

STABLE INTERPOLATION WITH ISOTROPIC AND ANISOTROPIC GAUSSIANS USING HERMITE GENERATING FUNCTION

KATHARINA KORMANN^{†‡}, CAROLINE LASSER[‡], AND ANNA YUROVA^{†‡}

ABSTRACT. Gaussian kernels can be an efficient and accurate tool for multivariate interpolation. For smooth functions, high accuracies are often achieved near the flat limit where the interpolation matrix becomes increasingly ill-conditioned. Stable evaluation algorithms for isotropic Gaussians (Gaussian radial basis functions) have been proposed based on a Chebyshev expansion of the Gaussians by Fornberg, Larsson & Flyer and based on a Mercer expansion with Hermite polynomials by Fasshauer & McCourt. In this paper, we propose a new stabilization algorithm for the multivariate interpolation with isotropic or anisotropic Gaussians for arbitrary number of dimensions derived from the generating function of the Hermite polynomials. We also derive and analyze a new analytic cut-off criterion for the generating function expansion that allows to automatically adjust the number of the stabilizing basis functions.

1. INTRODUCTION

Multivariate interpolation is a topic that is relevant for a vast number of applications. Gaussian radial basis functions (Gaussian RBFs) are a class of functions for which interpolation generalizes to higher dimensions in a simple way while yielding spectral accuracy [5]. However, it is known that rather small values of the *shape parameter* $\varepsilon > 0$ (the width of the Gaussians) are often required. In this case, the Gaussians become increasingly flat, and the interpolation matrix becomes ill-conditioned. This problem has been extensively studied in the literature (see the review [8] by Fornberg & Flyer and [22] for Tarwater’s description of this phenomenon in 1985). It has been quantified by Fornberg & Zuev [13], that the eigenvalues of the interpolation matrix are proportional to powers of the shape parameter, causing the notorious ill-conditioning in the flat limit regime $\varepsilon \rightarrow 0$.

A direct collocation solution of the interpolation problem, referred to as RBF-Direct in the literature, computes the expansion coefficients of the Gaussian interpolant by inverting the collocation matrix and then evaluating the expansion. Several algorithms have been proposed to stabilize this procedure in the flat limit regime, see [12, 11, 9, 7, 17, 19, 6, 10, 21, 24]. A common idea of many of the stabilization algorithms—including the one proposed in this paper—is to evaluate the interpolant in a sequence of well-conditioned steps by a transformation to a different basis so that the ill-conditioning is isolated in a diagonal matrix that can be inverted analytically.

1.1. The new HermiteGF stabilization approach. In this paper, we propose a stabilizing expansion of isotropic Gaussian functions, later referred to as HermiteGF expansion, built on the exponential generating function of the classic Hermite polynomials. For certain classes of functions, anisotropic Gaussians yield improved accuracy as shown in [2]. To include these cases in our description, we use an anisotropic generating function recently obtained by Dietert, Keller,

Date: October 18, 2019

Funding: This work has been carried out within the framework of the EUROfusion Consortium and has received funding from the Euratom research and training programme 2014-2018 and 2019-2020 under grant agreement No 633053. The views and opinions expressed herein do not necessarily reflect those of the European Commission.

[†]Max Planck Institute for Plasma Physics, Boltzmannstr. 2, 85748 Garching, Germany (katharina.kormann@ipp.mpg.de, anna.yurova@ipp.mpg.de).

[‡]Technical University of Munich, Department of Mathematics, Boltzmannstr. 3, 85748 Garching, Germany (classer@ma.tum.de).

& Troppmann [4] as well as by Hagedorn [15] and generalize our HermiteGF expansion to the anisotropic case. We also propose and analyze a novel cut-off criterion, that contrary to the existing ones does not only account for the diagonal contributions of the stabilization but measures the impact of the full stabilization basis.

1.2. Previous stabilization approaches. The first stabilization method was the Contour–Padé approximation proposed by Fornberg & Wright for multiquadrics [12]. The authors later enhanced it to the RBF-RA stabilization algorithm [24], which has a wide applicability but is limited to a small number of nodes. Later Fornberg & Piret [11] proposed the so-called RBF-QR method for stable interpolation with Gaussians on the sphere by expanding the Gaussians in spherical harmonics. The method has been extended to more general domains in one to three dimensions by Fornberg, Larsson, & Flyer [9]. This expansion is based on a combination of Chebyshev polynomials and spherical harmonics. This method will be referred to as Chebyshev-QR in this paper. The technique has also been used for the stable computation of RBF-generated finite differences by Larsson, Lehto, Heryudono, & Fornberg [17]. Fornberg, Lehto, & Powell [10] developed an alternative stabilization technique for the same problem. To treat complex domains, the Chebyshev-QR method has been combined with a partition-of-unity approach by Larsson, Shcherbakov, & Heryudono [18]. Fasshauer & McCourt [7] have developed another RBF-QR method, called Gauss-QR, that relies on a Mercer expansion of the Gaussian kernel. The basis transformation involves scaled Hermite polynomials. De Marchi and Santin [19] considered a different construction of a stable basis based on a factorization of the kernel matrix for general radial kernels. Our new basis is similar to the one in [7] with the difference that it can be extended to the interpolation with anisotropic Gaussians. Moreover, the generating function framework enables us to derive a new cut-off criterion that accounts for the full Hermite basis effect. We note that a multiscale analysis as provided by Griebel, Rieger, & Zwicknagl [14] might allow to estimate the tail of the Mercer expansion of a Gaussian kernel and subsequently to derive an alternative Mercer series based truncation criterion (see [6, Remark 13.12]).

1.3. Organization of the paper. The paper is organized as follows: In section 2, we introduce our HermiteGF expansion of isotropic and anisotropic Gaussians and derive important properties of the HermiteGF basis. In section 3, we propose a new RBF-QR method based on the HermiteGF basis. A cut-off criterion for the new HermiteGF expansion is derived in section 4. Numerical results show the accuracy of our method in section 5 and finally, section 6 concludes the paper.

2. HERMITEGF EXPANSION

In this section, similarly to the earlier stabilization approaches [11, 9, 7], we propose an expansion of the anisotropic Gaussians in a “better” basis, that spans the same space, but avoids instabilities related to the flat limit. For the sake of simplicity, we first derive the expansion for Gaussians in 1D. We then extend our expansion to the case of multivariate anisotropic Gaussians.

2.1. Interpolation problem. Before introducing our expansion of the Gaussian basis, let us briefly define the interpolation problem. Given a set $\{\phi_k(\cdot)\}_{k=1}^N$ of expansion functions and the values $\mathbf{f} = \{f_i\}_{i=1}^N$ of the function $f : \mathbb{R}^d \rightarrow \mathbb{R}$ at collocation points $\{\mathbf{x}_i^{\text{col}}\}_{i=1}^N$ we seek to find an interpolant of the form,

$$(2.1) \quad s(\mathbf{x}) = \sum_{k=1}^N \alpha_k \phi_k(\mathbf{x}),$$

such that it satisfies the N collocation conditions,

$$s(\mathbf{x}_i^{\text{col}}) = f_i \quad \text{for } i = 1, \dots, N.$$

The direct approach is to find the coefficients $\boldsymbol{\alpha} = \{\alpha_k\}_{k=1}^N$ as a solution of the linear system,

$$(2.2) \quad \Phi^{\text{col}} \boldsymbol{\alpha} = \mathbf{f}, \quad \text{with } \Phi_{ij}^{\text{col}} = \phi_j(\mathbf{x}_i^{\text{col}}).$$

The matrix $\Phi^{\text{col}} \in \mathbb{R}^{N \times N}$ is called *collocation matrix*. Then, after solving the linear system eq. (2.2), the interpolant eq. (2.1) can be evaluated at any point of the domain. In this paper, we consider Gaussian radial basis functions (isotropic Gaussians)

$$\phi_k(\mathbf{x}) = \exp(-\varepsilon^2 \|\mathbf{x} - \mathbf{x}_k\|^2)$$

with center points $X^{\text{cen}} = \{\mathbf{x}_k\}_{k=1}^N$, shape parameter $\varepsilon > 0$ and anisotropic Gaussians

$$\phi_k(\mathbf{x}) = \exp(-(\mathbf{x} - \mathbf{x}_k)^T E^T E (\mathbf{x} - \mathbf{x}_k))$$

with invertible shape matrix $E \in \mathbb{R}^{d \times d}$.

2.2. HermiteGF expansion in 1D. Let $\{h_\ell\}_{\ell \geq 0}$ be the Hermite polynomials in the physicists' version, that satisfy the recurrence relation,

$$h_{\ell+1}(x) = 2xh_\ell(x) - 2\ell h_{\ell-1}(x), \quad h_0(x) = 1, \quad h_{-1}(x) = 0.$$

The following upper bound holds for the magnitude of the Hermite polynomials [1, Expr. 22.14.17]: There exists a positive constant $c \approx 1.086435$ such that

$$|h_\ell(x)| \leq e^{\frac{x^2}{2}} c \sqrt{2^\ell \ell!}$$

for all $\ell \geq 0$ and all $x \in \mathbb{R}$. The combinatorial factors $\sqrt{2^\ell \ell!}$ grow very fast with ℓ . Therefore, in order to avoid overflow for large ℓ , we work with a scaled version of the Hermite polynomials, $\frac{1}{\sqrt{2^\ell \ell!}} h_\ell$. We introduce three parameters,

$$\varepsilon > 0, \quad \gamma > 0, \quad t \in (0, 1),$$

that is, the usual shape parameter $\varepsilon > 0$, a scaling parameter $\gamma > 0$ for varying the evaluation domain of the Hermite polynomials to improve conditioning, and a truncation parameter $t \in (0, 1)$ for controlling the truncation error of the stabilization expansion. We then define the following basis functions,

$$H_\ell^{\gamma, \varepsilon, t}(x) = \frac{t^{\ell/2}}{\sqrt{2^\ell \ell!}} h_\ell(\gamma x) e^{-\varepsilon^2 x^2},$$

that we refer to as *HermiteGF functions*.

Based on the generating function theory, we derive an infinite expansion of the one dimensional Gaussian RBFs in the new HermiteGF basis $\{H_\ell^{\gamma, \varepsilon, t}\}_{\ell \geq 0}$.

Proposition 2.1 (HermiteGF expansion (1D)). *For all parameters $\varepsilon > 0$, $\gamma > 0$, $t \in (0, 1)$, and for all shifts $x_0 \in \mathbb{R}$, we have a pointwise expansion*

$$(2.3) \quad \phi_k(x) = e^{-\varepsilon^2(x-x_k)^2} = \exp\left(\varepsilon^2 \Delta_k^2 \left(\frac{\varepsilon^2}{\gamma^2} - 1\right)\right) \sum_{\ell \geq 0} \frac{\varepsilon^{2\ell} \sqrt{2^\ell}}{\gamma^\ell \sqrt{t^\ell \ell!}} \Delta_k^\ell H_\ell^{\gamma, \varepsilon, t}(x - x_0),$$

for all $k = 1, \dots, N$, where $\Delta_k = x_k - x_0$. The RBF interpolant $s(x)$ can then be pointwise computed as

$$(2.4) \quad s(x) = \sum_{k=1}^N \alpha_k \exp\left(\varepsilon^2 \Delta_k^2 \left(\frac{\varepsilon^2}{\gamma^2} - 1\right)\right) \sum_{\ell \geq 0} \frac{\varepsilon^{2\ell} \sqrt{2^\ell}}{\gamma^\ell \sqrt{t^\ell \ell!}} \Delta_k^\ell H_\ell^{\gamma, \varepsilon, t}(x - x_0).$$

Proof. The Hermite polynomial's generating function is given by (see e.g. [1, Expr. 22.9.17]),

$$e^{2ba - a^2} = \sum_{\ell \geq 0} \frac{a^\ell}{\ell!} h_\ell(b), \quad \forall a, b \in \mathbb{R}.$$

Choosing $a = \frac{\varepsilon^2 \Delta_k}{\gamma}$ and $b = \gamma(x - x_0)$, we obtain

$$(2.5) \quad \sum_{\ell \geq 0} \frac{\varepsilon^{2\ell}}{\gamma^\ell \ell!} \Delta_k^\ell h_\ell(\gamma(x - x_0)) = \exp\left(2\varepsilon^2 \Delta_k(x - x_0) - \frac{\varepsilon^4 \Delta_k^2}{\gamma^2}\right)$$

Hence, we get

$$\begin{aligned} & \exp\left(\varepsilon^2 \Delta_k^2 \left(\frac{\varepsilon^2}{\gamma^2} - 1\right)\right) \sum_{\ell \geq 0} \frac{\varepsilon^{2\ell} \sqrt{2}^\ell}{\gamma^\ell \sqrt{t^\ell \ell!}} \Delta_k^\ell H_\ell^{\gamma, \varepsilon, t}(x - x_0) \\ &= \exp\left(\varepsilon^2 \Delta_k^2 \left(\frac{\varepsilon^2}{\gamma^2} - 1\right)\right) \sum_{\ell \geq 0} \frac{\varepsilon^{2\ell}}{\gamma^\ell \ell!} \Delta_k^\ell h_\ell(\gamma(x - x_0)) e^{-\varepsilon^2(x-x_0)^2} \stackrel{(2.5)}{=} e^{-\varepsilon^2(x-x_k)^2}, \end{aligned}$$

which proves expansion (2.3). Using expansion (2.3) in the interpolant (2.1), we get the representation (2.4). \square

Remark 2.1 (Basis centering). *Hermite polynomials are symmetric with respect to the axis $x = 0$. Due to its growth behavior, it is advantageous to have the basis centered around this point of symmetry, that is, to use the translation $x_0 = \frac{B-A}{2}$, where $[A, B]$ is the interval of interest for evaluating the function f .*

Remark 2.2 (The parameter γ). *The parameter $\gamma > 0$ in the basis $\{H_\ell^{\gamma, \varepsilon, t}\}_{\ell \geq 0}$ allows control over the evaluation domain of the Hermite polynomials. When choosing it, one has to consider two counteracting effects: For small values of γ , ill-conditioning appears since the values of the basis functions at the collocation points are too similar. On the other hand, Hermite polynomials take very large values on large domains which can lead to an overflow. An optimal balance depends on the particular function and the number of basis functions. The parameter plays a similar role to that of the “global scale parameter” α in [7].*

2.3. Multivariate HermiteGF expansion of anisotropic Gaussians. The HermiteGF expansion can be easily extended to higher dimensions for the case of isotropic Gaussians, using tensor products of 1D physicists’ Hermite polynomials,

$$h_\ell(\mathbf{x}) = h_{\ell_1}(x_1) \cdots h_{\ell_d}(x_d), \quad \ell \in \mathbb{N}^d, \quad \mathbf{x} \in \mathbb{R}^d.$$

However, finding a stable interpolant for anisotropic Gaussian functions of the type

$$\phi_{\mathbf{q}}(\mathbf{x}) = \exp(-(\mathbf{x} - \mathbf{q})^T E^T E(\mathbf{x} - \mathbf{q})), \quad \mathbf{x}, \mathbf{q} \in \mathbb{R}^d,$$

is a more challenging task. A similar question was raised in [7, § 8.5], however, without further investigation. McCourt & Fasshauer [20] considered anisotropic Gaussians with diagonal shape matrix E using Mercer expansion theory, but this result has not been extended to the case of arbitrary E . Analogously to the 1D case, we define the multivariate version of our *HermiteGF functions* by

$$H_\ell^{G, E, t}(\mathbf{x}) = \frac{t^{|\ell|/2}}{\sqrt{2^{|\ell|} \ell!}} h_\ell(G^T \mathbf{x}) \exp(-\mathbf{x}^T E^T E \mathbf{x}),$$

where $G, E \in \mathbb{R}^{d \times d}$ are arbitrary invertible matrices.

Proposition 2.2 (HermiteGF expansion of anisotropic Gaussians). *Let $\mathbf{q} \in \mathbb{R}^d$ and $E, G \in \mathbb{R}^{d \times d}$ be invertible matrices. Consider the anisotropic Gaussian $\phi_{\mathbf{q}}(\cdot)$. Then, for any shift $\mathbf{x}_0 \in \mathbb{R}^d$ the following relation holds pointwise in $\mathbf{x} \in \mathbb{R}^d$:*

$$(2.6) \quad \phi_{\mathbf{q}}(\mathbf{x}) = \exp(\Delta_{\mathbf{q}}^T (\tilde{G} - E^T E) \Delta_{\mathbf{q}}) \cdot \sum_{\ell \in \mathbb{N}^d} \frac{(G^{-1} E^T E \Delta_{\mathbf{q}})^\ell \sqrt{2^{|\ell|}}}{\sqrt{t^{|\ell|} \ell!}} H_\ell^{G, E, t}(\mathbf{x} - \mathbf{x}_0),$$

where $\Delta_{\mathbf{q}} = \mathbf{q} - \mathbf{x}_0$, $\tilde{G} = E^T E G^{-T} G^{-1} E^T E$.

Proof. We use the following anisotropic Hermite generating function (see [4, Lemma 5] or [15, Theorem 3.1] with $A = \text{Id}_d$):

$$(2.7) \quad \sum_{\ell \in \mathbb{N}^d} \frac{\mathbf{a}^\ell}{\ell!} h_\ell(\mathbf{b}) = \exp(2\mathbf{b}^T \mathbf{a} - \mathbf{a}^T \mathbf{a}), \quad \forall \mathbf{a}, \mathbf{b} \in \mathbb{R}^d.$$

We denote $\mathbf{b} = G^T(\mathbf{x} - \mathbf{x}_0)$ and $\mathbf{a} = G^{-1}E^T E \Delta_{\mathbf{q}}$. Then, using (2.7), we get

$$\sum_{\ell \in \mathbb{N}^d} \frac{(G^{-1}E^T E \Delta_{\mathbf{q}})^\ell}{\ell!} h_\ell(G^T(\mathbf{x} - \mathbf{x}_0)) = \exp(2(\mathbf{x} - \mathbf{x}_0)^T E^T E \Delta_{\mathbf{q}} - \Delta_{\mathbf{q}}^T \tilde{G} \Delta_{\mathbf{q}}).$$

We observe that

$$\begin{aligned} 2\mathbf{x}^T E^T E \mathbf{q} &= 2(\mathbf{x} - \mathbf{x}_0)^T E^T E \Delta_{\mathbf{q}} + 2\mathbf{x}^T E^T E \mathbf{x}_0 + 2\mathbf{x}_0^T E^T E \Delta_{\mathbf{q}}, \\ -\mathbf{x}^T E^T E \mathbf{x} &= -(\mathbf{x} - \mathbf{x}_0)^T E^T E (\mathbf{x} - \mathbf{x}_0) - 2\mathbf{x}^T E^T E \mathbf{x}_0 + \mathbf{x}_0^T E^T E \mathbf{x}_0. \end{aligned}$$

Putting everything together we arrive to (2.6). \square

This expansion provides a new powerful tool for dealing with anisotropic approximation. Note that the standard multidimensional isotropic Gaussian interpolation corresponds to the following matrix E :

$$E_{\text{isotropic}} = \varepsilon \text{Id}_d.$$

2.4. Mehler's formula (nD). In this section, we investigate the behavior of the HermiteGF basis functions with large index ℓ approaching infinity. To be able to decide where to cut the HermiteGF expansion for numerical purposes, it is useful to quantify the size of the tail of the truncated expansion. For that, we take a look at the magnitude of the values of the HermiteGF basis. We consider the infinite-dimensional vector that contains the values of all basis functions $H_\ell^{G,E,t}(\mathbf{x})$ at a certain point $\mathbf{x} \in \mathbb{R}^d$. Its Euclidean norm can be computed analytically using a multivariate extension of Mehler's formula for Hermite polynomials.

Theorem 2.1 (Bilinear generating function). *The following relation holds for all $t \in (0, 1)$ and all $\mathbf{x}, \mathbf{y} \in \mathbb{R}^d$*

$$(2.8) \quad \sum_{|\ell|=0}^{\infty} \frac{t^{|\ell|} h_\ell(\mathbf{x}) h_\ell(\mathbf{y})}{2^{|\ell|} \ell!} = \frac{\exp\left(\frac{t}{1-t^2}(\mathbf{x}^T \mathbf{y} + \mathbf{y}^T \mathbf{x}) - \frac{t^2}{1-t^2}(\|\mathbf{x}\|_2^2 + \|\mathbf{y}\|_2^2)\right)}{(1-t^2)^{d/2}}.$$

Proof. We extend the 1D proof proposed by Watson in [23, p. 4] to multiple dimensions. Applying the inverse Fourier transform to the Fourier transform of a normal distribution with $\sigma = 1/\sqrt{2}\text{Id}_d$, we get

$$e^{-\mathbf{x}^T \mathbf{x}} = \pi^{d/2} \int_{\mathbb{R}^d} e^{2\pi i \mathbf{x}^T \boldsymbol{\xi}} e^{-\pi^2 \boldsymbol{\xi}^T \boldsymbol{\xi}} d\boldsymbol{\xi} = \pi^{-d/2} \int_{\mathbb{R}^d} e^{2i \mathbf{x}^T \boldsymbol{\xi}} e^{-\boldsymbol{\xi}^T \boldsymbol{\xi}} d\boldsymbol{\xi}.$$

Hence,

$$h_\ell(\mathbf{x}) = e^{\mathbf{x}^T \mathbf{x}} (-\nabla)^\ell e^{-\mathbf{x}^T \mathbf{x}} = \frac{(-2i)^{|\ell|}}{\pi^{d/2}} e^{\mathbf{x}^T \mathbf{x}} \int_{\mathbb{R}^d} \boldsymbol{\xi}^\ell e^{2i \mathbf{x}^T \boldsymbol{\xi}} e^{-\boldsymbol{\xi}^T \boldsymbol{\xi}} d\boldsymbol{\xi},$$

where we used the Rodrigues formula for multivariate tensor-product Hermite polynomials (see [4, Expr. 11] with $M = \text{Id}$). Then,

$$\begin{aligned} (2.9) \quad & \sum_{|\ell|=0}^{\infty} \frac{t^{|\ell|} h_\ell(\mathbf{x}) h_\ell(\mathbf{y})}{2^{|\ell|} \ell!} = \\ & = \frac{e^{\mathbf{x}^T \mathbf{x} + \mathbf{y}^T \mathbf{y}}}{\pi^d} \sum_{|\ell|=0}^{\infty} \frac{(-2t)^{|\ell|}}{\ell!} \int_{\mathbb{R}^d} \int_{\mathbb{R}^d} \boldsymbol{\xi}_x^\ell \boldsymbol{\xi}_y^\ell e^{2i(\mathbf{x}^T \boldsymbol{\xi}_x + \mathbf{y}^T \boldsymbol{\xi}_y)} e^{-\boldsymbol{\xi}_x^T \boldsymbol{\xi}_x - \boldsymbol{\xi}_y^T \boldsymbol{\xi}_y} d\boldsymbol{\xi}_x d\boldsymbol{\xi}_y \\ & = \frac{e^{\mathbf{x}^T \mathbf{x} + \mathbf{y}^T \mathbf{y}}}{\pi^d} \int_{\mathbb{R}^d} \int_{\mathbb{R}^d} e^{-2t \boldsymbol{\xi}_x^T \boldsymbol{\xi}_y} e^{2i(\mathbf{x}^T \boldsymbol{\xi}_x + \mathbf{y}^T \boldsymbol{\xi}_y)} e^{-\boldsymbol{\xi}_x^T \boldsymbol{\xi}_x - \boldsymbol{\xi}_y^T \boldsymbol{\xi}_y} d\boldsymbol{\xi}_x d\boldsymbol{\xi}_y, \end{aligned}$$

where we used the Taylor series of the exponential function. Recall the following formula for a bivariate Gaussian Fourier integral (see [23, p. 4]):

$$(2.10) \quad \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} g_{(x,y)}(u,v) dv du = \frac{\pi e^{-(x^2+y^2)/2}}{\sqrt{1-t^2}} \exp\left(\frac{x^2-y^2}{2} - \frac{(x-yt)^2}{1-t^2}\right).$$

with

$$g_{(x,y)}(u, v) = \exp(-u^2 - 2tuv - v^2 + 2ixu + 2iyv).$$

Using (2.10) d times for the integral (2.9) together with the algebraic identity

$$-\frac{1}{2}(x^2 + y^2) + \frac{1}{2}(x^2 - y^2) - \frac{(x - yt)^2}{1 - t^2} = \frac{2txy - x^2 - y^2}{1 - t^2}$$

yields (2.8). □

With the help of the multidimensional Mehler's formula we can now compute the square of the Euclidean norm

$$H_{\lim}^{G,E,t}(\mathbf{x}) := \sum_{|\ell|=0}^{\infty} H_{\ell}^{G,E,t}(\mathbf{x})^2 = \sum_{|\ell|=0}^{\infty} \frac{t^{|\ell|}}{2^{|\ell|} \ell!} h_{\ell}^2(G^T \mathbf{x}) \exp(-2\mathbf{x}^T E^T E \mathbf{x})$$

of the infinite vector containing the values of all basis functions $H_{\ell}^{G,E,t}(\mathbf{x})$ with $\ell \in \mathbb{N}^d$ at some point $\mathbf{x} \in \mathbb{R}^d$. Indeed,

$$(2.11) \quad H_{\lim}^{G,E,t}(\mathbf{x}) = \frac{\exp\left(-2\mathbf{x}^T E^T E \mathbf{x} + \frac{2t\|G^T \mathbf{x}\|_2^2}{1+t}\right)}{(1-t^2)^{d/2}}.$$

We take a look if this value matches our numerical result. One can see in Figure 1 that for the square domain the limit value is matched for different values of ε , t .

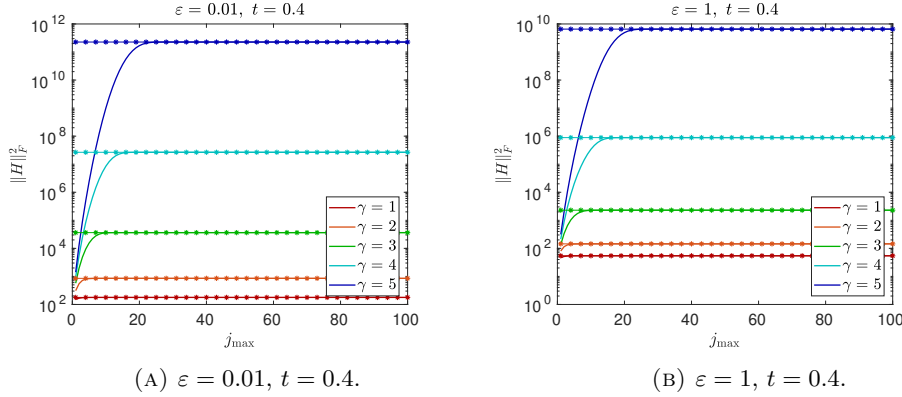


FIGURE 1. Frobenius norm of the matrix H of values of the basis functions at 100 Halton nodes \mathbf{x}_k on a square domain $[-1, 1] \times [-1, 1]$. The solid line is the value corresponding to only the basis functions with a degree up to j_{\max} . The * corresponds to the analytically computed value.

3. STABILIZATION OF THE RBF INTERPOLATION

In this section, we derive a numerical stabilization algorithm for RBF interpolation based on the HermiteGF expansion. The main idea is to represent the RBF interpolant in the more stable basis $\{H_{\ell}^{G,E,t}\}_{\ell \geq 0}$. For appropriately chosen parameters G and t , we expect it to be better conditioned. We now first derive the HermiteGF-QR algorithm in 1D and then generalize it to a multidimensional form.

3.1. **HermiteGF-QR (1D).** In 1D we denote by

$$\Phi(x, X^{\text{cen}}) = (\phi_1(x), \dots, \phi_N(x))$$

the vector of Gaussians with center points X^{cen} evaluated at $x \in \mathbb{R}$ and write the stabilization expansion (2.3) as an infinite matrix-vector product

$$(3.1) \quad \Phi(x, X^{\text{cen}}) = H^{\gamma, \varepsilon, t}(x - x_0) B(\varepsilon, \gamma, t, X^{\text{cen}})$$

where the vector

$$H^{\gamma, \varepsilon, t}(x - x_0) = (H_0^{\gamma, \varepsilon, t}(x - x_0), H_1^{\gamma, \varepsilon, t}(x - x_0), \dots)$$

contains all the elements of the polynomial basis $\{H_\ell^{\gamma, \varepsilon, t}\}_{\ell \geq 0}$ evaluated in the translated point $x - x_0$ and

$$B(\varepsilon, \gamma, t, X^{\text{cen}})_{\ell k} = \exp\left(\varepsilon^2 \Delta_k^2 \left(\frac{\varepsilon^2}{\gamma^2} - 1\right)\right) \frac{\varepsilon^{2\ell} \sqrt{2}^\ell}{\gamma^\ell \sqrt{t}^\ell \ell!} \Delta_k^\ell$$

is an $\infty \times N$ matrix. The major part of the ill-conditioning is now confined in the matrix B . Since B is independent of the point x where the basis function is evaluated, both the evaluation and interpolation matrix can be expressed in the form (3.1) with the same matrix B .

We follow the RBF-QR approach and further split

$$B^T = CD$$

into a well-conditioned full $N \times \infty$ matrix C and an infinite diagonal matrix D , where all harmful effects are confined in D . In the case of expansion (2.3), the following setup follows naturally from the Chebyshev-QR theory [9, § 4.1.3],

$$C_{k\ell} = \exp\left(\varepsilon^2 \Delta_k^2 \left(\frac{\varepsilon^2}{\gamma^2} - 1\right)\right) \frac{\Delta_k^\ell}{L^\ell}, \quad D_{\ell\ell} = \frac{\varepsilon^{2\ell} \sqrt{2}^\ell}{\gamma^\ell \sqrt{t}^\ell \ell!} L^\ell.$$

Here we also divide each coefficient by the radius of the domain L containing the center points in order to avoid ill-conditioning in C coming from taking high powers of x_k . That might be dangerous when the domain is too large, however, it still extends the range of domain diameters possible.

The goal is now to find a basis $\{\psi_j\}$ spanning the same space as $\{\phi_k\}$ but yielding a better conditioned collocation matrix. In particular, we need an invertible matrix X such that $X^{-1}\Phi(x)^T$ is better conditioned. Let us perform a QR-decomposition on $C = QR$. Then we get,

$$\Phi(x)^T = CDH^{\gamma, \varepsilon, t}(x - x_0)^T = Q \begin{pmatrix} R_1 & R_2 \end{pmatrix} \begin{pmatrix} D_1 & 0 \\ 0 & D_2 \end{pmatrix} H^{\gamma, \varepsilon, t}(x - x_0)^T$$

where R_1 and D_1 are $N \times N$ matrices containing the upper (left) block of the infinite matrices R and D , respectively, while R_2 and D_2 assemble the remaining entries. Consider $X = QR_1 D_1$.¹ The new basis $\Psi(\mathbf{x})^T := X^{-1}\Phi(x)^T$ can be formed as,

$$\begin{aligned} \Psi(x)^T &= D_1^{-1} R_1^{-1} Q^H \Phi(x)^T \\ &= D_1^{-1} R_1^{-1} Q^H Q \begin{pmatrix} R_1 D_1 & R_2 D_2 \end{pmatrix} H^{\gamma, \varepsilon, t}(x - x_0)^T \\ &= (\text{Id} \quad D_1^{-1} R_1^{-1} R_2 D_2) H^{\gamma, \varepsilon, t}(x - x_0)^T. \end{aligned}$$

To avoid under/overflow in the computation of $D_1^{-1} R_1^{-1} R_2 D_2$, we form the two matrices $\tilde{R} = R_1^{-1} R_2$ and \tilde{D} with elements

$$\tilde{D}_{i,j} = \left(\frac{\varepsilon^2 L}{\gamma} \sqrt{\frac{2}{t}}\right)^{N+j-i} \sqrt{\frac{i!}{(N+j)!}}, \quad 1 \leq i \leq N, \quad j \geq 1,$$

¹We assume that the matrix X is invertible. If this is not the case, then column pivoting in the QR decomposition has proved to be effective (see [9, § 5]).

and compute their Hadamard product, $\tilde{R} * \tilde{D}$. Despite the harmful effects contained in D , the resulting term $D_1^{-1}R_1^{-1}R_2D_2 = \tilde{R} * \tilde{D}$ is then harmless.

3.2. Multivariate anisotropic HermiteGF-QR. In this section, we derive an analog of the HermiteGF-QR algorithm for the multivariate case. Since we are now dealing with matrices, in the general case, it is impossible to separate E from \mathbf{x}_k in $(E\mathbf{x}_k)^\ell$ as we did before. Therefore, the flow of the HermiteGF-QR does not apply directly. However, it is still possible to tackle some part of the ill-conditioning analytically. Consider the following splitting of the matrix $G^{-1}E^TE \in \mathbb{R}^{d \times d}$:

$$G^{-1}E^TE = \text{Diag} + \text{Rem},$$

where Diag is the $d \times d$ diagonal matrix containing the diagonal elements of $G^{-1}E^TE$ and Rem contains the remaining off-diagonal terms. Denote

$$\mathbf{v}_k = (\text{Id} + \text{Diag}^{-1}\text{Rem})\Delta_k \quad \text{and} \quad \Delta_k = \mathbf{x}_k - \mathbf{x}_0.$$

Then, it holds that

$$\begin{aligned} (G^{-1}E^TE\Delta_k)^\ell &= ((\text{Diag} + \text{Rem})\Delta_k)^\ell = (\text{Diag}((\text{Id} + \text{Diag}^{-1}\text{Rem})(\mathbf{x}_k - \mathbf{x}_0)))^\ell \\ &= \prod_{i=1}^d (\text{Diag}_{ii}(\mathbf{v}_k)_i)^{\ell_i} = \prod_{i=1}^d \text{Diag}_{ii}^{\ell_i}(\mathbf{v}_k)_i^{\ell_i} = \left(\prod_{i=1}^d \text{Diag}_{ii}^{\ell_i} \right) \mathbf{v}_k^\ell, \end{aligned}$$

where $\text{Diag}^{-1}\text{Rem}$ can be computed analytically. Denote $\mathbf{d}_{\text{vec}} = \text{diag}(\text{Diag})$. Then, the generating function expansion (2.6) can be written as:

$$\begin{aligned} \phi_k(\mathbf{x}) &= \exp(-\Delta_k^T E^T E \Delta_k + \Delta_k^T \tilde{G} \Delta_k) \sum_{\ell \in \mathbb{N}^d} \frac{(G^{-1}E^TE\Delta_k)^\ell \sqrt{2^{|\ell|}}}{\sqrt{t^{|\ell|}\ell!}} H_\ell^{G,E,t}(\mathbf{x} - \mathbf{x}_0) \\ &= \exp(-\Delta_k^T E^T E \Delta_k + \Delta_k^T \tilde{G} \Delta_k) \sum_{\ell \in \mathbb{N}^d} \frac{\mathbf{d}_{\text{vec}}^\ell \mathbf{v}_k^\ell \sqrt{2^{|\ell|}}}{\sqrt{t^{|\ell|}\ell!}} H_\ell^{G,E,t}(\mathbf{x} - \mathbf{x}_0). \end{aligned}$$

As before, we can write the expansion above as the infinite matrix-vector product

$$\Phi(\mathbf{x}) = H^{G,E,t}(\mathbf{x} - \mathbf{x}_0) B(E, G, t, X^{\text{cen}})$$

with

$$B(E, G, t, X^{\text{cen}})_{\ell k} = \exp(-\Delta_k^T E^T E \Delta_k + \Delta_k^T \tilde{G} \Delta_k) \frac{\mathbf{d}_{\text{vec}}^\ell \mathbf{v}_k^\ell \sqrt{2^{|\ell|}}}{\sqrt{t^{|\ell|}\ell!}}.$$

As before, we write the transpose of the infinite matrix B as a product CD with

$$(3.2) \quad C_{k\ell} = \exp(-\Delta_k^T E^T E \Delta_k + \Delta_k^T \tilde{G} \Delta_k) \mathbf{v}_k^\ell, \quad D_{\ell\ell} = \frac{\mathbf{d}_{\text{vec}}^\ell \sqrt{2^{|\ell|}}}{\sqrt{t^{|\ell|}\ell!}}.$$

The $d \times d$ matrix product $\text{Diag}^{-1}\text{Rem}$ contained in the vectors \mathbf{v}_k can be computed analytically. The diagonal part of the matrix $G^{-1}E^TE$, that is now in the matrix D can be handled in the exact same fashion as it has been done in 1D. In particular, we perform the QR-decomposition of the matrix $C = QR$, block decompose

$$R = \begin{pmatrix} R_1 & R_2 \end{pmatrix}, \quad D = \begin{pmatrix} D_1 & 0 \\ 0 & D_2 \end{pmatrix},$$

such that the entries related to the first N stabilizing basis functions are contained in the $N \times N$ matrices R_1 and D_1 , and consider the preconditioner $X = QR_1D_1$. Hence, analogously to the HermiteGF-QR case, the new basis can be formed as

$$\Psi(\mathbf{x})^T := X^{-1}\Phi(\mathbf{x})^T = (\text{Id} \quad D_1^{-1}R_1^{-1}R_2D_2) H^{G,E,t}(\mathbf{x} - \mathbf{x}_0)^T.$$

The action of D_1^{-1} and D_2 can again be computed as the Hadamard product with

$$(3.3) \quad \tilde{D}_{ij} = \frac{D_{j+N,j+N}}{D_{ii}} \quad \text{with } i \in \{1, \dots, N\}, j \geq 1.$$

Remark 3.1. *When the magnitude of \mathbf{v}_k gets too large, the matrix C can also become ill-conditioned. To avoid that, one can increase the magnitude of the elements of G . Alternatively, one can add a scaling in C which should then be compensated in the matrix D , similarly to the scaling with the domain size L in the 1D version.*

3.2.1. *Isotropic HermiteGF-QR.* In the isotropic case, when $E = \varepsilon \text{Id}_d$ and $G = \gamma \text{Id}_d$, the expressions for the matrices C and D can be written in a simpler way. In particular, in this case, we have

$$\text{Diag} = \gamma^{-1} \varepsilon^2 \text{Id}_d, \quad \text{Rem} = \mathbf{0}_d, \quad \text{and} \quad \mathbf{v}_k = \mathbf{x}_k - \mathbf{x}_0 = \mathbf{\Delta}_k.$$

Hence, the diagonal vector simplifies to $\mathbf{d}_{\text{vec}} = \gamma^{-1} \varepsilon^2 (1, \dots, 1)$ and the elements of the matrices C and D take the form

$$C_{k\ell} = \exp\left(\varepsilon^2 \left(\frac{\varepsilon^2}{\gamma^2} - 1\right) \|\mathbf{\Delta}_k\|_{\ell_2}^2\right) \mathbf{\Delta}_k^\ell, \quad D_{\ell\ell} = \frac{(\sqrt{2}\gamma^{-1}\varepsilon^2)^{|\ell|}}{\sqrt{t^{|\ell|}\ell!}}.$$

3.3. **HermiteGF interpolant.** In the new basis, we can write the equivalent formulation of the interpolant (2.1) as

$$s(\mathbf{x}) = \Psi(\mathbf{x})(\Psi^{\text{col}})^{-1} \mathbf{f} \quad \text{with} \quad \Psi_{ij}^{\text{col}} = \psi_j(\mathbf{x}_i^{\text{col}}).$$

To compute the interpolant in the new formulation numerically, we have to cut the infinite expansion (2.6). We discuss the cut-off strategy in section 4.

3.4. **Alternative splitting based on the Vandemonde matrix.** Instead of using a QR-decomposition of the matrix $C \in \mathbb{R}^{N \times \infty}$ one could also split it as $C = \bar{E}W$, where $\bar{E} \in \mathbb{R}^{N \times N}$ is a diagonal matrix for the exponential part and $W \in \mathbb{R}^{N \times \infty}$ accounts for the polynomial contributions,

$$(3.4) \quad \bar{E}_{kk} = \exp(-\mathbf{\Delta}_k^T E^T E \mathbf{\Delta}_k + \mathbf{\Delta}_k^T \tilde{G} \mathbf{\Delta}_k) \quad \text{and} \quad W_{k\ell} = \mathbf{v}_k^\ell.$$

We now decompose the original basis using this splitting,

$$\begin{aligned} \Phi(\mathbf{x})^T &= C \begin{pmatrix} D_1 & 0 \\ 0 & D_2 \end{pmatrix} H^{G,E,t}(\mathbf{x} - \mathbf{x}_0)^T \\ &= \bar{E} \begin{pmatrix} W_1 & W_2 \end{pmatrix} \begin{pmatrix} D_1 & 0 \\ 0 & D_2 \end{pmatrix} H^{G,E,t}(\mathbf{x} - \mathbf{x}_0)^T \\ &= \bar{E} \begin{pmatrix} W_1 D_1 & W_2 D_2 \end{pmatrix} H^{G,E,t}(\mathbf{x} - \mathbf{x}_0)^T, \end{aligned}$$

where $W_1 \in \mathbb{R}^{N \times N}$ and $W_2 \in \mathbb{R}^{N \times \infty}$. With the preconditioner $X_V = \bar{E}W_1 D_1$, the new basis reads as

$$\Psi_V(\mathbf{x})^T = (\text{Id} \quad D_1^{-1} W_1^{-1} W_2 D_2) H^{G,E,t}(\mathbf{x} - \mathbf{x}_0)^T.$$

One can see that

$$X_V = \bar{E}W_1 D_1 = C_1 D_1 = QR_1 D_1 = X,$$

where C_1 is the first $N \times N$ block of C . Therefore, in exact arithmetic Ψ and Ψ_V are the same, however, in floating-point arithmetic the values of the bases might differ. However, we will use this alternative splitting for the derivation and analysis of a suitable cut-off criterion for the HermiteGF basis in the next section.

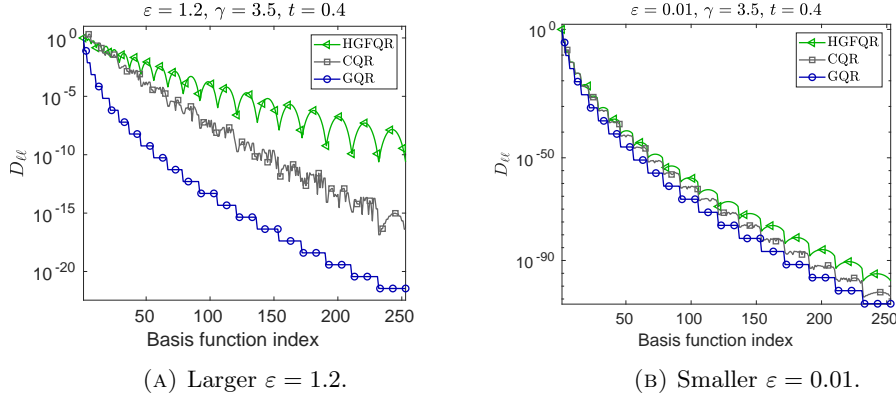


FIGURE 2. For $\varepsilon = 0.01$ the elements of D behave very similarly for all cases. For a larger value of ε , the elements of D in HermiteGF method show larger magnitude of oscillations which leads to higher j_{\max} .

4. CUT-OFF OF THE EXPANSION

To make the RBF-QR methods usable for numerical computations, one has to cut the expansion (2.6) at a certain polynomial degree $j_{\max} \in \mathbb{N}$ which in 1D also corresponds to the number of stabilizing basis functions M . In the multivariate setting, the number of basis functions M equals

$$M = \binom{j_{\max} + d}{d}.$$

However, choosing an efficient cut-off degree j_{\max} is not a trivial task. We first derive a criterion that is analogous to the state-of-the-art criteria for other RBF-QR methods ([9, Expr. 5.2], [7, Expr. 4.10]). However, especially in the HermiteGF case, it turns out to be inefficient, i.e. it overestimates the number j_{\max} . For the HermiteGF ansatz, we can derive a new cut-off criterion based on the theoretical framework presented in the previous sections. This new criterion allows to directly control the approximation error of the stable basis which is more efficient while still being effective.

4.1. State-of-the-art criterion. Similarly to [9, Expr. 5.2], [7, Expr. 4.10] we take a look at the matrix \tilde{D} that contains the effects of D_1^{-1} and D_2 . Recall that for the QR methods in section 3.1 the matrix \tilde{D} is then multiplied element-wise with the matrix \tilde{R} . We stop once all elements of the new block of \tilde{D} are below machine precision, i.e.

$$(4.1) \quad \max_{i=1 \dots N, |j| \geq j_{\max} + 1} \tilde{D}_{ij} < \varepsilon_{\text{mach}}.$$

The criterion (4.1) guarantees that all additional columns that could be added to D_2 would yield elements in \tilde{D} that are below machine precision. We now take a look at the behavior of the elements of the matrix D . Here, for the Gauss-QR method we used $\alpha_{\text{Gauss-QR}} = \gamma$. One can see in Figure 2 that for small values of ε the behavior is very similar for all three methods. However, for large ε the decay in D is particularly bad in our formulation. This criterion also neglects the matrix \tilde{R} and the effect of the polynomial vector $H^{\gamma, \varepsilon, t}(x - x_0)$. In particular, we know from section 2.4 that the tail of the polynomial vector has some decay.

4.2. New HermiteGF cut-off criterion. In this section, we derive a more holistic criterion for the cut-off in the HermiteGF expansion. We use the Vandermonde formulation of the method since it provides an explicit expression for the elements of all matrices which simplifies the analytical

study of the method. We cut the polynomial vector as

$$H^{G,E,t}(\mathbf{x} - \mathbf{x}_0) = \left(\hat{H}^{G,E,t}(\mathbf{x} - \mathbf{x}_0) \quad H_\infty^{G,E,t}(\mathbf{x} - \mathbf{x}_0) \right)$$

with $\hat{H}^{G,E,t} \in \mathbb{R}^{1 \times M}$, where the number of basis functions used is larger than the number of collocation points, that is, $M \geq N$. Analogously we cut the $N \times \infty$ Vandermonde matrix W_2 and the infinite diagonal matrix D_2 ,

$$W_2 = \left(\hat{W}_2 \quad W_\infty \right) \quad \text{and} \quad D_2 = \begin{pmatrix} \hat{D}_2 & 0 \\ 0 & D_\infty \end{pmatrix}$$

with $\hat{W}_2 \in \mathbb{R}^{N \times (M-N)}$ and $\hat{D}_2 \in \mathbb{R}^{(M-N) \times (M-N)}$. We note that the infinite matrix W_∞ contains the columns of the full Vandermonde matrix W from column $M + 1$ onward, while the infinite matrix D_∞ contains the diagonal entries of the full diagonal matrix D starting from the entry $M + 1$. We then rewrite the formulation of the method (see section 3.4) after the cut-off,

$$\hat{\Psi}(\mathbf{x})^T = \left(\text{Id}_{N \times N} \quad D_1^{-1} W_1^{-1} \hat{W}_2 \hat{D}_2 \right) \hat{H}^{G,E,t}(\mathbf{x} - \mathbf{x}_0)^T.$$

We want to make sure that

$$\delta\Psi(\mathbf{x}) = \Psi(\mathbf{x}) - \hat{\Psi}(\mathbf{x})$$

is small for all collocation points by choosing a sufficiently large but not too large truncation parameter j_{\max} . For estimating $\delta\Psi(\mathbf{x})$, we need the following lemma.

Lemma 4.1 (Exponential tail). *Consider $\mathbf{y} \in \mathbb{R}^d$ with $y_i \geq 0$ for all $i = 1, \dots, d$ and $j_{\max} \in \mathbb{N}$. Then,*

$$(4.2) \quad \sum_{|\ell| \geq j_{\max}} \frac{\mathbf{y}^\ell}{\ell!} \leq \sum_{|\ell|=j_{\max}} \exp(\|\mathbf{y}\|_1) \frac{\mathbf{y}^\ell}{\ell!},$$

where $\|\mathbf{y}\|_1 = |y_1| + \dots + |y_d|$.

Proof. Consider the function

$$f(\mathbf{y}) = \exp(y_1 + \dots + y_d), \quad \mathbf{y} \geq 0.$$

We note that the estimated sum coincides with the remainder of the Taylor series for the function f at the point $\mathbf{a} = 0$. Then, according to the multivariate Taylor's theorem, there exists $\boldsymbol{\xi} \in [0, \mathbf{y}]$ such that

$$\sum_{|\ell| \geq j_{\max}} \frac{\mathbf{y}^\ell}{\ell!} = \sum_{|\ell|=j_{\max}} \partial^\ell f(\boldsymbol{\xi}) \frac{\mathbf{y}^\ell}{\ell!}.$$

Noting that $\partial^\ell f(\boldsymbol{\xi}) = \exp(\boldsymbol{\xi}) \leq \exp(\|\mathbf{y}\|_1)$ for $\mathbf{y} \geq 0$, we arrive at (4.2). \square

Before proceeding to the estimation of the truncation error $\|\delta\Psi(\mathbf{x})\|_2$, we recall the definition of the vectors

$$\mathbf{d}_{\text{vec}} = \text{diag}(\text{Diag}) \quad \text{and} \quad \mathbf{v}_k = (\text{Id} + \text{Diag}^{-1} \text{Rem}) \boldsymbol{\Delta}_k$$

for $k = 1, \dots, N$. They will contribute to the upper bound of the following estimate. In the isotropic case, they have the particularly simple form

$$\mathbf{d}_{\text{vec}} = \gamma^{-1} \varepsilon^2 (1, \dots, 1) \quad \text{and} \quad \mathbf{v}_k = \boldsymbol{\Delta}_k \quad \text{for all } k = 1, \dots, N.$$

Theorem 4.1 (Truncation estimate). *For $k = 1, \dots, N$ we set*

$$\omega_k = \sum_{i=1}^N (W_1^{-1})_{ki}^2 > 0 \quad \text{and} \quad \mathbf{y}_k = \text{Diag } \mathbf{v}_k \in \mathbb{R}^d,$$

where W_1 is the upper left $N \times N$ block of the infinite Vandermonde matrix $W = (\mathbf{v}_k^\ell)$. For $j_{\max} \in \mathbb{N}$ we denote

$$\text{const}_{j_{\max}} := \left(\sum_{k=1}^N \frac{\omega_k \mathbf{k}! (t/2)^{|\mathbf{k}| - (j_{\max} + 1)}}{\mathbf{d}_{\text{vec}}^{2\mathbf{k}} (j_{\max} + 1)!} \right) \cdot \left(\sum_{i=1}^N \exp\left(\frac{2}{t} \|\mathbf{y}_i\|_2^2\right) \|\mathbf{y}_i\|_2^{2(j_{\max} + 1)} \right).$$

Then, the truncation error $\delta\Psi$ satisfies for all $\mathbf{x} \in \mathbb{R}^d$,

$$(4.3) \quad \|\delta\Psi(\mathbf{x})\|_2^2 \leq \text{const}_{j_{\max}} \cdot \left(H_{\text{lim}}^{G,E,t}(\mathbf{x} - \mathbf{x}_0) - \sum_{|\ell| \leq j_{\max}} H_{\ell}^{G,E,t}(\mathbf{x} - \mathbf{x}_0)^2 \right),$$

where $H_{\text{lim}}^{G,E,t}(\mathbf{x} - \mathbf{x}_0)$ can be evaluated via (2.11).

Proof. We start by observing that

$$D_1^{-1}W_1^{-1}W_2D_2 = D_1^{-1}W_1^{-1}(\hat{W}_2\hat{D}_2 - W_{\infty}D_{\infty}).$$

Hence, it holds that

$$\delta\Psi(\mathbf{x})^T = D_1^{-1}W_1^{-1}W_{\infty}D_{\infty}H_{\infty}^{G,E,t}(\mathbf{x} - \mathbf{x}_0)^T,$$

and due to compatibility of the Frobenius norm and the 2-norm

$$\|\delta\Psi(\mathbf{x})\|_2^2 \leq \|D_1^{-1}W_1^{-1}W_{\infty}D_{\infty}\|_F^2 \cdot \|H_{\infty}^{G,E,t}(\mathbf{x} - \mathbf{x}_0)\|_2^2.$$

We further consider the two norms on the right-hand side separately. We first take a look at the Frobenius norm. Recall that in the RBF-QR method we evaluate the effect of the impact of $D_1^{-1} \dots D_2$ analytically. We can do the same here:

$$D_1^{-1}W_1^{-1}W_{\infty}D_{\infty} = \tilde{D}_{\infty} \cdot *(W_1^{-1}W_{\infty}),$$

where $*$ denotes the Hadamard product and \tilde{D}_{∞} is constructed analogously to (3.3). We write the Frobenius norm as

$$\|D_1^{-1}W_1^{-1}W_{\infty}D_{\infty}\|_F^2 = \sum_{k=1}^N \sum_{\ell > M} \left(\tilde{D}_{k\ell}^2 \cdot \left(\sum_{i=1}^N (W_1^{-1})_{ki} W_{i\ell} \right)^2 \right)$$

and estimate with the help of the Cauchy-Schwarz inequality,

$$\tilde{D}_{k\ell}^2 \left(\sum_{i=1}^N (W_1^{-1})_{ki} W_{i\ell} \right)^2 \leq \frac{\omega_k}{D_{kk}^2} D_{\ell\ell}^2 \sum_{i=1}^N \mathbf{v}_i^{2\ell},$$

where $\ell \in \mathbb{N}^d$ is the ℓ -th multi-index corresponding to our basis enumeration. We used the explicit expression of $D_{\ell\ell}$ as defined in (3.2) and write

$$\frac{\omega_k}{D_{kk}^2} D_{\ell\ell}^2 \sum_{i=1}^N \mathbf{v}_i^{2\ell} = \frac{\omega_k \mathbf{k}! t^{|\mathbf{k}|}}{\mathbf{d}_{\text{vec}}^{2\mathbf{k}} 2^{|\mathbf{k}|}} \frac{2^{|\ell|}}{\ell! t^{|\ell|}} \sum_{i=1}^N (\text{Diag } \mathbf{v}_i)^{2\ell}.$$

We denote $\tilde{\mathbf{y}}_i = \left(\frac{2}{t}(\mathbf{y}_i)_1^2 \quad \dots \quad \frac{2}{t}(\mathbf{y}_i)_d^2 \right)$. Then, by lemma 4.1 we get

$$\begin{aligned} \|D_1^{-1}W_1^{-1}W_{\infty}D_{\infty}\|_F^2 &\leq \sum_{k=1}^N \frac{\omega_k \mathbf{k}! t^{|\mathbf{k}|}}{\mathbf{d}_{\text{vec}}^{2\mathbf{k}} 2^{|\mathbf{k}|}} \sum_{i=1}^N \sum_{|\ell| > j_{\max}} \frac{\tilde{\mathbf{y}}_i^{\ell}}{\ell!} \\ &\leq \sum_{k=1}^N \frac{\omega_k \mathbf{k}! t^{|\mathbf{k}|}}{\mathbf{d}_{\text{vec}}^{2\mathbf{k}} 2^{|\mathbf{k}|}} \sum_{i=1}^N \sum_{|\ell| = j_{\max} + 1} \exp(\|\tilde{\mathbf{y}}_i\|_1) \frac{\tilde{\mathbf{y}}_i^{\ell}}{\ell!}. \end{aligned}$$

Using the multinomial theorem and the fact that

$$\|H_{\infty}^{G,E,t}(\mathbf{x} - \mathbf{x}_0)\|_2^2 = H_{\text{lim}}^{G,E,t}(\mathbf{x} - \mathbf{x}_0) - \sum_{|\ell| \leq j_{\max}} H_{\ell}^{G,E,t}(\mathbf{x} - \mathbf{x}_0)^2.$$

we arrive to estimate (4.3). \square

The denominator $\mathbf{d}_{\text{vec}}^{2\mathbf{k}}$ in expression (4.3) can take extremely small values that can lead to underflow. To avoid this, it can be combined with $\|\mathbf{y}_i\|_2^{2(j_{\max}+1)}$. For this, we define the d -dimensional index $\mathbf{j}_d = (\frac{1}{d}, \dots, \frac{1}{d})$ and use the following transformation

$$\|\mathbf{y}_i\|_2 = \mathbf{d}_{\text{vec}}^{\mathbf{j}_d} \|\mathbf{y}_i ./ (\mathbf{d}_{\text{vec}}^{\mathbf{j}_d})\|_2 =: \mathbf{d}_{\text{vec}}^{\mathbf{j}_d} \|\mathbf{y}_i^D\|_2,$$

where $./$ denotes component-wise division. Pulling out $\text{Diag}^{\mathbf{j}_d}$ the constant of the estimate (4.3) can be rewritten as

$$\text{const}_{j_{\max}} = \left(\sum_{k=1}^N \frac{\omega_k \mathbf{k}! \left(\frac{2}{t} \mathbf{d}_{\text{vec}}^2\right)^{-\mathbf{k} + (j_{\max}+1)\mathbf{j}_d}}{(j_{\max}+1)!} \right) \left(\sum_{i=1}^N \exp\left(\frac{2}{t} \|\mathbf{y}_i\|_2^2\right) \|\mathbf{y}_i^D\|_2^{2(j_{\max}+1)} \right).$$

Note, that in the isotropic case, one can simplify $\mathbf{d}_{\text{vec}}^{-2\mathbf{k} + 2(j_{\max}+1)\mathbf{j}_d} = (\varepsilon^4/\gamma^2)^{j_{\max}+1-|\mathbf{k}|}$. We are now ready to formulate our cut-off criterion.

Criterion 1. We choose j_{\max} for the HermiteGF-QR method such that

$$\max_{k=1 \dots N} \frac{\|\delta\Psi(\mathbf{x}_k)\|_2}{\|\hat{\Psi}(\mathbf{x}_k)\|_2} \leq \text{TOL},$$

where $\{\mathbf{x}_k\}_{k=1}^N$ are the collocation points.

Since we are looking at the relative error, the tolerance TOL need not be machine precision. The crucial difference to the state-of-the-art criterion eq. (4.1) is that now the TOL directly controls the accuracy of the stable basis Ψ . Depending on the desired accuracy, the tolerance can be adjusted for the specific problem.

Remark 4.1. The state-of-the-art criterion that truncates diagonal elements below machine precision does not provide an error bound on the interpolant. On the one hand, the new criterion requires more computations for determining the cut-off degree. On the other hand, it allows to reduce the polynomial degree j_{\max} while still guaranteeing a given truncation error. This in turn reduces the computational cost of the interpolation step.

4.3. Automatic detection of t . One can use the criterion above also for determining the value of the parameter t . We scan the whole spectrum of the values of t and detect the one that yields the minimum amount of basis functions

$$\arg \min_{t \in (0,1)} j_{\max}(t) = t_{\text{auto}}.$$

Note that very small values of t can cause cancellations and should be excluded (see section 5.1.1). Even though this introduces additional computational cost in the determination of the suitable expansion, it could be profitable for the cases where the basis is used multiple times after having fixed the number j_{\max} as e.g. in a time loop.

5. NUMERICAL RESULTS

We have implemented the HermiteGF interpolation in MATLAB. The code is available at <https://gitlab.mpcdf.mpg.de/clapp/hermiteGF>. We compare the isotropic HermiteGF-based algorithm with the existing stabilization methods, the Chebyshev-QR method² and the Gauss-QR method³. We evaluate the influence of different parameters, such as ε , γ , number of collocation points N on the quality of the interpolation. For the Gauss-QR method, we take the free parameter α to be equal to our value of γ , i.e., $\alpha_{\text{Gauss-QR}} = \gamma$. Since there are no stabilization methods available for fully anisotropic interpolation, we test the anisotropic HermiteGF-QR only against the direct algorithm to verify the correctness. To determine the cut-off degree in the HermiteGF

²Code downloaded from http://www.it.uu.se/research/scientific_computing/software/rbf_qr on September 10, 2018.

³Code downloaded from <http://math.iit.edu/~mccomic/gaussqr/> on September 5, 2018.

method, we use the HermiteGF cut-off criterion with $\text{TOL} = 10^{-6}$, unless stated otherwise. For this tolerance, the HermiteGF-QR method provides results that match the results from Chebyshev-QR and Gauss-QR. The parameter t is detected automatically. In all tests we evaluate the interpolant at a set of evaluation points $\{\mathbf{z}_k\}_{k=1}^{N_{\text{ev}}}$ and look at the average error of the form [7, § 5.1, Expr. (5.2)]:

$$\text{error} = \frac{1}{N_{\text{ev}}} \sqrt{\sum_{k=1}^{N_{\text{ev}}} \left(\frac{f(\mathbf{z}_k) - s(\mathbf{z}_k)}{f(\mathbf{z}_k)} \right)^2}.$$

5.1. 2D isotropic interpolation. In this section, we take a look at the two-dimensional interpolation with HermiteGF-QR. We take multiples of the identity for both E and G . We look at a hyperbolic domain (see Figure 3) defined by the inequality

$$(5.1) \quad 0.04 \leq (x + 1.2)^2 - 4y^2 \leq 1$$

with a boundary condition $x^2 + y^2 \leq 1$. The hyperbola of type (5.1) can then be parameterized as

$$(x, y) = r(t) = (c \cosh(t) - 1.2, 0.5c \sinh(t)), \quad t \in \mathbb{R}, \quad c \in [0.2, 1].$$

We run the tests for the following function (f_4 from [9, § 6]):

$$f_4(x, y) = \sin(x^2 + 2y^2) - \sin(2x^2 + (y - 0.5)^2).$$

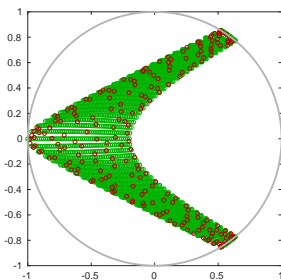


FIGURE 3. Hyperbolic domain. Evaluation grid (green) and $N = 210$ clustered Halton node points (red).

We investigate the behavior of the performance of the interpolation with respect to the parameters γ , ε , and number of functions N . We use $\gamma = 3.5$, $\varepsilon = 0.05$ and optimize t from the set `tvec = linspace(0.1, 0.99, 10)`, unless stated otherwise.

We sample the collocation points from Halton points that are clustered near the boundary to improve the conditioning of the polynomial interpolation and are then mapped to the hyperbolic domain. For all tests, we use $N_{\text{ev}} = 53^2$ evaluation points that are sampled similarly to the collocation points, but based on a uniform grid and without clustering. The nodes distribution is depicted in Figure 3. This domain and sampling strategy choice was inspired by [9, § 6.1.2].

5.1.1. The number of nodes N . Let us first look at the behavior of the method for different numbers of nodes, N . We take the values of $N \in [100, 410]$ of the form $\binom{P+2}{2}$ for some integer P , such that there are no same powers of ε present in both D_1 and D_2 . In Figure 4, we see that the error consistently decays for all the tested methods. Choosing the truncation parameter t in the interval $t \in (0.1, 1)$, the conditioning of the HermiteGF-QR method is slightly worse than for the other methods, since big powers of smaller values of t yield cancellations. Indeed, limiting the range of t to $t \in (0.3, 1)$ brings the conditioning to the level of the other methods. Using all integers in the interval $[100, 410]$ also provided consistent results for all three methods, however, the picture gets noisy. A snippet of that behavior can be seen in the zoomed regions in Figure 4. This can be related to the fact that for the values of N of the form above the limit of the RBF interpolant in the flat limit $\varepsilon \rightarrow 0$ is a *unique* polynomial of degree P [16, §4 Theorem 4.1] whereas for other values the uniqueness is not guaranteed.

5.1.2. Sensitivity to γ . Let us take a look at the influence of the parameter γ on the interpolation quality. We see in Figure 5 that for small N the interpolation quality is not sensitive to the value of γ . However, for larger N the parameter γ has to be chosen with care. One can see in the Figure 5b that the conditioning is worse for small γ . However, one should be careful while increasing γ since it also increases the evaluation domain of the Hermite polynomials, which take very large values on large domains which can lead to overflow. This effect becomes more pronounced as the degree

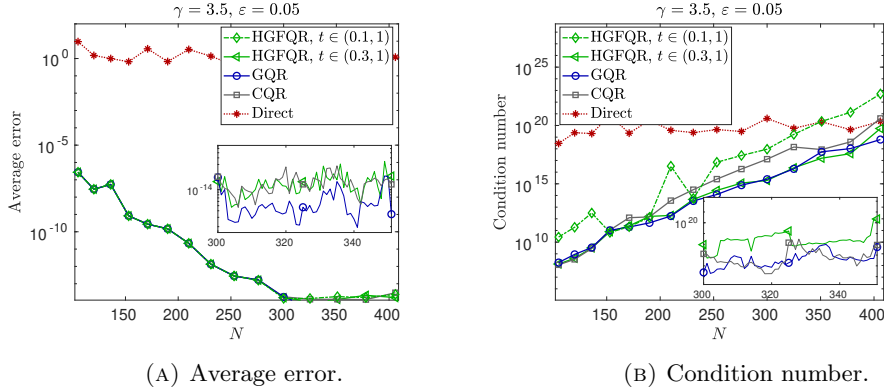


FIGURE 4. For the two-dimensional isotropic test case, the interpolation quality is the same for all three stabilization methods. The conditioning is slightly worse for small values of t . There is small noise for all methods when the number of radial basis functions N does not correspond to a number of all polynomials of a degree $\leq P$ for a certain P .

of the Hermite polynomials increases. The optimal balance depends on the particular function and the number of basis functions.

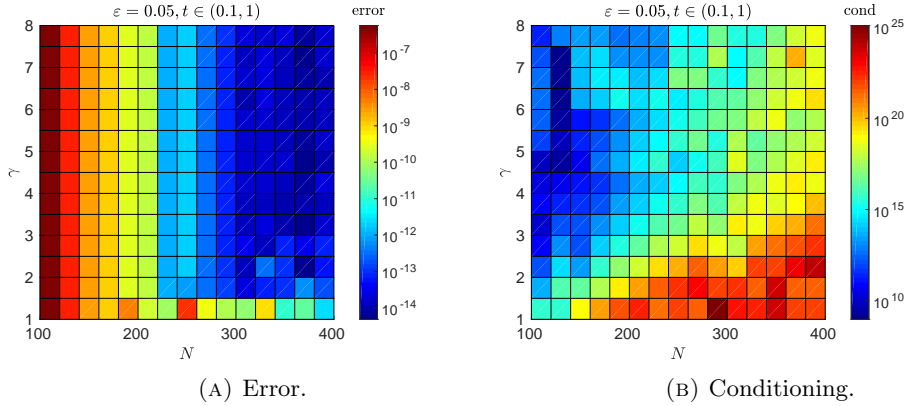


FIGURE 5. Average error and condition number for the isotropic two-dimensional test case. For small and moderate N the interpolation quality is not sensitive to the value of γ , whereas for the larger N one should carefully choose the value of γ .

5.1.3. *Cut-off degree j_{\max} .* Next, we look at the influence of the value of TOL on the quality of the interpolation. We compare the error only to the Gauss-QR method since the difference between the Chebyshev-QR and Gauss-QR results is down to machine precision. One can see in Figure 6 that for TOL = 10^{-6} the difference HermiteGF-QR and Gauss-QR is also down to machine precision. If we relax the tolerance to 10^{-2} , the error is still small compared to the magnitude of the interpolation error, while having smaller j_{\max} , which yields an improved computational efficiency. Also, the figure shows a general trend that the expansion decays rather fast for small ε while an increasing number of basis functions is needed for ε close to 1.

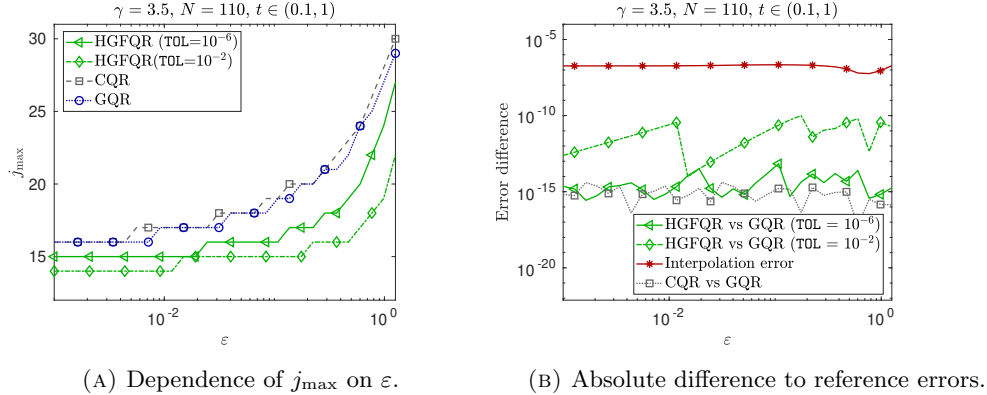


FIGURE 6. Optimized truncation value and error differences for the two-dimensional isotropic test case. For the coarser TOL= 10^{-2} we get fewer basis functions, with truncation error still much below the interpolation error.

5.2. 2D anisotropic interpolation. To test the performance of the HermiteGF expansion for anisotropic basis functions, we consider the function:

$$f_a(x, y) = \frac{1}{x^2 + xy + y^2} + 2, \quad x, y \in [-1, 1].$$

As collocation points, we use Halton points clustered toward the boundaries. For the evaluation grid we use 53×53 uniformly distributed points. As for the shape matrix E , we check whether the off-diagonal elements influence the quality of the results. We choose a non-diagonal matrix G of arbitrary pattern to demonstrate the robustness of the method. We fix E and G to be of the following form:

$$E = \varepsilon \begin{pmatrix} 1 & p \\ p & 1 \end{pmatrix} \quad \text{with } p \in [0, 0.8], \quad G = \gamma \begin{pmatrix} 1 & 0.3 \\ 0.1 & 1.3 \end{pmatrix} \quad \text{with } \gamma = 3.5.$$

We restrict t to the interval `tvec = linspace(0.3, 1, 10)` in order to improve the stability of the computations. Let us take a look on how much the off-diagonal elements of the matrix E influence the error. For our scan, we take 30 logarithmically distributed values of $\varepsilon \in [10^{-3}, 10^{0.1}]$. One can see from Figure 7 that certain choices of off-diagonal elements can improve the quality of the interpolation compared to purely diagonal shape matrices. However, for larger p slight instabilities occur which might be explained since E becomes singular for $p \rightarrow 1$.

5.3. Multivariate interpolation. In this section, we consider an example of the usage of HermiteGF-QR in higher dimension. For all tests, we use the function

$$f(\mathbf{x}) = \cos(|\mathbf{x}|), \quad \mathbf{x} \in [-1, 1]^d,$$

where $|\mathbf{x}| = \sum_{i=1}^d x_i$. We use Halton collocation points and 1000 Halton points, excluding the ones used for collocation, for the evaluation grid. We fix $G = 5\text{Id}_d$, $\varepsilon = \text{logspace}(-3, 0.1, 30)$, and we again optimize the parameter t over the set `tvec = linspace(0.3, 1, 10)`. As before, we choose the tolerance TOL = 10^{-6} . We first look whether the anisotropic HermiteGF-QR method converges to the results of the direct interpolation as ε increases. We choose an arbitrary pattern for E in order to verify that the stabilization works for a truly anisotropic interpolation,

$$E_a = \varepsilon \begin{pmatrix} 1 & 0.2 & 0.3 \\ 0.2 & 1 & 0.15 \\ 0.1 & 0.3 & 1 \end{pmatrix}.$$

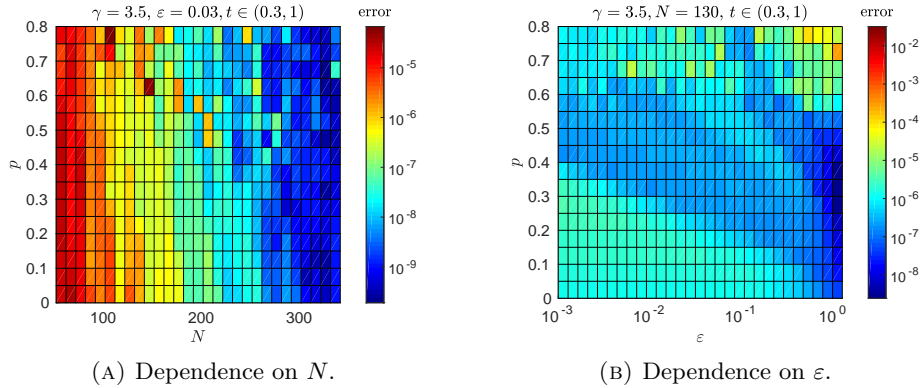


FIGURE 7. Two-dimensional anisotropic interpolation. For a fixed value of p , the error generally decreases with the growth of N as expected. Choosing an anisotropic shape matrix E ($p \neq 0$) often improves the interpolation quality.

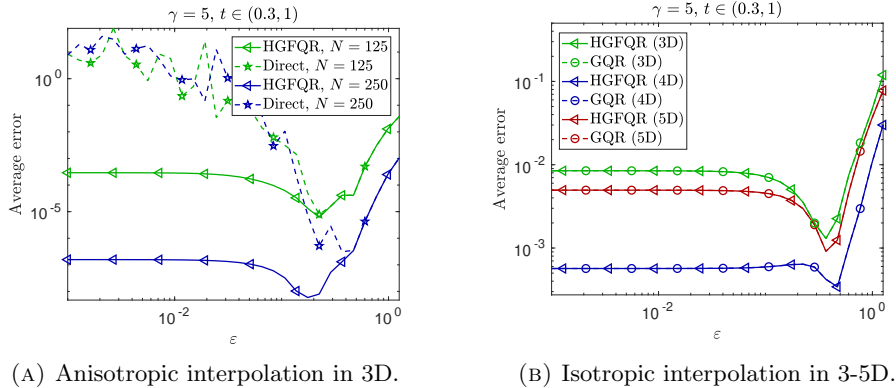


FIGURE 8. Interpolation in dimensions 3-5D: Both the anisotropic and the isotropic HermiteGF-QR method match the average error of the Gauss-QR method.

In Figure 8a, we can see that HermiteGF-QR interpolation works stably even for very small values of ε for different N . On the other hand, for larger values of ε the result matches the direct anisotropic interpolation.

In order to validate the HermiteGF-QR method in higher dimensions against the existing methods, we compare the isotropic HermiteGF-QR method with the Gauss-QR method in 3-5D. For that, we fix the shape matrix E and the number of interpolation points N as

$$E = \text{Id}_d \quad \text{and} \quad N = 4^d,$$

where d is the dimensionality. We choose the tolerance $\text{TOL} = 10^{-2}$ since it is enough to meet the overall accuracy of the method. One can see in Figure 8b that the HermiteGF-QR method matches the reference Gauss-QR method in 3-5D.

6. CONCLUSION AND OUTLOOK

In this paper, we derive a new stabilization algorithm for Gaussian RBF interpolation in the flat limit ($\varepsilon \rightarrow 0$). The main idea of “isolating” the ill-conditioning in a special matrix is the same as in the previous approaches [11, 7, 9]. However, our new algorithmic framework draws from generating functions that naturally extend to the interpolation with anisotropic Gaussians. We

introduce several parameters (ε/E , γ/G , t) for the new HermiteGF basis which have a distinct connotation: ε/E is the original shape parameter of the Gaussian basis, γ/G stands for the size of the evaluation domain of the Hermite polynomials, and the technical basis truncation parameter t can be chosen automatically thanks to a novel analytically derived truncation criterion. The interpolation quality is not sensitive to the precise value of γ/G . The generic formulation of the method essentially provides an algorithm in any dimension, and we have reported results for up to five dimensions.

For the cut-off of the HermiteGF expansion, we derive a novel truncation criterion, generalizing Mehler's theory for bilinear generating functions. In particular, we analytically estimate the truncation error $\delta\Psi$ in the stable HermiteGF basis Ψ . This allows adjusting the number of basis functions based on the desired accuracy in the basis Ψ .

The HermiteGF-QR algorithm has been implemented in MATLAB. For all isotropic test cases, the accuracy of the HermiteGF-QR method is consistent with the ones of established stabilization methods (Chebyshev-QR, Gauss-QR). For the anisotropic case, the results matches the RBF-Direct method, where the latter is applicable.

The stability of our algorithm could be further improved by employing special algorithms for the stable inversion of the Vandermonde type matrices. Based on the successful experience of the automatic detection of the truncation parameter t , it could be possible to develop an algorithm of choosing an optimal parameter matrix G defining the effective evaluation domain. We believe that our anisotropic method could be also of interest in statistical data fitting [6, § 17] or for continental size ice sheet simulations, see e.g. [3].

ACKNOWLEDGMENTS

Fruitful discussions with Elisabeth Larsson (Uppsala University) and Michael McCourt (SigOpt) are gratefully acknowledged.

REFERENCES

- [1] M. ABRAMOWITZ AND I. A. STEGUN, *Handbook of mathematical functions: with formulas, graphs, and mathematical tables*, vol. 55, Courier Corporation, 1964.
- [2] R. BEATSON, O. DAVYDOV, AND J. LEVESLEY, *Error bounds for anisotropic rbf interpolation*, J. Approx. Theory, 162 (2010), pp. 512 – 527.
- [3] G. CHENG AND V. SHCHERBAKOV, *Anisotropic radial basis function methods for continental size ice sheet simulations*, J. Comput. Phys., 372 (2018), pp. 161–177.
- [4] H. DIETERT, J. KELLER, AND S. TROPFMANN, *An invariant class of wave packets for the Wigner transform*, J. Math. Anal. Appl., 450 (2017), pp. 1317–1332.
- [5] G. E. FASSHAUER, F. J. HICKERNELL, AND H. WOŹNIAKOWSKI, *On dimension-independent rates of convergence for function approximation with Gaussian kernels*, SIAM J. Numer. Anal., 50 (2012), pp. 247–271.
- [6] G. E. FASSHAUER AND M. MCCOURT, *Kernel-based approximation methods using Matlab*, vol. 19, World Scientific Publishing Company, 2015.
- [7] G. E. FASSHAUER AND M. J. MCCOURT, *Stable evaluation of Gaussian radial basis function interpolants*, SIAM J. Sci. Comput., 34 (2012), pp. A737–A762.
- [8] B. FORNBERG AND N. FLYER, *Solving PDEs with radial basis functions*, Acta Numer., 24 (2015), p. 215258.
- [9] B. FORNBERG, E. LARSSON, AND N. FLYER, *Stable computations with Gaussian radial basis functions*, SIAM J. Sci. Comput., 33 (2011), pp. 869–892.
- [10] B. FORNBERG, E. LEHTO, AND C. POWELL, *Stable calculation of Gaussian-based RBF-FD stencils*, Comput. Math. Appl., 65 (2013), pp. 627–637.
- [11] B. FORNBERG AND C. PIRET, *A stable algorithm for flat radial basis functions on a sphere*, SIAM J. Sci. Comput., 30 (2007), pp. 60–80.
- [12] B. FORNBERG AND G. B. WRIGHT, *Stable computation of multiquadric interpolants for all values of the shape parameter*, Comput. Math. Appl., 48 (2004), pp. 853 – 867.
- [13] B. FORNBERG AND J. ZUEV, *The Runge phenomenon and spatially variable shape parameters in RBF interpolation*, Comput. Math. Appl., 54 (2007), pp. 379–398.
- [14] M. GRIEBEL, C. RIEGER, AND B. ZWICKNAGL, *Multiscale approximation and reproducing kernel Hilbert space methods*, SIAM J. Numer. Anal., 53 (2015), pp. 852–873.
- [15] G. A. HAGEDORN, *Generating function and a Rodrigues formula for the polynomials in d-dimensional semi-classical wave packets*, Ann. Physics, 362 (2015), pp. 603–608.

- [16] E. LARSSON AND B. FORNBERG, *Theoretical and computational aspects of multivariate interpolation with increasingly flat radial basis functions*, *Comput. Math. Appl.*, 49 (2005), pp. 103–130.
- [17] E. LARSSON, E. LEHTO, A. HERYUDONO, AND B. FORNBERG, *Stable computation of differentiation matrices and scattered node stencils based on Gaussian radial basis functions*, *SIAM J. Sci. Comput.*, 35 (2013), pp. A2096–A2119.
- [18] E. LARSSON, V. SHCHERBAKOV, AND A. HERYUDONO, *A least squares radial basis function partition of unity method for solving PDEs*, *SIAM J. Sci. Comput.*, 39 (2017), pp. A2538–A2563.
- [19] S. D. MARCHI AND G. SANTIN, *A new stable basis for radial basis function interpolation*, *J. Comput. Appl. Math.*, 253 (2013), pp. 1–13.
- [20] M. MCCOURT AND G. E. FASSHAUER, *Stable likelihood computation for Gaussian random fields*, in *Recent Applications of Harmonic Analysis to Function Spaces, Differential Equations, and Data Science*, Springer, 2017, pp. 917–943.
- [21] J. RASHIDINIA, G. E. FASSHAUER, AND M. KHASI, *A stable method for the evaluation of Gaussian radial basis function solutions of interpolation and collocation problems*, *Comput. Math. Appl.*, 72 (2016), pp. 178–193.
- [22] A. E. TARWATER, *Parameter study of Hardys multiquadric method for scattered data interpolation*, Tech. Report UCRL-53670, Lawrence Livermore National Lab., CA (USA), 1985.
- [23] G. WATSON, *Notes on generating functions of polynomials:(2) Hermite polynomials*, *J. Lond. Math. Soc.*, 1 (1933), pp. 194–199.
- [24] G. B. WRIGHT AND B. FORNBERG, *Stable computations with flat radial basis functions using vector-valued rational approximations*, *J. Comput. Phys.*, 331 (2017), pp. 137–156.