An Introduction to the Hilbert-Schmidt SVD using Iterated Brownian Bridge Kernels

**Roberto Cavoretto · Gregory E. Fasshauer ·
Michael McCourt**

Abstract Kernel-based approximation methods—often in the form of radial basis functions—have been used for many years now and usually involve setting up a kernel matrix which may be ill-conditioned when the shape parameter of the kernel takes on extreme values, i.e., makes the kernel "flat". In this paper we present an algorithm we refer to as the *Hilbert-Schmidt SVD* and use it to emphasize two important points which—while not entirely new—present a paradigm shift under way in the practical application of kernel-based approximation methods: (i) it is not necessary to form the kernel matrix (in fact, it might even be a bad idea to do so), and (ii) it is not necessary to know the kernel in closed form. While the Hilbert-Schmidt SVD and its two implications apply to general positive definite kernels, we introduce in this paper a class of so-called *iterated Brownian bridge kernels* which allow us to keep the discussion as simple and accessible as possible.

Keywords Hilbert-Schmidt SVD · Iterated Brownian bridge kernels · Positive definite kernels · Stable interpolation · Splines

Mathematics Subject Classification (2000) 65D05 · 65D07 · 41A15

1 Introduction

Kernel-based methods are popular tools for problems of interpolation and differential equations [8,12], statistics [40], machine learning [30,41], and other fields. Their popularity stems from their inherently meshfree nature, providing an avenue to potentially avoid the "curse of dimensionality" [13,27]. Many different kernels

R. Cavoretto
Department of Mathematics "G. Peano", University of Turin, Turin, Italy
E-mail: roberto.cavoretto@unito.it

G. E. Fasshauer
Department of Applied Mathematics, Illinois Institute of Technology, Chicago, USA
E-mail: fasshauer@iit.edu

M. McCourt
Department of Mathematical and Statistical Sciences, University of Colorado, Denver, USA
E-mail: mccomic@mcs.anl.gov

exist, and several factors may be considered when choosing the appropriate kernel for a given application, including the numerical stability of computations involving that kernel.

The presence of free parameters influencing the shape and smoothness of kernels allows for high levels of accuracy, but may also introduce ill-conditioning into the standard problem formulation. For many very smooth kernels, or kernels with an increasingly flat parametrization, the computational error may prevent realization of the theoretically optimal accuracy. This problem was analyzed in, e.g., [5,11,23,33,39] and techniques to circumvent this in certain circumstances were discussed in, e.g., [15,16,17].

In [14], a technique was developed using Hilbert-Schmidt theory [35] to create a new basis for the Gaussian interpolant derived from the eigenfunction expansion of the Gaussian in $\mathbb{R}^d$; the new basis is devoid of the standard ill-conditioning. This change of basis approach was first used in the pioneering work [16]. Once increasingly flat Gaussian interpolants could be stably computed, the polynomial limit, as predicted in [11,22,23,33], could be numerically confirmed in arbitrary dimensions. Therefore, Gaussians can always produce at least the same accuracy as polynomials because the polynomial result can be obtained in the limit.

Since the eigenfunction expansion associated with the Gaussian kernel is rather complex and its implementation (which can be found in the MATLAB library `http://math.iit.edu/~mccomic/gaussqr/`) is non-trivial we have decided to present in this paper a completely transparent implementation of the *Hilbert-Schmidt SVD* which uses so-called *iterated Brownian bridge kernels*. The advantage of introducing this new class of kernels, defined on the interval $[0,1]$ with very specific boundary conditions, lies in the fact that the resulting MATLAB code is almost trivial and can be included in this paper (see Appendix). Moreover, iterated Brownian bridge kernels generalize the Brownian bridge kernel which plays an important role in many applications in statistics or finance (see, e.g., [4, Sec. 2.2.1], [18, Sec. 3.1], [32, Sec. 3.7], [38, Sec. 4.7.4], and our explanation in Remark 1).

The two main messages we would like to communicate with this paper are

1. The kernel-based solution of interpolation, approximation or differential equations problems can—and perhaps frequently should—be achieved *without ever forming the so-called kernel matrix* K (sometimes also referred to as Gram or collocation matrix). This, in particular, implies that the Hilbert-Schmidt SVD (see (3.6)) is not obtained by factoring the kernel matrix. However, the Hilbert-Schmidt SVD does represent a factorization of the kernel matrix. In other words, we can—if needed—obtain K from the Hilbert-Schmidt SVD, but not vice versa (as is done with the traditional SVD in linear algebra).

2. Kernel-based methods can proceed accurately and efficiently *without knowing a closed form of the kernel K*. A series representation of the kernel suffices, and may sometimes represent the extent of our understanding of a given kernel. In this paper we introduce the family of univariate iterated Brownian bridge kernels which are in some sense analogous to the well-known family of Matérn kernels on $\mathbb{R}^d$, i.e., both families are defined using two parameters with one denoting the smoothness of the kernel and the other its flatness or scale. Both the smoothness and scale of the iterated Brownian bridge kernels are encoded in their Hilbert-Schmidt eigenvalues, while the eigenfunctions are invariant for the entire family (see (2.8)).

It should be noted that neither of these messages is entirely new. The idea to not work with the kernel matrix was the motivation of the research of Fornberg and co-workers on the Contour-Padé method or the RBF-QR method (see, e.g., [15, 16, 17]. Another change of basis technique was proposed for conditionally positive definite functions in [3]. Kernels given primarily (or solely) in series form were the topic of, e.g., the papers [20, 44, 45].

It is our hope that the present paper, with its simple family of kernels and straightforward MATLAB implementation of the Hilbert-Schmidt SVD, provides a more transparent introduction to the issues associated with modern computational techniques for kernel-based approximation methods and therefore makes this topic accessible to a wider range of practitioners.

Another topic that will be mentioned in this paper is the fact that the "flat" limit of our iterated Brownian bridge kernels corresponds to a certain class of piecewise polynomial splines that were already featured in the literature on L-splines of the 1960s and 70s [36, 42]. In Section 5 we will provide a modern proof of the convergence orders of this subfamily of piecewise polynomial iterated Brownian bridge kernels using the framework of sampling inequalities in reproducing kernel Hilbert spaces. Unfortunately, even the modern proof techniques do not seem to enable us to generalize the results for the piecewise polynomial case from [42, Sect. 8] to our larger family of iterated Brownian bridge kernels, i.e., from the piecewise polynomial $\varepsilon = 0$ case to the more general case with $\varepsilon > 0$. Among other numerical experiments, we will verify these convergence orders in Section 6, and end the paper with a set of closing remarks.

2 Definition of Iterated Brownian Bridge Kernels

Our definition of iterated Brownian bridge kernels is motivated by several factors. On the one hand we wanted to create a two-parameter family similar to the Matérn kernels on $\mathbb{R}^d$, which are popular in many applications—especially in statistics, where techniques such as maximum likelihood estimation are used to obtain the "best" kernel for a given set of data. On the other hand, we wanted our kernel to give rise to a very simple Hilbert-Schmidt series. As we will now demonstrate, iterated Brownian bridge kernels address both of these concerns.

2.1 Review of Hilbert-Schmidt and Sturm-Liouville Eigenvalue Theory

Given a domain Ω, a weight function $\rho : \Omega \to \mathbb{R}^+$ and a continuous square integrable positive definite kernel $K : \Omega \times \Omega \to \mathbb{R}$, a Hilbert-Schmidt integral operator $\mathcal{K}$ is defined as

$$\mathcal{K}f = \int_\Omega K(\cdot, z)f(z)\rho(z)\mathrm{d}z. \qquad (2.1)$$

This operator maps functions in $L_2(\Omega, \rho)$ to $\mathcal{H}_K$, the reproducing kernel Hilbert space induced by K. From Hilbert-Schmidt theory [35] (or, equivalently, Mercer's theorem [26]) we know that K has a representation of the form

$$K(x, z) = \sum_{n=1}^{\infty} \lambda_n \varphi_n(x) \varphi_n(z), \qquad x, z \in \Omega. \qquad (2.2)$$

The eigenvalues λ_n and eigenfunctions φ_n are defined by

$$\mathcal{K}\varphi = \lambda\varphi, \tag{2.3}$$

and we know $\lambda_n > 0$ because K is positive definite, and the eigenfunctions are orthonormal with respect to the $L_2(\Omega, \rho)$ inner product. They are also orthogonal in $\mathcal{H}_K$, but that is not relevant to our discussion.

In [9, Chap. V] the argument is made that the Hilbert-Schmidt integral eigenvalue problem is "inverse" to the self-adjoint regular Sturm-Liouville eigenvalue problem

$$\mathcal{L}\varphi = \frac{1}{\lambda}\varphi\rho, \tag{2.4}$$

where the differential operator $\mathcal{L}$ is defined so that K is its Green's kernel, i.e., $\mathcal{L}K(x,z) = \delta(x-z)$. Suitable boundary conditions (in the integral formulation reflected in the definition of $\mathcal{H}_K$, the range of $\mathcal{K}$) must also be imposed to uniquely define K through $\mathcal{L}$, but in doing so, the eigenvalues and eigenfunctions of (2.4) can be used in (2.2) to define K. This approach is used in Section 2.2 to find the eigenexpansion of the iterated Brownian bridge kernels.

2.2 Iterated Brownian Bridge Kernels as Green's Kernels of an Iterated Helmholtz BVP

The connection between the Hilbert-Schmidt and Sturm-Liouville eigenvalue problems reviewed in the previous subsection allows us to view the Mercer series as a generalized Fourier series. We now consider the following iterated modified Helmholtz differential operator of order 2β

$$\mathcal{L}_{\beta,\varepsilon} = \left(-\mathcal{D}^2 + \varepsilon^2\mathcal{I}\right)^\beta, \qquad \beta \in \mathbb{N}\setminus\{0\}, \quad \varepsilon \geq 0, \tag{2.5}$$

where $\mathcal{D}$ denotes the usual univariate derivative $\frac{\mathrm{d}}{\mathrm{d}x}$, $\mathcal{I}$ is the identity operator (i.e., $\mathcal{I}\varphi = \varphi$), and ε acts as a *shape parameter* (or *tension parameter*). In this paper we will focus on the domain $\Omega = [0,1]$, although more general domains can be addressed with an appropriate transformation.

We define the Sturm-Liouville eigenvalue problem

$$\mathcal{L}_{\beta,\varepsilon}\varphi(x) = \lambda^{-1}\varphi(x), \qquad x \in [0,1],$$

with boundary conditions conveniently chosen to be of the form

$$\varphi^{(2j)}(0) = \varphi^{(2j)}(1) = 0, \qquad j = 0, \ldots, \beta-1. \tag{2.6}$$

Note that the weight function, which appeared in the definition of $\mathcal{K}$, is $\rho \equiv 1$ for our purposes. For this ODE boundary value problem one can easily check that the corresponding eigenvalues and (normalized) eigenfunctions are given by

$$\lambda_n = \left(n^2\pi^2 + \varepsilon^2\right)^{-\beta}, \qquad \varphi_n(x) = \sqrt{2}\sin(n\pi x), \qquad n = 1, 2, \ldots, \tag{2.7}$$

so that the generalized Fourier series of the *iterated Brownian bridge kernels* $K_{\beta,\varepsilon}$ is

$$K_{\beta,\varepsilon}(x,z) = \sum_{n=1}^{\infty} \lambda_n \varphi_n(x)\varphi_n(z) = \sum_{n=1}^{\infty} \left(n^2\pi^2 + \varepsilon^2\right)^{-\beta} 2\sin(n\pi x)\sin(n\pi z). \quad (2.8)$$

As expected, the eigenfunctions are $L_2(\Omega)$ orthonormal, and because all the eigenvalues are positive, we know that the iterated Brownian bridge kernels are positive definite.

By using a tensor product form of our kernels, it is possible to work with them in higher dimensions, but no discussion of that setting is presented here.

Since the iterated Brownian bridge kernels are Green's functions of the iterated modified Helmholtz operator it follows that the free parameter $\beta \in \mathbb{N} \setminus \{0\}$ present in (2.5) generates a family of kernels of increasing smoothness, i.e., the kernel functions $K_{\beta,\varepsilon}(\cdot, z)$ of order β have $2\beta - 2$ smooth derivatives at their *center z*. The additional free parameter $\varepsilon \geq 0$ determines the shape of the kernels by providing a certain amount of *tension* that tends to localize the kernel around its center z as ε increases. This is illustrated in Figures 1 and 2.

Remark 1 We chose the name "iterated Brownian bridge" for our family of kernels since one could also derive the kernels by repeatedly applying the integral operator $\mathcal{K}$ to the basic kernel $K_{1,\varepsilon}$. One should note, however, that the Brownian bridge kernel used widely in the literature refers only to the specific choice of $\varepsilon = 0$, i.e., the kernel $K_{1,0}$, and therefore it would be more precise to refer to our kernels as *iterated generalized Brownian bridge kernels*, but that seems a bit unwieldy. The construction itself is very general and proceeds as follows. Using the integral operator (2.1) and the absolutely converging Hilbert-Schmidt series (2.2) of a symmetric positive definite kernel K defined on $\Omega \subset \mathbb{R}^d$ we can see that for a fixed $y \in \Omega$ one has

$$\mathcal{K}K(x,y) = \int_{\Omega} K(x,z)K(z,y)\rho(z)\mathrm{d}z$$

$$= \int_{\Omega} K(x,z) \sum_{n=1}^{\infty} \lambda_n \varphi_n(y)\varphi_n(z)\rho(z)\mathrm{d}z$$

$$= \sum_{n=1}^{\infty} \lambda_n \varphi_n(y) \int_{\Omega} K(x,z)\varphi_n(z)\rho(z)\mathrm{d}z.$$

Now, since the φ_n are the eigenfunctions of $\mathcal{K}$ corresponding to λ_n (see (2.3)) we can further conclude that

$$\mathcal{K}K(x,y) = \sum_{n=1}^{\infty} \lambda_n^2 \varphi_n(x)\varphi_n(y),$$

which is a new kernel with the same eigenfunctions as K, but whose eigenvalues are the squares of those of K. It should be noted that for a translation-invariant kernel—and therefore of course also for a radial kernel—the operation described above is a convolution. It is clear that the iterated kernel is smoother than the original one. The idea of constructing smoother kernels via iteration is a classical one and already described in [9, Sect. III.5.3]. For our purposes we prefer the derivation in terms of the iterated Helmholtz operator since the effect of the boundary conditions is explicit.

Remark 2 If, in particular, $\Omega = \mathbb{R}^d$ and $\mathcal{L}_{\beta,\varepsilon}^{(d)} = \left(-\Delta + \varepsilon^2 \mathcal{I}\right)^\beta$, $\beta > \frac{d}{2}$, with d-dimensional Laplacian Δ, then

$$K_{\beta,\varepsilon}^d(\boldsymbol{x}, \boldsymbol{z}) \doteq K_{\beta-d/2}\left(\varepsilon \|\boldsymbol{x} - \boldsymbol{z}\|\right) \left(\varepsilon \|\boldsymbol{x} - \boldsymbol{z}\|\right)^{\beta-d/2}, \qquad \beta > \frac{d}{2},$$

are the so-called *Matérn kernels* on $\mathbb{R}^d$. Here $K_{\beta-d/2}$ are modified Bessel functions of the second kind[1] and $\doteq$ denotes equality up to a constant factor.

Remark 3 While (2.8) accurately defines the kernel for any fixed ε and β, this infinite series representation may not seem as satisfying in comparison to the clean (closed) form representation available for the full space Matérn kernels. Among other things, we will show below that a iterated Brownian bridge kernel, given only by its series expansion, can be used just as effectively as a closed form kernel provided we combine it with the Hilbert-Schmidt SVD. Moreover, a time existed when modified Bessel functions of the second kind were considered only in series form, so defining our new kernels in series form is similarly appropriate.

Having said this, we will discuss those special cases for which closed forms of the kernel are known in Section 4.

3 Derivation of the Hilbert-Schmidt SVD

The discussion in Section 2.2 focused on the definition of iterated Brownian bridge kernels in terms of eigenvalues and eigenfunctions of a Sturm-Liouville eigenvalue problem with specific boundary conditions. These eigenvalues and eigenfunctions provide a representation of the kernel as a Hilbert-Schmidt series

$$K_{\beta,\varepsilon}(x, z) = \sum_{n=1}^{\infty} \lambda_n \varphi_n(x) \varphi_n(z).$$

3.1 An Infinite Matrix Decomposition of K based on Hilbert-Schmidt Series

Our Hilbert-Schmidt SVD (defined in (3.6) below) can be applied to any positive definite kernel K once it is written as a Hilbert-Schmidt series. Therefore, the discussion in this subsection involves arbitrary positive definite kernels and applies in full generality, in particular on arbitrary domains in arbitrary space dimensions d. As mentioned in the introduction, the standard or *direct* approach to the solution of most kernel-based approximation problems is to generate the kernel matrix $\mathsf{K} = \left(K(\boldsymbol{x}_i, \boldsymbol{x}_j)\right)_{i,j=1}^{N}$, where N is the number of given data as well as the number of kernel centers. However, as is well-known to practitioners, the matrix K is prone to ill-conditioning—especially if K contains a shape parameter related to "flatness" (potentially resulting in nearly identical rows or columns of K).

The main goal of the Hilbert-Schmidt SVD (and similar techniques such as RBF-QR) is to find a decomposition of the matrix K *without ever forming* K, and

---

[1] We apologize for overloading the letter K. However, the modified Bessel functions appear only in this remark. All other uses of K denote kernel *functions*, while $\mathcal{K}$ is used for the associated integral *operator* and K denotes the associated kernel *matrices*.

to use this decomposition to overcome the ill-conditioning associated with K. This is a transformation from the standard basis $\{K(\cdot, \boldsymbol{x}_1), \ldots, K(\cdot, \boldsymbol{x}_N)\}$ to a new, and better conditioned, basis which we will denote as $\{\psi_1, \ldots, \psi_N\}$ below. For us, the starting point of this change of basis process is the Hilbert-Schmidt series of the kernel.

In matrix terminology, a single kernel value (and thus a generic entry in the kernel matrix K) can be written as a weighted inner product involving a diagonal matrix containing all the eigenvalues

$$K(\boldsymbol{x}, \boldsymbol{z}) = \sum_{n=1}^{\infty} \lambda_n \varphi_n(x) \varphi_n(z) = \boldsymbol{\phi}(\boldsymbol{x})^T \Lambda \boldsymbol{\phi}(\boldsymbol{z}), \tag{3.1}$$

where we have used the notation

$$\boldsymbol{\phi}(\boldsymbol{x}) = \begin{pmatrix} \varphi_1(\boldsymbol{x}) \\ \vdots \\ \varphi_N(\boldsymbol{x}) \\ \vdots \end{pmatrix}, \qquad \Lambda = \begin{pmatrix} \lambda_1 & & & \\ & \ddots & & \\ & & \lambda_N & \\ & & & \ddots \end{pmatrix}.$$

Since the kernel matrix K is given by

$$\mathsf{K} = \begin{pmatrix} K(\boldsymbol{x}_1, \boldsymbol{x}_1) & \cdots & K(\boldsymbol{x}_1, \boldsymbol{x}_N) \\ & \vdots & \\ K(\boldsymbol{x}_N, \boldsymbol{x}_1) & \cdots & K(\boldsymbol{x}_N, \boldsymbol{x}_N) \end{pmatrix},$$

a generic row of K can be written using the eigenfunction expansion (3.1) as

$$\begin{aligned} \boldsymbol{k}(\boldsymbol{x})^T &= \begin{pmatrix} K(\boldsymbol{x}, \boldsymbol{x}_1) & \cdots & K(\boldsymbol{x}, \boldsymbol{x}_N) \end{pmatrix} \\ &= \begin{pmatrix} \boldsymbol{\phi}(\boldsymbol{x})^T \Lambda \boldsymbol{\phi}(\boldsymbol{x}_1) & \cdots & \boldsymbol{\phi}(\boldsymbol{x})^T \Lambda \boldsymbol{\phi}(\boldsymbol{x}_N) \end{pmatrix} \\ &= \boldsymbol{\phi}(\boldsymbol{x})^T \Lambda \begin{pmatrix} \boldsymbol{\phi}(\boldsymbol{x}_1) & \cdots & \boldsymbol{\phi}(\boldsymbol{x}_N) \end{pmatrix} \\ &= \boldsymbol{\phi}(\boldsymbol{x})^T \Lambda \Phi^T, \end{aligned} \tag{3.2}$$

where we have defined the matrix $\Phi^T = \begin{pmatrix} \boldsymbol{\phi}(\boldsymbol{x}_1) & \ldots & \boldsymbol{\phi}(\boldsymbol{x}_N) \end{pmatrix}$. The row vector $\boldsymbol{k}(\cdot)^T$ collects the data-dependent *basis functions* $K(\cdot, \boldsymbol{x}_1), \ldots, K(\cdot, \boldsymbol{x}_N)$, and (3.2) can be interpreted as an identity relating the N-dimensional data-dependent basis in $\boldsymbol{k}(\cdot)^T$ to the infinite-dimensional data-independent eigenfunction basis in $\boldsymbol{\phi}(\cdot)^T$. The transformation relating these two bases (spaces) is given by the matrix product $\Lambda \Phi^T$ and, of course, depends on the data locations $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_N$, that appear in $(\Phi)_{ij} = \varphi_j(\boldsymbol{x}_i)$.

With this additional notation we can now form the entire matrix K by stacking such rows of $\boldsymbol{k}(\cdot)^T$ evaluated at the different points $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_N$, i.e.,

$$\mathsf{K} = \begin{pmatrix} \boldsymbol{k}(\boldsymbol{x}_1)^T \\ \vdots \\ \boldsymbol{k}(\boldsymbol{x}_N)^T \end{pmatrix} = \begin{pmatrix} \boldsymbol{\phi}(\boldsymbol{x}_1)^T \\ \vdots \\ \boldsymbol{\phi}(\boldsymbol{x}_N)^T \end{pmatrix} \Lambda \Phi^T = \Phi \Lambda \Phi^T. \tag{3.3}$$

The eigen-decomposition of K given in (3.3) does not yet represent the Hilbert-Schmidt SVD. It has served to introduce our matrix-vector notation, but has not

done anything to alleviate the potential ill-conditioning in K associated with the eigenvalues in Λ. One could envision simply truncating the matrix Λ (and therefore the Hilbert-Schmidt series expansion) by zeroing out eigenvalues with an index n greater than some truncation index M, but then we would no longer be working in the standard kernel space $\text{span}\{K(\cdot, \boldsymbol{x}_1), \ldots, K(\cdot, \boldsymbol{x}_N)\}$. The Hilbert-Schmidt SVD derived in the next subsection maintains the standard kernel space without encountering the standard ill-conditioning.

3.2 Obtaining a New Basis via the Hilbert-Schmidt SVD

At this point, all we have done is to rewrite the kernel matrix as the product of three larger matrices. Although this helps us evaluate the kernel function using only the series expansion, the result of this process is no different than simply finding each kernel element individually via a weighted inner product, i.e.,

$$
\mathsf{K} = \begin{pmatrix} \boldsymbol{\phi}(\boldsymbol{x}_1)^T \Lambda \boldsymbol{\phi}(\boldsymbol{x}_1) & \cdots & \boldsymbol{\phi}(\boldsymbol{x}_1)^T \Lambda \boldsymbol{\phi}(\boldsymbol{x}_N) \\ & \vdots & \\ \boldsymbol{\phi}(\boldsymbol{x}_N)^T \Lambda \boldsymbol{\phi}(\boldsymbol{x}_1) & \cdots & \boldsymbol{\phi}(\boldsymbol{x}_N)^T \Lambda \boldsymbol{\phi}(\boldsymbol{x}_N) \end{pmatrix}.
$$

From a performance standpoint this already provides an improvement since computing the matrix via (3.3) is likely more efficient because it involves "level 3" BLAS operations (matrix-matrix products) rather than "level 1" BLAS operations (vector inner products $\boldsymbol{\phi}(\boldsymbol{x})^T \Lambda \boldsymbol{\phi}(\boldsymbol{z})$). See [19] for a review of this idea.

However, we have not yet addressed the potential ill-conditioning of K. We can improve on the matrix structure by considering blocks of Φ and Λ. Define

$$
\Phi = (\Phi_1 \ \ \Phi_2), \qquad \Lambda = \begin{pmatrix} \Lambda_1 & \\ & \Lambda_2 \end{pmatrix},
$$

where $\Phi_1, \Lambda_1 \in \mathbb{R}^{N \times N}$, $\Phi_2 \in \mathbb{R}^{N \times \infty}$, $\Lambda_2 \in \mathbb{R}^{\infty \times \infty}$. Starting with (3.2), we can write

$$
\boldsymbol{k}(\boldsymbol{x})^T = \boldsymbol{\phi}(\boldsymbol{x})^T \Lambda \Phi^T = \boldsymbol{\phi}(\boldsymbol{x})^T \begin{pmatrix} \Lambda_1 & \\ & \Lambda_2 \end{pmatrix} \begin{pmatrix} \Phi_1^T \\ \Phi_2^T \end{pmatrix}
$$

$$
= \boldsymbol{\phi}(\boldsymbol{x})^T \begin{pmatrix} \mathsf{I}_N \\ \Lambda_2 \Phi_2^T \Phi_1^{-T} \Lambda_1^{-1} \end{pmatrix} \Lambda_1 \Phi_1^T,
$$

where $\mathsf{I}_N \in \mathbb{R}^{N \times N}$ is the $N \times N$ identity matrix. To clean this up further, we define a new vector function

$$
\boldsymbol{\psi}(\boldsymbol{x})^T = \boldsymbol{\phi}(\boldsymbol{x})^T \begin{pmatrix} \mathsf{I}_N \\ \Lambda_2 \Phi_2^T \Phi_1^{-T} \Lambda_1^{-1} \end{pmatrix} \tag{3.4}
$$

so that

$$
\boldsymbol{k}(\boldsymbol{x})^T = \boldsymbol{\psi}(\boldsymbol{x})^T \Lambda_1 \Phi_1^T. \tag{3.5}
$$

It may be helpful to emphasize that we *do not* obtain the new basis functions in $\boldsymbol{\psi}(\cdot)^T$ from those in $\boldsymbol{k}(\cdot)^T$ as (3.5) might suggest. Instead, we rely entirely on the eigenvalues and eigenfunctions and use (3.4). The reader may confirm this in the

MATLAB code in the Appendix, e.g., `Psi_interp = Phi_interp*[I_N;Correction]` along with the definition of `Correction`.

Replacing the rows of K with (3.5) yields

$$\mathsf{K} = \begin{pmatrix} \boldsymbol{k}(\boldsymbol{x}_1)^T \\ \vdots \\ \boldsymbol{k}(\boldsymbol{x}_N)^T \end{pmatrix} = \begin{pmatrix} \boldsymbol{\psi}(\boldsymbol{x}_1)^T \\ \vdots \\ \boldsymbol{\psi}(\boldsymbol{x}_N)^T \end{pmatrix} \Lambda_1 \Phi_1^T = \Psi \Lambda_1 \Phi_1^T, \tag{3.6}$$

where Ψ is defined analogously to K. We call this factorization the *Hilbert-Schmidt SVD* of K because the diagonal matrix in the middle is filled with the first N Hilbert-Schmidt eigenvalues. The matrices Ψ and Φ_1 are not orthogonal, as in a standard SVD, but they are generated by the $L_2(\Omega, \rho)$ orthogonal eigenfunctions.

To shed a little more light on the nature of the new basis functions we look more carefully at the definition of $\boldsymbol{\psi}(\boldsymbol{x})$ in (3.4), i.e., we expand out $\boldsymbol{\phi}(\boldsymbol{x})^T$ to obtain

$$\boldsymbol{\psi}(\boldsymbol{x})^T = \big(\varphi_1(\boldsymbol{x}) \cdots \varphi_N(\boldsymbol{x})\big) + \big(\varphi_{N+1}(\boldsymbol{x}) \cdots\big) \Lambda_2 \Phi_2^T \Phi_1^{-T} \Lambda_1^{-1}.$$

Here we can see that the jth element of our new basis collected in $\boldsymbol{\psi}(\cdot)^T$ is the jth eigenfunction, plus a correction in the form of a linear combination of all the eigenfunctions with index greater than N. This infinite length correction (recall both Λ_2 and Φ_2 have infinite dimension) is guaranteed to make a finite contribution because the original Mercer series was uniformly convergent.

In Section 3.3 below we will address the practical question of determining at what point we can stop considering higher-order eigenfunctions in $\boldsymbol{\phi}(\cdot)^T$, i.e., how to choose a truncation length M for the infinite Hilbert-Schmidt series and the resulting infinite matrices Λ_2 and Φ_2. Note that such a truncation will—within the tolerance demanded for the truncation length M—lead to an alternate basis $\{\psi_1, \ldots, \psi_N\}$ for the standard kernel space span$\{K(\cdot, \boldsymbol{x}_1), \ldots, K(\cdot, \boldsymbol{x}_N)\}$. This new basis will hopefully be better conditioned than the standard kernel basis since it is generated from small "corrections" to the orthogonal eigenfunctions $\varphi_1, \ldots, \varphi_N$.

3.3 Truncating the Hilbert-Schmidt Series

Thus far, our discussion of the Hilbert-Schmidt SVD has been completely general and has resulted in a factorization of K containing a matrix Ψ that is built using the infinite matrices Λ_2 and Φ_2. For practical purposes these components need to be truncated. This can be accomplished by truncating the Hilbert-Schmidt series.

From now on our discussion specializes to iterated Brownian bridge kernels. Because the eigenvalues of the series (2.8) decay monotonically, and the eigenfunctions are bounded above by $\sqrt{2}$, at some point all the remaining terms in the series contribute a negligible amount to the summation and can be ignored. We choose our truncation point M by first picking a tolerance σ_{tol} and requiring that the ratio of the smallest and largest eigenvalues be less than the tolerance. When written out explicitly, the ratio of two eigenvalues

$$\frac{\lambda_M}{\lambda_N} < \sigma_{\text{tol}}, \qquad M > N,$$

can be solved for the necessary M:

$$M_{\text{tol}} = \frac{1}{\pi} \sqrt{\sigma_{\text{tol}}^{-1/\beta}(N^2\pi^2 + \varepsilon^2) - \varepsilon^2}. \tag{3.7}$$

Table 1 shows truncation values for various parametrizations, when N is set to 1. For example, the kernel $K_{3,10}$ can be represented with precision $\sigma_{\text{tol}} = 10^{-15}$ if we use a truncation length $M_{\text{tol}} = 1000$. For any desired precision σ_{tol}, a smoother kernel (i.e., greater β) can be represented accurately with a shorter series. The same is true as we decrease ε for a kernel of fixed smoothness.

Parameters		Precisions		
β	ε	10^{-5}	10^{-10}	10^{-15}
1	.1	3×10^2	1×10^5	3×10^7
1	1	3×10^2	1×10^5	3×10^7
1	10	1×10^3	3×10^5	1×10^8
3	10	2×10^1	2×10^2	1×10^3
5	10	1×10^1	3×10^2	1×10^2
7	10	7×10^0	2×10^1	4×10^1

Table 1: Value of the truncation length M_{tol} required to reach certain precisions σ_{tol} for series generated by various combinations of ε and β. The precision column is the ratio of the last and first eigenvalues, λ_M/λ_1.

A thorough analysis of truncation lengths for general kernels given in series form (which covers also much more general multiscale kernels such as those of [29]) is presented in [20].

In some sense, computing the kernel values with a series rather than closed form is entirely appropriate. Computing a value of the iterated Brownian bridge kernel with the series would be no different than evaluating $\Gamma(x)$ or $\text{erf}(x)$. Even evaluating the Gaussian kernel could be done with a series expansion. Since we show in the next section that iterated Brownian bridge kernels are special kinds of L-splines, this remark follows naturally from research which has shown that B-splines tend to Gaussians as the smoothness index β tends to infinity (see, e.g., [1,2,6] and [7, Chapter 8]). However, in using an eigenfunction series directly (as in (3.1)) we ignore the special structure of this series which is not present in a standard Taylor series. Exploiting this structure allows for interpolation without directly computing the kernel values. This is accomplished by using the Hilbert-Schmidt SVD or RBF-QR technique [14,16].

3.4 Relation to RBF-QR and Other Practical Considerations

The use of the term RBF-QR in designing a new basis for kernel-based methods comes from the computation of the $\Phi_2^T \Phi_1^{-T}$ term in the data-dependent correction in (3.4) using an additional QR factorization. Often it is preferable to first perform a QR factorization on the (presumably now M-length) Φ matrix to produce

$$\Phi = \begin{pmatrix} \Phi_1 & \Phi_2 \end{pmatrix} = Q \begin{pmatrix} R_1 & R_2 \end{pmatrix} = \begin{pmatrix} QR_1 & QR_2 \end{pmatrix},$$

which allows for $\Phi_2^T \Phi_1^{-T} = \mathsf{R}_2^T \mathsf{R}_1^{-T}$. This is a more stable computation than LU factorization, which may be preferable depending on the scale of the various eigenfunctions. The iterated Brownian bridge eigenfunctions are all on the same scale (because they are all sines), but other kernels which have employed the QR factorization have found the extra stability useful [16].

In the above derivation we have assumed that Φ_1 and Λ_1 are invertible, which should be obvious for Λ_1 since it is a diagonal matrix with positive eigenvalues on the diagonal. The inverse of Φ_1 exists because the eigenfunctions φ_j, $1 \leq j \leq N$, are linearly independent, and Φ_1 would be the interpolation matrix associated with interpolation at the N distinct data locations. Because that interpolant must be unique, the matrix Φ_1 must be invertible. Note that while this is straightforward for this set of eigenfunctions on this domain, other kernels may require more care— especially in higher dimensions.

A significant benefit to this HS-SVD approach is a more stable interpolation system. Traditionally, inverting K could introduce unacceptable amounts of instability because of its ill-conditioning. This was discussed in the context of RBF-QR applied to Gaussians in [14]. The HS-SVD (with or without the optional QR step) allows us to isolate the ill-conditioning primarily (although not entirely) in the Λ_1 factor and resolve the $\Lambda_2 \Phi_2^T \Phi_1^{-T} \Lambda_1^{-1}$ term safely. This effect will become apparent in Section 6 for larger values of β.

3.5 How the Hilbert-Schmidt SVD Transforms the Interpolation Problem

The scattered data interpolation problem presents N distinct data locations $\boldsymbol{x}_j$ and associated function values $y_j = f(\boldsymbol{x}_j)$, $1 \leq j \leq N$, and asks for an approximation s which is equal to f at the data locations. To solve this with kernels, we assume that the solution is written as

$$s(\boldsymbol{x}) = \sum_{j=1}^{N} c_j K(\boldsymbol{x}, \boldsymbol{x}_j).$$

Demanding equality to the function values at the data locations produces the linear system

$$\mathsf{K}\boldsymbol{c} = \boldsymbol{y}, \tag{3.8}$$

where $\boldsymbol{c} = (c_1, \ldots, c_N)^T$ and $\boldsymbol{y} = (y_1, \ldots, y_N)^T$. In this vector notation, values of s can be evaluated as

$$s(\boldsymbol{x}) = \boldsymbol{k}(x)^T \boldsymbol{c} = \boldsymbol{k}(x)^T \mathsf{K}^{-1} \boldsymbol{y}. \tag{3.9}$$

To see the benefit of using the Hilbert-Schmidt SVD of K, we can study the structure of our interpolant (3.9),

$$\begin{aligned} s(\boldsymbol{x}) &= \boldsymbol{k}(\boldsymbol{x})^T \mathsf{K}^{-1} \boldsymbol{y} \\ &= \boldsymbol{\psi}(\boldsymbol{x})^T \Lambda_1 \Phi_1^T \Phi_1^{-T} \Lambda_1^{-1} \Psi^{-1} \boldsymbol{y} \\ &= \boldsymbol{\psi}(\boldsymbol{x})^T \Psi^{-1} \boldsymbol{y}. \end{aligned} \tag{3.10}$$

This remaining expression suggests that we have developed a mechanism for evaluating our interpolant s in the basis $\boldsymbol{\psi}(\cdot)^T$ rather than its standard basis $\boldsymbol{k}(\cdot)^T$. Moreover, the *cardinal functions* associated with interpolation at $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_N$— usually obtained as $\boldsymbol{k}(\boldsymbol{x})^T \mathsf{K}^{-1}$—can be computed more stably via $\boldsymbol{\psi}(\boldsymbol{x})^T \boldsymbol{\Psi}^{-1}$ (see Fig. 3 and our further discussion there).

The system of interest for solving the scattered data interpolation problem now is

$$\boldsymbol{\Psi} \boldsymbol{b} = \boldsymbol{y}$$

rather than (3.8) as it would be in the standard basis.

4 Closed Form Representations of Iterated Brownian Bridge Kernels

In Section 2.2 we mentioned that a iterated Brownian bridge kernel $K_{\beta,\varepsilon}$ has smoothness order $2\beta - 2$, and that its shape is determined by the parameter ε. We will now show that the kernels $K_{\beta,0}$ are in fact piecewise polynomial splines, thereby establishing that the "flat" $\varepsilon \to 0$ limits of iterated Brownian bridge kernel interpolants are given as piecewise polynomial spline interpolants. This observation is in line with the findings of [39], where the "flat" limits of piecewise smooth RBF interpolants were found to be polyharmonic spline interpolants.

In this section we derive closed form representations for the piecewise polynomial spline kernels, and for the kernels $K_{\beta,\varepsilon}$ when $\beta = 1$ or 2.

4.1 Iterated Brownian Bridge Kernels for $\varepsilon = 0$: The Piecewise Polynomial Spline Case

If we restrict the differential operator in (2.5) to the case $\varepsilon = 0$, the eigenvalues and eigenfunctions from (2.7) become

$$\lambda_n = (n\pi)^{-2\beta}, \qquad \varphi_n(x) = \sqrt{2}\sin(n\pi x), \qquad n = 1, 2, \ldots.$$

Using these specific eigenvalues and eigenfunctions, a closed form for the Mercer series expansion of the kernel $K_{\beta,0}$ can be obtained with the help of the standard trigonometric identity $2\sin A \sin B = \cos(A - B) - \cos(A + B)$ with $A = n\pi x$ and $B = n\pi z$ along with two applications of the cosine series expansion of *Bernoulli polynomials* (see, e.g., [10, Eq. 24.8.1] and [28])

$$B_{2\beta}(t) = (-1)^{\beta+1} \frac{2(2\beta)!}{(2\pi)^{2\beta}} \sum_{n=1}^{\infty} n^{-2\beta} \cos(2\pi n t), \qquad 0 \le t \le 1, \ \beta = 1, 2, \ldots, \quad (4.1)$$

setting $t = \frac{x-z}{2}$ and $t = \frac{x+z}{2}$, respectively.

Since (4.1) requires $0 \le t \le 1$ we need to treat the cases $x \ge z$ and $z \ge x$ separately. However, it is possible to combine the two resulting formulas into the desired symmetric closed form representation

$$K_{\beta,0}(x,z) = (-1)^{\beta-1} \frac{2^{2\beta-1}}{(2\beta)!} \left[B_{2\beta}\left(\frac{|x-z|}{2}\right) - B_{2\beta}\left(\frac{x+z}{2}\right) \right], \qquad (4.2)$$

which is valid for any $0 \leq x, z \leq 1$.

Bernoulli polynomials of degree n can be expressed as (see, e.g., [10, Eq. 24.6.7] and [28])

$$B_n(x) = \sum_{k=0}^{n} \frac{1}{k+1} \sum_{j=0}^{k} (-1)^j \binom{k}{j} (x+j)^n, \qquad (4.3)$$

so that the first few Bernoulli polynomials are

$$B_0(x) = 1, \quad B_1(x) = x - \frac{1}{2}, \quad B_2(x) = x^2 - x + \frac{1}{6}, \quad B_3(x) = x^3 - \frac{3}{2}x^2 + \frac{1}{2}x.$$

Since it is apparent from (4.3) that the leading-order terms in x cancel, the kernel $K_{\beta,0}$ is indeed a piecewise polynomial of odd degree $2\beta - 1$. Knowing (4.3), we can easily verify that (4.2) for $\beta = 1, 2$ leads to piecewise linear polynomials of the form

$$K_{1,0}(x, z) = \min(x, z) - xz = \begin{cases} x - xz, & 0 \leq x \leq z, \\ z - xz, & z \leq x \leq 1, \end{cases}$$

(also known as the *Brownian bridge kernel*) and piecewise cubic polynomials

$$K_{2,0}(x, z) = \begin{cases} \frac{1}{6}x(1-z)(x^2 + z^2 - 2z), & 0 \leq x \leq z, \\ \frac{1}{6}(1-x)z(x^2 + z^2 - 2x), & z \leq x \leq 1. \end{cases}$$

For a fixed z, $K_{3,0}(x, z)$ gives piecewise quintic polynomials in x, and so on. Thus, (4.2) allows us to compute the piecewise polynomial kernel $K_{\beta,0}$ satisfying the boundary conditions (2.6) in closed form for any β.

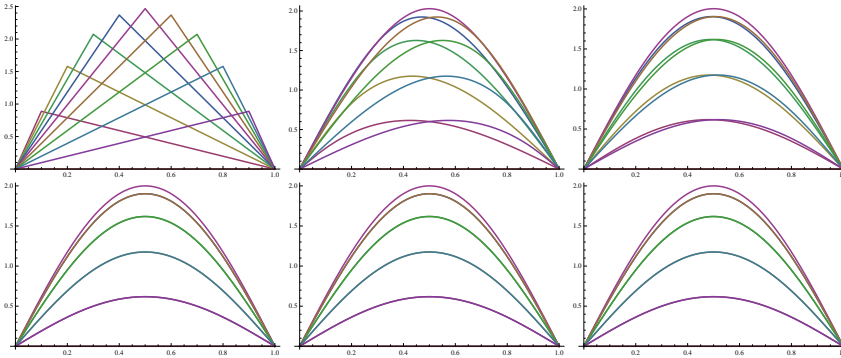Fig. 1: Copies of the piecewise polynomial spline kernel $K_{\beta,0}$. Top: $\beta = 1$ (left), $\beta = 2$ (middle), $\beta = 3$ (right); bottom: $\beta = 7$ (left), $\beta = 13$ (middle), $\beta = 20$ (right).

In Fig. 1 we provide plots of $K_{\beta,0}(\cdot, z)$ for different values of β and $z \in \{\frac{j}{10}, j = 1, \ldots, 9\}$. Note that the kernels corresponding to $z = 0$ and $z = 1$ are identically equal to zero due to the boundary conditions we have imposed. Since the kernels get rather small as β increases we have normalized all kernel plots by multiplying the kernels by $1/\lambda_1 = \pi^{2\beta}$. Clearly, this is a not a good basis for practical implementations since the set of translates of $K_{\beta,0}(\cdot, z)$ for different centers z becomes very nearly linearly dependent for larger β. In fact, the plots for $\beta \geq 7$ are nearly indistinguishable.

4.2 Iterated Brownian Bridge Kernels for $\varepsilon > 0$

By allowing ε to vary beyond 0, the resulting kernels $K_{\beta,\varepsilon}$ are no longer piecewise polynomials. This can be useful when attempting to approximate certain functions (as we show in Section 6) but it also renders the closed form (4.2) useless, requiring the derivation of a new formula. Unfortunately, as of this writing, the authors are unaware of any such closed form appropriate for all β when $\varepsilon > 0$.

Individual $K_{\beta,\varepsilon}$ closed forms can be derived by solving the Green's function boundary value problem

$$\mathcal{L}_{\beta,\varepsilon} K_{\beta,\varepsilon}(x,z) = \delta(x-z), \qquad x \in (0,1),$$
$$\frac{\partial^{2j}}{\partial x^{2j}} K_{\beta,\varepsilon}(x,z) = 0, \qquad x \in \{0,1\},\ j \in \{0,...,\beta-1\},$$

where $z \in (0,1)$ is fixed, the operator $\mathcal{L}_{\beta,\varepsilon}$ is applied to the first argument of K, and δ is the Dirac delta function. This is equivalent to the boundary value problem

$$\mathcal{L}_{\beta,\varepsilon} K_{\beta,\varepsilon}(x,z) = 0, \qquad x \in (0,1) \setminus \{z\},$$
$$\frac{\partial^{2j}}{\partial x^{2j}} K_{\beta,\varepsilon}(x,z) = 0, \qquad x \in \{0,1\},\ j \in \{0,1,...,\beta-1\},$$
$$\lim_{x \to z^-} \frac{\partial^j}{\partial x^j} K_{\beta,\varepsilon}(x,z) = \lim_{x \to z^+} \frac{\partial^j}{\partial x^j} K_{\beta,\varepsilon}(x,z), \qquad j \in \{0,1,...,2\beta-2\},$$
$$\lim_{x \to z^-} \frac{\partial^{2\beta-1}}{\partial x^{2\beta-1}} K_{\beta,\varepsilon}(x,z) = \lim_{x \to z^+} \frac{\partial^{2\beta-1}}{\partial x^{2\beta-1}} K_{\beta,\varepsilon}(x,z) + 1,$$

which can be solved for any fixed β using standard tools from classical differential equations.

When $\beta = 1$, the closed form solution is succinct,

$$K_{1,\varepsilon}(x,z) = \frac{\sinh\left(\varepsilon \min(x,z)\right) \sinh\left(\varepsilon(1-\max(x,z))\right)}{\varepsilon \sinh(\varepsilon)}$$
$$= \begin{cases} \frac{\sinh(\varepsilon x)\sinh(\varepsilon(1-z))}{\varepsilon\sinh(\varepsilon)}, & 0 \leq x \leq z, \\ \frac{\sinh(\varepsilon z)\sinh(\varepsilon(1-x))}{\varepsilon\sinh(L\varepsilon)}, & z \leq x \leq 1, \end{cases}$$

and this function can obviously be expressed in terms of (Fourier) sine series as shown in (2.8). A symmetric closed form expression for $K_{2,\varepsilon}$ was found recently by two summer REU students. It is quite a bit more complicated than $\beta = 1$,

$$K_{2,\varepsilon}(x,z) = \frac{\mathrm{e}^{-\varepsilon(x+z)}}{4\varepsilon^3(\mathrm{e}^{2\varepsilon}-1)^2}\Big[\mathrm{e}^{2\varepsilon}(2\varepsilon - \varepsilon(x+z) - 1) + \mathrm{e}^{4\varepsilon}\left(\varepsilon(x+z)+1\right)$$
$$+ \mathrm{e}^{2\varepsilon(1+x+z)}\left(2\varepsilon - \varepsilon(x+z)+1\right) + \mathrm{e}^{2\varepsilon(x+z)}\left(\varepsilon(x+z)-1\right)$$
$$+ \mathrm{e}^{2\varepsilon(2+\min(x,z))}\left(-\varepsilon|x-z|-1\right) + \mathrm{e}^{2\varepsilon\max(x,z)}\left(-\varepsilon|x-z|+1\right)$$
$$+ \mathrm{e}^{2\varepsilon(1+\min(x,z))}\left(1 - 2\varepsilon + \varepsilon|x-z|\right)$$
$$+ \mathrm{e}^{2\varepsilon(1+\max(x,z))}\left(1 + 2\varepsilon - \varepsilon|x-z|\right)\Big], \qquad (4.4)$$

but is found by similar means.

For $\beta > 2$ and $\varepsilon > 0$, closed forms of $K_{\beta,\varepsilon}$ should be derivable, but as of right now they are unknown. If the jump in complexity from $K_{1,\varepsilon}$ to $K_{2,\varepsilon}$ is indicative, writing the closed form for higher β values will quickly become impractical, even if they can be found using the same straightforward techniques. Fortunately, as we discussed in Section 3, it is actually preferable to work with the Hilbert-Schmidt SVD which is based on the series form of $K_{\beta,\varepsilon}$ from (2.8) rather than the closed form.
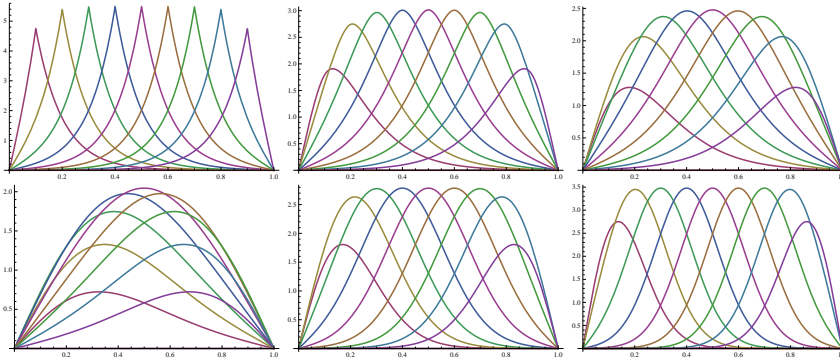
Fig. 2: Copies of the iterated Brownian bridge kernel. Top: $K_{\beta,10}$ for $\beta = 1$ (left), $\beta = 2$ (middle), $\beta = 3$ (right); bottom: $K_{\beta,\varepsilon}$ for $\beta = 7$, $\varepsilon = 10$ (left), $\beta = 13$, $\varepsilon = 30$ (middle), $\beta = 20$, $\varepsilon = 50$ (right).

Fig. 2 shows iterated Brownian bridge kernels using the same values of β as in Fig. 1. The plots in Fig. 2 illustrate how the iterated Brownian bridge kernels using large values of ε become more localized and more and more resemble translated Gaussian kernels as $\beta \to \infty$. However, by construction, these Gaussian-like iterated Brownian bridge kernels obey zero boundary conditions.

To see what the effect of these built-in boundary conditions may be, we plot the cardinal functions for interpolation by iterated Brownian bridge kernels and by Gaussians (which are simply translated across the domain of interest). The top row of Fig. 3 shows some of the cardinal functions for iterated Brownian bridge interpolation with $K_{20,50}$ at 22 equally spaced points in $(0,1)$ on the left, and for 22 Chebyshev points on the right. The bottom row provides the analogous plots of Gaussian cardinal functions, where the shape parameter of the Gaussian was chosen to match the shape of $K_{20,50}(\cdot, 1/2)$, i.e., $\varepsilon_{\text{Gauss}} = 5.75$. Iterated Brownian bridge cardinal functions are evaluated via $\boldsymbol{\psi}(\boldsymbol{x})^T \Psi^{-1}$ as discussed at the end of Section 3.5, while we use $\boldsymbol{k}(\boldsymbol{x})^T \mathsf{K}^{-1}$ for Gaussians. Note that the cardinal functions associated with $K_{20,50}$ have to be evaluated using the Hilbert-Schmidt SVD approach since a closed form is not even known for this kernel. Filling the K matrix via truncated Hilbert-Schmidt expansions would be possible, but not advisable as explained earlier. On the other hand, the direct approach is fine for Gaussians due to the relatively large value of the shape parameter used for this example. If one wants to look at cardinal functions associated with "flat" Gaussian then the Hilbert-Schmidt SVD of [14] can be employed.
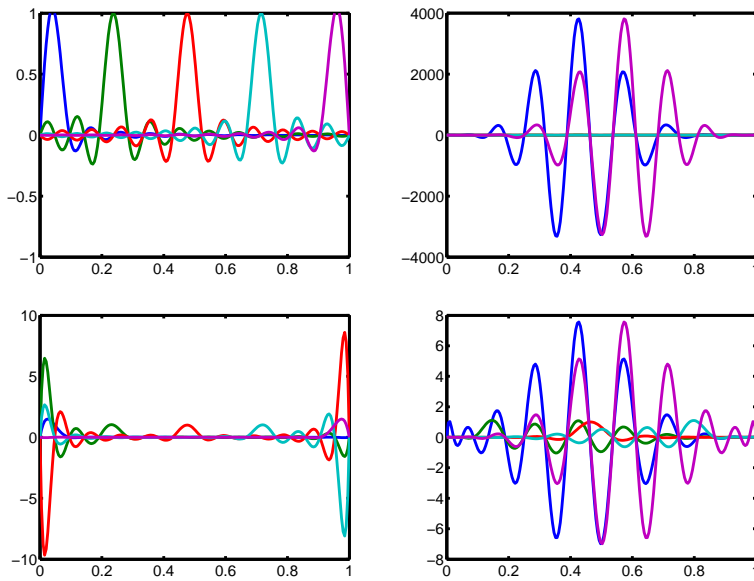
Fig. 3: Some cardinal functions for interpolation at 22 points in $(0, 1)$. Top: iterated Brownian bridge cardinal functions for equally spaced points (left), Chebyshev points (right); bottom: Gaussian cardinal functions for equally spaced points (left), Chebyshev points (right).

It is interesting to observe that, due to the built-in boundary conditions, the iterated Brownian bridge cardinal functions indicate that uniformly distributed data is much preferred over Chebyshev data, even though one might expect something like a Runge phenomenon to occur when interpolating on a compact interval. On the other hand, it is known that interpolation with cubic splines also favors evenly distributed data (see, e.g., [24, 25]). In the Gaussian case, the preference is not as clear, but will favor Chebyshev points as the shape parameter tends to zero, i.e., as one approaches polynomial interpolation. This justifies our use of equally spaced points for the experiments reported in Section 6.

5 Error Bounds via Sampling Inequalities

The sub-family of iterated Brownian bridge kernels with $\varepsilon = 0$ coincides with the special class of interpolatory L-splines with even-order derivative boundary conditions mentioned in [42, Sect. 8], provided those derivatives are zero also for the function f generating the data. We now use the much more recent framework of sampling inequalities in reproducing kernel Hilbert spaces (see, e.g. [31]) to obtain the same error bounds as already reported in [42]. The following discussion closely follows [31, 34] and also draws on some ingredients from [42].

The goal of a typical *sampling inequality* is to bound a weak continuous norm of a function which is either zero or small on a discrete sampling set $\mathcal{X}$ by a sum of two terms: one measuring the sampled data on $\mathcal{X}$, and the other given by

a stronger continuous norm of the function which is usually multiplied by some power of the *fill distance h* of $\mathcal{X}$.

In this paper we are concerned with error bounds for kernel-based interpolation, and the difference between the function f assumed to have generated the data and its kernel interpolant s is an ideal candidate for the application of a sampling inequality since this difference is zero on the set $\mathcal{X}$ of data points.

For the case $\varepsilon = 0$ we can take the (semi-)inner product and (semi-)norm to be

$$\langle f, g \rangle_\beta = \int_0^1 f^{(\beta)}(x) g^{(\beta)}(x) \mathrm{d}x, \quad |f|_\beta^2 = \int_0^1 \left(f^{(\beta)}(x) \right)^2 \mathrm{d}x,$$

the usual semi-norm associated with piecewise polynomial splines of degree $2\beta - 1$. We also introduce the finite set $\mathcal{X} = \{x_1, \ldots, x_N\} \subset [0,1]$ of discrete points with fill distance $h = \sup_{x \in [0,1]} \min_{x_j} |x - x_j|$. Using the L_2-norm (i.e., $\beta = 0$) as the weak norm one can then prove the following lemma using elementary estimates along with the Cauchy-Schwarz inequality.

Lemma 1 *[34, Lemma 8.33] Given $f \in C^1[0,1]$ and the set $\mathcal{X}$ defined as above, we have*

$$\|f\|_{L_2[0,1]} \leq h|f|_1 + \sqrt{2}|f|_{2,\mathcal{X}},$$
$$\|f\|_{L_\infty[0,1]} \leq \sqrt{h}|f|_1 + |f|_{\infty,\mathcal{X}},$$

where we define the discrete *(semi-)norms $|f|_{2,\mathcal{X}}^2 = h \sum_{x \in \mathcal{X}} f^2(x)$ and $|f|_{\infty,\mathcal{X}} = \sup_{x \in \mathcal{X}} |f(x)|$.*

A first—suboptimal—interpolation error bound which does not take into consideration the boundary conditions of the kernel or the data can easily be obtained by using induction based on Rolle's theorem and Lemma 1 along with the minimum norm property of the $K_{\beta,0}$ iterated Brownian bridge interpolant.

Theorem 1 *[34, Theorem 8.35] Let $f \in C^\beta[0,1]$ be interpolated by a $K_{\beta,0}$ iterated Brownian bridge spline s on the set $\mathcal{X}$ defined as above. Then there are constants $C_{\beta,2}$ and $C_{\beta,\infty}$ depending only on β, but not on f or h, such that*

$$\|f - s\|_{L_2[0,1]} \leq C_{\beta,2} h^\beta |f - s|_\beta \leq 2 C_{\beta,2} h^\beta |f|_\beta,$$
$$\|f - s\|_{L_\infty[0,1]} \leq C_{\beta,\infty} h^{\beta-1/2} |f - s|_\beta \leq 2 C_{\beta,\infty} h^{\beta-1/2} |f|_\beta.$$

If we also demand that the even-derivative boundary conditions of our iterated Brownian bridge kernels (in this $\varepsilon = 0$ case) are satisfied, then we can prove

Theorem 2 *Let $f \in C^{2\beta}[0,1]$ be interpolated by a $K_{\beta,0}$ iterated Brownian bridge spline s on the set $\mathcal{X}$ defined as above and also assume that f satisfies the same boundary conditions as the kernel $K_{\beta,0}$, i.e., $f^{(2j)}(0) = f^{(2j)}(1) = 0$, $j = 0, \ldots, \beta - 1$, then*

$$\|f - s\|_{L_2[0,1]} \leq C_{\beta,2}^2 h^{2\beta} |f|_{2\beta}, \tag{5.1}$$

$$\|f - s\|_{L_\infty[0,1]} \leq C_{\beta,\infty}^2 h^{2\beta-1} |f|_{2\beta}. \tag{5.2}$$

Proof The proof follows [34], but since the boundary conditions differ we need a slightly modified argument. Following [36,42], we introduce the *bilinear concomitant* $P_\beta(f, g)$ as

$$P_\beta(f, g) = \sum_{j=0}^{\beta-1} (-1)^j \mathcal{D}^{\beta-1-j} f(x) \, \mathcal{D}^{\beta+j} g(x).$$

It arises as the boundary contribution in repeated integration by parts, i.e.,

$$\int_0^1 \mathcal{D}^\beta f(x) \mathcal{D}^\beta g(x) \mathrm{d}x = P_\beta(f, g)\big|_0^1 + (-1)^\beta \int_0^1 f(x) \mathcal{D}^{2\beta} g(x) \mathrm{d}x.$$

Now, using orthogonality in the reproducing kernel Hilbert space of our kernel $K_{\beta,0}$ we have

$$|f - s|_\beta^2 = \langle f - s, f - s \rangle_\beta = \langle f - s, f \rangle_\beta.$$

Then, however,

$$|f - s|_\beta^2 = \langle f - s, f \rangle_\beta = \int_0^1 \mathcal{D}^\beta (f - s)(x) \mathcal{D}^\beta f(x) \mathrm{d}x$$

$$= P_\beta(f - s, f)\big|_0^1 + (-1)^\beta \int_0^1 (f - s)(x) \mathcal{D}^{2\beta} f(x) \mathrm{d}x.$$

Inspecting the bilinear concomitant we note that

$$P_1(f - s, f) = (f - s)(x) \mathcal{D} f(x)$$
$$P_2(f - s, f) = \mathcal{D}(f - s)(x) \mathcal{D}^2 f(x) - (f - s)(x) \mathcal{D}^3 f(x)$$
$$P_3(f - s, f) = \mathcal{D}^2(f - s)(x) \mathcal{D}^3 f(x) - \mathcal{D}(f - s)(x) \mathcal{D}^4 f(x) + (f - s)(x) \mathcal{D}^5 f(x)$$
$$\vdots$$

so that $P_\beta(f - s, f)$ is zero on the boundary since each product contains an even derivative of order up to $2\beta - 2$ which is zero since the corresponding derivative of f as well as that of s (and therefore their difference) is zero on the boundary.

This leaves us with

$$|f - s|_\beta^2 = (-1)^\beta \int_0^1 (f - s)(x) \mathcal{D}^{2\beta} f(x) \mathrm{d}x$$

$$\leq \|f - s\|_{L_2[0,1]} |f|_{2\beta}.$$

Theorem 1 together with the inequality just derived now gives us

$$\|f - s\|_{L_2[0,1]}^2 \leq C_{\beta,2}^2 h^{2\beta} |f - s|_\beta^2$$

$$\leq C_{\beta,2}^2 h^{2\beta} \|f - s\|_{L_2[0,1]} |f|_{2\beta}$$

or

$$\|f - s\|_{L_2[0,1]} \leq C_{\beta,2}^2 h^{2\beta} |f|_{2\beta}.$$

The bound in the sup-norm can be proved similarly.   □

Unfortunately, the case $\varepsilon > 0$ seems to be considerably more difficult to analyze since the inner product of the reproducing kernel Hilbert space of $K_{\beta,\varepsilon}$ is related to the differential operator $\mathcal{L}_{\beta,\varepsilon}$ as well as our specific boundary conditions. Thus, bounds analogous to (5.1)-(5.2) cannot be established since the corresponding bilinear concomitant no longer seems to vanish on the boundary. Nevertheless, the numerical experiments below will demonstrate that the same rate of convergence as in Theorem 2 can be achieved also in the non-polynomial case when $\varepsilon > 0$.

Remark 4 Our choice of boundary conditions for the iterated Brownian bridge kernels was motivated by the simplicity of the resulting eigenexpansion and therefore a particularly transparent implementation of the Hilbert-Schmidt SVD. One could, of course, work with the same differential operator and impose different boundary conditions, such as periodic boundary conditions, prescribed successive derivatives at the boundary, or setting odd derivatives to zero instead of even ones. All of these choices have been studied in the *L*-spline literature. In the literature on reproducing kernels the case of periodic boundary conditions was discussed in [43, Chapter 2], resulting in a reproducing kernel

$$\widetilde{K}_{\beta,0}(x,z) = \sum_{n=1}^{\infty} \frac{2}{(2n\pi)^{2\beta}} \cos(2n\pi(x-z))$$

that can be seen to be piecewise polynomial in nature by expressing it in terms of Bernoulli polynomials as

$$\widetilde{K}_{\beta,0}(x,z) = \frac{(-1)^{\beta-1}}{(2\beta)!} B_{2\beta}((x-z) \bmod 1).$$

For this specific case, the eigenfunctions are sines and cosines, $\sqrt{2}\sin(2n\pi x)$ and $\sqrt{2}\cos(2n\pi x)$, with associated double eigenvalues $(2n\pi)^{-2\beta}$. For this setup it is also rather straightforward to establish error bounds using either *L*-spline techniques or RKHS techniques combined with sampling inequalities as done above since the associated periodic boundary conditions work out nicely with the bilinear concomitant. In fact, error bounds for the case $\varepsilon > 0$ with periodic boundary conditions can already be found in the *L*-spline literature [36].

6 Numerical Experiments

The smoothness of the iterated Brownian bridge kernels is controlled by the free parameter $\beta \geq 1$ such that a iterated Brownian bridge interpolant has $2\beta - 2$ smooth derivatives. As was discussed in Section 5, we conjecture that the convergence order of interpolants based on these kernels, with respect to the fill distance h introduced in Section 5, should be $O\left(h^{2\beta}\right)$, regardless of the value of ε. Based on our observation about the cardinal functions in Section 4.2, we always use N evenly spaced points in $(0,1)$, meaning that $h \approx N^{-1}$. The experiments in this section demonstrate that iterated Brownian bridge interpolants converge at order $O\left(N^{-2\beta}\right)$ for appropriately smooth and homogeneous input functions f.

In addition to the smoothness parameter β, we have also discussed a shape parameter ε which influences the peakedness of the iterated Brownian bridge kernels.

It is known that interpolation performed with infinitely smooth kernels, equipped with a shape parameter ε, will yield polynomial interpolation as they approach their $\varepsilon \to 0$ "flat" limit (see, e.g., [11, 23, 33]). Because ε can take positive values, these methods have the potential to achieve accuracy superior to polynomials at essentially the same computational cost. A similar situation occurs for the iterated Brownian bridge functions: the presence of the free parameter ε allows iterated Brownian bridge interpolation to surpass piecewise polynomial interpolation for some interpolation problems.

This section studies the significance of this ε parameter, as well as the smoothness parameter β, in the setting of some interpolation problems. We demonstrate when the ε and β flexibility is valuable in improving the solution accuracy. For examples dealing with the order of convergence as a function of β, we compute the error of our interpolant s to the function f using the L_2-norm,

$$\text{error}(s) = \|s - f\|_2,$$

where the norm is approximated on 400 evenly spaced points in $[0, 1]$. The L_∞ norm is used in Section 6.2.2 only, where we are studying the effect of varying ε for a fixed β.

6.1 Stable Interpolation with the Hilbert-Schmidt SVD

Let us begin our numerical experiments by demonstrating how the Hilbert-Schmidt SVD brings numerical stability to kernel computations. Moreover, being able to stably compute the cardinal functions for all values of the kernel parameters as discussed at the end of Section 3.5 (and also Fig. 3) allows us to look at the growth of Lebesgue constant associated with iterated Brownian bridge kernels.

Section 3.4 introduced the idea that the basis collected in $\boldsymbol{\psi}(\cdot)$ is more stable than the standard basis in $\boldsymbol{k}(\cdot)$. Numerical evidence of this for Gaussians was presented in [14]. Here we show in an experiment that for small values of ε, the interpolation matrix K evaluated using the Mercer series representation (3.3) becomes unreasonably ill-conditioned even for small N. The function we choose to interpolate is

$$f(x) = \sin(2\pi x)$$

which satisfies all boundary conditions for any β value. This condition problem would still appear even for functions f which do not satisfy any boundary conditions, but the choice of this function f allows us to eliminate nonhomogeneity as a potential problem here. We perform interpolation with only $N = 10$ points, and compare the use of the $\boldsymbol{\psi}$ and $\boldsymbol{k}$ bases in Fig. 4a.

For suitably large values of ε, the two interpolation methods overlap, but as $\varepsilon \to 0$ the standard basis loses accuracy and stagnates as the condition of the K matrix overwhelms the solution. With the Hilbert-Schmidt SVD we can compute the interpolant through Ψ^{-1} using (3.10) instead, and we are able to safely interpolate for all values of ε. This new basis provides a stable mechanism to evaluate piecewise polynomial splines, of arbitrary degree, for which a Hilbert-Schmidt series expansion is available.

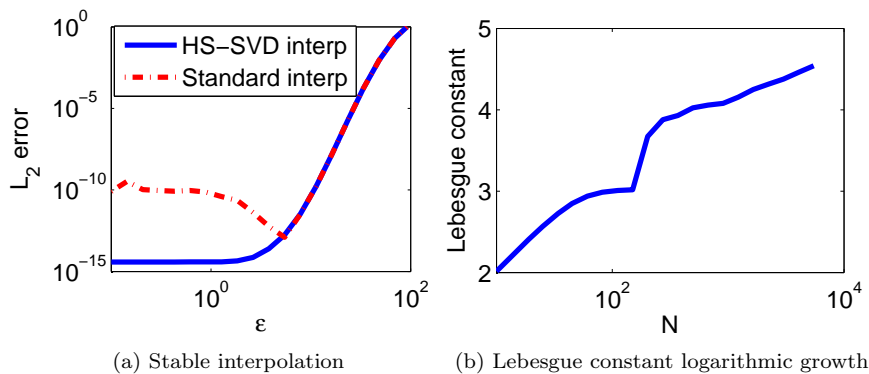(a) Stable interpolation   (b) Lebesgue constant logarithmic growth

Fig. 4: In Fig. 4a, even with only $N = 10$ points, the $\beta = 8$ standard basis is too ill-conditioned to stably interpolate the data as $\varepsilon \to 0$. The Hilbert-Schmidt SVD has no such stability problem for small ε, allowing it to reach the best accuracy possible subject to machine precision. In Fig. 4b, the Lebesgue constant for interpolation with $K_{20,50}$ on N equally spaced points in $(0,1)$ grows logarithmically. The maximum was computed using $10N$ equally spaced points in $[0,1]$.

Although this eigenexpansion is only applicable for the piecewise polynomial splines presented in Section 4.1, this technique could be used to stably evaluate splines generated by other differential operators and boundary conditions. In this context it serves as a useful complement to the theory of B-splines (see, e.g., [37]), which also allow for stable evaluation.

Stability of interpolation methods is frequently also measured by studying the growth of the *Lebesgue constant*

$$\Lambda_{N,\mathcal{X}} = \max_{x \in [0,1]} \sum_{j=1}^{N} |\ell_j(x)|,$$

where ℓ_j is the jth Lagrange or cardinal function used for interpolation on the set $\mathcal{X}$ consisting of N points in $(0,1)$. As pointed out earlier, we can evaluate the entire vector of these cardinal functions for interpolation with iterated Brownian bridge kernels using the Hilbert-Schmidt SVD, i.e.,

$$\boldsymbol{\ell}(x)^T = \boldsymbol{\psi}(x)^T \boldsymbol{\Psi}^{-1},$$

where $\boldsymbol{\ell}(x)^T = (\ell_1(x), \ldots, \ell_N(x))$. In Fig. 4b we demonstrate how the Lebesgue constant for interpolation on N equally spaced points in $(0,1)$ using the kernel $K_{20,50}$ seems to grow logarithmically, i.e., at the optimal rate, for values of N from 10 to 5485 This gives further support to the claim that iterated Brownian bridge kernels, through the Hilbert-Schmidt SVD, provide a stable interpolation method.

6.2 Interpolation for Functions Satisfying Homogeneous Boundary Conditions

In the derivation of the iterated Brownian bridge and piecewise polynomial spline kernels in Section 2.2, we imposed boundary conditions (2.6) to ensure unique

Green's functions. However, in doing so, we have restricted the set of functions which can be recovered with arbitrary precision on $[0, 1]$ to functions which satisfy these boundary conditions. This section studies the interpolation properties of these kernels on suitably homogeneous functions.
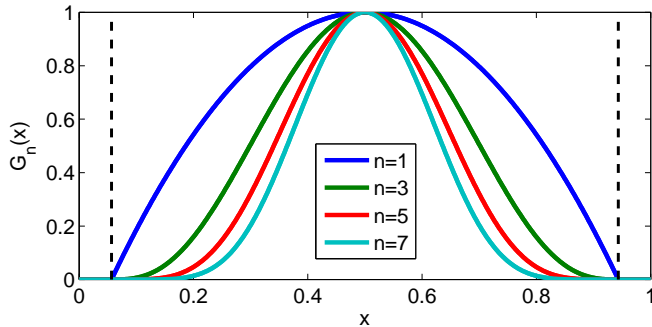

Fig. 5: Sample plots of the G_n function from (6.1) with $n = 1, 3, 5, 7$. As with all G_n in this section, $\gamma = .0567$. The vertical dashed lines are at $x = \gamma$ and $x = 1 - \gamma$, and indicate the point beyond which the function is 0.

The first function that we consider is related to a function from [21] and was specifically designed to study the effect of error near the boundaries. This family of functions,

$$G_n(x) = (1/2 - \gamma)^{-2n} (x - \gamma)_+^n (1 - \gamma - x)_+^n, \qquad x \in [0, 1], \qquad (6.1)$$

has all its derivatives at $x = 0$ and $x = 1$ equal to 0, meaning that it definitely satisfies (2.6). The parameter $\gamma \in (0, 1/2)$ causes the function to be identically zero outside of the interval $(\gamma, 1 - \gamma)$, with discontinuous n^{th} derivatives at $x = \{\gamma, 1 - \gamma\}$. By manipulating n we can consider functions that both satisfy and violate the necessary smoothness conditions of the underlying Green's kernel differential operator (2.5). For these experiments we will fix $\gamma = .0567$ rather arbitrarily but with the intention of keeping the points of discontinuity away from any data point. G_n is plotted in Fig. 5 for some values of n.

6.2.1 Example 1: Convergence Orders

Our first example will study the order of convergence of these kernel methods for a fixed $\varepsilon = 1$ but different β values. We choose the smoothness parameter in our test function G_n to be $n = 6$ and perform iterated Brownian bridge interpolation with $\beta = 1, \ldots, 5$. For $\beta \leq 3$, the necessary smoothness conditions are satisfied, but for $\beta > 3$, the interpolating functions assume a higher level of smoothness than the function G_6 provides. Fig. 6a shows the expected improvement in convergence order until the smoothness assumption is violated (when $\beta = 4$); at which point, subsequent increases in β yield negligible convergence improvements.

These results support our belief, discussed in Section 5, that functions which satisfy the 2β homogeneity conditions (2.6) and have at least 2β smooth derivatives

β	exponent
1	-1.96
2	-4.01
3	-6.35
4	-6.75
5	-6.66

(b) Convergence order table
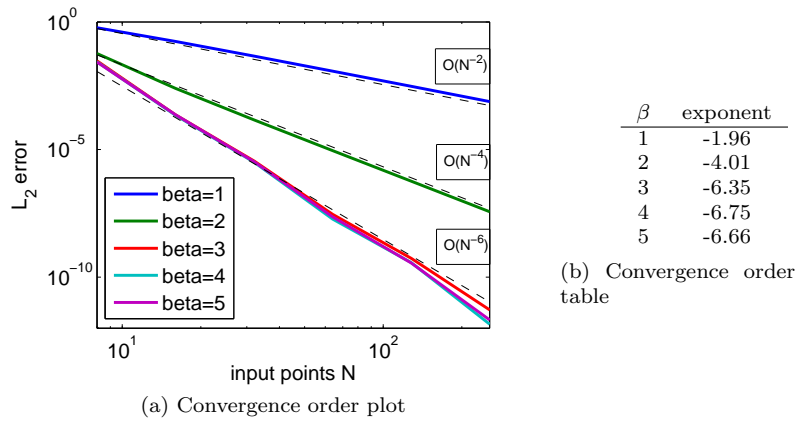
(a) Convergence order plot

Fig. 6: Error results for iterated Brownian bridge interpolation on the function G_6 from (6.1). As β increases, the order of convergence increases, until the interpolating basis functions reach the smoothness of G_6. Beyond that point (which occurs at $\beta = 3$) we encounter saturation, and there is no value in further increasing β. The N input points were evenly spaced in $(0, 1)$ as were the 400 points at which the error was computed.

can be interpolated using order β iterated Brownian bridge basis functions with an $O(N^{-2\beta})$ order of convergence. Recall that we were able only to prove that result in the $\varepsilon = 0$ setting. The table in Fig. 6b shows the results of lines of best fit on the log scale graph from Fig. 6a under the assumption that the error of the interpolant follows

$$\text{error}(s) = CN^{-2\beta}, \quad \text{or} \quad \log(\text{error}(s)) = \log C - 2\beta \log N.$$

The exponent column should contain the values -2β if this error formula is valid. This is the case when $\beta \le 3$, but for larger β the convergence order deteriorates.

6.2.2 Example 2: Piecewise Polynomial Splines as Limits of Iterated Brownian Bridge Interpolants and the Suboptimality of Splines

Earlier, we claimed that the kernels associated with (2.5) for $\varepsilon = 0$ could be used to produce piecewise polynomial splines with appropriate boundary conditions. We now demonstrate that fact experimentally, and also show that optimal accuracy of the iterated Brownian bridge kernels may be achieved for $\varepsilon > 0$. Fig. 7 studies the use of $\beta = 1$ iterated Brownian bridge kernels to approximate G_1 sampled at evenly spaced points. This confirms that the iterated Brownian bridge kernels are converging to piecewise linear splines as $\varepsilon \to 0$, and the table in Fig. 7b displays the ε_{opt} value which produces optimal error.

For this $\beta = 1$ example, the optimal error for iterated Brownian bridge interpolation is only marginally better than the spline result. This is not always the case, as is demonstrated for a $\beta = 2$ interpolant in Fig. 8. We consider a function

$$\hat{G}_2(x) = G_2(x) \exp(-36(x - .4)^2), \tag{6.2}$$
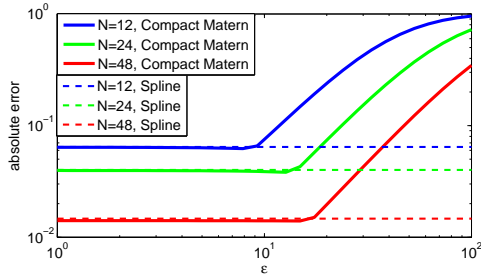
(a) Error as a function of ε

N	$\varepsilon_{\mathrm{opt}}$	error ratio
12	8.9	.96
24	13.9	.95
48	16.8	.95

(b) ε parameterization for optimal error

Fig. 7: Error results for iterated Brownian bridge interpolation on the function G_1 from (6.1). The effect of ε is studied for various N values, and each time an optimal ε, denoted $\varepsilon_{\mathrm{opt}}$, produces a smaller error than the spline which occurs in the $\varepsilon \to 0$ limit. In the table on the right, the $\varepsilon_{\mathrm{opt}}$ value and the ratio of the optimal error to the spline error is displayed. The N input points were evenly spaced in $(0,1)$ as were the 400 points at which the error was computed. The spline results were computed with a piecewise linear spline.

where G_2 was defined in (6.1), using the parameter $\gamma = .0567$. This function satisfies the necessary smoothness and convergence criteria, but is concentrated around $x = .4$ rather than $x = .5$. In Fig. 8a, we see clear benefits to using $\varepsilon > 0$ to perform interpolation, but not for all N values.
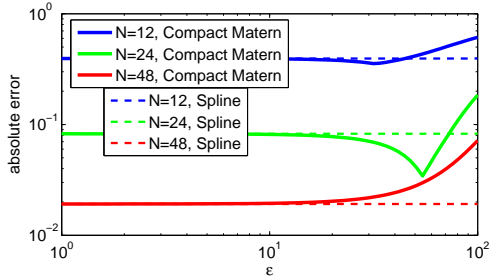
(a) Error as a function of ε

N	$\varepsilon_{\mathrm{opt}}$	error ratio
12	32.2	.90
24	54.4	.42
48	$\to 0$	1

(b) ε parameterization for optimal error

Fig. 8: Error results for iterated Brownian bridge interpolation on the function $\hat{G}_2$ from (6.2). The effect of ε is studied for various N values, and for smaller N an optimal ε, denoted $\varepsilon_{\mathrm{opt}}$, produces a smaller error than the spline which occurs in the $\varepsilon \to 0$ limit. In the table on the right, the $\varepsilon_{\mathrm{opt}}$ value and the ratio of the optimal error to the spline error is displayed. For $N = 48$, the optimal error seems to occur for $\varepsilon = 0$. The N input points were evenly spaced in $(0,1)$ as were the 400 points at which the error was computed. The spline results were computed with a natural cubic spline.

It seems that the potential for improved accuracy by allowing $\varepsilon \neq 0$ is strongly related to the number, and thus location, of the interpolation points. This flex-

ibility would likely provide even greater value should the scattered data not be located at evenly spaced points. We conclude only that the "flat" limit of a piecewise polynomial spline is achieved in all cases.

6.2.3 Summary for Functions Satisfying Homogeneous Boundary Conditions

The results of this section suggest two points:

- The order of the iterated Brownian bridge interpolation scheme is $O(N^{-2\beta})$ for sufficiently smooth and appropriately homogeneous functions.
- The $\varepsilon \to 0$ limit of the iterated Brownian bridge interpolant is the interpolating piecewise polynomial spline of degree $2\beta - 1$ with boundary conditions specified by (2.6). This observation is in agreement with the theoretical results of [39] on interpolation in the "flat" limit for kernels with finite smoothness.

6.3 Interpolating Functions Which Do Not Satisfy the Boundary Conditions

In the previous section, Fig. 6 studied the effect of choosing β greater than the requisite level of smoothness of the underlying function. This is relevant, because we may not know a priori the appropriate smoothness when interpolating functions with limited smoothness. Likewise, many functions of interest do not satisfy the boundary conditions (2.6), and in this section we study the effect of violating those boundary conditions on the resulting interpolant.

One function which satisfies the $\beta = 1$ and $\beta = 2$ conditions is the polynomial

$$p(x) = x - 2x^3 + x^4. \tag{6.3}$$

This function violates the $\beta = 3$ conditions because $p^{(4)}(0) \neq 0$, but all higher derivatives are 0, so only the $\beta = 3$ condition is violated. Fig. 9 studies convergence using different $K_{\beta,1}$ kernels, and shows that there is stagnation beyond $\beta = 2$.

Up until $\beta = 2$, we observe the expected order of convergence, namely $O(N^{-2\beta})$. Beyond the first nonhomogeneous boundary condition at $\beta = 3$, there is no improvement in convergence; this happens despite the fact that all other boundary conditions for $\beta \neq 3$ are satisfied. Higher order kernels make no progress, much in the same way that increasing β beyond the appropriate smoothness was ineffective in Fig. 6.

7 Conclusions and Future Work

In this paper we demonstrated that, despite the infinite length of the Hilbert-Schmidt series, the Hilbert-Schmidt SVD can be used to perform accurate and stable interpolation with positive definite kernels—even in their "flat" $\varepsilon \to 0$ limit. Iterated Brownian bridge kernels, given in terms of the eigenvalues and eigenfunctions of the iterated modified Helmholtz differential operator applied on the interval $[0, 1]$, were developed in order to provide a simple and transparent implementation of the Hilbert-Schmidt SVD. The kernels are defined by using the eigenvalues and eigenfunctions in a Hilbert-Schmidt (or Mercer's) series. Boundary conditions enforcing homogeneity of even derivatives at 0 and 1 differentiate these

(a) Convergence order plot

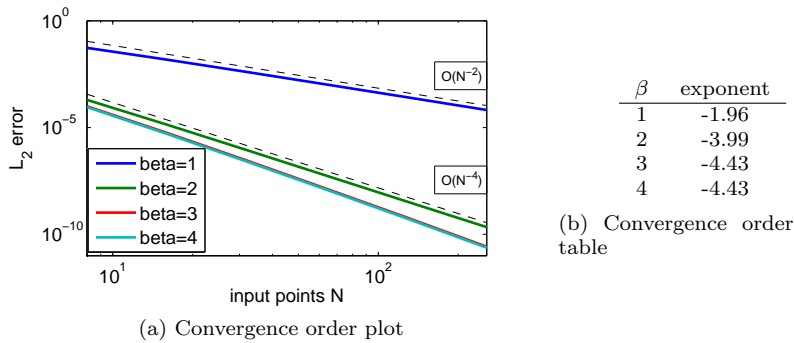| β	exponent
1	-1.96
2	-3.99
3	-4.43
4	-4.43

(b) Convergence order table

Fig. 9: Error results for iterated Brownian bridge interpolation on the function p from (6.3). The expected order of convergence is observed for $\beta = 1$ and $\beta = 2$, until the $\beta = 3$ boundary conditions are violated. For $\beta = 3$, the order is the same as $\beta = 2$, and even though the $\beta = 4$ boundary conditions are satisfied, the $\beta = 4$ interpolant converges no more quickly than $\beta = 2$ because of the $\beta = 3$ barrier. The N input points were evenly spaced in $(0, 1)$ as were the 400 points at which the error was computed. Only $\varepsilon = 1$ was considered here.

kernels from the traditional full-space Matérn kernels. Free parameters ε and β control the locality and smoothness of these kernels, and in the $\varepsilon \to 0$ "flat" limit they can be used to produce piecewise polynomial splines.

Numerical results show convergence of the interpolant at order roughly twice the smoothness of the kernel, which is a result predicted theoretically for polynomial splines when $\varepsilon = 0$ using sampling inequalities. This convergence behavior only applies to functions which satisfy the aforementioned homogeneity conditions at the boundary; failure to comply impedes the improvement otherwise commensurate with a smoother kernel.

Future work must address the transition to higher dimensions, likely through the use of tensor product kernels, and the implication of the boundary conditions in higher dimensions. The presence of both a locality and smoothness parameter adds another layer of complexity to the common problem of choosing the *right* kernel parametrization. Future work should study the statistical or analytic methods for optimally choosing ε and β simultaneously. During that process, one may also consider the interpretation of fractional β—in the operator sense (2.5), a fractional β seems nonsensical, but the final series computations can handle non-integer β without difficulty.

Some work has studied the distribution of error on $[0, 1]$ for iterated Brownian bridge interpolants, but more work will help understand the effect of violating the boundary conditions, and perhaps even provide a mechanism to circumvent them without losing the higher convergence. Furthermore, different boundary conditions may be preferable for certain problems, so the impact of other boundary conditions must be studied. It is, of course, also possible to define alternative families of kernels using the same differential operators we have used, but with different boundary conditions. For example, a family of periodic kernels is discussed in [43, Chapter 2]. Finally, one can also investigate other differential operators leading, for example, to kernels in terms of classical orthogonal polynomials such as Leg-

endre or Chebyshev polynomials. For those cases, our Hilbert-Schmidt SVD can easily be adapted.

In general, applicability of the Hilbert-Schmidt SVD is limited by the availability of the eigenfunctions and eigenvalues of (the Hilbert-Schmidt integral operator associated with) K. Therefore, one might instead consider kernels that are entirely defined in terms of series expansions of orthonormal bases or frames (as, e.g., in [20]) with expansion coefficients that decay sufficiently quickly. Our matrix factorization approach should be applicable to those kernels as well.

Appendix

To illustrate the simplicity of the implementation of the Hilbert-Schmidt SVD for iterated Brownian bridge kernels we include the MATLAB code for a typical interpolation problem.

```
function yy = HSSVDInterp(x,y,ep,beta,xx)
  phifunc = @(n,x) sqrt(2)*sin(pi*x*n);
  lambdafunc = @(n) ((n*pi).^2+ep^2).^(-beta);
  N = length(x);
  M = ceil(1/pi*sqrt(eps^(-1/beta)*(N^2*pi^2+ep^2)-ep^2));
  I_N = eye(N);
  Lambda = diag(lambdafunc(1:M));
  Phi_interp = phifunc(1:M,x);
  Phi_eval = phifunc(1:M,xx);
  Phi_1 = Phi_interp(:,1:N);
  Phi_2 = Phi_interp(:,N+1:end);
  Lambda_1 = Lambda(1:N,1:N);
  Lambda_2 = Lambda(N+1:M,N+1:M);
  Correction = Lambda_2*(Phi_1\Phi_2)'/Lambda_1;
  Psi_interp = Phi_interp*[I_N;Correction];
  Psi_eval = Phi_eval*[I_N;Correction];
  yy = Psi_eval*(Psi_interp\y);
end
```

References

1. Allasia, G., Cavoretto, R., De Rossi, A.: Lobachevsky spline functions and interpolation to scattered data. Comput. Appl. Math. **32**(1), 71–87 (2013)
2. Allasia, G., Cavoretto, R., De Rossi, A.: Numerical integration on multivariate scattered data by Lobachevsky splines. Int. J. Comput. Math. **90**(9), 2003–2018 (2013)
3. Beatson, R.K., Light, W.A., Billings, S.: Fast solution of the radial basis function interpolation equations: Domain decomposition methods. SIAM J. Sci. Comput. **22**, 1717–1740 (2000)
4. Berlinet, A., Thomas-Agnan, C.: Reproducing Kernel Hilbert Spaces in Probability and Statistics. Springer (2004)
5. de Boor, C.: On interpolation by radial polynomials. Adv. Comput. Math. **24**(1–4), 143–153 (2006)
6. Brinks, R.: On the convergence of derivatives of B-splines to derivatives of the Gaussian function. Comput. Appl. Math. **27**(1), 79–92 (2008)

7. Cavoretto, R.: Meshfree approximation methods, algorithms and applications. Ph.D. thesis, University of Turin (2010)
8. Chen, W., Fu, Z.J., Chen, C.S.: Recent Advances on Radial Basis Function Collocation Methods. Springer Briefs in Applied Sciences and Technology. Springer (2014)
9. Courant, R., Hilbert, D.: Methods of Mathematical Physics, Vol. 1. Wiley (1953). Reprinted 1989
10. NIST Digital Library of Mathematical Functions. http://dlmf.nist.gov/, Release 1.0.5 of 2012-10-01 (2012). Online companion to [28]
11. Driscoll, T.A., Fornberg, B.: Interpolation in the limit of increasingly flat radial basis functions. Comput. Math. Appl. **43**(3–5), 413–422 (2002)
12. Fasshauer, G.E.: Meshfree Approximation Methods with MATLAB. World Scientific Publishing Co., Inc., River Edge, NJ, USA (2007)
13. Fasshauer, G.E., Hickernell, F.J., Woźniakowski, H.: On dimension-independent rates of convergence for function approximation with Gaussian kernels. SIAM J. Numer. Anal. **50**(1), 247–271 (2012)
14. Fasshauer, G.E., McCourt, M.: Stable evaluation of Gaussian RBF interpolants. SIAM J. Sci. Comput. **34**(2), A737—A762 (2012)
15. Fornberg, B., Larsson, E., Flyer, N.: Stable computations with Gaussian radial basis functions. SIAM J. Sci. Comput. **33**(2), 869–892 (2011)
16. Fornberg, B., Piret, C.: A stable algorithm for flat radial basis functions on a sphere. SIAM J. Sci. Comput. **30**(1), 60–80 (2008)
17. Fornberg, B., Wright, G.: Stable computation of multiquadric interpolants for all values of the shape parameter. Comput. Math. Appl. **48**(5–6), 853–867 (2004)
18. Glasserman, P.: Monte Carlo Methods in Financial Engineering, *Applications of Mathematics*, vol. 53. Springer-Verlag, New York (2004)
19. Golub, G.H., Van Loan, C.F.: Matrix Computations (3rd ed.). Johns Hopkins University Press, Baltimore, MD, USA (1996)
20. Griebel, M., Rieger, C., Zwicknagl, B.: Multiscale approximation and reproducing kernel Hilbert space methods, INS Preprint No. 1312, University of Bonn (2013)
21. Hubbert, S., Müller, S.: Thin plate spline interpolation on the unit interval. Numer. Algorithms **45**(1–4), 167–177 (2007)
22. Larsson, E., Fornberg, B.: Theoretical and computational aspects of multivariate interpolation with increasingly flat radial basis functions. Comput. Math. Appl. **49**(1), 103–130 (2005)
23. Lee, Y.J., Yoon, G.J., Yoon, J.: Convergence of increasingly flat radial basis interpolants to polynomial interpolants. SIAM J. Numer. Anal. **39**(2), 537–553 (2007)
24. Lyche, T., Schumaker, L.L.: On the convergence of cubic interpolating splines. In: Spline functions and approximation theory (Proc. Sympos., Univ. Alberta, Edmonton, Alta., 1972), *Internat. Ser. Numer. Math.*, vol. 21, pp. 169–189. Birkhäuser, Basel (1973)
25. Marsden, M.: Cubic spline interpolation of continuous functions. J. Approx. Theory **10**(2), 103–111 (1974)
26. Mercer, J.: Functions of positive and negative type, and their connection with the theory of integral equations. Philosophical Transactions of the Royal Society of London. Series A, Containing Papers of a Mathematical or Physical Character **209**(441–458), 415–446 (1909)
27. Novak, E., Woźniakowski, H.: Tractability of Multivariate Problems Volume 1: Linear Information. No. 6 in EMS Tracts in Mathematics. European Mathematical Society (2008)
28. Olver, F.W.J., Lozier, D.W., Boisvert, R.F., Clark, C.W. (eds.): NIST Handbook of Mathematical Functions. Cambridge University Press, New York, NY (2010). Print companion to [10]
29. Opfer, R.: Multiscale kernels. Adv. Comput. Math. **25**(4), 357–380 (2006)
30. Rasmussen, C.E., Williams, C.K.I.: Gaussian Processes for Machine Learning (Adaptive Computation and Machine Learning). The MIT Press (2005)
31. Rieger, C., Schaback, R., Zwicknagl, B.: Sampling and stability. In: M. Dæhlen, M. Floater, T. Lyche, J.L. Merrien, K. Mørken, L.L. Schumaker (eds.) Mathematical Methods for Curves and Surfaces, *Lecture Notes in Computer Science*, vol. 5862, pp. 347–369. Springer, Berlin, Heidelberg (2010)
32. Ritter, K.: Average-Case Analysis of Numerical Problems, *Lecture Notes in Mathematics*, vol. 1733. Springer (2000)
33. Schaback, R.: Limit problems for interpolation by analytic radial basis functions. J. Comput. Appl. Math. **212**(2), 127–149 (2008)

34. Schaback, R.: Kernel-Based Meshless Methods. Manuscript (2011)
35. Schmidt, E.: Über die Auflösung linearer Gleichungen mit unendlich vielen Unbekannten. Rend. Circ. Mat. Palermo **25**(1), 53–77 (1908)
36. Schultz, M.H., Varga, R.S.: *L*-splines. Numer. Math. **10**(4), 345–369 (1967)
37. Schumaker, L.L.: Spline Functions: Basic Theory. John Wiley & Sons (New York) (1981). Reprinted by Krieger Publishing 1993.
38. Shreve, S.E.: Stochastic Calculus for Finance II: Continuous-time Models, vol. 11. Springer (2004)
39. Song, G., Riddle, J., Fasshauer, G.E., Hickernell, F.J.: Multivariate interpolation with increasingly flat radial basis functions of finite smoothness. Adv. Comput. Math. **36**(3), 485–501 (2012)
40. Stein, M.L.: Interpolation of Spatial Data: Some Theory for Kriging. Springer Series in Statistics. Springer (1999)
41. Steinwart, I., Christmann, A.: Support Vector Machines. Information Science and Statistics. Springer (2008)
42. Swartz, B.K., Varga, R.S.: Error bounds for spline and *L*-spline interpolation. J. Approx. Theory **6**(1), 6–49 (1972)
43. Wahba, G.: Spline Models for Observational Data. SIAM, Philadelphia, PA (1990)
44. Zwicknagl, B.: Power series kernels. Constr. Approx. **29**(1), 61–84 (2008)
45. Zwicknagl, B., Schaback, R.: Interpolation and approximation in Taylor spaces. J. Approx. Theory **171**, 65–83 (2013)