TECHNISCHE UNIVERSITÄT
CHEMNITZ

Master Thesis

Fast Cross-validation in Harmonic Approximation

Felix Bartel

Supervisors: Dr. Ralf Hielscher and Prof. Dr. Daniel Potts

June 12, 2019

Contents

1 Introduction

A central problem in approximation theory is the reconstruction of functions $f\colon X \to \mathbb{C}$ from noisy function values. That is, for a finite set of nodes $\mathcal{X} \subset X$, we are given function values

$$\boldsymbol{f} = (f(x) + \varepsilon_x)_{x \in \mathcal{X}} \in \mathbb{C}^{|\mathcal{X}|}$$

where we assume ε_x to be zero mean Gaussian noise. Considering a family of basis functions

$$\varphi_{\boldsymbol{n}}\colon X \to \mathbb{C}, \quad \boldsymbol{n} \in \mathcal{I}$$

for an index set $\mathcal{I}$, we are seeking Fourier coefficients $\hat{g}_{\boldsymbol{n}}$, $\boldsymbol{n} \in \mathcal{I}$ such that

$$\sum_{\boldsymbol{n} \in \mathcal{I}} \hat{g}_{\boldsymbol{n}} \varphi_{\boldsymbol{n}}$$

reconstructs the original function "reasonably well". To put this in concrete terms we use the common method of minimizing the Tikhonov functional

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

for $\boldsymbol{F} = \boldsymbol{F}_{\mathcal{X},\mathcal{I}} = (\varphi_{\boldsymbol{n}}(x))_{x \in \mathcal{X}, \boldsymbol{n} \in \mathcal{I}}$ the Fourier matrix, $\|\cdot\|_{\boldsymbol{W}}$ a weighted norm in space domain, $\|\cdot\|_{\hat{\boldsymbol{W}}}$ a weighted norm in frequency domain with $\boldsymbol{W} = \operatorname{diag}(w_x)_{x \in \mathcal{X}}$, $w_x > 0$ and $\hat{\boldsymbol{W}} = \operatorname{diag}(w_{\boldsymbol{n}})_{\boldsymbol{n} \in \mathcal{I}}$, $w_{\boldsymbol{n}} > 0$, and a regularization parameter $\lambda > 0$.

For a fixed λ we obtain a reconstruction by minimizing (1.1), which can be done by solving a system of equations and will be recapped in the beginning of Chapter 2.

The term $\boldsymbol{F}\hat{\boldsymbol{g}}$ is simply the evaluation of our approximation in the given nodes $\mathcal{X}$ and, thus, the first term can be thought of as a data fitting term. The second term controls the smoothing of the approximation by penalizing the Fourier coefficients. By using this approach the degrees of freedom shrink from $|\mathcal{I}|$ of the Fourier coefficients $\hat{\boldsymbol{g}}$ to one of the "optimal" λ. Choosing λ optimal means that the data fitting term and the smoothing term in (1.1) are weighted properly with respect to the given problem. There are many different strategies like the discrepancy principle or the L-curve method to do that but we will focus on a cross-validation approach. In cross-validation the set of nodes is split into one for computing an approximation and one for the validation. We are interested in the special case of "leave-one-out" cross-validation, i.e., for a fixed regularization parameter λ and any node $x \in \mathcal{X}$ we compute the optimal $\hat{\boldsymbol{g}}_{\lambda,(x)}$ of the functional (1.1) restricted to the set of nodes $\mathcal{X} \setminus \{x\}$ and use

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

as cross-validation score, where $(\boldsymbol{F}\hat{\boldsymbol{g}}_{\lambda,(x)})_x$ is the approximation and f_x is the given function value in the node x, respectively.

Originated is this approach in [15] from Golub, Heath and Whaba where splines were used but it spread to many parameter estimating problems since then. The biggest criticism is that we have to compute $|\mathcal{X}|$ Tikhonov minimizers, which is computationally too expensive for most applications. There are several approaches to lower this computational cost. For instance in spline interpolation on the interval or higher-dimensional domains Monte Carlo approximations [8], matrix decomposition methods [54, 46] and Krylov space methods [34] have been used. 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 [48] which requires to solve the minimization problem only once for each regularization parameter λ. The idea of this thesis is to generalize the approach in [48] to arbitrary sampling nodes and other domains like the unit interval or the two-dimensional sphere.

The thesis is based on our paper [3] and is organized as follows. The second chapter deals with the details of the cross-validation score and proves in Theorem 2.4 a reformulation which allows to separate the computation of the cross-validation score into the calculation of one Tikhonov minimizer of (1.1) and $|\mathcal{X}|$ function-independent values $h_{x,x}$. These values $h_{x,x}$ are the diagonal entries of the so called hat matrix $\boldsymbol{H} = \boldsymbol{F}(\boldsymbol{F}^{\mathsf{H}}\boldsymbol{W}\boldsymbol{F}+\lambda\hat{\boldsymbol{W}})^{-1}\boldsymbol{F}^{\mathsf{H}}\boldsymbol{W}$ and their efficient (approximate) computation is subject to the Chapters 4–7.

In between, Chapter 3 illuminates the given approximation problem from the view of reproducing kernel Hilbert spaces and provides a different interpretation on the weights $\hat{\boldsymbol{W}}$ in frequency domain.

Chapters 4–7 deal with the fast computation of the diagonal entries $h_{x,x}$ in the cases of the d-dimensional torus $\mathbb{T}^d$, the unit interval $[-1, 1]$, the two-dimensional sphere $\mathbb{S}^2$, and the rotation group $SO(3)$. They are divided into one case where quadrature nodes and weights are given and we obtain with Theorems 4.2, 5.2, 6.1, and 7.1 the corresponding formulae for the diagonal entries $h_{x,x}$. This requirement is satisfied for regular tensor product grids and rank-1 lattices on the d-dimensional torus, Chebyshev nodes on the unit interval $[-1, 1]$, e.g., Gauss Legendre nodes on the two-dimensional unit sphere $\mathbb{S}^2$, and Gauss quadrature rules on the rotation group $SO(3)$.

In the case that no quadrature nodes and weights are given we suggest approximating them by the volume of the corresponding Voronoi cells and carry out the same computations. Our numerical tests in Sections 4.4, 5.3, 6.2, and 7.2 indicate a good approximation of the true cross-validation score, which is much more expensive to compute.

Together with fast Fourier algorithms on the torus [29, 27], for rank-1 lattices [24, 25], on the interval [10, 27], on the sphere [28, 27], and the rotation group [43] 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.

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

2.1 Basic Concepts

Let us start this chapter 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*

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

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

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

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 $\hat{\boldsymbol{g}}_\lambda = (\boldsymbol{F}^{\mathsf{H}}\boldsymbol{W}\boldsymbol{F} + \lambda\hat{\boldsymbol{W}})^{-1}\boldsymbol{F}^{\mathsf{H}}\boldsymbol{W}\boldsymbol{f}$. Using the positive definiteness we see that $\hat{\boldsymbol{g}}_\lambda$ fulfills the required minimizing property. ∎

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 to $x \in \mathcal{X}$ 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

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

Definition 2.2. *The* ordinary cross-validation score *for the Tikhonov functional* (1.1) *is defined as*

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

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

Interpreting (2.3), we are comparing the predicted or smoothened value $(\boldsymbol{F}\hat{\boldsymbol{g}}_{(x)})_x$ with the noisy data f_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. 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 $\hat{\boldsymbol{g}}$ and $\hat{\boldsymbol{g}}_{(x)}$, respectively, and show the following relationship between their inverse, cf. [15].

Lemma 2.3. *Let* $\boldsymbol{A}$ *and* $\boldsymbol{A}_{(x)}$ *be defined as in* (2.4) *and*

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

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*

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

which when applied to a data vector $\boldsymbol{f}$ solves the normal equation (2.1) and evaluates the resulting function in the nodes $\mathcal{X}$. The next lemma is a generalization of [15, equation 2.2] and [53, 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 2.4. *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).

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

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

Next we apply Lemma 2.3 and observe that the denominator in (2.5) can be expressed in terms of the diagonal entries $h_{x,x}$ of the hat matrix $\boldsymbol{H}$:

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

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

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

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

Remark 2.5. *According to Theorem 2.4 the ordinary cross-validation score is nothing more than the weighted norm of the residue*

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

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. [53, section 4.3].

Definition 2.6. *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 2.7. *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. Furthermore, 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 $\hat{\boldsymbol{g}}_\lambda$ of (1.1) converges to the zero vector as $\lambda \to \infty$, we obtain for $\boldsymbol{f} = \boldsymbol{e}_x$ and $\lambda \to \infty$

$$h_{x,x} = (\boldsymbol{F}\hat{\boldsymbol{g}}_\lambda)_x \to 0.$$

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

2.2 Algorithm to Compute the Cross-validation Scores

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 $\boldsymbol{W} = \operatorname{diag}(w_x)_{x\in\mathcal{X}} \in \mathbb{R}^{|\mathcal{X}|\times|\mathcal{X}|}$
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 $\hat{\boldsymbol{g}} \coloneqq \boldsymbol{A}^{-1}\boldsymbol{F}^{\mathsf{H}}\boldsymbol{W}\boldsymbol{f}$, where $\boldsymbol{A}$ is given in (2.4).
2. Compute $\boldsymbol{g} \coloneqq \boldsymbol{H}\boldsymbol{f} = \boldsymbol{F}\hat{\boldsymbol{g}}$
3. Compute $h_{x,x} \coloneqq w_x \boldsymbol{F}_{x,:}\boldsymbol{A}^{-1}\boldsymbol{F}^{\mathsf{H}}_{x,:}$ for $x \in \mathcal{X}$ and $h \coloneqq \dfrac{1}{|\mathcal{X}|}\displaystyle\sum_{x\in\mathcal{X}} h_{x,x}$.
4. Evaluate $P(\lambda) \coloneqq \displaystyle\sum_{x\in\mathcal{X}} \frac{|\tilde{f}_x - f_x|^2}{(1-h_{x,x})^2}$ and $V(\lambda) \coloneqq \displaystyle\sum_{x\in\mathcal{X}} \frac{|\tilde{f}_x - f_x|^2}{(1-h)^2}$.

Lemma 2.8. *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*

$$\left(\boldsymbol{M}^{\mathsf{H}}\boldsymbol{M}\right)\hat{\boldsymbol{g}} = \boldsymbol{M}^{\mathsf{H}}\boldsymbol{b},$$

which the LSQR method solves, is equivalent to (2.1).

Proof. With basic linear algebra we obtain

$$\begin{aligned} & \left(\boldsymbol{M}^{\mathsf{H}}\boldsymbol{M}\right)\hat{\boldsymbol{g}} = \boldsymbol{M}^{\mathsf{H}}\boldsymbol{b} \\ \Leftrightarrow \quad & \left(\begin{pmatrix}\boldsymbol{F}^{\mathsf{H}}\boldsymbol{W}^{1/2} & \sqrt{\lambda}\hat{\boldsymbol{W}}^{1/2}\end{pmatrix}\begin{pmatrix}\boldsymbol{W}^{1/2}\boldsymbol{F} \\ \sqrt{\lambda}\hat{\boldsymbol{W}}^{1/2}\end{pmatrix}\right)\hat{\boldsymbol{g}} = \begin{pmatrix}\boldsymbol{F}^{\mathsf{H}}\boldsymbol{W}^{1/2} & \sqrt{\lambda}\hat{\boldsymbol{W}}^{1/2}\end{pmatrix}\begin{pmatrix}\boldsymbol{W}^{1/2}\boldsymbol{f} \\ \boldsymbol{0}\end{pmatrix} \\ \Leftrightarrow \quad & \left(\boldsymbol{F}^{\mathsf{H}}\boldsymbol{W}\boldsymbol{F} + \lambda\hat{\boldsymbol{W}}\right)\hat{\boldsymbol{g}} = \boldsymbol{F}^{\mathsf{H}}\boldsymbol{W}\boldsymbol{f}. \end{aligned}$$

■

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

2.3 Derivative of the Cross-validation Scores

In the end, we want to minimize the cross-validation scores. Therefore it is reasonable to seek for an efficient way to compute the derivative with respect to λ.

Theorem 2.9. *Let λ be fixed and $\boldsymbol{f} \in \mathbb{C}^{|\mathcal{X}|}$ function values. Then the derivative of the cross-validation score evaluates to*

$$\frac{\partial}{\partial\lambda}P(\lambda) = 2\sum_{x\in\mathcal{X}}\left(\frac{((\boldsymbol{H}\boldsymbol{f})_x - f_x)(\frac{\partial}{\partial\lambda}(\boldsymbol{H}\boldsymbol{f})_x)}{(1-h_{x,x})^2} + \frac{((\boldsymbol{H}\boldsymbol{f})_x - f_x)^2(\frac{\partial}{\partial\lambda}h_{x,x})}{(1-h_{x,x})^3}\right)$$

with

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

Proof. We simply have to use the quotient rule to the ordinary cross-validation score after applying Theorem 2.4

$$P(\lambda) = \sum_{x\in\mathcal{X}}\left(\frac{(\boldsymbol{H}\boldsymbol{f})_x - f_x}{1-h_{x,x}}\right)^2$$

and use

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

to obtain $\frac{\partial}{\partial\lambda}H\boldsymbol{f}$. ∎

Remark 2.10. *(i) Similar to Lemma 2.8 we can perform the computation of $\frac{\partial}{\partial\lambda}\boldsymbol{H}\boldsymbol{f}$ from Theorem 2.9 efficient and numerically more stable by using $\boldsymbol{M}$ from Lemma 2.8 and*

$$\boldsymbol{c} = \begin{pmatrix}\boldsymbol{0}\\ \frac{1}{\sqrt{\lambda}}\hat{\boldsymbol{W}}^{1/2}\hat{\boldsymbol{g}},\end{pmatrix}$$

where $\boldsymbol{0}$ is a vector containing $|\mathcal{X}|$ zeros and

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

which is a byproduct of Algorithm 1. Now we can apply the LSQR method with the coefficient matrix $\boldsymbol{M}$ and the right-hand side $\boldsymbol{c}$. Multiplying the result with $-\boldsymbol{F}$ we obtain $\frac{\partial}{\partial\lambda}\boldsymbol{H}\boldsymbol{f}$.

(ii) Given $\frac{\partial}{\partial\lambda}h_{x,x}$, the computational effort to compute $P(\lambda)$ and $\frac{\partial}{\partial\lambda}P(\lambda)$ approximately doubles compared to computing $P(\lambda)$ alone.

(iii) Replacing $h_{x,x}$ and $\frac{\partial}{\partial\lambda}h_{x,x}$ with their respective mean value we can compute $\frac{\partial}{\partial\lambda}V(\lambda)$ by the same calculations.

To efficiently apply Theorem 2.9 we further need the derivative of the diagonal entries $\frac{\partial}{\partial\lambda}h_{x,x}$. But their computation will be postponed to the specific settings in Chapters 4–7.

3 The Reproducing Kernel Hilbert Space

This chapter combines key facts from [55, 38, 40] and serves as an interlude to get a second perception of the given Tikhonov problem (1.1). We introduce reproducing kernel Hilbert spaces and give some thoughts on approximating within these.

3.1 Basic Concepts

Motivated by the fact that we act on point evaluations the definition of a reproducing kernel Hilbert space seems very natural.

Definition 3.1. *Let $\mathscr{H}$ be a Hilbert space of complex-valued functions with domain X. We call $\mathscr{H}$* reproducing kernel Hilbert space*, if the point evaluation operator*

$$E_x \colon \mathscr{H} \to \mathbb{C},\ f \mapsto f(x)$$

is continuous for all $x \in X$.

Remark 3.2. *Applying the Riesz representation theorem to the point evaluation operator of a reproducing kernel Hilbert space we obtain the unique existence of a $\varphi_x \in \mathscr{H}$ such that*

$$f(x) = E_x(f) = (f, \varphi_x)_{\mathscr{H}}$$

with $(\cdot,\cdot)_{\mathscr{H}}$ being the inner product of $\mathscr{H}$. Because φ_x is an element of $\mathscr{H}$ itself, we can write by the same argument

$$\varphi_x(y) = E_y(\varphi_x) = (\varphi_x, \varphi_y)_{\mathscr{H}} =: K(x,y). \tag{3.1}$$

Definition 3.3. *(i) The kernel function $K(\cdot,\cdot) \colon X \times X \to \mathbb{C}$ in (3.1) is called* reproducing kernel *of the Hilbert space $\mathscr{H}$.*

(ii) A kernel function K is called positive definite *or* strictly positive definite *if for every finite choice $\mathcal{X} \subset X$ of nodes and $c_x \in \mathbb{C}, x \in \mathcal{X}$*

$$\sum_{x\in\mathcal{X}}\sum_{y\in\mathcal{X}} c_x\overline{c_y}K(x,y) \geq 0 \quad \text{or} \quad \sum_{x\in\mathcal{X}}\sum_{y\in\mathcal{X}} c_x\overline{c_y}K(x,y) > 0$$

holds, respectively.

Remark 3.4. *The following properties characterize a reproducing kernel and validate its naming*

(i) $K(x,y) = \overline{K(y,x)}$ for all $x, y \in X$,

(ii) $K(\cdot,x) \in \mathscr{H}$ for all $x \in X$ and

(iii) (reproducing property) $(f, K(\cdot, x))_{\mathscr{H}} = f(x)$ *for all* $x \in X$ *and* $f \in \mathscr{H}$.

Remark 3.5 (Uniqueness of the reproducing kernel)**.** *Assume* $\mathscr{H}$ *has two reproducing kernels* K_1 *and* K_2*. Then*

$$(f, K_1(\cdot, x) - K_2(\cdot, x))_{\mathscr{H}} = f(x) - f(x) = 0$$

for all $f \in \mathscr{H}$ *and* $x \in X$*. In particular this holds for* $f = K_1(\cdot, x) - K_2(\cdot, x)$ *such that*

$$\|K_1(\cdot, x) - K_2(\cdot, x)\|_{\mathscr{H}} = 0$$

for all $x \in X$*. That is,* $K_1 = K_2$.

So every reproducing kernel Hilbert space has its own unique reproducing kernel. We are also interested in the reverse statement which is covered with the following theorem.

Theorem 3.6 (Moore-Aronszajn)**.** *Let* X *be a set and* $K \colon X \times X \to \mathbb{C}$ *a positive definite kernel function, then there exists an unique reproducing kernel Hilbert space of functions on* X *such that* K *is its reproducing kernel.*

Proof. **1.** Let $\mathscr{H}_0 \coloneqq \operatorname{span}\{K(\cdot, x) \colon x \in X\}$, $\mathcal{X}, \mathcal{Y} \subset X$ two finite subsets, and

$$(f, g)_{\mathscr{H}_0} = \left(\sum_{x \in \mathcal{X}} c_x K(\cdot, x), \sum_{y \in \mathcal{Y}} d_y K(\cdot, y) \right)_{\mathscr{H}_0} \coloneqq \sum_{x \in \mathcal{X}} \sum_{y \in \mathcal{Y}} c_x \overline{d_y} K(x, y). \tag{3.2}$$

At first we need to show that (3.2) is indeed a valid inner product on $\mathscr{H}_0$. For the well-definedness of $(\cdot, \cdot)_{\mathscr{H}_0}$ on $\mathscr{H}_0$ we have to show that if $f = \sum_{n=1}^{N} \alpha_n K(\cdot, x_n)$ is the zero function, then $(f, g)_{\mathscr{H}_0} = 0$ for all $g \in \mathscr{H}_0$. Since $\mathscr{H}_0$ is spanned by $K(\cdot, y)$ for $y \in X$ it is sufficient to show $(f, K(\cdot, y))_{\mathscr{H}_0} = 0$ for all $y \in X$ which holds by

$$(f, K(\cdot, y))_{\mathscr{H}_0} = \sum_{x \in \mathcal{X}} \alpha_n K(x, y) = f(y) = 0.$$

The symmetry and the bilinearity follow immidiately from the definition and as K is a positive definite kernel function we have that $(f, f)_{\mathscr{H}_0} = \sum_{n=1}^{N} \sum_{m=1}^{N} \alpha_n \alpha_m K(\boldsymbol{x}_n, \boldsymbol{x}_m) \geq 0$. Equality occurs if and only if $f \equiv 0$, since
'$\Rightarrow$':

$$0 = (f, f)_{\mathscr{H}_0} (K(\cdot, x), K(\cdot, x))_{\mathscr{H}_0} \geq (f, K(\cdot, x))_{\mathscr{H}_0}^2 = f(x)^2,$$

where the inequality follows from the Cauchy-Schwarz inequality and
'$\Leftarrow$':

$$(f, f)_{\mathscr{H}_0} = (0 \cdot f, f)_{\mathscr{H}_0} = 0 \cdot (f, f)_{\mathscr{H}_0} = 0.$$

Therefore $(\cdot, \cdot)_{\mathscr{H}_0}$ is a valid inner product.

2. By taking equivalence classes of Cauchy sequences we complete $\mathscr{H}_0$ to get $\mathscr{H}$. We must show that every element of $\mathscr{H}$ is a function on X. For this let $\{f_n\}_{n \in \mathbb{N}} \subset \mathscr{H}_0$ be

the Cauchy sequence that converges to $f \in \mathscr{H}$. By the Cauchy-Schwarz inequality we achieve

$$|f_n(x) - f_m(x)| = |(f_n - f_m, K(\cdot, x))_{\mathscr{H}_0}| \leq \|f_n - f_m\|_{\mathscr{H}_0} \sqrt{K(x,x)}.$$

Hence the sequence is pointwise Cauchy. We may define $f(x) = \lim_{n\to\infty} f_n(x)$ and obtain

$$(f, K(\cdot, x))_{\mathscr{H}} = \lim_{n\to\infty} (f_n, K(\cdot, x))_{\mathscr{H}_0} = \lim_{n\to\infty} f_n(x) = f(x).$$

Thus, $\mathscr{H}$ is a reproducing kernel Hilbert space on X and K its reproducing kernel.

3. Since $\mathscr{H}_0$ is dense in $\mathscr{H}$, $\mathscr{H}$ is the unique reproducing kernel Hilbert space containing $\mathscr{H}_0$. And considering $K(\cdot, x) \in \mathscr{H}$ for all $x \in X$, it is clear that any reproducing kernel Hilbert space with reproducing kernel K must contain $\mathscr{H}_0$. Therefore we have shown uniqueness. ■

Remark 3.7. *The proof of Theorem 3.6 shows on the one hand the existence of a corresponding reproducing kernel Hilbert space and on the other hand gives a construction scheme for the reproducing kernel Hilbert space which is called* native space.

3.2 Smoothing Approximation

In this section we give a brief introduction into the specialties of the approximation within reproducing kernel Hilbert spaces. Given the problem of finding a function $f \in \mathscr{H}$ to approximate given, potentionally noisy function values $f(x)$ in nodes $x \in \mathcal{X}$ we could try to minimize the Tikhonov functional

$$J_\lambda(g) = \sum_{x\in\mathcal{X}} w_x |g(x) - f(x)|^2 + \lambda \|g\|_{\mathcal{H}}^2 \tag{3.3}$$

for $g \in \mathscr{H}$ and $\boldsymbol{W} = \operatorname{diag}(w_x)_{x\in\mathcal{X}}$, $w_x > 0$ for all $x \in \mathcal{X}$. The first term is called *discrepancy*, which measures how well the values are fitted, and the second term is a *smoothing term.*

Theorem 3.8. *The minimum g of the Tikhonov functional in* (3.3) *satisfies*

$$g = \sum_{x\in\mathcal{X}} c_x K(\cdot, x) \tag{3.4}$$

for suitable $c_x \in \mathbb{C}$, $x \in \mathcal{X}$.

Proof. At first we define the subspace $\mathscr{H}_K := \operatorname{span}\{K(\cdot, x)\}_{x\in\mathcal{X}}$. Each $f \in \mathscr{H}$ can be decomposed into a component along and one perpendicular to $\mathscr{H}_K$: $g = g_K + g_{K^\perp}$. With this decomposition and the use of the reproducing kernel we can reformulate the Tikhonov functional (3.3) as follows

$$\begin{aligned} J_\lambda(g) &= \sum_{x\in\mathcal{X}} |(g_K, K(\cdot, x))_{\mathscr{H}} + \underbrace{(g_{K^\perp}, K(\cdot, x))_{\mathscr{H}}}_{=0} - f(x)|^2 + \lambda \|g_K + g_{K^\perp}\|_{\mathscr{H}}^2 \\ &\geq \sum_{x\in\mathcal{X}} |g_K(x) - f(x)|^2 + \lambda \|g_K\|_{\mathscr{H}}^2 = J_\lambda(g_K). \end{aligned}$$

Thus the minimum μ_λ must belong to the subspace $\mathscr{H}_K$. ■

Using Theorem 3.8 we see that g depends only on $\boldsymbol{c} = (c_x)_{x\in\mathcal{X}}$. Thus we can reformulate (3.3) as follows

$$J_\lambda(\boldsymbol{c}) = \|\boldsymbol{K}\boldsymbol{c} - \boldsymbol{f}\|_{\boldsymbol{W}}^2 + \lambda\|\boldsymbol{c}\|_{\boldsymbol{K}}^2$$

with $\|\boldsymbol{c}\|_{\boldsymbol{K}}^2 = \boldsymbol{c}^{\mathsf{H}}\boldsymbol{K}\boldsymbol{c}$, $\|\boldsymbol{f}\|_{\boldsymbol{W}}^2 = \boldsymbol{f}^{\mathsf{H}}\boldsymbol{W}\boldsymbol{f}$, and $\boldsymbol{K} = (K(x,y))_{x,y\in\mathcal{X}}$.

Lemma 3.9. *Let K be a strictly positive definite kernel. Then the unique Tikhonov minimizer of (3.3) is given by*

$$\boldsymbol{c} = (\boldsymbol{W}\boldsymbol{K} + \lambda\boldsymbol{I})^{-1}\boldsymbol{W}\boldsymbol{f}.$$

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

$$\nabla J_\lambda(\boldsymbol{c}) = 2\boldsymbol{K}\boldsymbol{W}\boldsymbol{K}\boldsymbol{c} - 2\boldsymbol{K}\boldsymbol{W}\boldsymbol{f} + 2\lambda\boldsymbol{K}\boldsymbol{c} \overset{!}{=} 0$$

Because K is stricly positive definite, and therefore $\boldsymbol{K}$ is positive definite, we obtain

$$(\boldsymbol{W}\boldsymbol{K} + \lambda\boldsymbol{I})\boldsymbol{c} \overset{!}{=} \boldsymbol{W}\boldsymbol{f}.$$

Since additionally $\boldsymbol{W}$ and λ are strictly postive $\boldsymbol{W}\boldsymbol{K} + \lambda\boldsymbol{I}$ is positive definite. In particular it is invertible such that we achieve the proposed formula and using the positive definitness the stated unique minimization property holds. ∎

3.3 Connection to the Fourier setting

Let $\mathscr{H}$ be a separable Hilbert space of functions over X, then we can consider an orthonormal set of basis functions $\varphi_{\boldsymbol{n}}\colon X \to \mathbb{C}$, $\boldsymbol{n} \in \mathcal{I}$ for a countable index set $\mathcal{I}$. Using this basis we can express every $f \in \mathscr{H}$ like

$$f = \sum_{\boldsymbol{n}\in\mathcal{I}} \hat{f}_{\boldsymbol{n}}\varphi_{\boldsymbol{n}}$$

for $\hat{f}_{\boldsymbol{n}} = (f, \varphi_{\boldsymbol{n}})_{\mathscr{H}}$, $\boldsymbol{n} \in \mathcal{I}$. The above sum converges in the Hilbert space itself but pointwise convergence is not given a priori. This is where the reproducing kernel Hilbert spaces come into play with the property of a continuous point evaluation. The following theorem gives an equivalent characterization of reproducing kernel Hilbert spaces in terms of the basis functions.

Theorem 3.10. *Let $\mathscr{H}$ be a separable Hilbert space of functions over $\mathcal{X}$ with an orthonormal basis $\{\varphi_{\boldsymbol{n}}\}_{\boldsymbol{n}\in\mathcal{I}}$.*

(i) Then $\mathscr{H}$ is a reproducing kernel Hilbert space if and only if

$$\sum_{\boldsymbol{n}\in\mathcal{I}} \varphi_{\boldsymbol{n}}(x)\overline{\varphi_{\boldsymbol{n}}(y)}$$

converges for all $x, y \in \mathcal{X}$.

(ii) If one of the two equivalent conditions in (i) holds the corresponding kernel has the form

$$K(x,y) = \sum_{\boldsymbol{n}\in\mathcal{I}} \varphi_{\boldsymbol{n}}(x)\overline{\varphi_{\boldsymbol{n}}(y)}. \tag{3.5}$$

Proof. In order to prove (i) we first assume that $\mathscr{H}$ is a reproducing kernel Hilbert space with reproducing kernel K. Since $K(\cdot,x) \in \mathscr{H}$ we can apply the reproducing property to its Fourier expansion

$$K(\cdot,x) = \sum_{\boldsymbol{n}\in\mathcal{I}} (K(\cdot,x),\varphi_{\boldsymbol{n}})_{\mathcal{H}}\varphi_{\boldsymbol{n}} = \sum_{\boldsymbol{n}\in\mathcal{I}} \overline{(\varphi_{\boldsymbol{n}},K(\cdot,x))_{\mathcal{H}}}\varphi_{\boldsymbol{n}} = \sum_{\boldsymbol{n}\in\mathcal{I}} \overline{\varphi_{\boldsymbol{n}}(x)}\varphi_{\boldsymbol{n}}$$

Due to the continuity of the point evaluation operator in a reproducing kernel Hilbert space, the left-hand side is finite for all arguments $y \in \mathcal{X}$. Therefore the right-hand sum converges for all $x, y \in \mathcal{X}$. Note that this also proves (ii).

To see that $\mathscr{H}$ is a reproducing kernel Hilbert space from the pointwise convergence of $\sum_{\boldsymbol{n}\in\mathcal{I}} \varphi_{\boldsymbol{n}}(x)\overline{\varphi_{\boldsymbol{n}}(y)}$ we have

$$\sum_{\boldsymbol{n}\in\mathcal{I}} |\varphi_{\boldsymbol{n}}(x)|^2 = \sum_{\boldsymbol{n}\in\mathcal{I}} \varphi_{\boldsymbol{n}}(x)\overline{\varphi_{\boldsymbol{n}}(x)} < \infty.$$

In combination with

$$|E_x(f)|^2 = |f(x)|^2 \leq \left(\sum_{\boldsymbol{n}\in\mathcal{I}} \left|\hat{f}_{\boldsymbol{n}}\right|^2\right)\left(\sum_{\boldsymbol{n}\in\mathcal{I}} |\varphi_{\boldsymbol{n}}(x)|^2\right) = \left(\sum_{\boldsymbol{n}\in\mathcal{I}} |\varphi_{\boldsymbol{n}}(x)|^2\right)\|f\|_{\mathscr{H}}^2$$

this shows the continuity of the evaluation operator. Thus $\mathscr{H}$ is a reproducing kernel Hilbert space by Definition 3.1.

■

To obtain a connection between the coefficients c_x in (3.4) and the Fourier coefficients $\hat{f}_{\boldsymbol{n}}$ we state the followong lemma.

Lemma 3.11. *For a finite set of nodes $\mathcal{X}$ and $f = \sum_{x\in\mathcal{X}} c_x K(\cdot,x) \in \mathscr{H}$ we additionally have the representation $f = \sum_{\boldsymbol{n}\in\mathcal{I}} \hat{f}_{\boldsymbol{n}}\varphi_{\boldsymbol{n}}$. One can switch between these two representaitons by the following relation between c_x and $\hat{f}_{\boldsymbol{n}}$*

$$\hat{f}_{\boldsymbol{n}} = \sum_{y\in\mathcal{X}} c_y\overline{\varphi_{\boldsymbol{n}}(y)}.$$

Proof. Using the kernel representation in Fourier space (3.5) we achieve

$$f(x) = \sum_{y\in\mathcal{X}} c_y K(x,y) = \sum_{y\in\mathcal{X}} c_y \sum_{\boldsymbol{n}\in\mathcal{I}} \varphi_{\boldsymbol{n}}(x)\overline{\varphi_{\boldsymbol{n}}(y)} = \sum_{\boldsymbol{n}\in\mathcal{I}} \underbrace{\left(\sum_{y\in\mathcal{X}} c_y\overline{\varphi_{\boldsymbol{n}}(y)}\right)}_{=\hat{f}_{\boldsymbol{n}}} \varphi_{\boldsymbol{n}}(x).$$

■

Remark 3.12. *Let the index set $\mathcal{I}$ be finite and K a positive definite kernel. Then we can set up $\boldsymbol{F} = (\varphi_{\boldsymbol{n}}(x))_{x\in\mathcal{X},\boldsymbol{n}\in\mathcal{I}}$. Function evaluations can now be expressed in terms of matrices from the kernel setting as well as for the Fourier setting, i.e., $\boldsymbol{K}\boldsymbol{c} = \boldsymbol{F}\hat{\boldsymbol{f}}$. Using that $\|f\|_{\mathscr{H}} = \|\hat{\boldsymbol{f}}\|_2$ we may write the Tikhonov functional* (3.3) *in the Fourier setting*

$$J\left(\hat{\boldsymbol{f}}\right) = \left\|\boldsymbol{F}\hat{\boldsymbol{f}} - \boldsymbol{f}\right\|_{\boldsymbol{W}}^2 + \lambda \left\|\hat{\boldsymbol{f}}\right\|_2^2$$

Remark 3.13. *By introducing a weight matrix $\hat{\boldsymbol{W}}$ in frequency domain and substituting*

$$\boldsymbol{F} \to \boldsymbol{F}\hat{\boldsymbol{W}}^{-1/2} \quad \textit{and} \quad \hat{\boldsymbol{f}} \to \hat{\boldsymbol{W}}^{1/2}\hat{\boldsymbol{f}}$$

we extract weights from the Fourier matrix $\boldsymbol{F}$ into a weight matrix $\hat{\boldsymbol{W}}$ and end up with the same functional as in Chapter 2.

Therefore we have an additional interpretation of the weights $\hat{\boldsymbol{W}}$ in frequency domain, namely the form of the kernel we are approximating with in the given nodes.

4 Cross-validation on the torus

In this chapter, 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 functions $\{\mathrm{e}^{2\pi\mathrm{i}\boldsymbol{n}\cdot\boldsymbol{x}}\}_{\boldsymbol{n}\in\mathbb{Z}^d}$ in $L_2(\mathbb{T}^d)$. With this setting the Fourier matrix $\boldsymbol{F}$ becomes

$$\boldsymbol{F} = \boldsymbol{F}_{\mathcal{X},\mathcal{I}} = \left(\mathrm{e}^{2\pi\mathrm{i}\boldsymbol{n}\cdot\boldsymbol{x}}\right)_{\boldsymbol{x}\in\mathcal{X},\boldsymbol{n}\in\mathcal{I}} \tag{4.1}$$

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 [48] and [14]. In fact, our approach in this section can be seen as a generalization of [48] 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 2.4 efficiently. The idea is to choose $\boldsymbol{W}$ such that $\boldsymbol{F}^{\mathsf{H}}\boldsymbol{W}\boldsymbol{F}$ attains 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.

4.1 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 4.1. *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*

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

For the tours we obtain the following

Theorem 4.2. *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 index set $\mathcal{D}(\mathcal{I}) \coloneqq \{\boldsymbol{n}_1 - \boldsymbol{n}_2 : \boldsymbol{n}_1, \boldsymbol{n}_2 \in \mathcal{I}\}$. Then*

(i) the inverse of $\boldsymbol{A}$, given in (2.4)*, where $\boldsymbol{F}$ is the Fourier matrix* (4.1) *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*

$$h_{\boldsymbol{x},\boldsymbol{x}} = w_{\boldsymbol{x}} \sum_{\boldsymbol{n}\in\mathcal{I}} \frac{1}{1+\lambda\hat{w}_{\boldsymbol{n}}}. \tag{4.2}$$

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

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

■

Corollary 4.3. *Given the prerequisites of Theorem 4.2*

(i) we can compute $P(\lambda)$ *and* $V(\lambda)$ *by Algorithm 1 in the same complexity as multiplying a vector with* $\boldsymbol{F}$ *and* $\boldsymbol{F}^{\mathsf{H}}$*,*

(ii) additionally, we can compute $\frac{\partial}{\partial\lambda}P(\lambda)$ *and* $\frac{\partial}{\partial\lambda}V(\lambda)$ *with the cost of one further multiplication with* $\boldsymbol{F}$

for a fixed regularization parameter λ*.*

Proof. (i) follows immediately from Theorem 4.2 (ii). With the same argument the derivative of the diagonal elements evaluates to

$$\frac{\partial}{\partial\lambda} h_{x,x} = -w_x \sum_{\boldsymbol{n}\in\mathcal{I}} \frac{\hat{w}_{\boldsymbol{n}}}{(1+\lambda\hat{w}_{\boldsymbol{n}})^2}.$$

Using $\boldsymbol{F}^{\mathsf{H}}\boldsymbol{W}\boldsymbol{F} = \boldsymbol{I}$ we obtain

$$\frac{\partial}{\partial\lambda}\boldsymbol{H}\boldsymbol{f} = -\boldsymbol{F}\operatorname{diag}\left(\frac{1}{1+\lambda\hat{w}_{\boldsymbol{n}}}\right)\hat{\boldsymbol{W}}\hat{\boldsymbol{g}},$$

where $\hat{\boldsymbol{g}}$ is a byproduct of Algorithm 1. Finally, we can use Theorem 2.9 and Remark 2.10 to compute $\frac{\partial}{\partial\lambda}P(\lambda)$ and $\frac{\partial}{\partial\lambda}V(\lambda)$. ■

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

4.2 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 $\mathcal{I}_{2N} \coloneqq \mathbb{Z}^d \cap \prod_{t=1}^{d} [-N, N) = \mathcal{D}(\mathcal{I}_N)$. Thus we can apply Theorem 4.2 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 [48] 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.

Another related approach is covered in [14] where the penalty term consists of finite differences of second order in time domain. In contrast to computing the diagonal elements of the hat matrix $\boldsymbol{H}$ they used an eigenvalue decomposition to compute the trace of the hat matrix for the generalized cross-validation score.

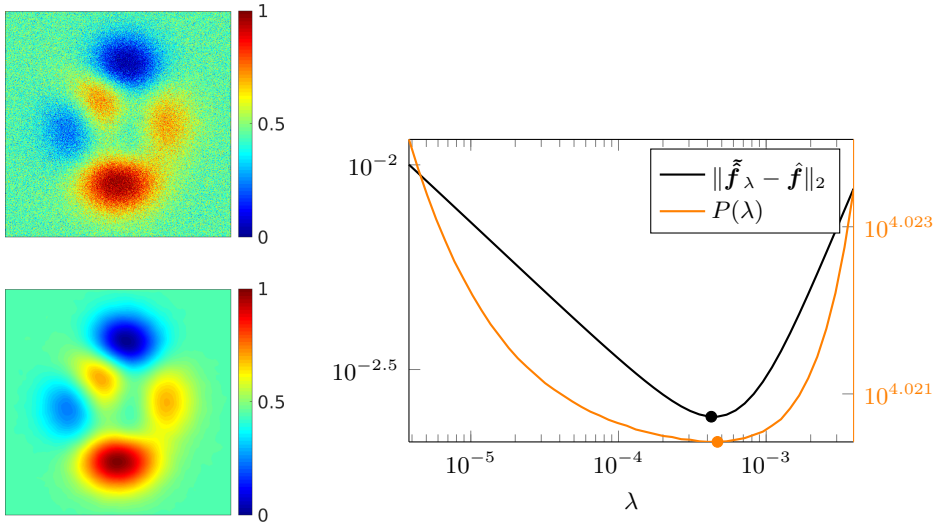
In order to illustrate our approach we chose as the test function f the peaks function from MATLAB, which is a sum of translated and scaled Gaussian bells. We evaluated this function on a grid of 1024×1024 equispaced nodes $\mathcal{X}$ and corrupted the data by 10% Gaussian noise $\varepsilon_{\boldsymbol{x}}$,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

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

for all $\boldsymbol{x} \in \mathcal{X}$ as depicted in Figure 4.1 (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 $L_2(\mathbb{T}^d)$. Varying the regularization parameter $\lambda \in [2^{-18}, 2^{-8}]$ we computed the Tikhonov minimizers $\hat{\boldsymbol{g}}_\lambda$ according to (2.1) and the cross-validation score. We then applied Parseval to the original $\hat{\boldsymbol{f}}$ and $\hat{\boldsymbol{g}}_\lambda$ 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 4.1 (b). Note that according to (4.2) 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 $\boldsymbol{g}_\lambda$ with respect to the minimizer of the cross-validation score $P(\lambda)$ is depicted in Figure 4.1 (a).

In Figure 4.1 (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).



(a) noisy input data and reconstruction

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

Figure 4.1: Approximation from two-dimensional equispaced data: Comparison of the ordinary cross-validation score $P(\lambda)$ and the approximation error.

4.3 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 [49, 9] 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

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

where $M\mathbf{1} = (M, \dots, M)^\mathsf{T} \in \mathbb{Z}^d$. 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 $\boldsymbol{W} = \operatorname{diag}(1/M)_{\boldsymbol{x}\in\mathcal{X}}$, cf. [26, 24, 42]. The advantage of rank-1 lattices is the variable index set $\mathcal{I}$ instead of the tensor-product approach like in Section 4.2. 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., $d = 7$,

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

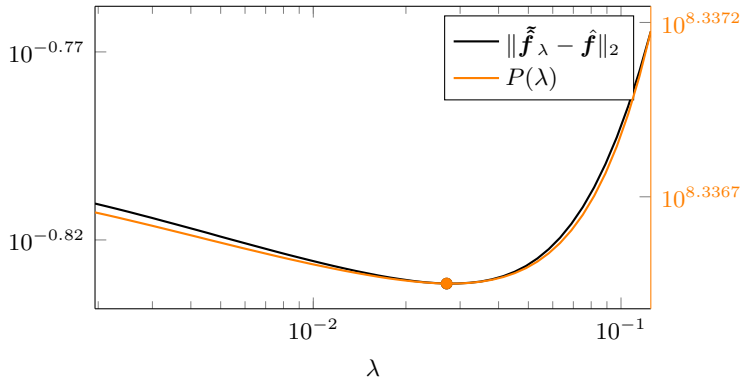


Figure 4.2: Approximation in $\mathbb{T}^7$ from data at a rank-1 lattice: Comparison of the cross-validation score $P(\lambda)$ with the approximation error.

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

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

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 (4.2) 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 4.2. 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.

4.4 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 $\boldsymbol{W} = \operatorname{diag}(w_{\boldsymbol{x}})_{\boldsymbol{x}\in\mathcal{X}}$ as a solution of the linear system

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

where $\boldsymbol{e_0}$ is the vector which only contains zeros, except in the $\mathbf{0}$-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

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

of the nodes $\mathcal{X}$, cf. [11]. 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 4.4. *Let $\mathcal{M}$ be a Riemannian manifold with a distance function* $\operatorname{dist}(\cdot,\cdot)$*. 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*

$$V_{\boldsymbol{x}} \coloneqq \left\{\boldsymbol{y}\in\mathcal{M} : \operatorname{dist}(\boldsymbol{x},\boldsymbol{y}) \le \operatorname{dist}(\boldsymbol{x'},\boldsymbol{y}),\ \forall \boldsymbol{x'}\in\mathcal{X}\right\}.$$

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

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

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.

Theorem 4.5. *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*

$$\delta_{\mathcal{X}} \coloneqq \max_{\boldsymbol{y}\in\mathcal{M}} \min_{\boldsymbol{x}\in\mathcal{X}} \operatorname{dist}(\boldsymbol{y},\boldsymbol{x})$$

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

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

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

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

■

Remark 4.6. *(i) Theorem 4.5 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 [20] 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

$$\begin{aligned}\boldsymbol{F}^{\mathsf{H}}\boldsymbol{W}\boldsymbol{F} &= \left(\sum_{\boldsymbol{x}\in\mathcal{X}} w_{\boldsymbol{x}} \mathrm{e}^{2\pi\mathrm{i}\boldsymbol{n}_1\boldsymbol{x}}\overline{\mathrm{e}^{2\pi\mathrm{i}\boldsymbol{n}_2\boldsymbol{x}}}\right)_{\boldsymbol{n}_1,]_2\in\mathcal{I}} \\ &\approx \left(\int_{\mathbb{T}^d} \mathrm{e}^{2\pi\mathrm{i}\boldsymbol{n}_1\boldsymbol{x}}\mathrm{e}^{-2\pi\mathrm{i}\boldsymbol{n}_2\boldsymbol{x}}\,\mathrm{d}\boldsymbol{x}\right)_{\boldsymbol{n}_1,\boldsymbol{n}_2\in\mathcal{I}} = \boldsymbol{I} \in \mathbb{C}^{\mathcal{I}\times\mathcal{I}}\end{aligned}$$

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

$$\boldsymbol{H} = \boldsymbol{F}\left(\boldsymbol{F}^{\mathsf{H}}\boldsymbol{W}\boldsymbol{F} + \lambda\hat{\boldsymbol{W}}\right)^{-1}\boldsymbol{F}^{\mathsf{H}}\boldsymbol{W} \approx \boldsymbol{F}\left(\boldsymbol{I} + \lambda\hat{\boldsymbol{W}}\right)^{-1}\boldsymbol{F}^{\mathsf{H}}\boldsymbol{W} =: \tilde{\boldsymbol{H}}. \tag{4.3}$$

Analogously to Theorem 4.2 we obtain for the diagonal entries $\tilde{h}_{\boldsymbol{x},\boldsymbol{x}}$ of the *approximated hat matrix* $\tilde{\boldsymbol{H}}$,

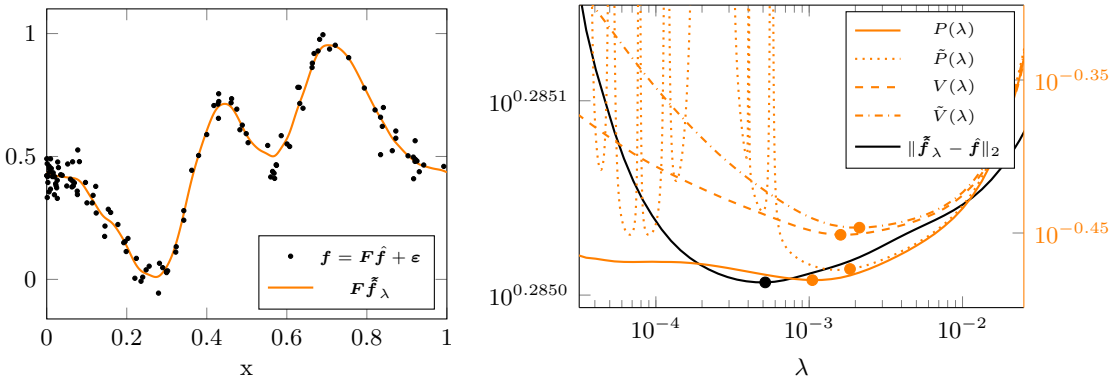
$$\tilde{h}_{\boldsymbol{x},\boldsymbol{x}} = w_{\boldsymbol{x}} \sum_{\boldsymbol{n}\in\mathcal{I}} \frac{1}{1+\lambda\hat{w}_{\boldsymbol{n}}}.$$

Together with Theorem 2.4 and Definition 2.6 this motivates the following definition of approximated cross-validation scores.

Definition 4.7. *The approximated cross-validation score $\tilde{P}(\lambda)$ and the approximated generalized cross-validation score $\tilde{V}(\lambda)$ are defined by*

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

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



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

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

Figure 4.3: 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.

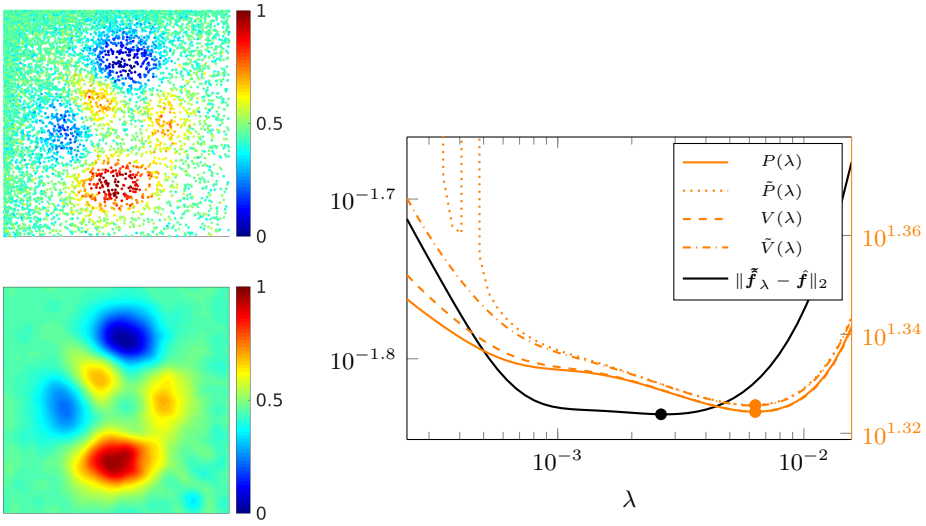
Remark 4.8. *(i) Algorithm 1 is easily modified to compute the approximated scores by replacing all occurrences of $h_{\boldsymbol{x},\boldsymbol{x}}$ by $\tilde{h}_{\boldsymbol{x},\boldsymbol{x}}$. The computationally most expensive part remains the computation of the Tikhonov minimizer $\boldsymbol{g} = \boldsymbol{H}\boldsymbol{f} = \boldsymbol{F}(\boldsymbol{F}^{\mathsf{H}}\boldsymbol{W}\boldsymbol{F} + \lambda\hat{\boldsymbol{W}})^{-1}\boldsymbol{F}^{\mathsf{H}}\boldsymbol{W}\boldsymbol{f}$. 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.*

(ii) As in Corollary 4.3 (ii), we can compute $\frac{\partial}{\partial\lambda}\tilde{P}(\lambda)$ and $\frac{\partial}{\partial\lambda}\tilde{V}(\lambda)$ using Theorem 2.9 but have to solve one additional system of equations.

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 $\mathcal{I}_{64}^{1d} = \{-32, \dots, 31\}$. 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 Section 4.2 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 4.3 (a) and Figure 4.4 (a).

As in both cases the weights $w_{\boldsymbol{x}}$ are far from uniform, we may expect a big 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 4.3. In the



(a) noisy input data and reconstruction

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

Figure 4.4: 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.

two-dimensional example both functionals coincide surprisingly well, cf. Figure 4.4.

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 summands of the ordinary cross-validation score (2.7) diverge. In contrast Lemma 2.7 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 $P(\lambda)$ and $V(\lambda)$ took approximately 1.61 seconds for the one-dimensional and 1278 seconds for the two-dimensional case, while the computation of $\tilde{P}(\lambda)$ and $\tilde{V}(\lambda)$ was performed within 0.02 and 0.16 seconds averaged over all tested λ.

5 Cross-validation on the unit interval

In this chapter, 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 $\mathcal{X}$.

5.1 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

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

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 explicitly using the following theorem.

Theorem 5.1. *Let the nodes $\mathcal{X}$ and the weights $\boldsymbol{W} = \operatorname{diag}(w_x)_{x \in \mathcal{X}}$ form a quadrature rule which is exact up to polynomial degree $2N-2$ and let $\hat{\boldsymbol{W}} = \operatorname{diag}(\hat{w}_0, \dots, \hat{w}_{N-1})$ 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 4.2 we obtain $\boldsymbol{F}^{\mathsf{H}}\boldsymbol{W}\boldsymbol{F}+\lambda\hat{\boldsymbol{W}} = \operatorname{diag}(\pi+\lambda\hat{w}_0, \pi/2+\lambda\hat{w}_1, \dots, \pi/2+\lambda\hat{w}_{N-1})$. Putting this into the formula for the diagonal elements of the hat matrix we find

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

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

5.2 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 5.1 for this case we require the discrete cosine transforms from first up to third kind

$$\begin{aligned} \boldsymbol{C}^{\mathrm{I}}_{N+1} &\coloneqq \sqrt{\frac{2}{N}} \left(\boldsymbol{\gamma}_N(n)\boldsymbol{\gamma}_N(m) \cos\left(\frac{nm\pi}{N}\right) \right)_{n,m=0}^{N}, \\ \boldsymbol{C}^{\mathrm{II}}_{N} &\coloneqq \sqrt{\frac{2}{N}} \left(\boldsymbol{\gamma}_N(n) \cos\left(\frac{n(2m+1)\pi}{2N}\right) \right)_{n,m=0}^{N-1}, \quad \boldsymbol{C}^{\mathrm{III}}_{N} \coloneqq \left(\boldsymbol{C}^{\mathrm{II}}_{N}\right)^{\mathsf{T}} \end{aligned}$$

with $\boldsymbol{\gamma}(0) = \boldsymbol{\gamma}(N) \coloneqq \sqrt{2}/2$ and $\boldsymbol{\gamma}(n) \coloneqq 1$ for $n = 1, \dots, N-1$. The corresponding matrix-vector products can be calculated using $\mathcal{O}(N \log N)$ arithmetic operations, cf. [42, Chapter 6].

Using the fact that $\boldsymbol{C}^{\mathrm{III}}_N$ is orthonormal, cf. [42, Sec. 3.5], we acquire

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

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

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

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

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

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 5.1 simplifies to:

Theorem 5.2. *Let $\mathcal{X} = \{x_1, \dots, x_m\}$ 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*

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

with

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

Proof. According to Theorem 5.1 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 5.3. *Given Chebyshev nodes of first kind*

(i) The ordinary cross-validation score $P(\lambda)$ and the generalized cross-validation score $V(\lambda)$ can be computed in $\mathcal{O}(N \log N)$ using Algorithm 1 for fixed λ.

(ii) The derivative of the diagonal elements evaluates to

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

for $m = 0, \dots, N-1$ with

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

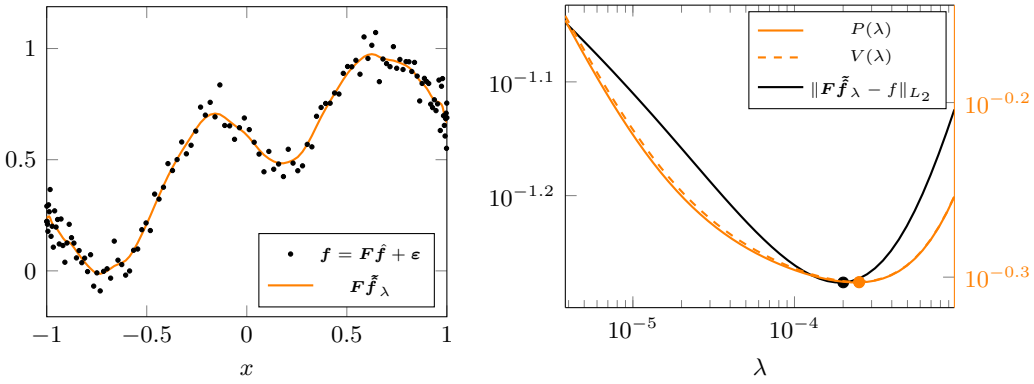
(iii) Additionally, we can compute $\frac{\partial}{\partial\lambda} P(\lambda)$ and $\frac{\partial}{\partial\lambda} V(\lambda)$ with the cost of two further discrete cosine transforms.

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 $\mathcal{O}(N \log N)$. Theorem 5.2 allows us to compute the diagonal entries of the hat matrix in $\mathcal{O}(N \log N)$ which proves (i). (ii) can be derived by standard calculus.

To obtain (iii) we use Theorem 2.9 and the given quadrature to derive

$$\frac{\partial}{\partial\lambda} \boldsymbol{H}\boldsymbol{f} = -\boldsymbol{F} \operatorname{diag}\left(\frac{1}{\pi + \lambda\hat{w}_0}, \frac{1}{\pi/2 + \lambda\hat{w}_1}, \dots, \frac{1}{\pi/2 + \lambda\hat{w}_{N-1}} \right) \hat{\boldsymbol{W}}\hat{\boldsymbol{g}}.$$

Because $\hat{\boldsymbol{g}}$ is a byproduct from Algorithm 1 we can calculate $\frac{\partial}{\partial\lambda}\boldsymbol{H}\boldsymbol{f}$ using one further cosine transform. Using this and (ii) in Remark 2.10 we obtain the assertion. ∎



(a) noisy input data $\boldsymbol{f} = \boldsymbol{F}\hat{\boldsymbol{f}} + \boldsymbol{\varepsilon}$ and reconstruction $\tilde{\boldsymbol{f}}_\lambda$ 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 5.1: 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.

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 5.1 (a). To choose $\hat{\boldsymbol{W}}$ we made use of the following statement from [51, 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 regularization for the λ with the smallest corresponding ordinary cross-validation score as one can see in Figure 5.1 (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.

5.3 Approximative Quadrature

In this section, we consider arbitrary, ordered nodes $x_m \in \mathcal{X} \subset [-1,1]$, $m = 0,\dots,M$. The corresponding cosine transforms can be computed using the nonequispaced discrete cosine transform, cf. [10], in $\mathcal{O}(N \log N + |\mathcal{X}|)$ where N is the bandwidth. As in Section 4.4 we determine approximate quadrature weights w_{x_m} for $m = 0,\dots,M$ 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.,

$$w_{x_m} = \begin{cases} \frac{y_0+y_1}{2}, & m = 0, \\ \frac{y_{m+1}-y_{m-1}}{2}, & m = 1, \dots, M-1, \\ \pi - \frac{y_{M-1}+y_M}{2}, & m = M. \end{cases} \tag{5.1}$$

Remark 5.4. *Let x_m be the Chebyshev nodes of first kind. Then the quadrature weights* (5.1) *coincide with the exact quadrature weights from Section 5.1.*

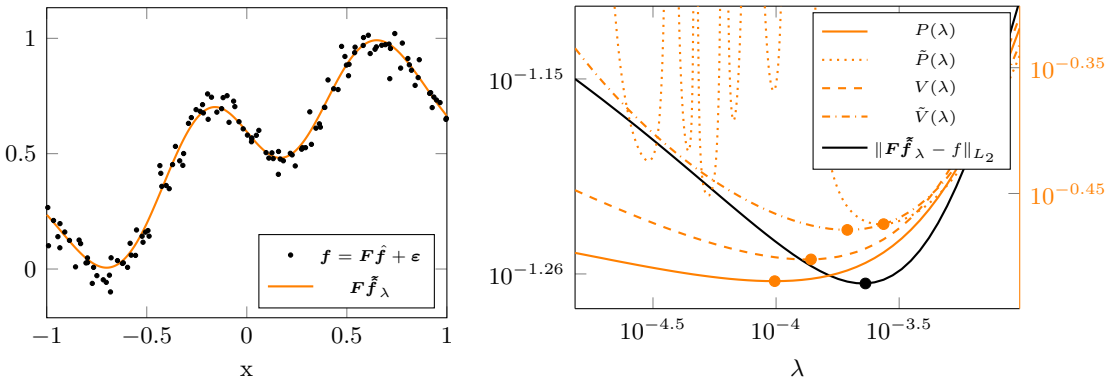
Analogously to Section 4.4 we use the approxmiated hat matrix $\tilde{\boldsymbol{H}}$ from (4.3) for ease of computation.

Remark 5.5. *(i) Using Theorem 5.1 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 4.7 in $\mathcal{O}(N \log N + |\mathcal{X}|)$ arithmetic operations given a fixed number of iterations to compute the Tikhonov minimizer.

(ii) Similar to Corollary 5.3 (ii), we can calculate $\frac{\partial}{\partial\lambda} h_{x,x}$ from Theorem 5.1 with one cosine transform. Now we can compute $\frac{\partial}{\partial\lambda}\tilde{P}(\lambda)$ and $\frac{\partial}{\partial\lambda}\tilde{V}(\lambda)$ using Theorem 2.9 and Remark 2.10 but have to solve one additional system of equations.

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 bandwidth equal to the number of nodes, i.e., $N = |\mathcal{X}|$. As the Voronoi weights resemble 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 5.2.

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.



(a) noisy input data $\boldsymbol{f} = \boldsymbol{F}\hat{\boldsymbol{f}} + \boldsymbol{\varepsilon}$ and reconstruction $\boldsymbol{F}\hat{\boldsymbol{g}}_\lambda$ 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 5.2: 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.

6 Cross-validation on the two-dimensional sphere

Approximation on the two-dimensional sphere $\mathbb{S}^2 \coloneqq \{\boldsymbol{x} \in \mathbb{R}^3 : \|\boldsymbol{x}\|_2 = 1\}$ 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 [57, 36, 33, 6], and by bounds for best approximations [47, 56, 21]. Based on these results the relationship between the mesh norm, the separation distance of the sampling points, and optimal approximation rates has been analyzed in the papers [12, 31, 28, 50]. 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 [41].

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. [13, 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 $\hat{\boldsymbol{W}} = \operatorname{diag}(\hat{w}_{n,k})_{h=0,\dots,N,\,k=-n,\dots,n}$ that depend only the polynomial degree, i.e., $\hat{w}_{n,k} = \hat{w}_n$.

6.1 Exact Quadrature

There are several approaches for 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. [42, 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. [45].

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 non-negative, 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 [19] with the resulting spherical designs being available in [16]. Finally, one can try to compute both quadrature nodes and weights in an optimization scheme, cf. [17].

For this section it is sufficient that the nodes $\mathcal{X}$ and the weights $\boldsymbol{W} = \operatorname{diag}(w_{\boldsymbol{x}})_{\boldsymbol{x}\in\mathcal{X}}$ 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 6.1. *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 $\boldsymbol{x} \in \mathcal{X}$ 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 4.2

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

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

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

■

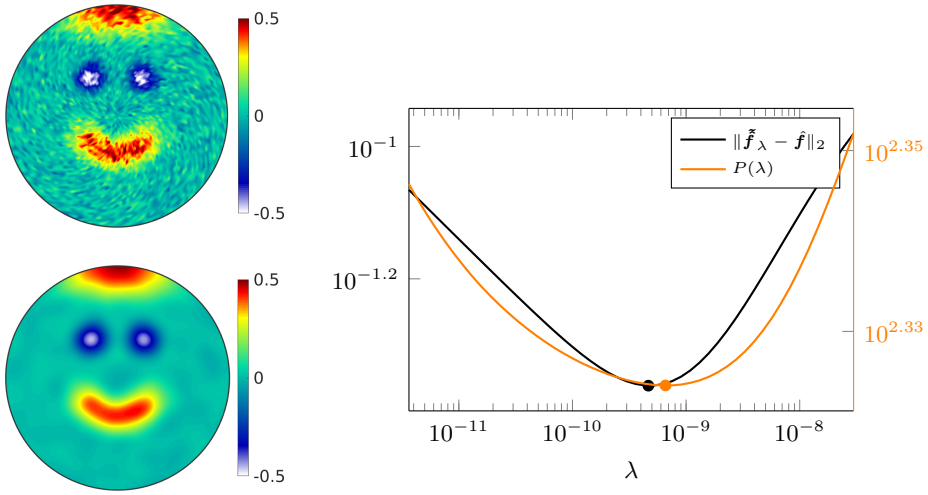
Corollary 6.2. *(i) For fixed λ the ordinary cross-validation score $P(\lambda)$ and the generalized cross-validation score $V(\lambda)$ on the two-dimensional sphere, given quadrature nodes and weights, can be computed in $\mathcal{O}(N^2\log N + |\mathcal{X}|)$ using Algorithm 1.*

(ii) From

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

in combination with Theorem 2.9 and Remark 2.10 we see that we can compute $\frac{\partial}{\partial\lambda}P(\lambda)$ and $\frac{\partial}{\partial\lambda}V(\lambda)$ with one further multiplication with $\boldsymbol{F}$

Proof. Due to Theorem 6.1 we can compute $h_{\boldsymbol{x},\boldsymbol{x}}$ in linear time. Using equation (6.1) 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 $\mathcal{O}(N^2\log N + |\mathcal{X}|)$. ■



(a) noisy input data and reconstruction

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

Figure 6.1: Approximation from equispaced data on the two-dimensional sphere: Comparison of the cross-validation score $P(\lambda)$ and the approximation error.

In order to illustrate Theorem 6.1 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 6.1 the diagonal entries $h_{\boldsymbol{x},\boldsymbol{x}}$ of the hat matrix are 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 $\hat{w}_n = (2n+1)^{2s}$ for $s = 3$ which corresponds to a function with 3 derivatives in $L_2(\mathbb{S}^2)$.

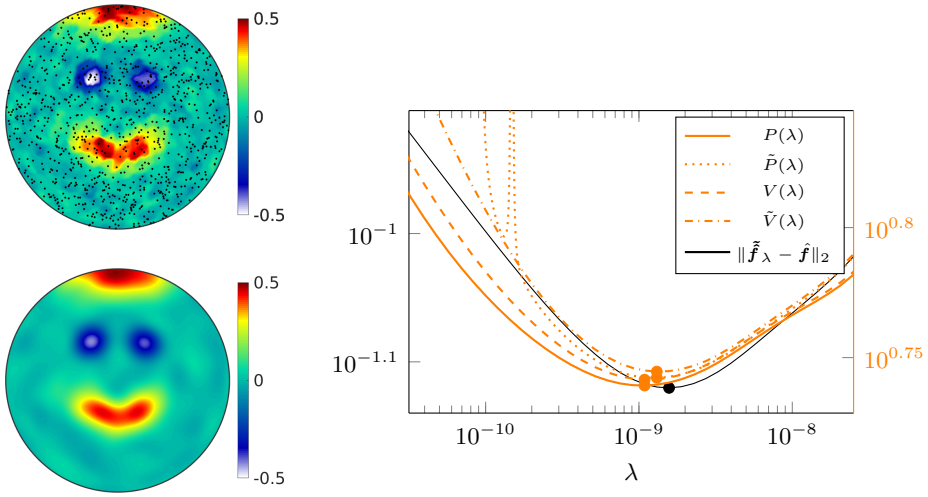
The test function consists of a sum of quadratic B-splines to which we added an error of 5% Gaussian noise for each node as one can see in Figure 6.1 (a). This function was suggested in [23]. We calculated $V(\lambda)$ and $P(\lambda)$ for $\lambda \in [2^{-38}, 2^{-25}]$ using Algorithm 1 with the help 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 $\hat{\boldsymbol{g}}$ which is a byproduct of Algorithm 1.

As it is illustrated in the Figure 6.1 (b) the minimum of the cross-validation score is very close to the minimum of the approximation error.

6.2 Approximative Quadrature

In the case function values are provided at nodes not forming a suitable quadrature rule we follow the previous ideas of Section 4.4 and 5.3 and use the approximated hat matrix $\tilde{\boldsymbol{H}}$ from (4.3) instead of $\boldsymbol{H}$ itself. This way we acquire $\tilde{P}(\lambda)$ and $\tilde{V}(\lambda)$ as in Definition 4.7. In place of quadrature weights we use a spherical Voronoi decomposition, cf. [44].

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



(a) noisy input data and reconstruction

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

Figure 6.2: Approximation from random nodes on the two-dimensional sphere: 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.

minimizer $\hat{\boldsymbol{g}}$. Again, we can compute $\frac{\partial}{\partial\lambda}\tilde{P}(\lambda)$ and $\frac{\partial}{\partial\lambda}\tilde{V}(\lambda)$ similar to Remark 4.8.

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 the previous example 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 6.2 compares the different cross-validation scores $P(\lambda)$, $V(\lambda)$, $\tilde{P}(\lambda)$, and $\tilde{V}(\lambda)$ for $\lambda \in [2^{-38}, 2^{-25}]$. 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 $\tilde{V}(\lambda)$.

7 Cross-validation on the rotation group

Now we look into cross-validation on the rotation group $SO(3)$, where approximating functions plays an important role with applications in crystallographic texture analysis, chemical physics, molecular biology, and robotics, cf. [39, 5, 35].

The Fourier basis functions in this setting are the *Wigner-D functions* $D_n^{kk'}$ for $n \in \mathbb{N}_0$ and $k, k' = -n, \dots, n$, which are orthogonal and normalized such that

$$\left\| D_n^{kk'} \right\|^2_{L_2(SO(3))} = \frac{8\pi^2}{2n+1},$$

cf. [52, Section 4.10, equation (5)], where $\| \cdot \|_{L_2(SO(3))}$ is the L_2-norm on the rotation group. The corresponding truncated Fourier matrix $\boldsymbol{F}$ becomes

$$\boldsymbol{F} = \left(D_n^{kk'}(\boldsymbol{x}) \right)_{\boldsymbol{x} \in \mathcal{X}; n=0,\dots,N, k,k'=-n,\dots,n}$$

for some maximal polynomial degree N. We consider weights in Fourier space which only depend on the polynomial degree n, namely $\hat{\boldsymbol{W}} = \operatorname{diag}(\hat{w}_n^{kk'})_{\boldsymbol{x} \in \mathcal{X}; n=0,\dots,N, k,k'=-n,\dots,n}$ with $\hat{w}_n^{kk'} = \hat{w}_n$.

7.1 Exact Quadrature

Following the structure of the previous chapters we deal with the case of exact quadrature first. One can derive such quadrature rules by considering a parametrization for the tensor product of $\mathbb{S}^2 \times \mathbb{S}^1$ and use the quadrature rules from these, cf. [18].

We will assume that the nodes $\mathcal{X}$ and the weights $\boldsymbol{W} = \operatorname{diag}(w_{\boldsymbol{x}})_{x \in \mathcal{X}}$ form a quadrature rule with exactness up to degree $2N$, i.e., $\boldsymbol{F}^{\mathsf{H}} \boldsymbol{W} \boldsymbol{F} = \operatorname{diag}(8\pi^2/(2n+1))_{n=0,\dots,N, k,k'=-n,\dots,n}$. 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 now be computed efficiently using the following theorem.

Theorem 7.1. *Let $\mathcal{X}$ and $\boldsymbol{W}$ form a quadrature formula $Q_{\mathcal{X},\boldsymbol{W}}$ that is exact for all Wigner-D functions up to polynomial degree $2N$. Then the diagonal entry corresponding to $\boldsymbol{x}$ of $\boldsymbol{H}$ satisfies*

$$h_{\boldsymbol{x},\boldsymbol{x}} = w_{\boldsymbol{x}} \sum_{n=0}^{N} \frac{2n+1}{\frac{8\pi^2}{2n+1} + \lambda \hat{w}_n}.$$

Proof. Like in the previous chapters we can use the exact quadrature to obtain

$$\boldsymbol{H} = \boldsymbol{F}\left(\boldsymbol{F}^{\mathsf{H}}\boldsymbol{W}\boldsymbol{F} + \lambda\hat{\boldsymbol{W}}\right)^{-1}\boldsymbol{F}^{\mathsf{H}}\boldsymbol{W} = \boldsymbol{F}\operatorname{diag}\left(\frac{1}{\frac{8\pi^2}{2\cdot 0+1} + \lambda\hat{w}_0}, \dots, \frac{1}{\frac{8\pi^2}{2N+1} + \lambda\hat{w}_N}\right)\boldsymbol{F}^{\mathsf{H}}\boldsymbol{W}.$$

Now we apply a special case of the addition theorem, cf. [52, Section 4.7, equation (4)], for Wigner-D functions to obtain

$$\sum_{k''=-n}^{n} D_n^{kk''}(\boldsymbol{x})\overline{D_n^{k'k''}(\boldsymbol{x})} = D_n^{kk'}(\boldsymbol{0}) = \delta_{k,k'}.$$

Ultimately, we can write the diagonal entries as follows

$$\begin{aligned} h_{\boldsymbol{x},\boldsymbol{x}} &= w_{\boldsymbol{x}} \sum_{n=0}^{N} \frac{1}{\frac{8\pi^2}{2n+1} + \lambda\hat{w}_n} \sum_{k=-n}^{n} \sum_{k'=-n}^{n} D_{k,k'}^{n}(\boldsymbol{x})\overline{D_{k,k'}^{n}(\boldsymbol{x})} \\ &= w_{\boldsymbol{x}} \sum_{n=0}^{N} \frac{2n+1}{\frac{8\pi^2}{2n+1} + \lambda\hat{w}_n}. \end{aligned}$$

■

Corollary 7.2. *(i) Fixing λ, we can compute the ordinary cross-validation score $P(\lambda)$ and the generalized cross-validation score $V(\lambda)$ within $\mathcal{O}(M + N^3 \log^2 N)$ numerical operations on the rotation group $SO(3)$ using Algorithm 1.*

(ii) Similar to Corollary 4.3 we can compute $\frac{\partial}{\partial\lambda}P(\lambda)$ and $\frac{\partial}{\partial\lambda}V(\lambda)$ by one further multiplication with $\boldsymbol{F}$

Proof. Using Theorem 7.1 we can compute the diagonal entries $h_{\boldsymbol{x},\boldsymbol{x}}$ and their derivatives in linear time and using the nonequispaced fast $SO(3)$ Fourier transform, cf. [43], the multiplication with $\boldsymbol{F}$ and $\boldsymbol{F}^{\mathsf{H}}$ can be done in $\mathcal{O}(M + N^3 \log^2 N)$. ■

To illustrate our approach numerically we chose $M = 5\,880$ approximately equidistributed nodes with quadrature weights $8\pi^2/M$, which form an exact quadrature rule up to polynomial degree $N = 23$. Due to Theorem 7.1 the diagonal entries of $\boldsymbol{H}$ are multiples of the spatial weights $w_{\boldsymbol{x}}$ which has the equality of ordinary and generalized cross-validation score as a consequence. We chose $\hat{w}_n = (n + 1/2)^{2s}$ for $s = 3$ which corresponds to a function with three derivatives in $L_2(SO(3))$.

The test function is an orientation density function of the data set for the mineral olivine from mtex, cf. [22]. We calculated $P(\lambda)$ and $V(\lambda)$ for $\lambda \in [2^{-18}, 2^{-15}]$ using Algorithm 1 and the $L_2(SO(3))$-error using Parseval from $\hat{\boldsymbol{f}}$ and $\hat{\boldsymbol{g}}$.

We can see in Figure 7.1 that the cross-validation score is close to the approximation error.

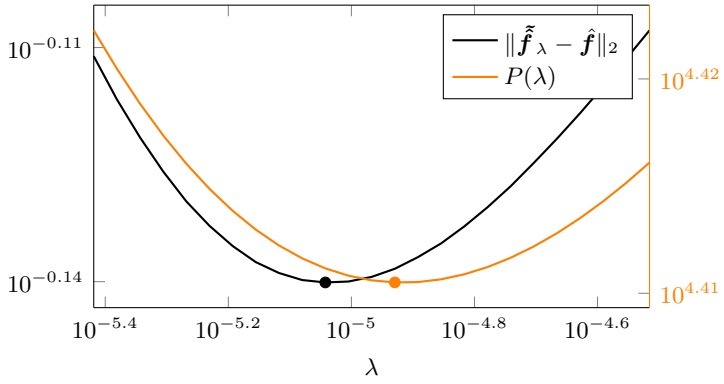


Figure 7.1: Approximation in $SO(3)$ given a quadrature rule: Comparison of the ordinary cross-validation score $P(\lambda)$ with the approximation error.

7.2 Approximative Quadrature

If no exact quadrature rule is given we can use the approximated hat matrix $\tilde{\boldsymbol{H}}$ form (4.3) instead or $\boldsymbol{H}$. Using Voronoi weights in $\boldsymbol{W}$ we obtain $\tilde{P}(\lambda)$ and $\tilde{V}(\lambda)$ as in Definition 4.7 in a fast manner.

We have to change Algorithm 1 to calculate the Voronoi weights before and calculate the Tikhonov minimizer $\hat{\boldsymbol{g}}$ by solving a linear system of equations.

Remark 7.3. *(i) Algorithm 1 can be used to compute the approximated cross-validation scores. The computationally most expensive part remains the computation of the Tikhonov minimizer* $\boldsymbol{g} = \boldsymbol{H}\boldsymbol{f} = \boldsymbol{F}(\boldsymbol{F}^{\mathsf{H}}\boldsymbol{W}\boldsymbol{F} + \lambda\hat{\boldsymbol{W}})^{-1}\boldsymbol{F}^{\mathsf{H}}\boldsymbol{W}\boldsymbol{f}$. *Using an iterative method with a fixed number of iterations and making use of the nonequispaced fast* $SO(3)$ *Fourier transform, this can be done in* $\mathcal{O}(M + N^3 \log^2 N)$.

(ii) As in Corollary 4.3 (ii), we can compute $\frac{\partial}{\partial\lambda}\tilde{P}(\lambda)$ *and* $\frac{\partial}{\partial\lambda}\tilde{V}(\lambda)$ *using Theorem 2.9 and Remark 2.10, but have to solve one additional system of equations.*

For a numerical experiment we used $|\mathcal{X}| = 46\,852$ random nodes on $SO(3)$ and polynomial degree $N = 25$ to get an oversampling factor of two. As sample function we used the same as in the quadrature case and perturbed the function values with 5 % Gaussian noise. The weights in frequency domain $\hat{\boldsymbol{W}}$ are chosen as before: $\hat{w}_n = (n + 1/2)^{2s}$ for $s = 3$.

Figure 6.2 compares the approximated cross-validation scores $\tilde{P}(\lambda)$ and $\tilde{V}(\lambda)$ for $\lambda \in [2^{-22}, 2^{-20}]$. Again, we observe a similar behavior of the different cross-validation scores to the $L_2(SO(3))$-approximation error. Surprisingly we do not get the peaks in the ordinary cross-validation score as in the other chapters.

We only calculated the approximated scores which took 9.25 seconds per λ. The observed factor equal to the number of nodes $|\mathcal{X}|$ observed in the previous chapters would lead to 120 hours per λ for the exact cross-validation scores which is why we omitted their computation.

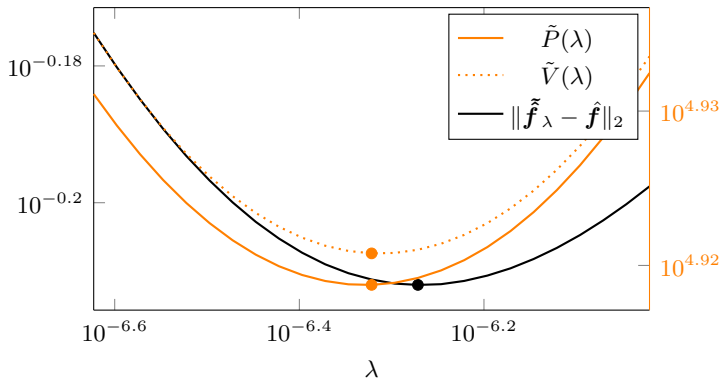


Figure 7.2: Approximation in $SO(3)$ without a given quadrature rule: Comparison of the ordinary cross-validation score $P(\lambda)$ with the approximation error.

8 Conclusion

In this thesis we presented a fast algorithm for the computation of the leave-one-out cross-validation score $P(\lambda)$ and its derivative $\frac{\partial}{\partial\lambda}P(\lambda)$ for the Tikhonov regularizer (1.1). 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 4.2, 5.2, 6.1, and 7.1 for approximation on the torus, the interval, the two-dimensional sphere, and the rotation group, respectively. Generalizations to other domains are possible following the framework presented in this thesis. 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 4.3, 5.3, 6.2, and 7.2. For arbitrary nodes we accomplished in Remarks 4.8, 5.5, Corollary 6.2, and Remark 7.3 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. Section 4.4. 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 thesis can be found on the GitHub repository `https://github.com/felixbartel/fcv`.

Bibliography

[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] F. Bartel, R. Hielscher, and D. Potts. Fast Cross-validation in Harmonic Approximation. *arXiv e-prints 1903.10206*, 2019.

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

[5] S. Bernstein and H. Schaeben. A one-dimensional Radon transform on $SO(3)$ and its application to texture goniometry. *Math. Methods Appl. Sci.*, 28(11):1269–1289, 2005.

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

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

[8] 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.

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

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

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

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

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

[14] D. Garcia. Robust smoothing of gridded data in one and higher dimensions with missing values. *Comput. Statist. Data Anal.*, 54(4):1167–1178, 2010.

[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. Sampling sets and quadrature formulae on the rotation group. *Numer. Funct. Anal. Optim.*, 30:665–688, 2009.

[19] 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.

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

[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® 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.*, 2007. accepted.

[34] 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.

[35] A. Makadia, L. Sorgi, and K. Daniilidis. Rotation estimation from spherical images. In *Proceedings of the 17th International Conference on Pattern Recognition, 2004. ICPR 2004.*, volume 3, pages 590–593 Vol.3, Aug 2004.

[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] F. J. Narcowich, R. Schaback, and J. D. Ward. Approximation in sobolev spaces by kernel expansions. *J. Approx. Theory*, 114:70–83, 2002.

[39] V. P. Palamodov. Reconstruction from a sampling of circle integrals in SO(3). *Inverse Problems*, 26(9):095008, 2010.

[40] V. I. Paulsen and M. Raghupathi. *An introduction to the theory of reproducing kernel Hilbert spaces*, volume 152 of *Cambridge Studies in Advanced Mathematics.* Cambridge University Press, Cambridge, 2016.

[41] 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.

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

[43] D. Potts, J. Prestin, and A. Vollrath. A fast algorithm for nonequispaced Fourier transforms on the rotation group. *Numer. Algorithms*, 52:355–384, 2009.

[44] 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`.

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

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

[47] 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.

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

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

[50] W. Themistoclakis and M. V. Barel. Uniform approximation on the sphere by least squares polynomials. *Numer. Alg.*, accepted, 2019.

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

[52] D. Varshalovich, A. Moskalev, and V. Khersonskii. *Quantum Theory of Angular Momentum*. World Scientific Publishing, Singapore, 1988.

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

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

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

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

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

TECHNISCHE UNIVERSITÄT CHEMNITZ

Studentenservice – Zentrales Prüfungsamt

Selbstständigkeitserklärung

Name: Vorname: geb. am: Matr.-Nr.:	<u>Bitte beachten:</u> 1. Bitte binden Sie dieses Blatt am Ende Ihrer Arbeit ein.

Selbstständigkeitserklärung*

Ich erkläre gegenüber der Technischen Universität Chemnitz, dass ich die vorliegende selbstständig und ohne Benutzung anderer als der angegebenen Quellen und Hilfsmittel angefertigt habe.

Die vorliegende Arbeit ist frei von Plagiaten. Alle Ausführungen, die wörtlich oder inhaltlich aus anderen Schriften entnommen sind, habe ich als solche kenntlich gemacht.

Diese Arbeit wurde in gleicher oder ähnlicher Form noch nicht als Prüfungsleistung eingereicht und ist auch noch nicht veröffentlicht.

Datum: ……. Unterschrift: ……

* Statement of Authorship

I hereby certify to the Technische Universität Chemnitz that this thesis is all my own work and uses no external material other than that acknowledged in the text.

This work contains no plagiarism and all sentences or passages directly quoted from other people's work or including content derived from such work have been specifically credited to the authors and sources.

This paper has neither been submitted in the same or a similar form to any other examiner nor for the award of any other degree, nor has it previously been published.