient. *Journal of Computational Physics*, 227(20):8769 – 8775, 2008.

[30] B. Jumarhon, S. Amini, and K. Chen. The Hermite collocation method using radial basis functions. *Engineering Analysis with Boundary Elements*, 24(78):607 – 611, 2000.

[31] E. J. Kansa. Multiquadrics–A scattered data approximation scheme with applications to computational fluid-dynamics–II solutions to parabolic, hyperbolic and elliptic partial differential equations. *Computers & Mathematics with Applications*, 19(89):147 – 161, 1990.

[32] R. Krasny and L. Wang. Fast evaluation of multiquadric RBF sums by a Cartesian treecode. *SIAM J. Scientific Computing*, 33(5):2341–2355, 2011.

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

[34] E. Larsson and A. Heryudono. A partition of unity radial basis function collocation method for partial differential equations. 2013. in preparation.

[35] Y. J. Lee, G. J. Yoon, and J. Yoon. Convergence of increasingly flat radial basis interpolants to polynomial interpolants. *SIAM Journal on Mathematical Analysis*, 39(2):537–553, 2007.

[36] J. Li and Y. C. Hon. Domain decomposition for radial basis meshless methods. *Numerical Methods for Partial Differential Equations*, 20(3):450–462, 2004.

[37] X. Li and C. Pozrikidis. The effect of surfactants on drop deformation and on the rheology of dilute emulsions in Stokes flow. *Journal of Fluid Mechanics*, 341:165–194, 1997.

[38] Leevan Ling and Manfred R. Trummer. Adaptive multiquadric collocation for boundary layer problems. *Journal of Computational and Applied Mathematics*, 188(2):265 – 282, 2006.

[39] M. McCourt. *Building Infrastructure for Multiphysics Simulations.* PhD thesis, Cornell University, 2012.

[40] A. Miele and R. R. Iyer. General technique for solving nonlinear, two-point boundary-value problems via the method of particular solutions. *Journal of Optimization Theory and Applications*, 5:382–399, 1970. 10.1007/BF00928674.

[41] P. W. Partridge, C. A. Brebbia, and L. C. Wrobel. *The dual reciprocity boundary element method.* International series on computational engineering. Computational Mechanics Publications, 1992.

[42] Maryam Pazouki and Robert Schaback. Bases for kernel-based spaces. *J. Comput. Appl. Math.*, 236(4):575–588, September 2011.

[43] C. E. Rasmussen and C. K. I. Williams. *Gaussian Processes for Machine Learning (Adaptive Computation and Machine Learning).* The MIT Press, 2005.

[44] B. Rodhe. A discontinuous Galerkin method with local radial basis function interpolation. Uptec report f 07 066, School of Engineering, Uppsala Univ., Uppsala, Sweden, 2007, 2007.

[45] S. A. Sarra. A numerical study of the accuracy and stability of symmetric and asymmetric RBF collocation methods for hyperbolic pdes. *Numerical Methods for Partial Differential Equations*, 24(2):670–686, 2008.

[46] R. Schaback. Convergence of unsymmetric kernel-based meshless collocation methods. *SIAM J. Numer. Anal.*, 45:333–351, January 2007.

[47] R. Schaback. Unsymmetric meshless methods for operator equations. *Numerische Mathematik*, 114:629–651, 2010. 10.1007/s00211-009-0265-z.

[48] A. Shokri and M. Dehghan. A Not-a-Knot meshless method using radial basis functions and predictorcorrector scheme to the numerical solution of improved Boussinesq equation. *Computer Physics Communications*, 181(12):1990 – 2000, 2010.

[49] M. L. Stein. *Interpolation of Spatial Data: Some Theory for Kriging.* Springer Series in Statistics. Springer, 1999.

[50] L. N. Trefethen. *Spectral Methods in Matlab.* Software, Environments, Tools. Society for Industrial and Applied Mathematics, 2000.

[51] C. F. Van Loan. The ubiquitous Kronecker product. *Journal of Computational and Applied Mathematics*, 123(12):85 – 100, 2000.

[52] H. Wendland. *Scattered Data Approximation.* Cambridge Monographs on Applied and Computational Mathematics. Cambridge University Press, 2005.