Fast Cross-validation in Harmonic Approximation

Felix Bartel[*]   Ralf Hielscher[†]   Daniel Potts[‡]

Finding a good regularization parameter for Tikhonov regularization problems is a though yet often asked question. One approach is to use leave-one-out cross-validation scores to indicate the goodness of fit. This utilizes only the noisy function values but, on the downside, comes with a high computational cost. In this paper we present a general approach to shift the main computations from the function in question to the node distribution and, making use of FFT and FFT-like algorithms, even reduce this cost tremendously to the cost of the Tikhonov regularization problem itself. We apply this technique in different settings on the torus, the unit interval, and the two-dimensional sphere. Given that the sampling points satisfy a quadrature rule our algorithm computes the cross-validations scores in floating-point precision. In the cases of arbitrarily scattered nodes we propose an approximating algorithm with the same complexity. Numerical experiments indicate the applicability of our algorithms.

Keywords and phrases: cross-validation, regularization, fast evaluation of cross-validation score, discrete Fourier transforms, spherical Fourier transform, NFFT

2010 AMS Mathematics Subject Classification: 65Txx, 65F22, 42A10.

1 Introduction

Estimating a good regularization parameter is a frequent problem in approximation, statistics, and inverse problems. In this paper we restrict ourselves to the basic example of approximating a function from discrete function values. To make the setting concrete

---

[1]Chemnitz University of Technology, Faculty of Mathematics, 09107 Chemnitz, Germany, felix.bartel@mathematik.tu-chemnitz.de, Phone:+49-371-531-35409, Fax:+49-371-531-835409

[2]Chemnitz University of Technology, Faculty of Mathematics, 09107 Chemnitz, Germany, hielscher@mathematik.tu-chemnitz.de, Phone:+49-371-531-38556, Fax:+49-371-531-838556

[3]Chemnitz University of Technology, Faculty of Mathematics, 09107 Chemnitz, Germany, potts@mathematik.tu-chemnitz.de, Phone:+49-371-531-32150, Fax:+49-371-531-832150

we fix for a finite index set $\mathcal{I}$ a family of basis functions $\varphi_{\boldsymbol{n}}\colon X \to \mathbb{C}$, $\boldsymbol{n} \in \mathcal{I}$ on a domain $X \subset \mathbb{R}^d$. Given a finite set of nodes $\mathcal{X} \subset X$ and the corresponding Fourier matrix

$$\boldsymbol{F} = \boldsymbol{F}_{\mathcal{X},\mathcal{I}} = (\varphi_{\boldsymbol{n}}(x))_{x\in\mathcal{X},\boldsymbol{n}\in\mathcal{I}},$$

we consider the problem of recovering Fourier coefficients $\hat{\boldsymbol{f}} = (\hat{f}_{\boldsymbol{n}})_{\boldsymbol{n}\in\mathcal{I}} \in \mathbb{C}^{|\mathcal{I}|}$ from noisy data

$$\boldsymbol{f} = (f_x)_{x\in\mathcal{X}} = \boldsymbol{F}\hat{\boldsymbol{f}} + \boldsymbol{\varepsilon} \in \mathbb{C}^{|\mathcal{X}|},$$

where $\boldsymbol{\varepsilon}$ is zero mean Gaussian noise. More specifically, we look for minimizers of the Tikhonov functional

$$J_\lambda\left(\tilde{\boldsymbol{f}}\right) = \left\|\boldsymbol{F}\tilde{\boldsymbol{f}} - \boldsymbol{f}\right\|_{\boldsymbol{W}}^2 + \lambda\left\|\tilde{\boldsymbol{f}}\right\|_{\hat{\boldsymbol{W}}}^2 \tag{1.1}$$

and ask for the optimal regularization parameter $\lambda > 0$, where $\|\boldsymbol{f}\|_{\boldsymbol{W}}^2 = \boldsymbol{f}^{\mathsf{H}}\boldsymbol{W}\boldsymbol{f}$ and $\|\hat{\boldsymbol{f}}\|_{\hat{\boldsymbol{W}}}^2 = \hat{\boldsymbol{f}}^{\mathsf{H}}\hat{\boldsymbol{W}}\hat{\boldsymbol{f}}$ for the strictly positive diagonal weight matrices $\boldsymbol{W} \in \mathbb{R}^{|\mathcal{X}|\times|\mathcal{X}|}$ and $\hat{\boldsymbol{W}} \in \mathbb{R}^{|\mathcal{I}|\times|\mathcal{I}|}$ in space, respectively frequency domain.

Because of its importance to many practical problems there is a vast literature on many different strategies to determine an optimal regularization parameter λ, e.g. [8, 6, 15, 38]. The idea of so called cross-validation methods is to divide the set of nodes into a subset used to compute the approximation and a subset used for validating the goodness of fit. This procedure might be repeated for different splittings and eventually results for a fixed regularization parameter λ in a cross-validation score. By minimizing this score an "optimal" regularization parameter is found. This approach was initially proposed by Golub, Heath and Wahba [15] in the setting of smoothing splines and since then has been applied to a wide range of parameter estimating problems.

In this paper we consider "leave-one-out" cross-validation, i.e., for fixed regularization parameter λ and any node $x \in \mathcal{X}$ we compute the minimizer $\tilde{\boldsymbol{f}}_{\lambda,(x)}$ of the functional (1.1) restricted to the set of nodes $\mathcal{X} \setminus \{x\}$ and use

$$P(\lambda) = \sum_{x\in\mathcal{X}} \left|\left[\boldsymbol{F}\tilde{\boldsymbol{f}}_{\lambda,(x)}\right]_x - f_x\right|^2$$

as cross-validation score. A drawback with a purly data-driven regularization method like this is, that there is no guarantee for a good approximation of the solution of the regularized problem as formulated in the Bakushinskiĭ veto, [3]. Another difficulty of this approach is its numerical complexity. Indeed, computing $P(\lambda)$ for a single value of λ requires solving the minimization problem (1.1) $|\mathcal{X}|$ times, which is too expensive for most applications. For spline interpolation on the interval or in higher dimensional domains different algorithms have been proposed to lower the computational costs. Those include Monte Carlo approximations [9], matrix decomposition methods [50, 43] and Krylow space methods [35].

For the specific setting of Fourier approximation on the torus $\mathbb{T}^d$ at regular lattice points a fast algorithm has been proposed by Tasche and Weyrich [45] which requires to solve the minimization problem only once for each regularization parameter λ. The idea

of this paper is to generalize the approach in [45] to arbitrary sampling nodes and other domains like the unit interval or the two-dimensional sphere.

The paper is organized as follows. In the second chapter we introduce the necessary notations and prove in Theorem 2.5 a representation of the cross-validation score that depends only on one solution of (1.1), but includes the diagonal entries $h_{x,x}$ of the so called hat matrix $\boldsymbol{H} = \boldsymbol{F}(\boldsymbol{F}^{\mathsf{H}}\boldsymbol{W}\boldsymbol{F} + \lambda\hat{\boldsymbol{W}})^{-1}\boldsymbol{F}^{\mathsf{H}}\boldsymbol{W}$. The efficient approximate computation of those diagonal entries for different settings is subject to the remaining chapters.

Our essential requirement for the exact fast computation of the diagonal entries $h_{x,x}$ is that the nodes $\mathcal{X}$ together with the weights $\boldsymbol{W}$ form an exact quadrature rule. This requirement is satisfied for regular tensor product grids and rank-1 lattices on the d-dimensional torus, Chebyshev nodes on the interval $[-1, 1]$, and, e.g., Gauss Legendre nodes on the two-dimensional unit sphere $\mathbb{S}^2$. The corresponding formulae for the diagonal entries $h_{x,x}$ are given in the Theorems 3.2, 4.2, and 5.1. Together with fast Fourier algorithms on the torus [29, 27], for rank-1 lattices [24, 25], on the interval [11, 27], and on the sphere [28, 27] this allows the efficient evaluation of the cross-validation score $P(\lambda)$ with a numerical complexity close to $\mathcal{O}(|\mathcal{I}| + |\mathcal{X}|)$. Numerical examples for all these settings illustrate our findings.

In the case that no exact quadrature rule is known for the given interpolation nodes we suggest approximating them by the volume of the corresponding Voronoi cells. Our numerical tests in Section 3.4, 4.3, and 5.2 indicate a good approximation of the true cross-validation score, which is much more expensive to compute. The MATLAB code of our algorithm as well as for all numerical experiments can be found on the GitHub repository `https://github.com/felixbartel/fcv`.

2 Cross-validation

Lets start this section by reminding that the minimizer of the Tikhonov functional (1.1) can be given explicitly.

Lemma 2.1. *The unique Tikhonov minimizer of* (1.1) *is*

$$\tilde{\boldsymbol{f}}_\lambda = \left(\boldsymbol{F}^{\mathsf{H}}\boldsymbol{W}\boldsymbol{F} + \lambda\hat{\boldsymbol{W}}\right)^{-1}\boldsymbol{F}^{\mathsf{H}}\boldsymbol{W}\boldsymbol{f}. \tag{2.1}$$

Proof. We look for stationary points by calculating the roots of the gradient of J_λ

$$\nabla J_\lambda\left(\tilde{\boldsymbol{f}}_\lambda\right) = 2\boldsymbol{F}^{\mathsf{H}}\boldsymbol{W}\boldsymbol{F}\tilde{\boldsymbol{f}}_\lambda - 2\boldsymbol{F}^{\mathsf{H}}\boldsymbol{W}\boldsymbol{f} + 2\lambda\hat{\boldsymbol{W}}\tilde{\boldsymbol{f}}_\lambda \stackrel{!}{=} \boldsymbol{0}.$$

Because $\boldsymbol{F}^{\mathsf{H}}\boldsymbol{F}$ is positive semidefinite, $\boldsymbol{W}, \hat{\boldsymbol{W}}$, and λ are strictly positive we find that $\boldsymbol{F}^{\mathsf{H}}\boldsymbol{W}\boldsymbol{F} + \lambda\hat{\boldsymbol{W}}$ is positive definite. In particular it is invertible such that the stationary point can be written as $\tilde{\boldsymbol{f}}_\lambda = (\boldsymbol{F}^{\mathsf{H}}\boldsymbol{W}\boldsymbol{F} + \lambda\hat{\boldsymbol{W}})^{-1}\boldsymbol{F}^{\mathsf{H}}\boldsymbol{W}\boldsymbol{f}$. Using the positive definiteness we see that $\tilde{\boldsymbol{f}}_\lambda$ fulfills the required minimizing property. ∎

Remark 2.2. *Since for random nodes $\mathcal{X}$ and many important function systems the matrix $\boldsymbol{F}^{\mathsf{H}}\boldsymbol{W}\boldsymbol{F}$ is invertible with probability one, see e.g. [4], we may relax the assumption*

on the frequency weights $\hat{w}_n$ to be only non negative. Especially, the zero order frequency weight is often set to zero to avoid penalizing constant functions.

As the leave-one-out cross-validation score depends on solving (2.1) for sets of nodes of the form $\mathcal{X} \setminus \{x\}$ we introduce the following notations for omitting a single node $x \in \mathcal{X}$. For $x \in \mathcal{X}$ and $\boldsymbol{f} \in \mathbb{C}^{|\mathcal{X}|}$ we denote by

$$\boldsymbol{f}_{(x)} = (f_y)_{y \in \mathcal{X} \setminus \{x\}} \in \mathbb{C}^{|\mathcal{X}|-1}$$

the vector of function values $\boldsymbol{f}$ with one node $x \in \mathcal{X}$ omitted. Accordingly, we denote by

$$\boldsymbol{F}_{(x)} = (\varphi_n(y))_{y \in \mathcal{X} \setminus \{x\}, n \in \mathcal{I}} \in \mathbb{C}^{(|\mathcal{X}|-1) \times |\mathcal{I}|}$$

the Fourier matrix $\boldsymbol{F}$ with the row corresponding $x \in \mathcal{X}$ omitted and by

$$\boldsymbol{W}_{(x)} = (W_{y,y'})_{y,y' \in \mathcal{X} \setminus \{x\}} \in \mathbb{C}^{(|\mathcal{X}|-1) \times (|\mathcal{X}|-1)}$$

the restriction of the spatial weight matrix $\boldsymbol{W}$ to the set of nodes $\mathcal{X} \setminus \{x\}$. With these notations the minimizer of the Tikhonov functional (1.1) reduced to the nodes $\mathcal{X} \setminus \{x\}$ can be written as

$$\tilde{\boldsymbol{f}}_{(x)} = \left(\boldsymbol{F}_{(x)}^{\mathsf{H}} \boldsymbol{W}_{(x)} \boldsymbol{F}_{(x)} + \lambda \hat{\boldsymbol{W}} \right)^{-1} \boldsymbol{F}_{(x)}^{\mathsf{H}} \boldsymbol{W}_{(x)} \boldsymbol{f}_{(x)} \in \mathbb{C}^{|\mathcal{I}|}. \tag{2.2}$$

Definition 2.3. *The ordinary cross-validation score for the Tikhonov functional (1.1) is defined as*

$$P(\lambda) = \sum_{x \in \mathcal{X}} \left| \left[\boldsymbol{F} \tilde{\boldsymbol{f}}_{(x)} \right]_x - f_x \right|^2 \tag{2.3}$$

where $\tilde{\boldsymbol{f}}_{(x)}$ is defined by (2.2) and $[\boldsymbol{F}\tilde{\boldsymbol{f}}_{(x)}]_x$ denotes the entry of $\boldsymbol{F}\tilde{\boldsymbol{f}}_{(x)}$ corresponding to the node $x \in \mathcal{X}$.

Interpreting (2.3), we are comparing the predicted or smoothened value $[\boldsymbol{F}\tilde{\boldsymbol{f}}_{(x)}]_{\boldsymbol{x}}$ with the noisy data $f_{\boldsymbol{x}}$ for each node. They intuitively differ more in the case of under- or oversmoothing. So its minimum is a candidate for the smoothing parameter λ. Indeed, probabilistic optimality results of the form

$$\frac{\|\boldsymbol{f} - \boldsymbol{F}\tilde{\boldsymbol{f}}_{\lambda^\star}\|_2^2}{\inf_{\lambda \geq 0} \|\boldsymbol{f} - \boldsymbol{F}\tilde{\boldsymbol{f}}_\lambda\|_2^2} \to 1$$

in probability for $\lambda^\star$ being the minimum of the cross-validation score have been shown for $|\mathcal{X}| \to \infty$ in [34] under the assumption of homogenious noise, i.e., ε_x has the same variance for all $x \in \mathcal{X}$. In Theorem 3.5 of [20], weights were incoporated and the condition was generalized to the case where all $\sqrt{w_x \varepsilon_x}$ have a common variance for $x \in \mathcal{X}$. These optimality results are with respect to a LOSS function similar to the data-fitting term in the Tikhonov functional (1.1). However, in our numerical experiments, we will have a look at the unweighted L_2-error as we use the weights w_x for numerical purposes and want to analyze how this disstorts the results. Unflattering is the fact that, for a single

regularization parameter λ, the direct computation of the ordinary cross-validation score requires to solve $|\mathcal{X}|$ times the normal equation (2.1).

Our first goal is to relate the solution of the reduced problem (2.2) to the solution of the full problem (2.1). To this end we define the matrices

$$
\begin{aligned}
\boldsymbol{A} &= \boldsymbol{F}^{\mathsf{H}} \boldsymbol{W} \boldsymbol{F} + \lambda \hat{\boldsymbol{W}}, \\
\boldsymbol{A}_{(x)} &= \boldsymbol{F}_{(x)}^{\mathsf{H}} \boldsymbol{W}_{(x)} \boldsymbol{F}_{(x)} + \lambda \hat{\boldsymbol{W}}
\end{aligned}
\tag{2.4}
$$

which are decisive for the computation of $\tilde{\boldsymbol{f}}$ and $\tilde{\boldsymbol{f}}_{(x)}$, respectively, and show the following relationship between their inverse, cf. [15].

Lemma 2.4. *Let $\boldsymbol{A}$ and $\boldsymbol{A}_{(x)}$ be defined as in (2.4) and*

$$
\boldsymbol{F}_{x,:} = (\varphi_n(x))_{n \in \mathcal{I}} \in \mathbb{C}^{1 \times |\mathcal{I}|}
$$

denote the row of the matrix $\boldsymbol{F}$ which corresponds to the node $x \in \mathcal{X}$. Then we have

$$
\boldsymbol{A}_{(x)}^{-1} = \boldsymbol{A}^{-1} + \frac{\boldsymbol{A}^{-1} w_x \boldsymbol{F}_{x,:}^{\mathsf{H}} \boldsymbol{F}_{x,:} \boldsymbol{A}^{-1}}{1 - w_x \boldsymbol{F}_{x,:} \boldsymbol{A}^{-1} \boldsymbol{F}_{x,:}^{\mathsf{H}}}.
\tag{2.5}
$$

Proof. The assertion of the lemma follows immediately by applying the Sherman-Morrison formula to

$$
\boldsymbol{A}_{(x)} = \boldsymbol{A} - w_x \boldsymbol{F}_{x,:}^{\mathsf{H}} \boldsymbol{F}_{x,:}.
$$

■

Our next goal is to make the repetitive solving of the normal equation (2.2) in (2.3) independent of the right-hand side $\boldsymbol{f}$. To this end we define the so called *hat matrix*

$$
\boldsymbol{H} := \boldsymbol{F} \boldsymbol{A}^{-1} \boldsymbol{F}^{\mathsf{H}} \boldsymbol{W} = \boldsymbol{F} \left(\boldsymbol{F}^{\mathsf{H}} \boldsymbol{W} \boldsymbol{F} + \lambda \hat{\boldsymbol{W}} \right)^{-1} \boldsymbol{F}^{\mathsf{H}} \boldsymbol{W}
\tag{2.6}
$$

which when applied to a data vector $\boldsymbol{f}$ solves the normal equation (2.1) and evaluates the resulting function at the nodes $\mathcal{X}$. The next lemma is a generalization of [15, equation 2.2] and [49, equation 4.2.9], and shows that for the computation of (2.3), with given diagonal entries $h_{x,x}$ of the hat matrix $\boldsymbol{H}$, it is sufficient to solve the normal equation (2.1) with respect to the data vector $\boldsymbol{f}$ only once.

Theorem 2.5. *The ordinary cross-validation score (2.3) can be written as*

$$
P(\lambda) = \sum_{x \in \mathcal{X}} \frac{[\boldsymbol{H} \boldsymbol{f} - \boldsymbol{f}]_x^2}{(1 - h_{x,x})^2}
\tag{2.7}
$$

with $h_{x,x}$, $x \in \mathcal{X}$ being the diagonal entries of the hat matrix $\boldsymbol{H}$ defined in (2.6).

5

Proof. Let $\boldsymbol{b} = \boldsymbol{F}^\mathsf{H} \boldsymbol{W} \boldsymbol{f}$. Then

$$\tilde{\boldsymbol{f}}_{(x)} = \boldsymbol{A}_{(x)}^{-1} \boldsymbol{F}_{(x)}^\mathsf{H} \boldsymbol{W}_{(x)} \boldsymbol{f}_{(x)} = \boldsymbol{A}_{(x)}^{-1} \left(\boldsymbol{b} - \boldsymbol{F}_{x,:}^\mathsf{H} w_x f_x \right).$$

Next we apply Lemma 2.4 and observe that the denominator in (2.5) can be expressed in terms of the diagonal entries $h_{x,x}$ of the hat matrix $\boldsymbol{H}$:

$$\begin{aligned}
\tilde{\boldsymbol{f}}_{(x)} &= \boldsymbol{A}_{(x)}^{-1} \left(\boldsymbol{b} - \boldsymbol{F}_{x,:}^\mathsf{H} w_x f_x \right) \\
&= \left(\boldsymbol{A}^{-1} + \frac{w_x \boldsymbol{A}^{-1} \boldsymbol{F}_{x,:}^\mathsf{H} \boldsymbol{F}_{x,:} \boldsymbol{A}^{-1}}{1 - h_{x,x}} \right) \left(\boldsymbol{b} - \boldsymbol{F}_{x,:}^\mathsf{H} w_x f_x \right) \\
&= \tilde{\boldsymbol{f}} + \frac{w_x \boldsymbol{A}^{-1} \boldsymbol{F}_{x,:}^\mathsf{H} f_x (h_{x,x} - 1) + w_x \boldsymbol{A}^{-1} \boldsymbol{F}_{x,:}^\mathsf{H} \boldsymbol{F}_{x,:} \tilde{\boldsymbol{f}} - w_x \boldsymbol{A}^{-1} \boldsymbol{F}_{x,:}^\mathsf{H} h_{x,x} f_x}{1 - h_{x,x}} \\
&= \tilde{\boldsymbol{f}} + w_x \boldsymbol{A}^{-1} \boldsymbol{F}_{x,:}^\mathsf{H} \frac{\boldsymbol{F}_{x,:} \tilde{\boldsymbol{f}} - f_x}{1 - h_{x,x}} \\
&= \tilde{\boldsymbol{f}} + w_x \boldsymbol{A}^{-1} \boldsymbol{F}_{x,:}^\mathsf{H} \frac{[\boldsymbol{H}\boldsymbol{f} - \boldsymbol{f}]_x}{1 - h_{x,x}}.
\end{aligned}$$

Multiplying with $\boldsymbol{F}$ from the left-hand side and subtracting f_x results in

$$\begin{aligned}
\left[\boldsymbol{F} \tilde{\boldsymbol{f}}_{(x)} \right]_x - f_x &= \boldsymbol{F}_{x,:} \tilde{\boldsymbol{f}}_{(x)} - f_x \\
&= \boldsymbol{F}_{x,:} \tilde{\boldsymbol{f}} + w_x \boldsymbol{F}_{x,:} \boldsymbol{A}^{-1} \boldsymbol{F}_{x,:}^\mathsf{H} \frac{[\boldsymbol{H}\boldsymbol{f} - \boldsymbol{f}]_x}{1 - h_{x,x}} - f_x \\
&= \left[\boldsymbol{F} \tilde{\boldsymbol{f}} \right]_x + h_{x,x} \frac{[\boldsymbol{H}\boldsymbol{f} - \boldsymbol{f}]_x}{1 - h_{x,x}} - [\boldsymbol{f}]_x \\
&= [\boldsymbol{H}\boldsymbol{f}]_x + \frac{[\boldsymbol{H}\boldsymbol{f} - \boldsymbol{f}]_x}{1 - h_{x,x}} + [\boldsymbol{f} - \boldsymbol{H}\boldsymbol{f}]_x - [\boldsymbol{f}]_x \\
&= \frac{[\boldsymbol{H}\boldsymbol{f} - \boldsymbol{f}]_x}{1 - h_{x,x}}
\end{aligned}$$

and hence each summand in (2.3) is equal to the corresponding summand in (2.7). ∎

Remark 2.6. *According to Theorem 2.5 the ordinary cross-validation score is nothing more than the weighted norm of the residue*

$$\boldsymbol{r} = \boldsymbol{F}\tilde{\boldsymbol{f}} - \boldsymbol{f} = \boldsymbol{H}\boldsymbol{f} - \boldsymbol{f}.$$

Although this means that the normal equation (2.1) has to be solved only once with respect to the data vector $\boldsymbol{f}$ the most expensive part remains, namely the computation of the diagonal entries

$$h_{x,x} = w_x \boldsymbol{F}_{x,:} \boldsymbol{A}^{-1} \boldsymbol{F}_{x,:}^\mathsf{H}$$

for $x \in \mathcal{X}$, which again requires repetitive solving of the normal equation.

Replacing the diagonal entries $h_{x,x}$ with their mean value

$$h = \frac{1}{|\mathcal{X}|} \sum_{x \in \mathcal{X}} h_{x,x} = \frac{1}{|\mathcal{X}|} \operatorname{trace} \boldsymbol{H}$$

we obtain the so called generalized cross-validation score, cf. [49, section 4.3].

Definition 2.7. *The* generalized cross-validation score *is defined as*

$$V(\lambda) = \sum_{x \in \mathcal{X}} \frac{[\boldsymbol{H}\boldsymbol{f} - \boldsymbol{f}]_x^2}{(1-h)^2} = \left(\frac{|\mathcal{X}| \, \|\boldsymbol{H}\boldsymbol{f} - \boldsymbol{f}\|_2}{\operatorname{trace}(\boldsymbol{I} - \boldsymbol{H})} \right)^2.$$

Obviously, if all diagonal entries $h_{x,x}$ of $\boldsymbol{H}$ coincide we have $P(\lambda) = V(\lambda)$.

Lemma 2.8. *The diagonal elements $h_{x,x}$ of the hat matrix $\boldsymbol{H}$ satisfy*

$$h_{x,x} < 1$$

for all $\lambda > 0$ and $x \in \mathcal{X}$.

Proof. Since $\boldsymbol{F}_{(x)}^{\mathsf{H}} \boldsymbol{W}_{(x)} \boldsymbol{F}_{(x)}$ is positive semidefinite and $\lambda \hat{\boldsymbol{W}}$ is strictly positive definite we see that $\boldsymbol{A}_{(x)} = \boldsymbol{F}_{(x)}^{\mathsf{H}} \boldsymbol{W}_{(x)} \boldsymbol{F}_{(x)} + \lambda \hat{\boldsymbol{W}}$ is invertible. On the other hand we know by the Sherman-Morrison formula that $\boldsymbol{A}_{(x)} = \boldsymbol{A} - w_x \boldsymbol{F}_{x,:}^{\mathsf{H}} \boldsymbol{F}_{x,:}$ is invertible if and only if $w_x \boldsymbol{F}_{x,:} A^{-1} \boldsymbol{F}_{x,:}^{\mathsf{H}} \neq 1$. Therefore

$$h_{x,x} = w_x \boldsymbol{F}_{x,:} \boldsymbol{A}^{-1} \boldsymbol{F}_{x,:}^{\mathsf{H}} \neq 1.$$

Since the minimizer $\tilde{\boldsymbol{f}}_\lambda$ of (1.1) converges to the zero vector as $\lambda \to \infty$, we obtain for $\boldsymbol{f} = \boldsymbol{e}_x$ and $\lambda \to \infty$

$$h_{x,x} = \left[\boldsymbol{F} \tilde{\boldsymbol{f}}_\lambda \right]_x \to 0.$$

Together with the fact that the diagonal entries $h_{x,x}$ depend continuously on λ this proves the assertion. ∎

2.1 Algorithm to compute the ordinary or generalized cross-validation score

Concluding the previous statements we end up with a sheme to compute the cross-validation scores.

Algorithm 1: generic computation of the cross-validation scores
---

Input:

    nodes $\mathcal{X}$

    spatial weights $\boldsymbol{W} = \mathrm{diag}(w_x)_{x \in \mathcal{X}} \in \mathbb{R}^{|\mathcal{X}| \times |\mathcal{X}|}$

    Fourier weights $\hat{\boldsymbol{W}} \in \mathbb{R}^{|\mathcal{I}| \times |\mathcal{I}|}$

    function values $\boldsymbol{f} = (f_x)_{x \in \mathcal{X}}$

    regularization parameter λ

Output:

    ordinary cross-validation score $P(\lambda)$

    generalized cross-validation score $V(\lambda)$

1. Compute $\tilde{\boldsymbol{f}} := \boldsymbol{H}\boldsymbol{f} = \boldsymbol{F}\boldsymbol{A}^{-1}\boldsymbol{F}^{\mathsf{H}}\boldsymbol{W}\boldsymbol{f}$, where $\boldsymbol{A}$ is given in (2.4).

2. Compute $h_{x,x} := w_x \boldsymbol{F}_{x,:} \boldsymbol{A}^{-1} \boldsymbol{F}_{x,:}^{\mathsf{H}}$ for $x \in \mathcal{X}$ and $h := \dfrac{1}{|\mathcal{X}|} \displaystyle\sum_{x \in \mathcal{X}} h_{x,x}$.

3. Evaluate $P(\lambda) := \displaystyle\sum_{x \in \mathcal{X}} \frac{|\tilde{f}_x - f_x|^2}{1 - h_{x,x}}$ and $V(\lambda) := \displaystyle\sum_{x \in \mathcal{X}} \frac{|\tilde{f}_x - f_x|^2}{1 - h}$.

Remark 2.9. *For computing the Tikhonov-minimizer of* (1.1) *one can use the LSQR method for numerical stability. This can be accomplished with the coefficient matrix*

$$\boldsymbol{M} = \begin{pmatrix} \boldsymbol{W}^{1/2}\boldsymbol{F} \\ \sqrt{\lambda}\hat{\boldsymbol{W}}^{1/2} \end{pmatrix}$$

and the right-hand side

$$\boldsymbol{b} = \begin{pmatrix} \boldsymbol{W}^{1/2}\boldsymbol{f} \\ \boldsymbol{0} \end{pmatrix},$$

where $\boldsymbol{0}$ *is a column vector containing* $|\mathcal{I}|$ *zeros. The resulting system of equations* $(\boldsymbol{M}^{\mathsf{H}}\boldsymbol{M})^{-1}\boldsymbol{M}^{\mathsf{H}}\boldsymbol{b}$*, which the LSQR method solves, is equivalent to* (2.1).

The computationally most expensive part of Algorithm 1 is the computation of the values $\tilde{\boldsymbol{f}}$ and $h_{x,x}$ for all $x \in \mathcal{X}$. In the subsequent sections we discuss some specific settings to speed up the process and propose an approximation of the ordinary and the generalized cross-validation score in more general cases.

3 Cross-validation on the torus

In this section, we seek for fast algorithms to compute the cross-validation score on the d-dimensional torus $\mathbb{T}^d$ with respect to the Fourier basis $\{\mathrm{e}^{2\pi\mathrm{i}\boldsymbol{n}\cdot\boldsymbol{x}}\}_{\boldsymbol{n}\in\mathbb{Z}^d}$ in $L_2(\mathbb{T}^d)$. With this setting the Fourier matrix $\boldsymbol{F}$ becomes

$$\boldsymbol{F} = \boldsymbol{F}_{\mathcal{X},\mathcal{I}} = \left(\mathrm{e}^{2\pi\mathrm{i}\boldsymbol{n}\cdot\boldsymbol{x}}\right)_{\boldsymbol{x}\in\mathcal{X}, \boldsymbol{n}\in\mathcal{I}} \tag{3.1}$$

for a finite node set $\mathcal{X} \subset \mathbb{T}^d$, a finite multi-index set $\mathcal{I} \subset \mathbb{Z}^d$ and $\boldsymbol{n} \cdot \boldsymbol{x}$ the Euclidean inner product. So $\mathcal{I}$ determines all possible frequencies and $\mathcal{X}$ the nodes of the transform. For

the specific case of equispaced nodes $\mathcal{X}$ fast algorithms have been reported in [45]. In fact, our approach in this section can be seen as a generalization of [45] to more general sampling sets and leads to the same algorithm for equispaced data.

Our central goal is to find a simpler expression for the diagonal entries of the hat matrix $\boldsymbol{H} = \boldsymbol{F}(\boldsymbol{F}^{\mathsf{H}}\boldsymbol{W}\boldsymbol{F} + \lambda\hat{\boldsymbol{W}})^{-1}\boldsymbol{F}^{\mathsf{H}}\boldsymbol{W}$ that allows us to apply Theorem 2.5 efficiently. The idea is to choose $\boldsymbol{W}$ such that $\boldsymbol{F}^{\mathsf{H}}\boldsymbol{W}\boldsymbol{F}$ has diagonal form because the inverse of $\boldsymbol{A} = \boldsymbol{F}^{\mathsf{H}}\boldsymbol{W}\boldsymbol{F} + \lambda\hat{\boldsymbol{W}}$ could then be calculated entry-wise.

3.1 Exact Quadrature

The first approach is to use quadrature rules. Because they are not limited to the torus we define them for general measure spaces so we can make use of them in subsequent sections.

Definition 3.1. *Let* $(\mathcal{M}, \mu)$ *be a measure space and* $\mathcal{P} \subset L_1(\mathcal{M})$ *a set of integrable functions. We call a set of nodes* $\mathcal{X} \subset \mathcal{M}$ *and a list of weights* $\boldsymbol{W} = \operatorname{diag}(w_{\boldsymbol{x}})_{\boldsymbol{x}\in\mathcal{X}}$ *an exact quadrature rule for* $\mathcal{P}$, *if for all* $f \in \mathcal{P}$ *we have*

$$Q_{\mathcal{X},\boldsymbol{W}}f := \sum_{\boldsymbol{x}\in\mathcal{X}} w_{\boldsymbol{x}}f(\boldsymbol{x}) = \int_{\mathcal{M}} f(\boldsymbol{x})\,\mathrm{d}\mu(\boldsymbol{x}).$$

For the tours we obtain the following

Theorem 3.2. *Let* $\mathcal{I} \subset \mathbb{Z}^d$ *be a finite multi-index set with Fourier weights* $\hat{\boldsymbol{W}} = \operatorname{diag}(\hat{w}_{\boldsymbol{n}})_{\boldsymbol{n}\in\mathcal{I}}$, $\mathcal{X} \subset \mathbb{T}^d$ *a set of nodes with* $\boldsymbol{W}$ *their corresponding quadrature weights such that* $(\mathcal{X}, \boldsymbol{W})$ *forms a quadrature rule which is exact for all trigonometric polynomials* $\mathrm{e}^{2\pi\mathrm{i}\boldsymbol{n}\cdot}$ *with frequencies* $\boldsymbol{n}$ *in the difference set* $\mathcal{D}(\mathcal{I}) := \{\boldsymbol{n}_1 - \boldsymbol{n}_2 : \boldsymbol{n}_1, \boldsymbol{n}_2 \in \mathcal{I}\}$. *Then*

(i) *the inverse of* $\boldsymbol{A}$, *given in* (2.4), *where* $\boldsymbol{F}$ *is the Fourier matrix* (3.1) *on* $\mathbb{T}^d$ *is*

$$\boldsymbol{A}^{-1} = \left(\boldsymbol{F}^{\mathsf{H}}\boldsymbol{W}\boldsymbol{F} + \lambda\hat{\boldsymbol{W}}\right)^{-1} = \operatorname{diag}\left(\frac{1}{1 + \lambda\hat{w}_{\boldsymbol{n}}}\right)_{\boldsymbol{n}\in\mathcal{I}},$$

(ii) *the diagonal entry corresponding to the node* $\boldsymbol{x} \in \mathcal{X}$ *of the hat matrix* $\boldsymbol{H} = \boldsymbol{F}\boldsymbol{A}^{-1}\boldsymbol{F}^{\mathsf{H}}\boldsymbol{W}$ *becomes*

$$h_{\boldsymbol{x},\boldsymbol{x}} = w_{\boldsymbol{x}} \sum_{\boldsymbol{n}\in\mathcal{I}} \frac{1}{1 + \lambda\hat{w}_{\boldsymbol{n}}}. \tag{3.2}$$

Proof. Since the product of two exponential functions supported on the frequency set $\mathcal{I}$ has only frequencies in $\mathcal{D}(\mathcal{I})$, where the quadrature nodes and weights are exact, we have

$$\left[\boldsymbol{F}^{\mathsf{H}}\boldsymbol{W}\boldsymbol{F}\right]_{\boldsymbol{n}_1,\boldsymbol{n}_2} = \sum_{\boldsymbol{x}\in\mathcal{X}} w_{\boldsymbol{x}}\overline{\mathrm{e}^{2\pi\mathrm{i}\boldsymbol{n}_1\boldsymbol{x}}}\mathrm{e}^{2\pi\mathrm{i}\boldsymbol{n}_2\boldsymbol{x}} = \int_{\mathbb{T}^d} \overline{\mathrm{e}^{2\pi\mathrm{i}\boldsymbol{n}_1\boldsymbol{x}}}\mathrm{e}^{2\pi\mathrm{i}\boldsymbol{n}_2\boldsymbol{x}}\,\mathrm{d}\boldsymbol{x}.$$

and, hence,

$$\boldsymbol{F}^{\mathsf{H}}\boldsymbol{W}\boldsymbol{F} + \lambda\hat{\boldsymbol{W}} = \operatorname{diag}\left(1 + \lambda\hat{w}_{\boldsymbol{n}}\right)_{\boldsymbol{n}\in\mathcal{I}}.$$

This implies (i). For (ii) we compute the diagonal entries of $\boldsymbol{H}$ as

$$h_{\boldsymbol{x},\boldsymbol{x}} = w_{\boldsymbol{x}} \sum_{\boldsymbol{n}\in\mathcal{I}} \frac{1}{1+\lambda\hat{w}_{\boldsymbol{n}}} \mathrm{e}^{2\pi\mathrm{i}\boldsymbol{n}\boldsymbol{x}} \mathrm{e}^{-2\pi\mathrm{i}\boldsymbol{n}\boldsymbol{x}} = w_{\boldsymbol{x}} \sum_{\boldsymbol{n}\in\mathcal{I}} \frac{1}{1+\lambda\hat{w}_{\boldsymbol{n}}}.$$

$\blacksquare$

Corollary 3.3. *With the prerequisites of Theorem 3.2 we can compute $P(\lambda)$ and $V(\lambda)$ by Algorithm 1 in the same complexity as multiplying a vector with $\boldsymbol{F}$ or $\boldsymbol{F}^{\mathsf{H}}$ for a fixed regularization parameter λ.*

Remark 3.4. *Theorem 3.2 requires $(\mathcal{X},\boldsymbol{W})$ to form an exact quadrature rule which directly alters the weighting of the data-fitting term in the underlying Tikhonov functional (1.1). Preferebly, one wants to have the weight w_x proportional to the level of noise ε_x for every node $x \in \mathcal{X}$, cf. [40]. In most of the following examples the weights as well as the Gaussian noise are uniform, thus fulfilling this property.*

Case studies for specific exact quadrature rules on the torus are discussed in the following two subsections.

3.2 Equispaced Nodes

The simplest example of quadrature on the torus $\mathbb{T}^d$ is Gauss quadrature which consists of N^d equispaced nodes

$$\mathcal{X} = \left\{ \frac{\boldsymbol{m}}{N} \in \mathbb{T}^d : \boldsymbol{m} \in \mathbb{Z}^d \cap \prod_{t=1}^{d}[0,N) \right\}$$

with uniform weights $w_{\boldsymbol{x}} = N^{-d}$. The resulting quadrature formula is exact for all trigonometric polynomials supported on $\mathcal{I}_{2N} := \mathbb{Z}^d \cap \prod_{t=1}^{d}[-N,N) = \mathcal{D}(\mathcal{I}_N)$. Thus we can apply Theorem 3.2 for $\mathcal{X}$ and $\mathcal{I} = \mathcal{I}_N$. The corresponding Fourier matrix $\boldsymbol{F} = \boldsymbol{F}_{\mathcal{X},\mathcal{I}_N}$ describes the ordinary discrete Fourier transform for which the matrix-vector product can be computed in $\mathcal{O}(N^d \log N)$ with the fast Fourier transform.

For $d = 1, 2$ our algorithm coincides with the algorithm proposed in [45] with the only difference that the authors evaluated the data fitting term in the frequency domain and used specific Fourier weights $w_n = n^4$ as regularization term.

In order to illustrate our approach we chose as the test function f the peaks function from MATLAB, which is a sum of translated and scaled Gaussian bells. We evaluated this function on a grid of 1024×1024 equispaced nodes $\mathcal{X}$ and corrupted the data by 10% Gaussian noise $\varepsilon_{\boldsymbol{x}}$, i.e., we set

$$f_{\boldsymbol{x}} = f(\boldsymbol{x}) + \varepsilon_{\boldsymbol{x}}$$

for all $\boldsymbol{x} \in \mathcal{X}$ as depicted in Figure 3.1, (a).

As regularization term we fixed isotropic Sobolev weights $\hat{w}_{\boldsymbol{n}} = 1 + \|\boldsymbol{n}\|_2^s$ for $\boldsymbol{n} \in \mathcal{I}_N$ and $s = 3$ in Fourier space, which correspond to a function with 3 derivatives in

(a) noisy input data and reconstruction

(b) approximation error $\|\tilde{\boldsymbol{f}}_\lambda - \hat{\boldsymbol{f}}\|_2$ (black) and cross-validation scores (orange)
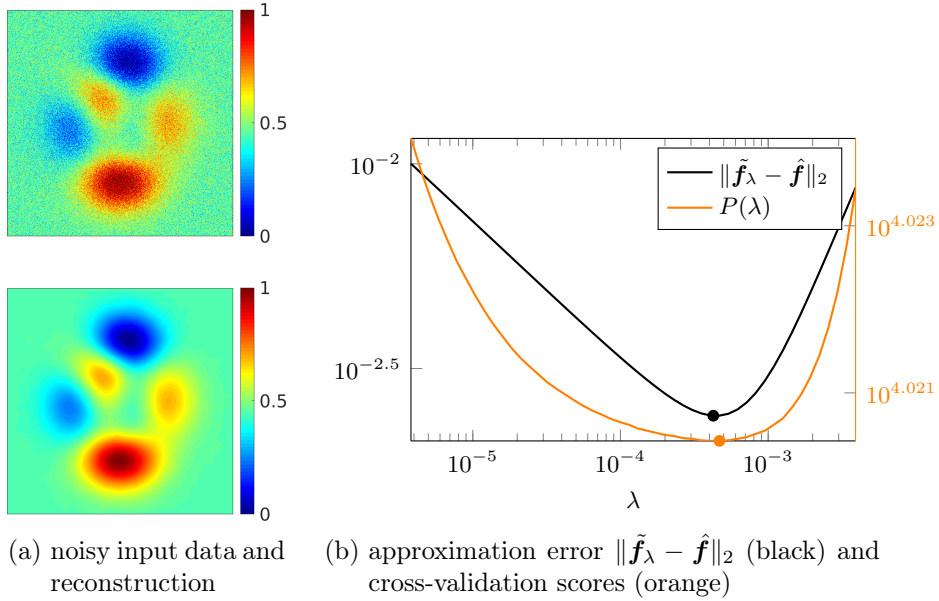
Figure 3.1: Approximation from two-dimensional equispaced data: Comparison of the ordinary cross-validation score $P(\lambda)$ and the approximation error.

$L_2(\mathbb{T}^d)$. Varying the regularization parameter $\lambda \in [2^{-18}, 2^{-8}]$ we computed the Tikhonov minimizers $\tilde{\boldsymbol{f}}_\lambda$ according to (2.1). We then applied Parseval to the original $\hat{\boldsymbol{f}}$ and $\tilde{\boldsymbol{f}}_\lambda$ which is a byproduct from Algorithm 1 to compute the $L_2(\mathbb{T}^d)$-approximation errors as a function of the regularization parameter λ. The resulting curve is depicted in Figure 3.1, (b). Note that according to (3.2) all diagonal entries of the hat matrix are equal and, hence, the ordinary cross-validation score coincides with the generalized cross-validation score. The reconstruction $\tilde{\boldsymbol{f}}_\lambda$ with respect to the minimizer of the cross-validation score $P(\lambda)$ is depicted in Figure 3.1, (a).

In Figure 3.1, (b) the actual $L_2(\mathbb{T}^d)$-approximation error is compared to the cross-validation score $P(\lambda)$ computed according to Algorithm 1 with use of the fast Fourier transform. We observe that the minimizers of both functionals coincide surprisingly well. For this numerical experiment the average running time for the evaluation of $P(\lambda)$ for a single value of λ was 0.06 seconds with the fast algorithm and more than 14 hours for a direct implementation of (2.3).

3.3 Rank-1 Lattices

The approximation of high-dimensional multivariate periodic functions by trigonometric polynomials using particular finite index sets $\mathcal{I}$ in frequency domain is possible using special index sets [46, 10] on the domain $\mathcal{X}$. The most efficient method uses samples along rank-1 lattices and is based on a simple univariate FFT [26]. Rank-1 lattices are

defined by

$$\mathcal{X} = \Lambda(\boldsymbol{z}, M) := \left\{ \boldsymbol{x} = \frac{1}{M}(m\boldsymbol{z} \bmod M\boldsymbol{1}) \in \mathbb{T}^d : m = 0, \ldots, M - 1 \right\}$$

where $M\boldsymbol{1} = (M, \ldots, M)^{\mathsf{T}} \in \mathbb{Z}^d$. They are fully characterized by the *generating vector* $\boldsymbol{z} \in \mathbb{Z}^d$ and the *lattice size* M. There exist algorithms which, given a frequency index set $\mathcal{I}$ and M, compute a generating vector $\boldsymbol{z}$ such that $\boldsymbol{F}^{\mathsf{H}}\boldsymbol{W}\boldsymbol{F}$ equals the identity matrix for $\boldsymbol{W} = \mathrm{diag}(1/M)_{\boldsymbol{x} \in \mathcal{X}}$, cf. [26, 24, 39]. The advantage of rank-1 lattices is the variable index set $\mathcal{I}$ instead of the tensor-product approach like in Section 3.2. So depending on the function we can adapt to different decay properties of the Fourier coefficients. Furthermore there exist fast algorithms which evaluate the matrix-vector product with $\boldsymbol{F}$ or $\boldsymbol{F}^{\mathsf{H}}$ in $\mathcal{O}(M \log M + d|\mathcal{I}|)$ using only one one-dimensional fast Fourier transform.

To exemplify these ideas we looked at a sample function consisting of a tensor product of $L_2(\mathbb{T}^d)$-normed B-splines of order two in seven dimensions, i.e., $d = 7$,

$$f(\boldsymbol{x}) = \prod_{j=1}^{d} B_2(x_j), \quad B_2(x) = 2\sqrt{3}\left(\mathcal{X}_{[0,0.5)}x + \mathcal{X}_{[0.5,1)}(1-x)\right)$$

where $\mathcal{X}_A$ denotes the indicator function. The Fourier coefficients of f are

$$\hat{f}_{\boldsymbol{n}} = \prod_{j=1}^{d} \begin{cases} \sqrt{3/4} & : n_j = 0 \\ \sqrt{3/4}\left(\frac{\sin(n_j\pi/2)}{n_j\pi/2}\right)^2 \cos(n_j\pi) & : \text{otherwise.} \end{cases}$$

Therefore the Fourier coefficients decay like $\mathcal{O}(\prod_{j=1}^{d} n_j^{-2})$ and a candidate for an index set $\mathcal{I}$ would be a d-dimensional hyperbolic cross

$$\mathcal{I}_N^{d,\mathrm{hc}} := \left\{ \boldsymbol{n} \in \mathbb{Z}^d : \prod_{j=1}^{d} \max(1, |n_j|) \leq N \right\}.$$

In particular we used a radius of $N = 16$, a reconstructing rank-1 lattice $\mathcal{X}$ from [26, table 6.2] with $M = 1\,105\,193$ nodes and set the weights in Fourier space to $\hat{w}_{\boldsymbol{n}} = \prod_{j=1}^{d} \max(|n_j|^2, 1)$.

Applying Algorithm 1 to $f_{\boldsymbol{x}} = f(\boldsymbol{x}) + \varepsilon$, $\boldsymbol{x} \in \mathcal{X}$, where ε denotes 5% Gaussian noise we computed the cross-validation scores $P(\lambda) = V(\lambda)$ for $\lambda \in [2^{-9}, 2^0]$. Again both scores coincide since the diagonal entries (3.2) of the hat matrix are multiples of the constant weights $w_{\boldsymbol{x}}$ in spatial domain. For the multiplications with $\boldsymbol{F}$ and $\boldsymbol{F}^{\mathsf{H}}$ we made use of fast rank-1 lattices Fourier transforms. A comparison of the actual $L_2(\mathbb{T}^d)$-approximation error with the cross-validation score is plotted in Figure 3.2. We observe that the optimal λ with respect to the $L_2(\mathbb{T}^d)$-error and the λ chosen by cross-validation are very close in this example.
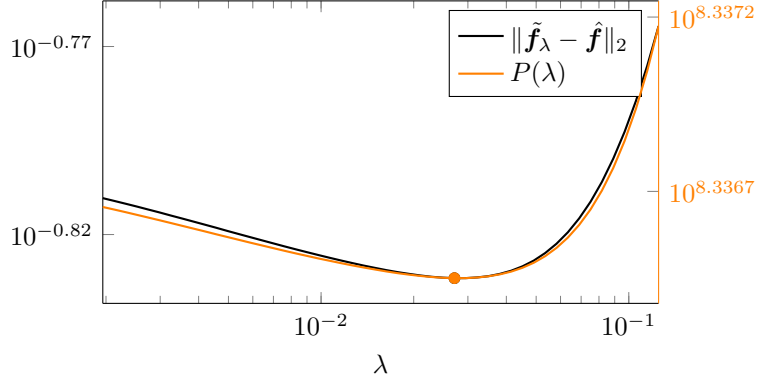
Figure 3.2: Approximation in $\mathbb{T}^7$ from data at a rank-1 lattice: Comparison of the ordinary cross-validation score $P(\lambda)$ with the approximation error.

3.4 Approximative quadrature

In the case of scattered data approximation exact quadrature rules are typically not available. In principle, one can compute exact quadrature rules by determining the weighs $\boldsymbol{W} = \mathrm{diag}(w_{\boldsymbol{x}})_{\boldsymbol{x} \in \mathcal{X}}$ as a solution of the linear system

$$\boldsymbol{F}_{\mathcal{X},\mathcal{D}(\mathcal{I})}^{\mathsf{H}} \boldsymbol{w} = \left(\mathrm{e}^{2\pi \mathrm{i}\boldsymbol{x} \cdot \boldsymbol{n}}\right)_{\boldsymbol{x} \in \mathcal{X}, \boldsymbol{n} \in \mathcal{D}(\mathcal{I})}^{\mathsf{H}} \boldsymbol{w} = \boldsymbol{e_0},$$

where $\boldsymbol{e_0}$ is the vector which only contains zeros, except in the $\boldsymbol{0}$-th position where it is equal to one. These weights can be guaranteed to be non negative under certain conditions on the frequency index set $\mathcal{I}$ and the mesh norm

$$\delta_{\mathcal{X}} := \max_{\boldsymbol{y} \in \mathcal{M}} \min_{\boldsymbol{x} \in \mathcal{X}} |\boldsymbol{y} - \boldsymbol{x}|$$

of the nodes $\mathcal{X}$, cf. [12]. However, those conditions strongly restrict the polynomial degree and do not guaranty the quadrature weights to be non-oscillating. This may be problematic, since the weights directly alter the problem (1.1) we want to solve.

Our idea is to replace the exact quadrature weights $\boldsymbol{W}$ by approximative weights derived from the Voronoi tessellation of the node set $\mathcal{X}$.

Definition 3.5. *Let $\mathcal{M}$ be a Riemannian manifold with a distance function* $\mathrm{dist}(\cdot, \cdot)$. *For a set of nodes $\mathcal{X} \subset \mathcal{M}$ we define the* Voronoi cell $V_{\boldsymbol{x}}$ *corresponding to $\boldsymbol{x} \in \mathcal{X}$ by*

$$V_{\boldsymbol{x}} := \left\{ \boldsymbol{y} \in \mathcal{M} : \mathrm{dist}(\boldsymbol{x}, \boldsymbol{y}) \leq \mathrm{dist}(\boldsymbol{x'}, \boldsymbol{y}), \ \forall \boldsymbol{x'} \in \mathcal{X} \right\}.$$

The Voronoi weight $w_{\boldsymbol{x}}$ *is the area of the Voronoi cell $V_{\boldsymbol{x}}$*

$$w_{\boldsymbol{x}} := \int_{\mathcal{M}} \mathcal{X}_{V_{\boldsymbol{x}}}(\boldsymbol{y}) \, \mathrm{d}\boldsymbol{y} = \int_{V_{\boldsymbol{x}}} \mathrm{d}\boldsymbol{y}.$$

To emphasize the choice of the Voronoi weights as approximative quadrature weights we make a rough error estimate for the approximated quadrature using the Voronoi weights.

13

Theorem 3.6. *Let $f \colon \mathcal{M} \to \mathbb{C}$ be Lipschitz continuous with the Lipschitz constant L. Let further $\mathcal{X}$ be a set of nodes with mesh norm*

$$\delta_{\mathcal{X}} := \max_{\boldsymbol{y} \in \mathcal{M}} \min_{\boldsymbol{x} \in \mathcal{X}} \mathrm{dist}(\boldsymbol{y}, \boldsymbol{x})$$

and Voronoi weights $w_{\boldsymbol{x}}$. Then

$$\left| \sum_{\boldsymbol{x} \in \mathcal{X}} w_{\boldsymbol{x}} f(\boldsymbol{x}) - \int_{\mathcal{M}} f(\boldsymbol{y}) \, \mathrm{d}\boldsymbol{y} \right| \leq L \delta_{\mathcal{X}} \int_{\mathcal{M}} \mathrm{d}\boldsymbol{y}.$$

Proof. Since the disjoint union of all Voronoi cells $V_{\boldsymbol{x}}$ is $\mathcal{M}$ itself we can decompose the integral as follows

$$\int_{\mathcal{M}} f(\boldsymbol{y}) \, \mathrm{d}\boldsymbol{y} = \sum_{\boldsymbol{x} \in \mathcal{X}} \int_{V_{\boldsymbol{x}}} f(\boldsymbol{y}) \, \mathrm{d}\boldsymbol{y} = \sum_{\boldsymbol{x} \in \mathcal{X}} \left(w_{\boldsymbol{x}} f(\boldsymbol{x}) + \int_{V_{\boldsymbol{x}}} f(\boldsymbol{y}) - f(\boldsymbol{x}) \, \mathrm{d}\boldsymbol{y} \right).$$

Noting that the maximal distance of $\boldsymbol{x}$ to any other node of the corresponding Voronoi cell $V_{\boldsymbol{x}}$ cannot exceed $\delta_{\mathcal{X}}$, we use the Lipschitz continuity to estimate the leftover integrand

$$\left| \sum_{\boldsymbol{x} \in \mathcal{X}} w_{\boldsymbol{x}} f(\boldsymbol{x}) - \int_{\mathcal{M}} f(\boldsymbol{y}) \, \mathrm{d}\boldsymbol{y} \right| = \left| \sum_{\boldsymbol{x} \in \mathcal{X}} \int_{V_{\boldsymbol{x}}} f(\boldsymbol{y}) - f(\boldsymbol{x}) \, \mathrm{d}\boldsymbol{y} \right| \leq L \delta_{\mathcal{X}} \int_{\mathcal{M}} \mathrm{d}\boldsymbol{y}.$$

$\blacksquare$

Remark 3.7. (i) *Theorem 3.6 states that the error of the quadrature formula gets small for smooth functions in the sense of a small Lipschitz constant and for small mesh norms, like for approximately equidistributed nodes.*

(ii) *Deterministic and probabilistic error estimates are available from [19] and [4], respectively.*

(iii) *The Voronoi decomposition is dual to the Delaunay triangulation and thus can be computed in $\mathcal{O}(|\mathcal{X}| \log |\mathcal{X}|)$ for the Euclidean distance in $\mathrm{dist}(\cdot, \cdot)$.*

Given that the error of the Voronoi quadrature is small we obtain approximately

$$\boldsymbol{F}^{\mathsf{H}} \boldsymbol{W} \boldsymbol{F} = \left[\sum_{\boldsymbol{x} \in \mathcal{X}} w_{\boldsymbol{x}} \mathrm{e}^{2\pi \mathrm{i} \boldsymbol{n}_1 \boldsymbol{x}} \overline{\mathrm{e}^{2\pi \mathrm{i} \boldsymbol{n}_2 \boldsymbol{x}}} \right]_{\boldsymbol{n}_1, \boldsymbol{n}_2 \in \mathcal{I}}$$

$$\approx \left[\int_{\mathbb{T}^d} \mathrm{e}^{2\pi \mathrm{i} \boldsymbol{n}_1 \boldsymbol{x}} \mathrm{e}^{-2\pi \mathrm{i} \boldsymbol{n}_2 \boldsymbol{x}} \, \mathrm{d}\boldsymbol{x} \right]_{\boldsymbol{n}_1, \boldsymbol{n}_2 \in \mathcal{I}} = \boldsymbol{I} \in \mathbb{C}^{\mathcal{I} \times \mathcal{I}}$$

where $\boldsymbol{I}$ denotes the identity matrix. Inserting this into the definition of the hat matrix $\boldsymbol{H}$ we have formally

$$\boldsymbol{H} = \boldsymbol{F} \left(\boldsymbol{F}^{\mathsf{H}} \boldsymbol{W} \boldsymbol{F} + \lambda \hat{\boldsymbol{W}} \right)^{-1} \boldsymbol{F}^{\mathsf{H}} \boldsymbol{W} \approx \boldsymbol{F} \left(\boldsymbol{I} + \lambda \hat{\boldsymbol{W}} \right)^{-1} \boldsymbol{F}^{\mathsf{H}} \boldsymbol{W} =: \tilde{\boldsymbol{H}}. \qquad (3.3)$$

Analogously to Theorem 3.2 we obtain for the diagonal entries $\tilde{h}_{\boldsymbol{x},\boldsymbol{x}}$ of the *approximated hat matrix* $\tilde{\boldsymbol{H}}$,

$$\tilde{h}_{\boldsymbol{x},\boldsymbol{x}} = w_{\boldsymbol{x}} \sum_{\boldsymbol{n} \in \mathcal{I}} \frac{1}{1 + \lambda \hat{w}_{\boldsymbol{n}}}.$$

Together with Theorem 2.5 and Definition 2.7 this motivates the following definition of approximated cross-validation scores.

Definition 3.8. *The approximated cross-validation score $\tilde{P}(\lambda)$ and the approximated generalized cross-validation score $\tilde{V}(\lambda)$ are defined by*

$$\tilde{P}(\lambda) = \sum_{\boldsymbol{x} \in \mathcal{X}} \frac{[\boldsymbol{H}\boldsymbol{f} - \boldsymbol{f}]_{\boldsymbol{x}}^2}{(1 - \tilde{h}_{\boldsymbol{x},\boldsymbol{x}})^2} \quad and \quad \tilde{V}(\lambda) = \sum_{\boldsymbol{x} \in \mathcal{X}} \frac{[\boldsymbol{H}\boldsymbol{f} - \boldsymbol{f}]_{\boldsymbol{x}}^2}{(1 - \tilde{h})^2},$$

where $\tilde{h} = \frac{1}{|\mathcal{X}|} \sum_{\boldsymbol{x} \in \mathcal{X}} \tilde{h}_{\boldsymbol{x},\boldsymbol{x}}$.

Remark 3.9. *Algorithm 1 is easily modified to compute the approximated scores by replacing all occurrences of $h_{\boldsymbol{x},\boldsymbol{x}}$ by $\tilde{h}_{\boldsymbol{x},\boldsymbol{x}}$. The computationally most expensive part remains the computation of the Tikhonov minimizer $\tilde{\boldsymbol{f}} = \boldsymbol{H}\boldsymbol{f} = \boldsymbol{F}(\boldsymbol{F}^{\mathsf{H}}\boldsymbol{W}\boldsymbol{F} + \lambda\hat{\boldsymbol{W}})^{-1}\boldsymbol{F}^{\mathsf{H}}\boldsymbol{W}\boldsymbol{f}$. Making use of the NFFT [29, 27] the matrix-vector multiplications with $\boldsymbol{F}$ and $\boldsymbol{F}^{\mathsf{H}}$ can be performed with $\mathcal{O}(|\mathcal{I}| \log |\mathcal{I}| + |\mathcal{X}|)$ numerical operations.*

In the remainder of this sections we illustrate that the approximated cross-validation scores can be computed drastically faster while providing a good approximation to the minimizer of the actual cross-validation score. To this end we considered scattered sampling points on the one-dimensional torus $\mathbb{T}$ as well as on the two-dimensional torus $\mathbb{T}^2$. In order generate sufficiently nonuniform sampling points we drew random samples with respect to the uniform distribution on $[0, 1]$ and $[0, 1]^2$ and squared them. This leads to sampling sets that are more dense towards the point 0 and the edges $0 \times [0, 1]$ and $[0, 1] \times 0$.

In the one-dimensional example we used $|\mathcal{X}| = 128$ sampling points and the index set $\mathcal{I}_{64}^{1d} = \{-32, \ldots, 31\}$. In the two-dimensional example we chose $|\mathcal{X}| = 8192$ and $\mathcal{I}_{64}^{2d} = \mathcal{I}_{64}^{1d} \times \mathcal{I}_{64}^{1d}$. In both cases this corresponds to an oversampling factor of two. As in Subsection 3.2 we chose as a test function the MATLAB peaks function with fixed second argument zero in the one-dimensional case. Finally, we added 5% Gaussian noise as depicted in Figure 3.3, (a) and Figure 3.4, (a).

As in both cases the weights $w_{\boldsymbol{x}}$ are far from uniform we may expect some difference between the ordinary cross-validation score $P(\lambda)$ and the generalized cross-validation score $V(\lambda)$. This can be observed in the one-dimensional example, cf. Figure 3.3. In the two-dimensional example both functionals coincide surprisingly well, cf. Figure 3.4. Furthermore, the weights are completely uncorrelated to the homogeneous noise level and, thus, contradict with Remark 3.4.

Judging the approximation of the exact cross-validation scores $P(\lambda)$ and $V(\lambda)$ by the approximated scores $\tilde{P}(\lambda)$ and $\tilde{V}(\lambda)$ we observe that for small regularization parameters λ the score $\tilde{P}(\lambda)$ contains several peaks for both examples. These peaks occur because we overestimate the diagonal entries $h_{x,x}$ such that they attain values around one and
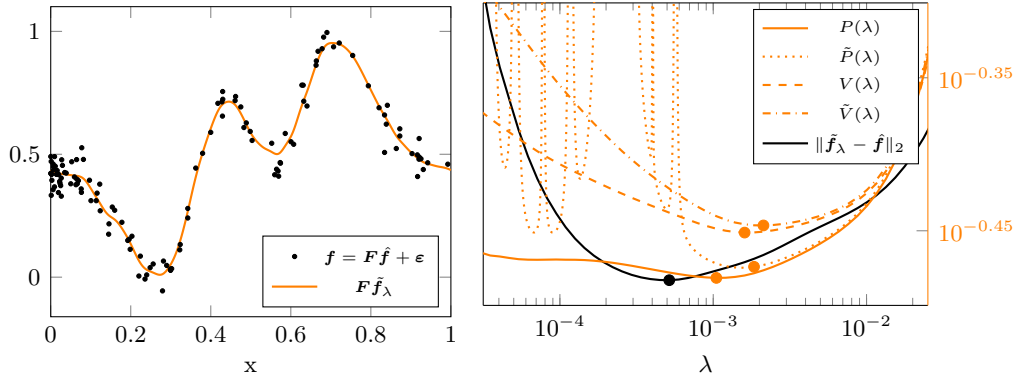
(a) noisy input data $\boldsymbol{f} = \boldsymbol{F}\hat{\boldsymbol{f}} + \boldsymbol{\varepsilon}$ and reconstruction $\boldsymbol{F}\tilde{\boldsymbol{f}}_\lambda$ with λ set to the minimizer of $\tilde{P}(\lambda)$

(b) approximation error $\|\tilde{\boldsymbol{f}}_\lambda - \hat{\boldsymbol{f}}\|_2$ (black) and cross-validation scores (orange)

Figure 3.3: Approximation from nonequispaced data: Comparison of the ordinary cross-validation score $P(\lambda)$ and the generalized cross-validation score $V(\lambda)$ with their approximations $\tilde{P}(\lambda)$ and $\tilde{V}(\lambda)$ and the approximation error.
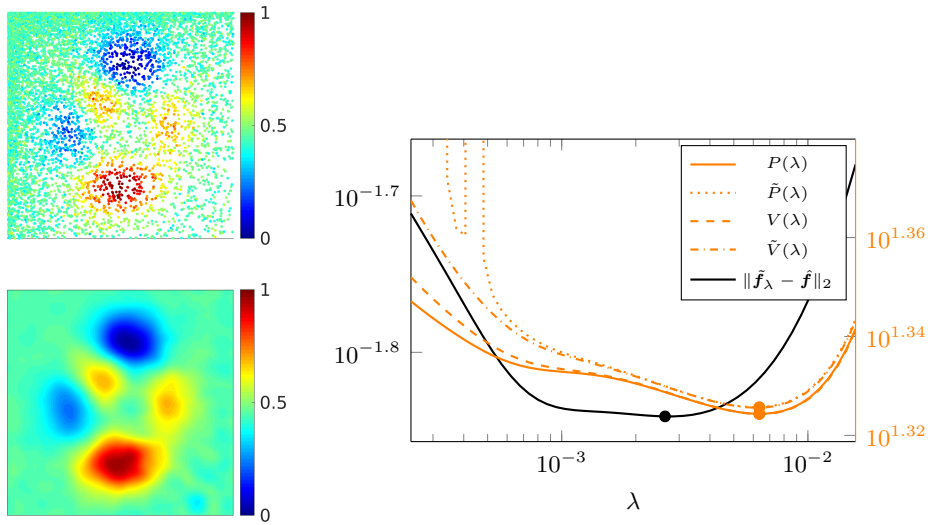
(a) noisy input data and reconstruction

(b) approximation error $\|\tilde{\boldsymbol{f}}_\lambda - \hat{\boldsymbol{f}}\|_2$ (black) and cross-validation scores (orange)

Figure 3.4: Approximation from two-dimensional nonequispaced data: Comparison of the ordinary cross-validation score $P(\lambda)$ and the generalized cross-validation score $V(\lambda)$ with their approximations $\tilde{P}(\lambda)$ and $\tilde{V}(\lambda)$ and the approximation error.

summands of the ordinary cross-validation score (2.7) diverge. In contrast Lemma 2.8 says that these diagonal entries are always smaller than one. Nevertheless, the minimizer of all four functionals $P, \tilde{P}, V, \tilde{V}$ are very close together for the two-dimensional example while for the one-dimensional example the minimizer of P and $\tilde{P}$ are closer to the $L_2(\mathbb{T}^d)$-optimal regularization parameter compared to the minimizer of V and $\tilde{V}$. A natural idea to avoid the oscillatory regions of the functional $\tilde{P}$ would be to use the minimizer of $\tilde{V}$ as initial guess for minimizing $\tilde{P}$.

The central reason for preferring the functional $\tilde{P}$ and $\tilde{V}$ over the functionals P, V is that they are faster to compute. Indeed, if we fix the number of iterations for computing the Tikhonov minimizer, $P(\lambda)$ and $V(\lambda)$ can be acquired in $\mathcal{O}(|\mathcal{I}||\mathcal{X}|\log|\mathcal{I}| + |\mathcal{X}|^2)$ numerical operations for a single regularization parameter λ, which compares to $\mathcal{O}(|\mathcal{I}|\log|\mathcal{I}| + |\mathcal{X}|)$ numerical operations for the evaluation of $\tilde{P}$ and $\tilde{V}$. In our toy example the computation of P and V took approximately 1.61 for the one-dimensional and 1278 seconds for the two-dimensional case, while the computation of $\tilde{P}$ and $\tilde{V}$ was performed within 0.02 and 0.16 seconds averaged over all tested λ.

4 Cross-validation on the unit interval

In this section, we consider the cross-validation scores for nonperiodic functions on the unit interval $[-1, 1]$ with respect to the *Chebyshev polynomials*

$$T_n(x) = \cos(n \arccos x) \quad n = 0, \dots, N-1$$

up to polynomial degree $N \in \mathbb{N}$. In this setting the Fourier matrix becomes

$$\boldsymbol{F} = [T_n(x)]_{x \in \mathcal{X}, n=0,\dots,N-1}$$

for a set of nodes $\mathcal{X}$.

4.1 Exact Quadrature

Similarly as for functions on the torus we consider the case of exact quadrature first. Therefore we remind that the Chebyshev polynomials are orthogonal with respect to the inner product

$$(f, g) = \int_{-1}^{1} \frac{f(x)g(x)}{\sqrt{1-x^2}} \, \mathrm{d}x$$

and are normalized such that

$$(T_{n_1}, T_{n_2}) = \begin{cases} \pi & : n_1 = n_2 = 0, \\ \pi/2 & : n_1 = n_2 \neq 0, \\ 0 & : n_1 \neq n_2. \end{cases}$$

Assuming that the nodes $\mathcal{X}$ and weights $\boldsymbol{W}$ form a quadrature rule that is exact up to polynomial degree $2N - 2$ the diagonal entries of the hat matrix $\boldsymbol{H}$ can be given expicitly using the following theorem.

17

Theorem 4.1. *Let the nodes $\mathcal{X}$ and the weights $\boldsymbol{W} = \mathrm{diag}(w_x)_{x \in \mathcal{X}}$ form a quadrature rule which is exact up to polynomial degree $2N - 2$ and let $\hat{\boldsymbol{W}} = \mathrm{diag}(\hat{w}_0, \ldots, \hat{w}_{N-1})$ be the weights in frequency domain. Then the diagonal entries $h_{x,x}$ of the hat matrix $\boldsymbol{H} = \boldsymbol{F}(\boldsymbol{F}^{\mathsf{H}}\boldsymbol{W}\boldsymbol{F} + \lambda\hat{\boldsymbol{W}})^{-1}\boldsymbol{F}\boldsymbol{W}$ corresponding to the nodes $x \in \mathcal{X}$ satisfy*

$$h_{x,x} = \frac{w_x}{2}\left(\sum_{n=1}^{N-1}\frac{1}{\pi/2 + \lambda\hat{w}_n}\cos(2n\arccos x) + \sum_{n=1}^{N-1}\frac{1}{\pi/2 + \lambda\hat{w}_n} + \frac{2}{\pi + \lambda\hat{w}_0}\right).$$

Proof. Similar to Theorem 3.2 we obtain $\boldsymbol{F}^{\mathsf{H}}\boldsymbol{W}\boldsymbol{F} + \lambda\hat{\boldsymbol{W}} = \mathrm{diag}(\pi + \lambda\hat{w}_0, \pi/2 + \lambda\hat{w}_1, \ldots, \pi/2 + \lambda\hat{w}_{N-1})$. Putting this into the formula for the diagonal elements of the hat matrix obtain

$$h_{x,x} = w_x\left(\sum_{n=1}^{N-1}\frac{1}{\pi/2 + \lambda\hat{w}_n}\cos^2(n\arccos x) + \frac{1}{\pi + \lambda\hat{w}_0}\right).$$

In combination with the addition theorem $\cos(2x) = 2\cos^2 x - 1$ this proves the assertion. ∎

4.2 Chebyshev nodes

The most basic examples of an exact quadrature formula on the interval is probably quadrature at Chebyshev nodes. In order to restate Theorem 4.1 for this case we require the discrete cosine transforms from first up to third kind

$$\boldsymbol{C}_{N+1}^{\mathrm{I}} := \sqrt{\frac{2}{N}}\left[\boldsymbol{\gamma}_N(n)\boldsymbol{\gamma}_N(m)\cos\left(\frac{nm\pi}{N}\right)\right]_{n,m=0}^{N},$$

$$\boldsymbol{C}_N^{\mathrm{II}} := \sqrt{\frac{2}{N}}\left[\boldsymbol{\gamma}_N(n)\cos\left(\frac{n(2m+1)\pi}{2N}\right)\right]_{n,m=0}^{N-1}, \quad \boldsymbol{C}_N^{\mathrm{III}} := (\boldsymbol{C}_N^{\mathrm{II}})^{\mathsf{T}}$$

with $\boldsymbol{\gamma}(0) = \boldsymbol{\gamma}(N) := \sqrt{2}/2$ and $\boldsymbol{\gamma}(n) := 1$ for $n = 1, \ldots, N - 1$. The corresponding matrix-vector products can be calculated using $\mathcal{O}(N \log N)$ arithmetic operations, cf. [39, Chapter 6].

Using the fact that $\boldsymbol{C}_N^{\mathrm{III}}$ is orthonormal, cf. [39, Sec. 3.5], we acquire

$$\boldsymbol{I} = C_N^{\mathrm{II}}C_N^{\mathrm{III}} = \left[\boldsymbol{\gamma}_N(n)\cos\left(\frac{n(2m+1)\pi}{2N}\right)\right]_{n,m=0}^{N-1}\frac{2}{N}\left[\boldsymbol{\gamma}_N(n)\cos\left(\frac{n(2m+1)\pi}{2N}\right)\right]_{m,n=0}^{N-1}.$$

If we multiply with $\mathrm{diag}(\sqrt{\pi}, \sqrt{\pi/2}, \ldots, \sqrt{\pi/2})$ from both sides we obtain

$$\mathrm{diag}(\pi, \pi/2, \ldots, \pi/2) = \left[\cos\left(\frac{n(2m+1)\pi}{2N}\right)\right]_{n,m=0}^{N-1}\frac{\pi}{N}\left[\cos\left(\frac{n(2m+1)\pi}{2N}\right)\right]_{m,n=0}^{N-1}.$$

Putting this into the form $\boldsymbol{F}^{\mathsf{H}}\boldsymbol{W}\boldsymbol{F}$ we see that the *Chebyshev nodes of first kind*

$$x_m = \cos\left(\frac{(2m+1)\pi}{2N}\right), \quad m = 0, \ldots, N - 1$$

and the uniform weights $w_x = \pi/N$ form a quadrature rule which is exact up to degree $2N - 2$.

For these specific nodes Theorem 4.1 simplifies to:

Theorem 4.2. *Let $\mathcal{X} = \{x_1, \ldots, x_m\}$ be the Chebyshev nodes of first kind and $w_{x_m} = \pi/N$. Then the diagonal entries h_{x_m, x_m} of the hat matrix (2.6) can be written as*

$$h_{x_m, x_m} = \frac{w_{x_m}}{2} \left(\frac{\sqrt{N/2}}{\gamma_{2N}(m)} \left[\boldsymbol{C}_{2N+1}^{\mathsf{I}} \boldsymbol{b} \right]_{2m+1} + \sum_{n=1}^{N-1} \frac{1}{\pi/2 + \lambda \hat{w}_n} \right), \quad m = 0, \ldots, N-1$$

with

$$\boldsymbol{b} = (b_0, \ldots, b_{2N})^{\mathsf{T}} = \left(\frac{2\sqrt{2}}{\pi + \lambda \hat{w}_0}, 0, \frac{1}{\pi/2 + \lambda \hat{w}_1}, 0, \ldots, \frac{1}{\pi/2 + \lambda \hat{w}_{N-1}}, 0, 0 \right)^{\mathsf{T}}.$$

Proof. According to Theorem 4.1 we have

$$h_{x_m, x_m} = \frac{w_{x_m}}{2} \left(\sum_{n=1}^{N-1} \frac{1}{\pi/2 + \lambda \hat{w}_n} \cos\left(\frac{n(2m+1)\pi}{N} \right) + \sum_{n=1}^{N-1} \frac{1}{\pi/2 + \lambda \hat{w}_n} + \frac{2}{\pi + \lambda \hat{w}_0} \right).$$

Using the coefficients $\boldsymbol{b}$ the first sum can be expressed with twice the frequency

$$h_{x_m, x_m} = \frac{w_{x_m}}{2} \left(\frac{1}{\gamma_{2N}(m)} \sum_{n=0}^{2N} b_n \gamma_{2N}(n) \gamma_{2N}(m) \cos\left(\frac{n(2m+1)\pi}{2N} \right) + \sum_{n=1}^{N-1} \frac{1}{\pi/2 + \lambda \hat{w}_n} \right),$$

which is the cosine transform of first kind. ∎

Corollary 4.3. *For fixed λ the ordinary cross-validation score $P(\lambda)$ and the generalized cross-validation score $V(\lambda)$ on the unit intervall for Chebyshev nodes of first kind can be computed in $\mathcal{O}(N \log N)$ using Algorithm 1.*

Proof. Because multiplying with $\boldsymbol{F}$ and $\boldsymbol{F}^{\mathsf{H}}$ can be done using the fast discrete cosine transform, we see that applying the hat matrix can be done in $\mathcal{O}(N \log N)$ and Theorem 4.2 allows us to compute the diagonal entries of the hat matrix in $\mathcal{O}(N \log N)$. ∎

To emphasize our results numerically we chose the peaks sample function f from MATLAB and fixed the second argument to zero. We evaluated f in $N = 128$ Chebyshev nodes and added 5% Gaussian noise as one can see in Figure 4.1, (a). To choose $\hat{\boldsymbol{W}}$ we made use of the following statement from [48, Theorem 7.1] which relates the smoothness of f to the decay of the Chebyshev coefficients $\boldsymbol{a}$: If for $\nu \geq 0$ the derivatives up to $f^{(\nu-1)}$ are absolute continuous and $f^{(\nu)}$ has bounded variation V then $|a_k| \leq 2V/(\pi(k-\nu)^{\nu+1})$. Because in general we do not know anything about the smoothness of the function f we chose $\hat{w}_n = n^3$ as weights which corresponds to a function with one absolute continuous derivative. We used Algorithm 1 to calculate the ordinary cross-validation score $P(\lambda)$ and the generalized cross-validation score $V(\lambda)$ for $\lambda \in [2^{-16}, 2^{-11}]$ and plotted the

(a) noisy input data $\boldsymbol{f} = \boldsymbol{F}\hat{f} + \boldsymbol{\varepsilon}$ and reconstruction $\boldsymbol{F}\tilde{f}_\lambda$ with λ set to the minimizer of $P(\lambda)$

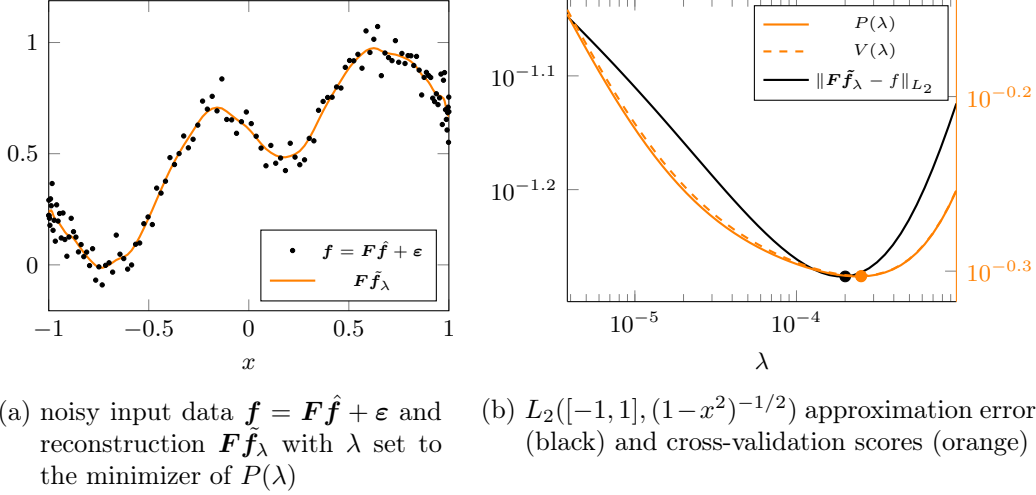(b) $L_2([-1,1], (1-x^2)^{-1/2})$ approximation error (black) and cross-validation scores (orange)

Figure 4.1: Approximation on the unit interval from data at Chebyshev nodes: Comparison of the ordinary cross-validation score $P(\lambda)$ and the generalized cross-validation score $V(\lambda)$ with the approximation error.

regularization for the λ with the smallest corresponding ordinary cross-validation score as one can see in Figure 4.1, (b).

We observe that the ordinary cross-validation score and the generalized cross-validation score differ only slightly and their minima are close to the $L_2([-1,1], (1-x^2)^{-1/2})$ optimal λ.

4.3 Approximative Quadrature

In this section, we consider arbitrary, ordered nodes $x_m \in \mathcal{X} \subset [-1,1]$, $m = 0, \ldots, M$. The corresponding cosine transforms can be computed using the nonequispaced discrete cosine transform, cf. [11], in $\mathcal{O}(N \log N + |\mathcal{X}|)$ where N is the bandwidth. As in Section 3.4 we determine approximate quadrature weights w_{x_m} for $m = 0, \ldots, M$ that allow us to efficiently estimate the diagonal entries of the hat matrix $\boldsymbol{H}$. Since we consider the unit interval with the non-uniform weight function $(1-x^2)^{-\frac{1}{2}}$ it is not a good idea to compute Voronoi weights directly. Instead, we consider the corresponding periodic approximation problem on the unit circle with constant weight by substituting $y_m = \arccos x_m \in [0, \pi]$ and use Voronoi weights with respect to y_m, i.e.,

$$w_{x_m} = \begin{cases} \frac{y_0 + y_1}{2}, & m = 0, \\ \frac{y_{m+1} - y_{m-1}}{2}, & m = 1, \ldots, M-1, \\ \pi - \frac{y_{M-1} + y_M}{2}, & m = M. \end{cases} \tag{4.1}$$

Remark 4.4. *Let x_m be the Chebyshev nodes of first kind. Then the quadrature weights (4.1) coincide with the exact quadrature weights from Section 4.1.*

Analogously to Section 3.4 we use the approxmiated hat matrix $\tilde{\boldsymbol{H}}$ from (3.3) for ease of computation.
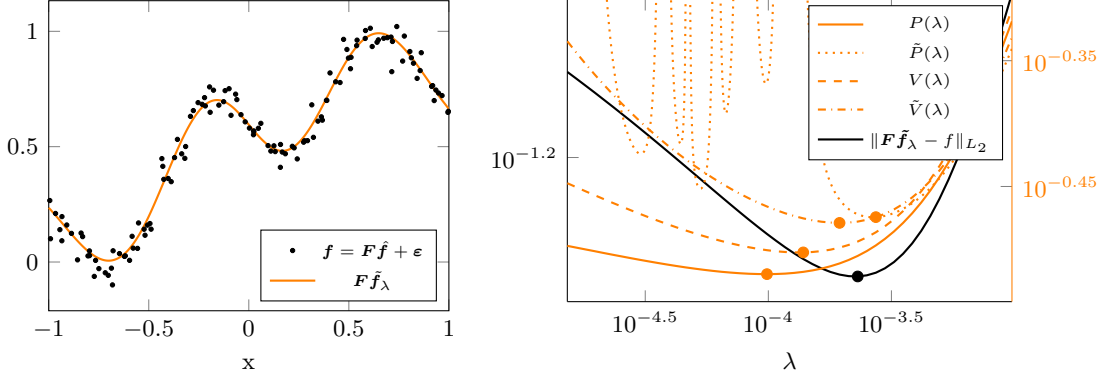
(a) noisy input data $\boldsymbol{f} = \boldsymbol{F}\hat{\boldsymbol{f}} + \boldsymbol{\varepsilon}$ and reconstruction $\boldsymbol{F}\tilde{\boldsymbol{f}}_\lambda$ with λ set to the minimizer of $\tilde{P}(\lambda)$

(b) $L_2([-1,1], (1-x^2)^{-1/2})$ approximation error (black) and cross-validation scores (orange)

Figure 4.2: Approximation from nonequispaced data: Comparison of the ordinary cross-validation score $P(\lambda)$ and the generalized cross-validation score $V(\lambda)$ with their approximations $\tilde{P}(\lambda)$ and $\tilde{V}(\lambda)$ and the approximation error.

Remark 4.5. *Using Theorem 4.1 the diagonal entries can be calculated efficiently with one nonequispaced discrete cosine transform.*

With the given tools we can modify Algorithm 1 to compute $\tilde{P}(\lambda)$ and $\tilde{V}(\lambda)$ from Definition 3.8 in $\mathcal{O}(N \log N + |\mathcal{X}|)$ arithmetic operations given a fixed number of iterations to compute the Tikhonov minimizer.

To exemplify our results we chose 128 uniformly distributed nodes on the unit interval which we perturbed by 5% Gaussian noise. Note that uniformly distributed nodes are far from optimal in the setting of polynomial interpolation on the interval. As in case of exact quadrature we set the bandwith equal to the number of nodes, i.e., $N = |\mathcal{X}|$. As the Voronoi weights resamble quadrature weights the choice of the bandwidth N is critical because in the case of $|\mathcal{X}| < N$ one can not expect to get an exact quadrature formula. As test function we used again the MATLAB peaks function with fixed second argument. Then we computed $P(\lambda)$, $\tilde{P}(\lambda)$, $V(\lambda)$, and $\tilde{V}(\lambda)$ for $\lambda \in [2^{-18}, 2^{-11}]$. The results can be seen in Figure 4.2.

We observe that all cross-validation scores follow the shape of the $L_2([-1,1], (1-x^2)^{-1/2})$-error and their minima are close to the optimal λ. Again, $\tilde{P}(\lambda)$ is affected by oscillations for small λ which are caused by diagonal entries h_{x_m, x_m} close to 1. The computation of the exact $P(\lambda)$ and $V(\lambda)$ averaged over all λ takes 4.07 seconds whereas the approximated $\tilde{P}(\lambda)$ and $\tilde{V}(\lambda)$ outperform this with 0.04 seconds.

5 Cross-validation on the two-dimensional sphere

Approximation on the two-dimensional sphere $\mathbb{S}^2 := \{\boldsymbol{x} \in \mathbb{R}^3 : \|\boldsymbol{x}\|_2 = 1\}$ has been subject of mathematical research for a long time. The base for approximation from scattered

data is formed by positive quadrature rules, Marcinkiewicz-Zygmund inequalities which are investigated in the papers [52, 36, 33, 5], and by bounds for best approximations [44, 51, 21]. Based on these result the relationship between the mesh norm, the separation distance of the sampling points, and optimal approximation rates has been analyzed in the papers [13, 31, 28, 47]. Approximation from noisy data has been considered in [1] and a priori and a posteriori estimates of the approximation error with respect to the regularization parameter have been proven in [38].

Following the approach of the previous sections we again consider the weighted Tikhonov functional (1.1). The analogue of the exponential functions become the spherical harmonics $\{Y_{n,k}\}_{n=0,\dots,\infty,k=-n,\dots,n}$, cf. [14, 2, 37, 7], which we assume to be normalized such that they form an orthonormal basis in $L_2(\mathbb{S}^2)$. For nodes $\mathcal{X} \subset \mathbb{S}^2$ and a maximum polynomial degree $N \in \mathbb{N}$ the Fourier matrix $\boldsymbol{F}$ becomes

$$\boldsymbol{F} = [Y_{n,k}(\boldsymbol{x})]_{\boldsymbol{x}\in\mathcal{X};n=0,\dots,N,k=-n,\dots,n}.$$

As for the weight matrix $\hat{\boldsymbol{W}}$ in Fourier space we consider isotropic weights $\hat{\boldsymbol{W}} = \mathrm{diag}(\hat{w}_{n,k})_{h=0,\dots,N,\,k=-n,\dots,n}$ that depend only the polynomial degree, i.e., $\hat{w}_{n,k} = \hat{w}_n$.

5.1 Exact Quadrature

There are several approaches to exact quadrature on the two-dimensional sphere. The most direct approach is to consider tensor products of Gauss quadrature rules on the circle and the unit interval $[-1, 1]$, cf. [39, Section 9.6]. A relaxation of this idea is to choose the points equally spaced at fixed latitudinal circles which also allows for an explicit computation of the quadrature weights, cf. [42].

A second approach is to choose the quadrature nodes approximately uniform and determine the quadrature weights by solving a linear system of equations. Given that the quadrature nodes are sufficiently well separated and the oversampling factor is sufficiently high, the resulting quadrature weights can guarantied to be nonnegative, cf. [36]. The computation of these quadrature weights can be implemented efficiently using fast spherical Fourier techniques, cf. [32, 30, 27].

A third approach, called Chebyshev quadrature, consists of fixing the weights to be constant and seeking quadrature nodes with a high degree of exactness. The resulting nodes are known as spherical t-designs. Efficient algorithms for their computation are described in [18] with the resulting spherical designs being available at [16]. Finally, one can try to compute both quadrature nodes and weights in an optimization schema, cf. [17].

For this section it is sufficient that the nodes $\mathcal{X}$ and the weights $\boldsymbol{W} = \mathrm{diag}(w_{\boldsymbol{x}})_{\boldsymbol{x}\in\mathcal{X}}$ form an exact quadrature rule of degree $2N$, i.e., $\boldsymbol{F}^{\mathsf{H}}\boldsymbol{W}\boldsymbol{F} = \mathbf{I}$. Under this assumption the diagonal entries of the hat matrix

$$\boldsymbol{H} = \boldsymbol{F}\left(\boldsymbol{F}^{\mathsf{H}}\boldsymbol{W}\boldsymbol{F} + \lambda\hat{\boldsymbol{W}}\right)^{-1}\boldsymbol{F}^{\mathsf{H}}\boldsymbol{W}$$

can by computed efficiently as it is stated in the following theorem.

Theorem 5.1. *Let the nodes $\mathcal{X}$ and the weights $\boldsymbol{W}$ form a quadrature formula $Q_{\mathcal{X},\boldsymbol{W}}$ that is exact for all spherical harmonics up to polynomial degree $2N$ then the diagonal entry corresponding to $\boldsymbol{x}$ of $\boldsymbol{H}$ satisfies*

$$h_{\boldsymbol{x},\boldsymbol{x}} = \frac{w_{\boldsymbol{x}}}{4\pi} \sum_{n=0}^{N} \frac{2n+1}{1+\lambda\hat{w}_n}.$$

Proof. Since $\boldsymbol{F}^{\mathsf{H}}\boldsymbol{W}\boldsymbol{F} = \boldsymbol{I}$ we obtain analogously to the proof of Theorem 3.2

$$\boldsymbol{H} = \boldsymbol{F} \operatorname{diag}\left(\frac{1}{1+\lambda\hat{w}_0}, \ldots, \frac{1}{1+\lambda\hat{w}_N}\right) \boldsymbol{F}^{\mathsf{H}}\boldsymbol{W}.$$

Looking into the diagonal entry corresponding to x and using the addition theorem of spherical harmonics, cf. [37, Theorem 5.11], we obtain the formula

$$h_{\boldsymbol{x},\boldsymbol{x}} = w_{\boldsymbol{x}} \sum_{n=0}^{N} \frac{1}{1+\lambda\hat{w}_n} \sum_{k=-n}^{n} Y_{n,k}(\boldsymbol{x})\overline{Y_{n,k}(\boldsymbol{x})} = \frac{w_{\boldsymbol{x}}}{4\pi} \sum_{n=0}^{N} \frac{2n+1}{1+\lambda\hat{w}_n}.$$

∎

Corollary 5.2. *For fixed λ the ordinary cross-validation score $P(\lambda)$ and the generalized cross-validation score $V(\lambda)$ on the two-dimensional sphere given quadrature nodes and weights can be computed in $\mathcal{O}(N^2 \log N + |\mathcal{X}|)$ using Algorithm 1.*

Proof. Due to Theorem 5.1 we can compute $h_{\boldsymbol{x},\boldsymbol{x}}$ in linear time. Using equation (5.1) applying the hat matrix has the same computational cost as one multiplication with $\boldsymbol{F}$ and one with $\boldsymbol{F}^{\mathsf{H}}$. Using the nonequispaced fast spherical Fourier transform (NFSFT, cf. [32]) this can be done in $\mathcal{O}(N^2 \log N + |\mathcal{X}|)$. ∎

In order to illustrate Theorem 5.1 we consider a quadrature rule consisting of $21\,000$ approximately equidistributed nodes and equal weights $w_{\boldsymbol{x}} = 4\pi/21\,000$ that is exact up to polynomial degree $2N = 200$, as reported in [16]. Since by Theorem 5.1 the diagonal entries $h_{\boldsymbol{x},\boldsymbol{x}}$ of the hat matrix are constant multiples of the constant spatial weights $w_{\boldsymbol{x}}$ the ordinary cross-validation score and the generalized cross-validation score coincide for this setting. For weights in frequency domain we have chosen $\hat{w}_n = (2n)^{2s}$ for $s = 3$ which corresponds to a function with 3 derivatives in $L_2(\mathbb{S}^2)$.

The test function consists of a sum of quadratic B-splines to which we added an error of 5% Gaussian noise for each node as one can see in Figure 5.1, (a). This function was suggested in [23]. We calculated $V(\lambda)$ and $P(\lambda)$ for $\lambda \in [2^{-38}, 2^{-25}]$ using Algorithm 1 with the help of the MATLAB toolbox MTEX, cf. [22]. Furthermore we calculated the $L_2(\mathbb{S}^2)$-error using Parseval from the original $\hat{\boldsymbol{f}}$ and $\hat{\tilde{\boldsymbol{f}}}$ which are a byproduct of Algorithm 1.

As it is illustrated in the Figure 5.1, (b) the minimum of the cross-validation score is very close to the minimum of the approximation error.
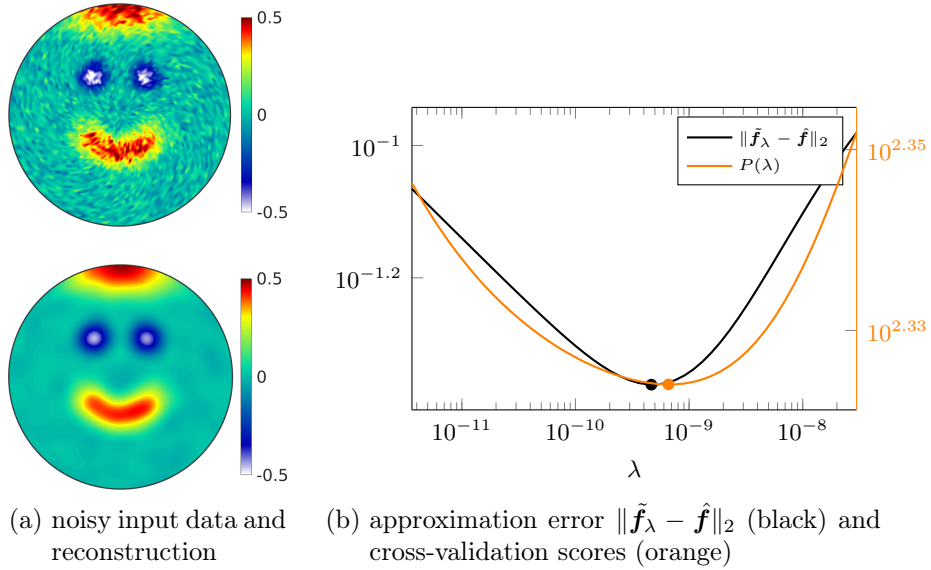
| (a) noisy input data and reconstruction | (b) approximation error $\|\tilde{\boldsymbol{f}}_\lambda - \hat{\boldsymbol{f}}\|_2$ (black) and cross-validation scores (orange) |

Figure 5.1: Approximation from two-dimensional equispaced data: Comparison of the cross-validation score $P(\lambda)$ and the approximation error.

5.2 Approximative quadrature

In the case function values are provided at nodes not forming a suitable quadrature rule we follow the previous ideas of Section 3.4 and 4.3 and use the approximated hat matrix $\tilde{\boldsymbol{H}}$ from (3.3) instead of $\boldsymbol{H}$ itself. This way we acquire $\tilde{P}(\lambda)$ and $\tilde{V}(\lambda)$ as in Definition 3.8. In place of quadrature weights we use a spherical Voronoi decomposition, cf. [41].

The only changes to Algorithm 1 are the prior computation of the Voronoi weights and the necessity of solving a linear system of equations for computing the Tikhonov minimizer $\tilde{\boldsymbol{f}}$. This system of linear equation was solved with a conjugate gradient method, as implemented in the MATLAB `lsqr` command, limited to 20 iterations.

In order to illustrate the efficiency of approximative quadrature weights for estimating the cross-validation score we consider the same test function and $\hat{\boldsymbol{W}}$ as in Example 5.1 and apply Algorithm 1 with polynomial degree $N = 30$ to $|\mathcal{X}| = 2(N+1)^2 = 1922$ random nodes, which corresponds to an oversampling factor of two. Figure 5.2 compares the different cross-validation scores $P(\lambda)$, $V(\lambda)$, $\tilde{P}(\lambda)$ and $\tilde{V}(\lambda)$ for $\lambda \in [2^{-38}, 2^{-25}]$. The interval has been choosen by hand in such a way that it reflects the characteristic features of the functionals. All scores have their minimum close to the minimum of the of the actual approximation error. On the downside, we again observe several peaks in the approximated ordinary cross-validation score for small values of λ. So it is important to start the minimization process with a large λ. We also want to note that the computation of the exact $P(\lambda)$ and $V(\lambda)$ took 227 seconds averaged over all λ in contrast to 0.12 seconds for the approximated $\tilde{P}(\lambda)$ and $\tilde{V}(\lambda)$.

(a) noisy input data and
reconstruction

(b) approximation error $\|\tilde{\boldsymbol{f}}_\lambda - \hat{\boldsymbol{f}}\|_2$ (black) and
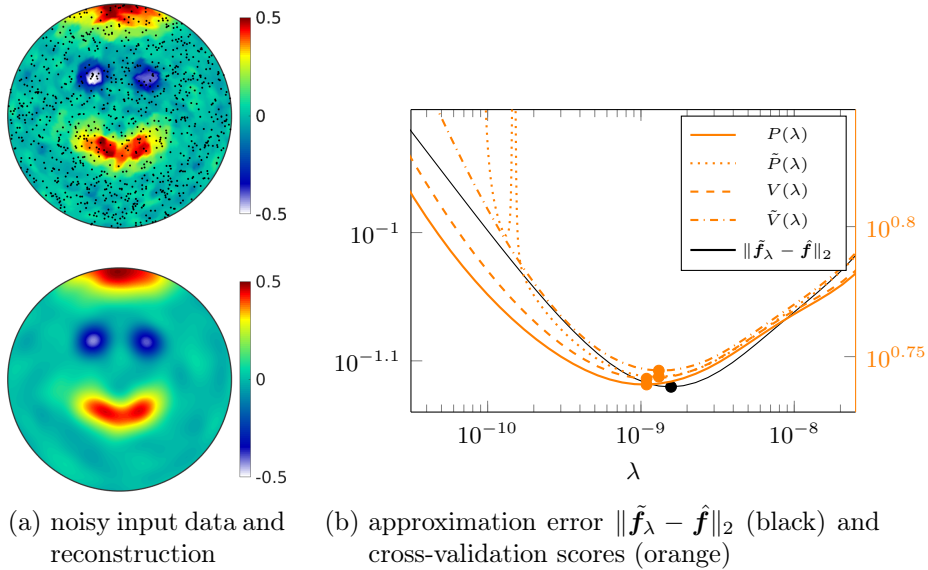cross-validation scores (orange)

Figure 5.2: Approximation from two-dimensional random nodes: Comparison of the ordinary cross-validation score $P(\lambda)$ and the generalized cross-validation score $V(\lambda)$ with their approximations $\tilde{P}(\lambda)$ and $\tilde{V}(\lambda)$ and the approximation error.

6 Conclusion

In this paper we presented a fast algorithm for the computation of the leave-one-out cross-validation score $P(\lambda)$ for the Tikhonov regularizer (1.1). While many approximations of the cross-validation score do not consider every node separately, but fewer and larger validation sets, c.f. [6], we considered the full leave-one-out cros-validation score. In contrast to other approaches we did not restrict ourselves to spline interpolation on the interval at equispaced nodes, but considered more general domains and samplings. The key points of Algorithm 1 are explicit formulas for the diagonal elements $h_{x,x}$ of the hat matrix $\boldsymbol{H}$ which we were able to derive in the Theorems 3.2, 4.2, and 5.1, for approximation on the torus, the interval, and the two-dimensional sphere, respectively. Generalizations to other domains, e.g., the rotation group $SO(3)$, are possible following the framework presented in this paper. For all these domains FFT-like algorithms can be applied to achieve quasilinear complexity with respect to the number of nodes for the computation of the Tikhonov minimizer as well as for the leave-one-out cross-validation score.

The efficiency of our approach has been illustrated in several numerical experiments with respect to the different domains. For the nodes we distinguished two settings. For nodes belonging to a quadrature rule, like equispaced nodes or rank-1 lattices on the torus, our Algorithm 1 computes the cross-validation score $P(\lambda)$ with floating point precision, cf. Corollaries 3.3, 4.3, and 5.2. For arbitrary nodes we accomplished in Remarks 3.9, 4.5 and Corollary 5.2 a good approximation using Voronoi weights in place of the quadrature weights. The numerical experiments confirm our theoretical results.

In all test scenarios our algorithm was several orders of magnitude faster then the direct reference implementation.

In some cases the approximated leave-one-out cross-validation score $\tilde{P}(\lambda)$ suffered from peaks for λ smaller than the optimal one, cf. Subsection 3.4. Anyway, in our test cases we had no problems finding the global minimum by initializing the line search algorithm with a sufficiently large λ and thus avoiding the oscillatory region.

All relevant MATLAB code, including the algorithm for the fast computation of the leave-out-one cross-validation score, its minimizer and all numerical examples of this paper can be found on the GitHub repository `https://github.com/felixbartel/fcv`.

References

[1] C. An, X. Chen, I. H. Sloan, and R. S. Womersley. Regularized least squares approximations on the sphere using spherical designs. *SIAM J. Numer. Anal.*, 50:1513–1534, 2012.

[2] K. Atkinson and W. Han. *Spherical Harmonics and Approximations on the Unit Sphere: An Introduction*, volume 2044 of *Lecture Notes in Mathematics*. Springer, Heidelberg, 2012.

[3] A. B. Bakushinskiĭ. Remarks on the choice of regularization parameter from quasioptimality and relation tests. *Zh. Vychisl. Mat. i Mat. Fiz.*, 24(8):1258–1259, 1984.

[4] R. F. Bass and K. Gröchenig. Random sampling of multivariate trigonometric polynomials. *SIAM J. Math. Anal.*, 36:773–795, 2004.

[5] A. Böttcher, S. Kunis, and D. Potts. Probabilistic spherical Marcinkiewicz-Zygmund inequalities. *J. Approx. Theory*, 157:113–126, 2009.

[6] A. Caponnetto and Y. Yao. Cross-validation based adaptation for regularization operators in learning theory. *Anal. Appl. (Singap.)*, 8(2):161–183, 2010.

[7] F. Dai and Y. Xu. *Approximation Theory and Harmonic Analysis on Spheres and Balls*. Springer Monographs in Mathematics. Springer, New York, 2013.

[8] E. De Vito, S. Pereverzyev, and L. Rosasco. Adaptive kernel methods using the balancing principle. *Found. Comput. Math.*, 10(4):455–479, 2010.

[9] L. N. Deshpande and D. Girard. Fast computation of cross-validated robust splines and other non-linear smoothing splines. *Curves and Surfaces*, pages 143–148, 1991.

[10] D. Dung, V. N. Temlyakov, and T. Ullrich. Hyperbolic Cross Approximation. *Advanced Courses in Mathematics. CRM Barcelona. Birkhäuser/Springer*, 2018.

[11] M. Fenn and D. Potts. Fast summation based on fast trigonometric transforms at nonequispaced nodes. *Numer. Linear Algebra Appl.*, 12:161–169, 2005.

[12] F. Filbir and H. Mhaskar. Marcinkiewicz-Zygmund measures on manifolds. *J. Complexity*, 27:568–598, 2011.

[13] F. Filbir and W. Themistoclakis. Polynomial approximation on the sphere using scattered data. *Math. Nachr.*, 281:650–668, 2008.

[14] W. Freeden, T. Gervens, and M. Schreiner. *Constructive Approximation on the Sphere*. Oxford University Press, Oxford, 1998.

[15] G. H. Golub, M. Heath, and G. Wahba. Generalized cross-validation as a method for choosing a good ridge parameter. *Technometrics*, 21(2):215–223, 1979.

[16] M. Gräf. Numerical spherical designs on S^2. `http://www.tu-chemnitz.de/~potts/workgroup/graef/quadrature`, 2010.

[17] M. Gräf. *Efficient Algorithms for the Computation of Optimal Quadrature Points on Riemannian Manifolds*. Dissertation. Universitätsverlag Chemnitz, 2013.

[18] M. Gräf and D. Potts. On the computation of spherical designs by a new optimization approach based on fast spherical Fourier transforms. *Numer. Math.*, 119:699–724, 2011.

[19] K. Gröchenig. Reconstruction algorithms in irregular sampling. *Math. Comput.*, 59:181–194, 1992.

[20] C. Gu. *Smoothing spline ANOVA models*, volume 297 of *Springer Series in Statistics*. Springer, New York, second edition, 2013.

[21] K. Hesse and I. H. Sloan. Hyperinterpolation on the sphere. In N. K. Govil, H. N. Mhaskar, R. N. Mohapatra, Z. Nashed, and J. Szabados, editors, *Frontiers in Interpolation and Approximation*, Pure and Applied Mathematics. Taylor & Francis Books, Boca Raton, Florida, 2006.

[22] R. Hielscher. MTEX 5.1 - A matlab toolbox for crystallographic texture analysis. `http://mtex-toolbox.github.io/`.

[23] R. Hielscher and M. Quellmalz. Optimal mollifiers for spherical deconvolution. *Inverse Problems*, 31(8):085001, 2015.

[24] L. Kämmerer. *High Dimensional Fast Fourier Transform Based on Rank-1 Lattice Sampling*. Dissertation. Universitätsverlag Chemnitz, 2014.

[25] L. Kämmerer. LFFT, Matlab$^\circledR$ toolbox for the lattice and generated set based FFT. `http://www.tu-chemnitz.de/~lkae/lfft`, 2014.

[26] L. Kämmerer, D. Potts, and T. Volkmer. Approximation of multivariate periodic functions by trigonometric polynomials based on rank-1 lattice sampling. *J. Complexity*, 31:543–576, 2015.

[27] J. Keiner, S. Kunis, and D. Potts. NFFT 3.5, C subroutine library. `http://www.tu-chemnitz.de/~potts/nfft`. Contributors: F. Bartel, M. Fenn, T. Görner, M. Kircheis, T. Knopp, M. Quellmalz, M. Schmischke, T. Volkmer, A. Vollrath.

[28] J. Keiner, S. Kunis, and D. Potts. Efficient reconstruction of functions on the sphere from scattered data. *J. Fourier Anal. Appl.*, 13:435–458, 2007.

[29] J. Keiner, S. Kunis, and D. Potts. Using NFFT3 - a software library for various nonequispaced fast Fourier transforms. *ACM Trans. Math. Software*, 36:Article 19, 1–30, 2009.

[30] J. Keiner and D. Potts. Fast evaluation of quadrature formulae on the sphere. *Math. Comput.*, 77:397–419, 2008.

[31] S. Kunis. A note on stability results for scattered data interpolation on Euclidean spheres. *Adv. Comput. Math.*, 30:303–314, 2009.

[32] S. Kunis and D. Potts. Fast spherical Fourier algorithms. *J. Comput. Appl. Math.*, 161:75–98, 2003.

[33] Q. T. Le Gia and H. N. Mhaskar. Quadrature formulas and localized linear polynomial operators on the sphere. *SIAM J. Numer. Anal.*, 47:440–466, 2008.

[34] K.-C. Li. Asymptotic optimality of C_L and generalized cross-validation in ridge regression with application to spline smoothing. *Ann. Statist.*, 14(3):1101–1112, 1986.

[35] M. A. Lukas, F. R. de Hoog, and R. S. Anderssen. Efficient algorithms for robust generalized cross-validation spline smoothing. *J. Comput. Appl. Math.*, 235:102–107, 2010.

[36] H. N. Mhaskar, F. J. Narcowich, and J. D. Ward. Spherical Marcinkiewicz-Zygmund inequalities and positive quadrature. *Math. Comput.*, 70:1113–1130, 2001. Corrigendum on the positivity of the quadrature weights in 71:453–454, 2002.

[37] V. Michel. *Lectures on Constructive Approximation: Fourier, Spline, and Wavelet Methods on the Real Line, the Sphere, and the Ball.* Birkhäuser, New York, 2013.

[38] S. Pereverzyev, I. Sloan, and P. Tkachenko. Parameter choice strategies for least-squares approximation of noisy smooth functions on the sphere. *SIAM J. Numer. Anal.*, 53:820–835, 2015.

[39] G. Plonka, D. Potts, G. Steidl, and M. Tasche. *Numerical Fourier Analysis.* Applied and Numerical Harmonic Analysis. Birkhäuser, 2018.

[40] C. H. Reinsch. Smoothing by spline functions. I, II. *Numer. Math.*, 10:177–183; ibid. 16 (1970/71), 451–454, 1967.

[41] R. J. Renka. Algorithm 772: Stripack: Delaunay triangulation and voronoi diagram on the surface of a sphere. *ACM Trans. Math. Softw.*, 23:416–434, 1997. For accompanying software, see `http://www.acm.org/pubs/calgo`.

[42] D. Rosca. Spherical quadrature formulas with equally spaced nodes on latitudinal circles. *Electron. Trans. Numer. Anal.*, 35:148–163, 2009.

[43] R. B. Sidje, A. B. Williams, and K. Burrage. Fast generalized cross validation using Krylov subspace methods. *Numer Algor*, 47:109–131, 2008.

[44] I. Sloan and R. Womersley. The uniform error of hyperinterpolation on the sphere. In W. Haußmann, K. Jetter, and M. Reimer, editors, *Advances in Multivariate Approximation*, volume 107 of *Mathematical Research*, pages 289–306. Wiley VCH, Berlin, 1999.

[45] M. Tasche and N. Weyrich. Smoothing inversion of Fourier series using generalized cross-validation. *Results Math.*, 29(1-2):183–195, 1996.

[46] V. N. Temlyakov. *Approximation of periodic functions*. Computational Mathematics and Analysis Series. Nova Science Publishers Inc., Commack, NY, 1993.

[47] W. Themistoclakis and M. V. Barel. Uniform approximation on the sphere by least squares polynomials. *Numer. Alg.*, 81:1089–1111, 2019.

[48] L. N. Trefethen. *Approximation theory and approximation practice*. Society for Industrial and Applied Mathematics (SIAM), Philadelphia, PA, 2013.

[49] G. Wahba. *Spline Models for Observational Data*. Springer, SIAM, 1990.

[50] H. L. Weinert. Efficient computation for Whittaker-Henderson smoothing. *Comp. Stat. & Data Analysis*, 52:959–974, 2007.

[51] R. S. Womersley and I. H. Sloan. How good can polynomial interpolation on the sphere be? *Adv. Comput. Math.*, 14:195–226, 2001.

[52] Y. Xu and E. W. Cheney. Strictly positive definite functions on spheres. *Proc. Amer. Math. Soc.*, 116:977–981, 1992.