Fast global optimization on the torus, the sphere and the rotation group

Manuel Gräf Ralf Hielscher

Math Subject Classifications. 65T40, 65K10 53B21 49M15, 33C55.

Keywords and Phrases. global optimization on manifolds, iterative methods, harmonic analysis, fast Fourier methods, Kikuchy pattern, crystallographic texture analysis.

Detecting all local extrema or the global extremum of a polynomial on the torus, the sphere or the rotation group is a tough yet often requested numerical problem. We present a heuristic approach that applies common descent methods like nonlinear conjugated gradients or Newtons methods simultaneously to a large number of starting points. The corner stone of our approach are FFT like algorithms, i.e., algorithms that scale almost linearly with respect to the sum of the dimension of the polynomial space and the number of evaluation points. These FFT like algorithms allow us to compute one step of a descent method simultaneously for all staring points at almost the same cost as for one single starting point. The effectiveness of the proposed algorithms is demonstrated in various applications. In particular, we apply it to the Radon transform of a spherical function which allows us to detect lines in spherical patterns.

1 Introduction

Detecting local extrema of a function f defined on a manifold $\mathcal{M}$ is a frequent problem in mathematical applications. As an important and practical example one might think of a function defined on some subgroup of the Euclidean transformation group describing the binding energy between some molecules, proteins, etc.. Since local extrema of the binding energy are related to stable bindings one is interested in fast algorithms for their determination, cf. [6].

Clearly there won't be any algorithm that determines all local extrema or the global extremum of an arbitrary function in finite time. A common heuristic approach is as follows:

i) Chose an arbitrary set of $M \in \mathbb{N}$ initial points $\boldsymbol{x}_1, \dots, \boldsymbol{x}_M \in X$.

ii) For each initial point $\boldsymbol{x}_j$, $j = 1, \dots, M$, perform a local search for an extrema $\boldsymbol{x}_j^*$, e.g. by applying a local optimization method such as the method of nonlinear conjugated gradients or Newton's method.

iii) Determine the global extremum from the local extrema $\boldsymbol{x}_j^*$, $j = 1, \dots, M$.

Let us assume that the function f is a trigonometric polynomial of degree N on the d-dimensional torus $\mathbb{T}^d$. Then the nonequispaced fast Fourier transform (NFFT) allows for the approximate evaluation of the polynomial f at all initial points $\boldsymbol{x}_j, \ldots, \boldsymbol{x}_M$ with arithmetic complexity $\mathcal{O}(N^d \log N + M)$, cf. [23]. We shall refer to such algorithms that scales linearly modulo a log term with respect to the sum of the dimension of the polynomial space and the number of evaluation points as *FFT like algorithms*. It is not difficult to see that also the gradient and the Hessian of f at all initial nodes $\boldsymbol{x}_j, \ldots, \boldsymbol{x}_M$ can be computed by FFT like algorithms, i.e. with the same arithmetic complexity as the evaluation of the function values. In particular, for $M \sim N^d$ the arithmetic complexity of one iteration step in commonly used optimization methods for all M initial points is almost the same as for one single initial point, $\mathcal{O}(N^d \log N)$, cf. Theorem 3.1.

Generalizations of the nonequispaced fast Fourier transform are known for the two-dimensional sphere $\mathbb{S}^2$, cf. [25] and the group $\mathrm{SO}(3)$ of rotations in three-dimensional Euclidean space, cf. [29, 24]. We show that the above observation generalizes to these settings as well, i.e., that there are FFT like algorithms for the evaluation of the gradient and the Hessian of polynomials on $\mathbb{S}^2$ and $\mathrm{SO}(3)$. More precisely, we prove in Theorem 4.3 and 5.3 that for polynomials on the sphere or the rotation group the simultaneous computation of one iteration step of a local descent method for as many initial points as the dimension of the polynomial space has almost the same arithmetic complexity as for one initial point.

We illustrate the applicability of our approach with two real world examples. The first example deals with line detection in so called Kikuchi pattern on the sphere, cf. [16], which we realize by first computing a spherical polynomial $f\colon \mathbb{S}^2 \to \mathbb{C}$ approximating the given data by an appropriate quadrature rule. Second, we compute the Radon transform

$$\mathcal{R}f(\boldsymbol{\eta}) = \frac{1}{2\pi} \int_{\boldsymbol{\xi} \perp \boldsymbol{\eta}} f(\boldsymbol{\xi}) \, \mathrm{d}\boldsymbol{\xi}$$

of the spherical polynomial f by using the fact that the spherical harmonics are eigenfunctions of the operator $\mathcal{R}$. Third, we apply our algorithm for simultaneous detection of the local extrema of $\mathcal{R}f$. And, finally, we compute the lines corresponding to these extrema. Those lines in a Kikuchi pattern correspond to crystallographic lattice planes and are instrumental for the identification of crystals and there orientation within a specimen [4].

As an example for the rotation group $\mathrm{SO}(3)$ we consider functions $f\colon \mathrm{SO}(3) \to \mathbb{R}$ which model the distribution of crystal orientations in a polycrystalline material. In practice, the function f is often given as a polynomial on the rotation group as it is derived from experimental data by solving an inverse problem. Local maxima of f correspond to preferred crystal orientation and are of great practical importance, cf. [21] .

We conclude that such simultaneous optimization is applicable whenever the function of interest has a finite representation in a basis that allow for FFT like algorithms for point evaluation as well as for the computation of the gradient and the Hessian.

2 Local Optimization Methods on Riemannian Manifolds

In what follows we consider compact Riemannian manifolds $\mathcal{M} \subset \mathbb{R}^d$ without boundary and sufficiently smooth (at least continuously differentiable) functions $f : \mathcal{M} \to \mathbb{R}$. In general we are interested in the determination of local extreme points, which can computed by straightforward generalizations of the well-known optimization methods in Euclidean space

$\mathbb{R}^d$. An excellent reference for optimization methods in Euclidean space is given in [28]. Generalizations to Riemannian manifolds have been investigated originally by Udrişte [34] and Smith [31].

For simplicity we restrict our attention to the determination of a local minimizer $\boldsymbol{x}_* \in \mathcal{M}$ of f by means of what is know as descent or line search methods. However, we like to mention that other optimization methods such as trust region methods have been generalized to Riemannian manifolds as well, cf. [3].

In Euclidean space $\mathbb{R}^d$ descent methods are iterative methods, which start from an initial point $\boldsymbol{x}^{(0)} \in \mathbb{R}^d$ and generate a sequence of points $\boldsymbol{x}^{(k)} \in \mathbb{R}^d$ accordingly to the iteration

$$\boldsymbol{x}^{(k+1)} := \boldsymbol{x}^{(k)} + \alpha^{(k)} \boldsymbol{d}^{(k)} \in \mathbb{R}^d, \qquad k = 0, 1, \ldots, \tag{2.1}$$

where $\boldsymbol{d}^{(k)}$ is the search direction and $\alpha^{(k)} \geq 0$ is the step length. We note that the well-known method of steepest descent, the conjugate gradient method, and Newton's method fit into this class of descent methods.

For Riemannian manifolds $\mathcal{M}$ the iteration scheme (2.1) does not make sense, since the lines $l_{\boldsymbol{x},\boldsymbol{d}} \colon \mathbb{R} \to \mathbb{R}^d$, $l_{\boldsymbol{x},\boldsymbol{d}}(t) := \boldsymbol{x} + t\boldsymbol{d}$, do not need to run in $\mathcal{M}$. However, we recall that for any point $\boldsymbol{x} \in \mathcal{M}$ we may associate a subspace of $\mathbb{R}^d$ which is called the tangent space $\mathrm{T}_{\boldsymbol{x}}\mathcal{M} \subset \mathbb{R}^d$ and that additionally for a given tangent vector $\boldsymbol{d} \in \mathrm{T}_{\boldsymbol{x}}\mathcal{M}$ there is a unique geodesic curve $\gamma_{\boldsymbol{x},\boldsymbol{d}} \colon \mathbb{R} \to \mathcal{M}$ with $\gamma_{\boldsymbol{x},\boldsymbol{d}}(0) = \boldsymbol{x}$ and $\dot{\gamma}_{\boldsymbol{x},\boldsymbol{d}}(0) = \boldsymbol{d}$, e.g. cf. [17]. Then, after replacing the straight lines on the right hand side in (2.1) by geodesic curves $\gamma_{\boldsymbol{x}^{(k)},\boldsymbol{d}^{(k)}}$, we obtain descent methods on Riemannian manifolds $\mathcal{M}$ given by

$$\boldsymbol{x}^{(k+1)} := \gamma_{\boldsymbol{x}^{(k)},\boldsymbol{d}^{(k)}}(\alpha^{(k)}) \in \mathcal{M}, \qquad k = 0, 1, \ldots, \tag{2.2}$$

where the search direction must be a tangent vector $\boldsymbol{d}^{(k)} \in \mathrm{T}_{\boldsymbol{x}^{(k)}}\mathcal{M}$. The determination of effective search directions $\boldsymbol{d}^{(k)}$ and step lengths $\alpha^{(k)}$ is crucial for the success of such iteration schemes. We note that the computation of a reasonable step size $\alpha^{(k)} > 0$ leads to a generic one dimensional optimization problem usually referred to as *the line search*. Before we explain in detail the line search, we briefly describe the generic generalizations of the standard descent methods.

Generic descent methods. The *method of steepest descent* is obtained by setting

$$\boldsymbol{x}^{(k+1)} := \gamma_{\boldsymbol{x}^{(k)},\boldsymbol{d}^{(k)}}(\alpha^{(k)}), \qquad \boldsymbol{d}^{(k)} := -\nabla_{\mathcal{M}} f(\boldsymbol{x}^{(k)}) \in \mathrm{T}_{\boldsymbol{x}^{(k)}}\mathcal{M}, \qquad k = 0, 1, \ldots, \tag{2.3}$$

where $\nabla_{\mathcal{M}} f$ is the gradient of f and $\alpha^{(k)}$ is determined by a line search. Since it requires only first order derivatives it is quite easily to implement. However, it suffers from very slow convergence (linear rate), due to the "zig-zagging" nature of this method, cf. [31].

In contrast, *Newton's method* is given by the more sophisticated scheme

$$\boldsymbol{x}^{(k+1)} := \gamma_{\boldsymbol{x}^{(k)},\boldsymbol{d}^{(k)}}(1), \qquad k = 0, 1, \ldots, \tag{2.4}$$

where the descent direction $\boldsymbol{d}^{(k)} \in \mathrm{T}_{\boldsymbol{x}^{(k)}}\mathcal{M}$ is determined by the requirement

$$\mathrm{H}_{\mathcal{M}} f(\boldsymbol{x}^{(k)})(\boldsymbol{d}^{(k)}, \boldsymbol{v}) = -\nabla_{\mathcal{M}} f(\boldsymbol{x}^{(k)})^{\top} \boldsymbol{v}, \qquad \boldsymbol{v} \in \mathrm{T}_{\boldsymbol{x}^{(k)}}\mathcal{M}, \tag{2.5}$$

where $\mathrm{H}_{\mathcal{M}} f$ is the Hessian bilinear form associated to f. We note that for the Euclidean space $\mathcal{M} = \mathbb{R}^d$ the relation (2.5) is equivalent to the more familiar description $\boldsymbol{d}^{(k)} :=$

$-\mathbf{H}f(\boldsymbol{x}^{(k)})^{-1}\nabla f(\boldsymbol{x}^{(k)})$. It can be shown that nearby a local minimizer with positive definite Hessian the convergence of Newton's method is very fast (quadratic rate), cf. [31]. However, the generic scheme (2.4) suffers from serious instability problems if the sufficient convergence conditions are not met. Furthermore, the determination of the Newton step, cf. (2.5), by solving an equation system for the Hessian matrix, might be far to expansive for large scale problems.

A reasonable trade-off between stability, convergence rate and computational complexity is given by the *nonlinear conjugate gradient method (CG method)* determined by

$$\boldsymbol{x}^{(k+1)} := \gamma_{\boldsymbol{x}^{(k)},\boldsymbol{d}^{(k)}}(\alpha^{(k)}), \qquad \boldsymbol{d}^{(k+1)} := -\boldsymbol{g}^{(k+1)} + \beta^{(k)}\tilde{\boldsymbol{d}}^{(k)} \in \mathrm{T}_{\boldsymbol{x}^{(k+1)}}\mathcal{M}, \qquad k = 0, 1, \ldots, \tag{2.6}$$

where $\boldsymbol{g}^{(k)} := \nabla_{\mathcal{M}} f(\boldsymbol{x}^{(k)}) \in \mathrm{T}_{\boldsymbol{x}^{(k)}}\mathcal{M}$ and $\boldsymbol{d}^{(0)} := -\boldsymbol{g}^{(0)} \in \mathrm{T}_{\boldsymbol{x}^{(0)}}\mathcal{M}$ is the initial search direction. The tangent vector $\tilde{\boldsymbol{d}}^{(k)} \in \mathrm{T}_{\boldsymbol{x}^{(k+1)}}\mathcal{M}$ is defined by the parallel transported tangent vector $\boldsymbol{d}^{(k)}$ due to

$$\tilde{\boldsymbol{d}}^{(k)} := \dot{\gamma}_{\boldsymbol{x}^{(k)},d^{(k)}}(\alpha^{(k)}) \in \mathrm{T}_{\boldsymbol{x}^{(k+1)}}\mathcal{M}. \tag{2.7}$$

Specific choices of the parameter $\beta^{(k)}$ in (2.6) lead to several well-known conjugate gradient methods. In the case of Euclidean space an overview is given in [15]. For instance the CG method proposed by Daniel in [11] adapted to Riemannian manifolds, replaces the scalar $\beta^{(k)}$ by

$$\beta_{\mathrm{D}}^{(k)} := \frac{\mathrm{H}_{\mathcal{M}} f(\boldsymbol{x}^{(k+1)})(\boldsymbol{g}^{(k+1)}, \tilde{\boldsymbol{d}}^{(k)})}{\mathrm{H}_{\mathcal{M}} f(\boldsymbol{x}^{(k+1)})(\tilde{\boldsymbol{d}}^{(k)}, \tilde{\boldsymbol{d}}^{(k)})}. \tag{2.8}$$

For this particular CG method the convergence is shown to behave superlinear, cf. [12]. In the reminder of this paper we restricted our self to this specific coefficient rule since the computation of the Hessian matrix for the functions we have in mind will not increases the arithmetic complexity compared to CG methods without Hessian evaluation.

The line search. The search of a reasonable step length in an iterative optimization method of the form (2.2) leads to a generic one-dimensional optimization problem. The ideal choice of a step length would be the nearest local minimizer of the function $f \circ \gamma_{\boldsymbol{x},\boldsymbol{d}}$, $\boldsymbol{x} \in \mathcal{M}$, $\boldsymbol{d} \in \mathrm{T}_{\boldsymbol{x}}\mathcal{M}$. However, the determination of an exact local minimum is usually impractically so that the step lengths in (2.2) are usually computed by algorithms known as inexact line search methods.

Reasonable inexact line search methods determine step lengths α which satisfy the Wolfe conditions

$$f(\gamma_{\boldsymbol{x},\boldsymbol{d}}(\alpha)) - f(\gamma_{\boldsymbol{x},\boldsymbol{d}}(0)) \quad \leq \mu\,\alpha\,\tfrac{\mathrm{d}}{\mathrm{d}t} f(\gamma_{\boldsymbol{x},\boldsymbol{d}}(t))\Big|_{t=0} \tag{2.9}$$

$$\frac{\mathrm{d}}{\mathrm{d}t} f(\gamma_{\boldsymbol{x},\boldsymbol{d}}(t))\Big|_{T=\alpha} \quad \geq \eta\,\tfrac{\mathrm{d}}{\mathrm{d}t} f(\gamma_{\boldsymbol{x},\boldsymbol{d}}(t))\Big|_{t=0}, \tag{2.10}$$

for fixed constants $0 < \mu < \eta$. We remark that the relatively mild Wolfe conditions (2.9)–(2.10) imply for a large class of functions f convergence results for the method of steepest descent, Newton's method, and the nonlinear conjugate gradient method, which establish the classical convergence rate estimates, cf. [28, 12].

For simplicity we will use a simple Armijo line search method, cf. Algorithm 1, which only tries to enforce condition (2.9) and requires only the evaluation of the function f and its

gradient $\nabla_{\mathcal{M}} f$ at points along a given geodesic $\gamma_{\boldsymbol{x},\boldsymbol{d}}$. Note, we observed in the test cases in Section 3–5 that for the most starting points the second condition (2.10) is satisfied as well. For more sophisticated line search algorithms we refer to [28] and the references therein.

Algorithm 1 (**Simultaneous Line Search - Armijo condition**)

Parameters: $0 < \mu < \frac{1}{2}$, $0 < \tau < 1$, maximal iterations $k_{\max} \in \mathbb{N}$;
Input: differentiable function $f\colon \mathcal{M} \to \mathbb{R}$, starting points $\boldsymbol{x}_i \in \mathcal{M}$, descent directions $\boldsymbol{d}_i \in \mathrm{T}_{\boldsymbol{x}_i}\mathcal{M}$, initial step lengths $\alpha_i^{(1)} > 0$, $i = 1, \ldots, M$;
Initialization: $f_i^{(0)} := f(\boldsymbol{x}_i)$, $g_i := (f \circ \gamma_{\boldsymbol{x}_i,\boldsymbol{d}_i})'(0)$, $i = 1, \ldots, M$, iteration counter $k := 1$;
while $k <= k_{\max}$ **do**
  $f_i^{(k)} := f \circ \gamma_{\boldsymbol{x}_i,\boldsymbol{d}_i}(\alpha_i^{(k)})$, $i = 1, \ldots, M$;
  allgood:=true;
  for $i = 1, \ldots, M$ **do**
    if $f_i^{(k)} - f_i^{(0)} > \mu \cdot \alpha_i^{(k)} \cdot g_i$ **then**
      $\alpha_i^{(k+1)} := \tau \cdot \alpha_i^{(k)}$;
      allgood:=false;
    else
      $\alpha_i^{(k+1)} := \alpha_i^{(k)}$;
    end if
  end for
  $k := k + 1$;
  if allgood==true **then**
    break;
  end if
end while
Output: step lengths $\alpha_i^{(k)} \geq 0$, $i = 1, \ldots, M$.

Simultaneous optimization methods. In practical applications one is often interested in finding all local extrema of a function f. Unless f has additional properties, e.g. is strictly convex, this is impossible to accomplish by an algorithm, as f may have infinitely many local extrema. A common heuristic approach is to apply local optimization methods repeatedly with different and evenly spread starting points $\boldsymbol{x}_i \in \mathcal{M}$, $i = 1, \ldots, M$. With this direct approach the numerical cost is proportional to the number of starting points M times the numerical cost of computing one local extremum. Usually, the evaluations of the function f as well as the evaluation of its gradient and Hessian are the most expensive steps in such algorithms.

In the remainder of this paper we will consider functions f belonging to functions spaces that allow for FFT like algorithms for function evaluations as well as for evaluations of the gradient and the Hessian, i.e., for algorithms that scales linearly, modulo a log factor, with respect to the sum $M + N^d$ of the number M of evaluation points and the dimension N^d of the function space. Under these requirements we can arrange the evaluations of the function, its gradient and its Hessian such that they are performed simultaneously for each iteration step of some local optimization method. As a consequence, the arithmetic complexity of one simultaneous iteration for $M \sim N^d$ starting points is the same, modulo a log factor, as for a single starting point. The following two concrete implementations, cf. Algorithm 1

Algorithm 2 (**Simultaneous CG Method (with restarts)**)

Parameters: regularization parameter $\lambda > 0$, tolerance $\tau > 0$, maximal iterations $k_{\max}$;
Input: twice differentiable function $f\colon \mathcal{M} \to \mathbb{R}$, initial points $\boldsymbol{x}_1^{(0)}, \dots, \boldsymbol{x}_M^{(0)} \in \mathcal{M}$;
Initialization: $\boldsymbol{g}_i^{(0)} := \nabla_{\mathcal{M}} f(\boldsymbol{x}_i^{(0)})$, $\boldsymbol{d}_i^{(0)} := -\boldsymbol{g}_i^{(0)}$, $i = 1, \dots, M$, dimension d of $\mathcal{M}$, iteration counter $k = 1$;
while $\frac{1}{M}\sum_{i=1}^{M} |g_i^{(k)}| > \tau$ and $k < k_{\max}$ **do**
  $h_i^{(k)} := \mathrm{H}_{\mathcal{M}} f(\boldsymbol{x}_i^{(k)})(\boldsymbol{d}_i^{(k)}, \boldsymbol{d}_i^{(k)})$, $i = 1, \dots, M$;
  $\tilde{\alpha}_i^{(k)} := \frac{|\boldsymbol{g}_i^{(k)} \cdot \boldsymbol{d}_i^{(k)}|}{h_{1,i}^{(k)} + \lambda |\boldsymbol{g}_i^{(k)} \cdot \boldsymbol{d}_i^{(k)}| \|\boldsymbol{d}_i^{(k)}|}$, $i = 1, \dots, M$, (initial step lengths);
  compute $\alpha_i^{(k)}$ by the simultaneous line search Algorithm 1 with starting points $\boldsymbol{x}_i^{(k)}$, descent directions $\boldsymbol{d}_i^{(k)}$, initial step lengths $\tilde{\alpha}_i^{(k)}$ for $i = 1, \dots, M$;
  $\boldsymbol{x}_i^{(k+1)} := \gamma_{\boldsymbol{x}_i^{(k)}, \boldsymbol{d}_i^{(k)}}(\alpha_i^{(k)})$, $i = 1, \dots, M$;
  $\boldsymbol{g}_i^{(k+1)} := \nabla_{\mathcal{M}} f(\boldsymbol{x}_i^{(k+1)})$, $i = 1, \dots, M$;
  $\tilde{\boldsymbol{d}_i}^{(k)} := \dot{\gamma}_{\boldsymbol{x}_i^{(k)}, d_i^{(k)}}(\alpha_i^{(k)})$, $i = 1, \dots, M$;
  $\beta_{\mathrm{n},i}^{(k)} := \mathrm{H}_{\mathcal{M}} f(\boldsymbol{x}_i^{(k+1)})(\boldsymbol{g}_i^{(k+1)}, \tilde{\boldsymbol{d}_i}^{(k)})$, $i = 1, \dots, M$;
  $\beta_{\mathrm{d},i}^{(k)} := \mathrm{H}_{\mathcal{M}} f(\boldsymbol{x}_i^{(k+1)})(\tilde{\boldsymbol{d}_i}^{(k)}, \tilde{\boldsymbol{d}_i}^{(k)})$, $i = 1, \dots, M$;
  for $i = 1, \dots, M$ **do**
    if $(k+1) \equiv 0 \mod d$ and $\beta_{\mathrm{d},i}^{(k)} \neq 0$ **then**
      $\beta_i^{(k)} := \frac{\beta_{\mathrm{n},i}^{(k)}}{\beta_{\mathrm{d},i}^{(k)}}$;
      $\boldsymbol{d}_i^{(k+1)} := -\boldsymbol{g}_i^{(k+1)} + \beta_i^{(k)} \tilde{\boldsymbol{d}_i}^{(k)}$;
      if $\boldsymbol{g}_i^{(k+1)} \cdot \boldsymbol{d}_i^{(k+1)} > 0$ **then**
        $\boldsymbol{d}_i^{(k+1)} := -\boldsymbol{g}_i^{(k+1)}$ (enforce descent direction);
      end if
    else
      $\boldsymbol{d}_i^{(k+1)} := -\boldsymbol{g}_i^{(k+1)}$;
    end if
  end for
  $k := k + 1$;
end while
Output: approximate local minimizers $\boldsymbol{x}_1^{(k)}, \dots, \boldsymbol{x}_M^{(k)} \in \mathcal{M}$ of f.

and Algorithms 2, of simultaneous optimization algorithms are used for later reference and demonstration purposes. We remark that other well-know optimization methods such as Newton's method or trust region methods may be generalized as well and should provide on the torus $\mathbb{T}^d$, the sphere $\mathbb{S}^2$, and the rotation group SO(3) similar complexity bounds as in Theorem 3.1, Theorem 4.3, and Theorem 5.3, respectively.

In Algorithm 1 we present a simple implementation of a simultaneous line search which consists of one simultaneous evaluation of the gradient of f at all starting points in the initialization step and at most $k_{\max} + 1$ simultaneous evaluations of the function f.

Algorithm 2 is the implementation of a simultaneous conjugated gradient method on Riemannian manifolds, which consist of one simultaneous gradient evaluation and three simul-

taneous Hessian evaluations per iteration step. We note that the first Hessian evaluations lead to particular useful step length estimates for the initial step lengths used in Algorithm 1, since we are able to compute regularized Newton steps for the one-dimensional optimization problem. The chosen regularization parameter λ prohibits to take too large step lengths, which may result from degenerated Hessian forms. In particular for evenly spread starting points on compact manifolds this is a reasonable restriction, which allows us to search only in a small neighborhood (within radius $1/\lambda$) of the starting points for local minimizers without limiting the fast convergence behavior near a local minimizer with positive definite Hessian.

In all the following numerical tests we set in Algorithm 1 the parameters $\mu := 0.4$, $\tau := 0.5$, and the maximal iterations $k_{\max} := 6$. In Algorithm 2 we use the regularization parameter $\lambda := \sqrt[d]{M}/10$ (since we assume that the starting points are evenly spread), the tolerance $\tau := 10^{-8}$, and the maximal iterations $k_{\max} := 10$.

3 Global Optimization on the Torus

The most simple case of a non-Euclidean manifold is probably the d-dimensional torus $\mathbb{T}^d := \mathbb{R}^d/\mathbb{Z}^d$. We recall that functions on the torus $\mathbb{T}^d$ can be considered as one-periodic functions in $\mathbb{R}^d$, i.e., $f\colon \mathbb{R}^d \to \mathbb{C}$ with $f(\boldsymbol{x}+\boldsymbol{z}) = f(\boldsymbol{x})$, $\boldsymbol{x} \in \mathbb{R}^d$, $\boldsymbol{z} \in \mathbb{Z}^d$. Hence, the geometry of the torus $\mathbb{T}^d$ inherits naturally the geometry of the Euclidean space $\mathbb{R}^d$. That is, the tangent space $\mathrm{T}_{\boldsymbol{x}}\mathbb{T}^d$ at a point $\boldsymbol{x} \in \mathbb{T}^d$ is simply the Euclidean space $\mathbb{R}^d$ and the geodesics $\gamma_{\boldsymbol{x},\boldsymbol{d}}$, $\boldsymbol{x} \in \mathbb{T}^d$, $\boldsymbol{d} \in \mathbb{R}^d$, are given by straight lines modulo one in each coordinate. It follows that the gradient $\nabla_{\mathbb{T}^d} f(\boldsymbol{x})$ as well as the Hessian form $\mathrm{H}_{\mathbb{T}^d} f(\boldsymbol{x})$, $\boldsymbol{x} \in \mathbb{T}^d$, of a sufficient smooth function on the torus $\mathbb{T}^d$ are also expressed by the usual gradient and Hessian in $\mathbb{R}^d$.

Functions on the torus are often approximated by its truncated Fourier expansion. Let $f\colon \mathbb{R}^d \to \mathbb{C}$ be a trigonometric polynomial of degree N,

$$f(\boldsymbol{x}) := \sum_{\boldsymbol{k}\in\mathbb{Z}^d,\, |\boldsymbol{k}|\le N} \hat{f}(\boldsymbol{k}) \mathrm{e}^{2\pi \mathrm{i}\, \boldsymbol{k}\cdot\boldsymbol{x}}, \qquad \boldsymbol{x} \in \mathbb{R}^d \tag{3.1}$$

with Fourier coefficients $\hat{f}(\boldsymbol{k}) \in \mathbb{C}$, $\boldsymbol{k} \in \mathbb{Z}^d$, $|\boldsymbol{k}| \le N$. Then the evaluation of f at arbitrary points $\boldsymbol{x}_1, \dots, \boldsymbol{x}_M \in \mathbb{R}^d$, $M \in \mathbb{N}$, can be interpreted as the multiplication of the vector of Fourier coefficients $\hat{\boldsymbol{f}} = (\hat{f}_{\boldsymbol{k}})_{|\boldsymbol{k}|\le N} \in \mathbb{C}^{(2N+1)^d}$ with the Fourier matrix

$$\boldsymbol{F}_{N,M} := (\mathrm{e}^{2\pi \mathrm{i} \boldsymbol{k}\cdot\boldsymbol{x}_j})_{j=1,\dots,M;|\boldsymbol{k}|\le N} \in \mathbb{C}^{M\times(2N+1)^d}.$$

The matrix vector product $\boldsymbol{F}_{N,M}\hat{\boldsymbol{f}}$ can be computed approximately with accuracy ε by the nonequispaced fast Fourier transform (NFFT), cf. [30, 23], with arithmetic complexity $\mathcal{O}(N^d \log N + M \log^d(\varepsilon^{-1}))$. Accordingly, we can approximately compute the gradient and the Hessian matrix of f in M points $\boldsymbol{x}_1, \dots, \boldsymbol{x}_M \in \mathbb{R}^d$,

$$\begin{aligned}
\nabla f(\boldsymbol{x}_j) &= 2\pi\mathrm{i} \sum_{|\boldsymbol{k}|\le N} \hat{f}(k) \mathrm{e}^{2\pi \mathrm{i}\, \boldsymbol{k}\cdot\boldsymbol{x}_j} \boldsymbol{k} \in \mathbb{C}^d,\\
\mathbf{H} f(\boldsymbol{x}_j) &= -4\pi^2 \sum_{|\boldsymbol{k}|\le N} \hat{f}(k) \mathrm{e}^{2\pi \mathrm{i}\, \boldsymbol{k}\cdot\boldsymbol{x}_j} \boldsymbol{k}\boldsymbol{k}^T \in \mathbb{C}^{d\times d}
\end{aligned}$$

by an NFFT for fixed d with arithmetic complexity $\mathcal{O}(N^d \log N + M \log^d(\varepsilon^{-1}))$. Thus the NFFT enables us to compute one step of a descent method (2.2) simultaneously for M starting points with arithmetic complexity $\mathcal{O}(N^d \log N + M \log^d(\varepsilon^{-1}))$ as long as the line search

requires only a fixed number of evaluations of the Hessian and the gradient of f. This is especially remarkable as for the number of starting points M being of the same magnitude as the number of Fourier coefficients, i.e., $M \sim N^d$, the arithmetic complexity of computing one step of a descent method (2.2) for one point $\boldsymbol{x} \in \mathbb{R}^d$ is almost the same (up to a logarithmic factor) as computing it for all M points.

Theorem 3.1. *Let $f\colon \mathbb{R}^d \to \mathbb{R}$, $d \in \mathbb{N}_0$ fixed, be a real-valued trigonometric polynomial of degree $N \in \mathbb{N}_0$ and $\boldsymbol{x}_1, \dots, \boldsymbol{x}_M \in \mathbb{R}^d$ some nodes. Then the gradients $\nabla f(\boldsymbol{x}_j)$ and the Hessian matrices $\mathbf{H}f(\boldsymbol{x}_j)$, $j = 1, \dots, M$, and therefore, one step of the simultaneous CG method, cf. Algorithm 2, can be computed numerically with error ε and arithmetic complexity $\mathcal{O}(N^d \log N + M \log^d \varepsilon^{-1})$.*

Numerical experiments. In order to demonstrate the performance of our approach we consider the global minimization problem for the function $f\colon [-1,1]^2 \to \mathbb{R}$ given by

$$\begin{aligned} f(x,y) := \left(\frac{x^2}{4} + e^{\sin(50x)} + \sin(70\sin(x))\right) + \left(\frac{x^2}{4} + \sin(60e^y) + \sin(\sin(80y))\right) \\ - \cos(10x)\sin(10y) - \sin(10x)\cos(10y), \qquad (x,y) \in [-1,1]^2. \end{aligned} \tag{3.2}$$

Note, that this optimization problem has been originally proposed by Nick Trefethen as part of the SIAM 100-digit challenge, so that the global minimizer is known up to very high precision, cf. [7]. The challenge of finding the global minimum results from the many local minima provided by the function f, cf. Figure 3.1. We also like to mention that this function has also been used as a test function for the two-dimensional global optimization algorithm presented in [33]. However, our approach is different, since we are able to compute all local minimizers simultaneously.

In order to apply the proposed optimization algorithms on the torus, we need to transform the given function $f\colon [-1,1]^2 \to \mathbb{R}$ into a one-periodic function $\tilde{f}\colon \mathbb{R}^2 \to \mathbb{R}$ and, finally, approximate that function $\tilde{f}$ by a trigonometric polynomial $\tilde{f}_N\colon \mathbb{R}^2 \to \mathbb{R}$ of degree $N \in \mathbb{N}_0$, cf. (3.1). Since we aim to approximate the function $\tilde{f}$ by a trigonometric polynomial, we take the usual approach and apply the cosine substitution

$$\tilde{f}(t_1,t_2) := f(\boldsymbol{x}(t_1,t_2)), \qquad \boldsymbol{x}(t_1,t_2) := (\cos(2\pi t_1), \cos(2\pi t_2))^\top, \quad (t_1,t_2)^\top \in \mathbb{R}^2.$$

The reason for this substitution results from the good approximation properties of the Chebyshev polynomials of the first kind $T_k\colon [-1,1] \to \mathbb{R}$ and their relations to trigonometric polynomials due to $T_k(\cos(t)) = \cos(kt)$. The Fourier coefficients $\hat{f}_{\boldsymbol{k}}$, $\boldsymbol{k} := (k_1,k_2)^\top \in \mathbb{Z}^2$, $|\boldsymbol{k}| \le N$, of the trigonometric polynomial

$$\tilde{f}_N(t_1,t_2) := \sum_{k_1=-N}^{N-1} \sum_{k_2=-N}^{N-1} \hat{f}_{(k_1,k_2)} \mathrm{e}^{2\pi \mathrm{i}(k_1 t_1 + k_2 t_2)}$$

are simply determined by a corresponding discrete Fourier transform

$$\hat{f}_{(k_1,k_2)} := \frac{1}{4N^2} \sum_{i=1}^{2N} \sum_{j=1}^{2N} \tilde{f}(\boldsymbol{t}_{i,j}) \mathrm{e}^{-2\pi \mathrm{i} \boldsymbol{k} \cdot \boldsymbol{t}_{i,j}}, \qquad \boldsymbol{t}_{i,j} := \left(\frac{i-N}{2N}, \frac{j-N}{2N}\right)^\top, \ i,j = 1, \dots, 2N,$$

which can be performed in $\mathcal{O}(N^2 \log N)$ arithmetic operations.

For degree $N = 250$ we obtain an approximation error $\|\tilde{f} - \tilde{f}_N\|_\infty / \|\tilde{f}\|_\infty < 2 \cdot 10^{-3}$. After applying the simultaneous CG method, cf. Algorithm 2, on the torus $\mathbb{T}^2$ to $\tilde{f}_N$ with starting points $\boldsymbol{x}_j$, $j = 1, \dots, 10^4$, lying on an equispaced grid of size 100×100, we where able to determine 719 local minimizers, illustrated in Figure 3.1. The computed global minimizer $\tilde{\boldsymbol{t}}_*$ of $\tilde{f}_N$ satisfies $|\boldsymbol{x}(\tilde{\boldsymbol{t}}_*) - \boldsymbol{x}_*| < 10^{-4}$, where the global minimum of f is given by

$$\boldsymbol{x}_* := (-0.02440307969437517\dots, 0.2106124271553557\dots)^\top.$$

The error of the global minimizer can be readily reduced by applying a standard local minimization method to the original function f with starting point $\boldsymbol{x}(\tilde{\boldsymbol{t}}_*)$.

4 Global optimization on the two-dimensional sphere

In this section we want to apply the same ideas as in the previous one in order to develop a simultaneous optimization scheme for functions defined on the sphere

$$\mathbb{S}^2 = \left\{ \boldsymbol{\xi} \in \mathbb{R}^3 \mid |\boldsymbol{\xi}| = 1 \right\}.$$

Most of the calculations presented in this section lead to formulas which can be found in [14] where they have been applied for optimization on the sphere as well.

Geometry. As parameterisation of the sphere $\mathbb{S}^2$ we use spherical coordinates

$$\boldsymbol{\xi}(\theta, \rho) = (\cos\rho \sin\theta, \sin\rho \sin\theta, \cos\theta)^\top, \quad \theta \in [0, \pi],\ \rho \in [0, 2\pi).$$

Then the canonical basis of the tangent space

$$\mathrm{T}_{\boldsymbol{\xi}}\mathbb{S}^2 = \{\boldsymbol{x} \in \mathbb{R}^3 \mid \boldsymbol{x} \cdot \boldsymbol{\xi} = 0\}$$

at a point $\boldsymbol{\xi}(\theta, \rho) \in \mathbb{S}^2 \setminus \{(0, 0, \pm 1)^\top\}$ is given by

$$\boldsymbol{\xi}_\theta = \frac{\partial}{\partial\theta}\boldsymbol{\xi}(\theta, \rho), \quad \boldsymbol{\xi}_\rho = \frac{\partial}{\partial\rho}\boldsymbol{\xi}(\theta, \rho). \tag{4.1}$$

With respect to this basis the metric tensor $\boldsymbol{G}_{\mathbb{S}^2}$ takes the form

$$\boldsymbol{G}_{\mathbb{S}^2}(\boldsymbol{\xi}(\theta, \rho)) = \begin{pmatrix} 1 & 0 \\ 0 & \sin^2\theta \end{pmatrix},$$

which leads for any two tangent vectors $\boldsymbol{x} = x_\theta \boldsymbol{\xi}_\theta + x_\rho \boldsymbol{\xi}_\rho$ and $\boldsymbol{y} = y_\theta \boldsymbol{\xi}_\theta + y_\rho \boldsymbol{\xi}_\rho$ to the inner product

$$\boldsymbol{x} \cdot \boldsymbol{y} = (x_\theta\ x_\rho)\, \boldsymbol{G}_{\mathbb{S}^2}(\boldsymbol{\xi}(\theta, \rho)) \begin{pmatrix} y_\theta \\ y_\rho \end{pmatrix}.$$

The geodesics on the sphere are exactly the great circles. That is, for a starting point $\boldsymbol{\xi} \in \mathbb{S}^2$ and a direction $\boldsymbol{d} \in \mathrm{T}_{\boldsymbol{\xi}}\mathbb{S}^2$ the unique geodesics $\gamma_{\boldsymbol{\xi},\boldsymbol{d}} \colon \mathbb{R} \to \mathbb{S}^2$ with $\gamma_{\boldsymbol{\xi},\boldsymbol{d}}(0) = \boldsymbol{\xi}$ and $\dot{\gamma}_{\boldsymbol{\xi},\boldsymbol{d}}(0) = \boldsymbol{d}$ is given by, (cf. e.g. [34, p.19]),

$$\gamma_{\boldsymbol{\xi},\boldsymbol{d}}(\lambda) = \cos(\lambda\,|\boldsymbol{d}|)\boldsymbol{\xi} + \sin(\lambda\,|\boldsymbol{d}|)\boldsymbol{d}/\,|\boldsymbol{d}|\,. \tag{4.2}$$

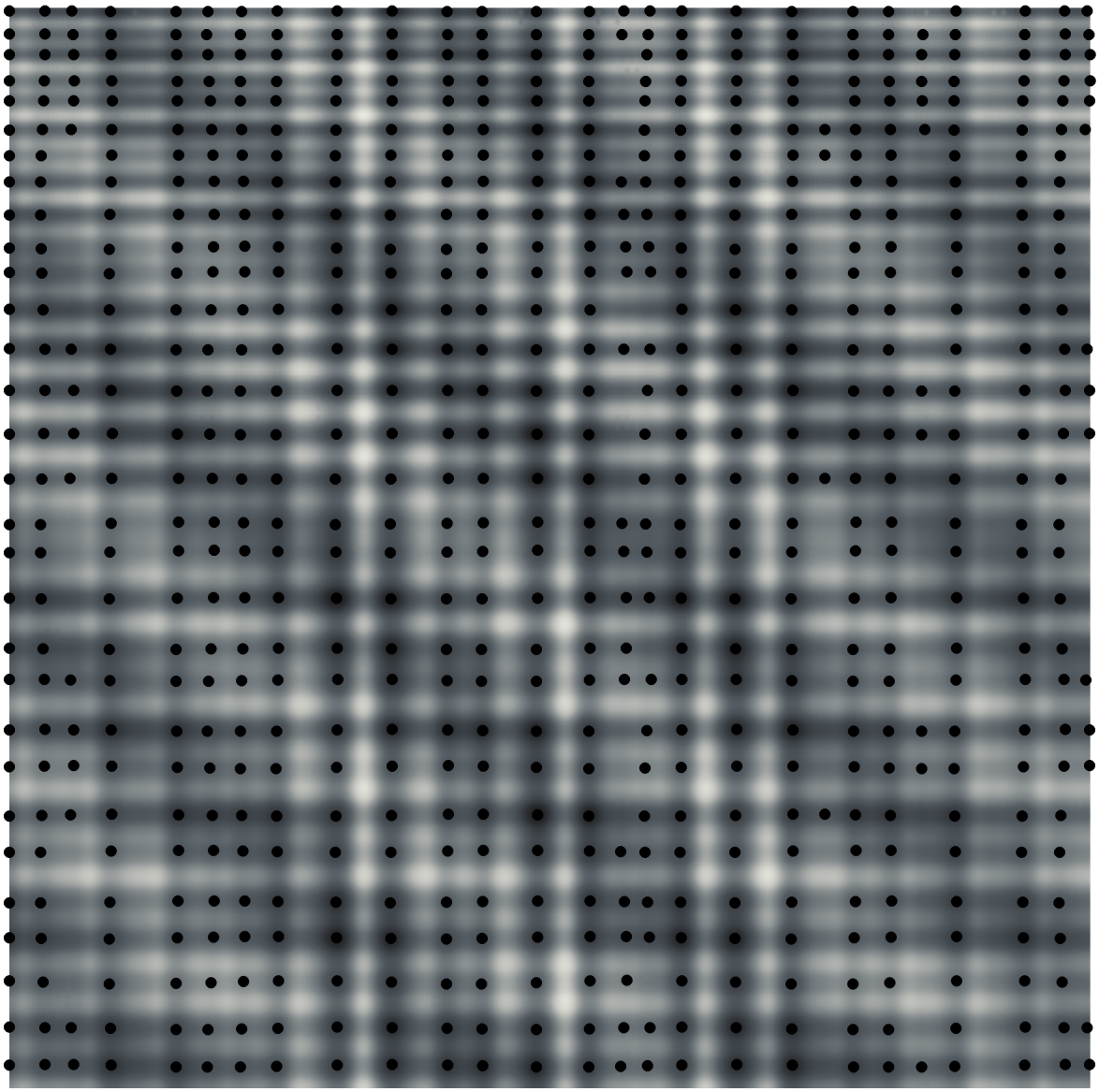

Figure 3.1: Illustration of the function $f\colon [-1,1]^2 \to \mathbb{R}$ given by (3.2) as a density plot and 719 local minima (black dots) detected by the simultaneous CG method applied to the trigonometric polynomial $\tilde{f}_N$, $N = 250$, and $10,000$ starting points lying on an equispaced 100×100 grid.

Gradient and Hessian. Let $f_\theta := \frac{\partial}{\partial\theta} f(\boldsymbol{\xi}(\theta,\rho))$ and $f_\rho := \frac{\partial}{\partial\rho} f(\boldsymbol{\xi}(\theta,\rho))$ be the partial derivatives of a function $f \in C^1(\mathbb{S}^2)$ with respect to θ and ρ. Then the spherical gradient $\nabla_{\mathbb{S}^2} f(\boldsymbol{\xi}(\theta,\rho)) \in \mathrm{T}_{\boldsymbol{\xi}(\theta,\rho)}\mathbb{S}^2$ has the following representation with respect to the basis given in (4.1),

$$\nabla_{\mathbb{S}^2} f(\boldsymbol{\xi}(\theta,\rho)) = (\boldsymbol{\xi}_\theta, \boldsymbol{\xi}_\rho)\, \boldsymbol{G}_{\mathbb{S}^2}^{-1}(\theta,\rho) \begin{pmatrix} f_\theta \\ f_\rho \end{pmatrix} = f_\theta\, \boldsymbol{\xi}_\theta + \sin^2\theta\, f_\rho\, \boldsymbol{\xi}_\rho. \tag{4.3}$$

In order to define the spherical Hessian matrix $\mathbf{H}_{\mathbb{S}^2} f(\boldsymbol{\xi}(\theta,\rho)) \in \mathbb{C}^{2\times 2}$ of a function $f \in C^2(\mathbb{S}^2)$ we need the spherical Christoffel symbols Γ_{ij}^{θ} , Γ_{ij}^{ρ}, $i,j = 1,2$, which can be written in matrix form as

$$\mathbf{\Gamma}^{\theta}(\theta,\rho) := \begin{pmatrix} 0 & 0 \\ 0 & -\sin\theta\cos\theta \end{pmatrix}, \quad \mathbf{\Gamma}^{\rho}(\theta,\rho) := \begin{pmatrix} 0 & \cot\theta \\ \cot\theta & 0 \end{pmatrix}.$$

Let $f_{\theta,\theta}$, $f_{\theta,\rho}$, $f_{\rho,\rho}$ be the partial derivatives of f with respect to the parameterization in polar coordinates. Then the spherical Hessian of f with respect to the canonical basis (4.1) in the tangent space $\mathrm{T}_{\boldsymbol{\xi}(\theta,\rho)}$ reads as, cf. [34],

$$\mathbf{H}_{\mathbb{S}^2} f(\boldsymbol{\xi}(\theta,\rho)) = \begin{pmatrix} f_{\theta,\theta} & f_{\theta,\rho} \\ f_{\theta,\rho} & f_{\rho,\rho} \end{pmatrix} - f_{\theta}\mathbf{\Gamma}^{\theta} - f_{\rho}\mathbf{\Gamma}^{\rho}. \tag{4.4}$$

Spherical harmonics. The analog of the exponential functions on the torus are the spherical harmonics on the sphere. Let the Legendre polynomials $P_n\colon [-1,1] \to \mathbb{R}$ and the associated Legendre functions $P_n^k\colon [-1,1] \to \mathbb{R}$ be defined by

$$\begin{aligned} P_n(x) &:= \frac{1}{2^n n!}\frac{\mathrm{d}^n}{\mathrm{d}x^n}\left(x^2-1\right)^n, && n \in \mathbb{N}_0, \\ P_n^k(x) &:= \left(\frac{(n-k)!}{(n+k)!}\right)^{1/2}\left(1-x^2\right)^{k/2}\frac{\mathrm{d}^k}{\mathrm{d}x^k}P_n(x), && n \in \mathbb{N}_0,\ k = 0,\dots,n. \end{aligned} \tag{4.5}$$

Then the spherical harmonics $Y_n^k\colon \mathbb{S}^2 \to \mathbb{C}$, $n \in \mathbb{N}_0$, $k = -n,\dots,n$,

$$Y_n^k(\boldsymbol{\xi}(\theta,\rho)) := \sqrt{\frac{2n+1}{4\pi}}P_n^{|k|}(\cos\theta)\mathrm{e}^{\mathrm{i}k\rho} \tag{4.6}$$

constitute an orthonormal basis in the space $L^2(\mathbb{S}^2)$ with respect to the canonical surface measure of the sphere $\mu_{\mathbb{S}^2}$ satisfying $\mu_{\mathbb{S}^2}(\mathbb{S}^2) = 4\pi$. By a spherical polynomial of degree $N \in \mathbb{N}_0$ we denote a function $f\colon \mathbb{S}^2 \to \mathbb{C}$ that has a finite series expansion in spherical harmonics

$$f(\boldsymbol{\xi}) := \sum_{n=0}^{N}\sum_{k=-n}^{n}\hat{f}(n,k)Y_n^k(\boldsymbol{\xi}), \quad \boldsymbol{\xi} \in \mathbb{S}^2. \tag{4.7}$$

As for the torus we can write the evaluation of f at arbitrary nodes $\boldsymbol{\xi}_1,\dots,\boldsymbol{\xi}_M \in \mathbb{S}^2$ as the product of a Fourier matrix

$$\boldsymbol{F}_{N,M} := \left(Y_n^k(\boldsymbol{\xi}_j)\right)_{j=1,\dots,M;(n,k)\in I_N} \in \mathbb{C}^{M\times|I_N|}$$

with the vector of Fourier coefficients $\hat{\boldsymbol{f}} := (\hat{f}(n,k))_{(n,k)\in I_N} \in \mathbb{C}^{|I_N|}$, where we use the index set

$$I_N := \{(n,k) \in \mathbb{N}\times\mathbb{Z} \,:\, n = 0,\dots,N;\ k = -n,\dots,n\}.$$

The matrix vector product $\boldsymbol{F}_{N,M}\hat{\boldsymbol{f}}$ can be computed approximately with error ε by an nonequispaced fast spherical Fourier transform (NFSFT) with arithmetic complexity $\mathcal{O}(N^2\log^2 N + M\log^2(\varepsilon^{-1}))$, cf. [25, 22].

In what follows we derive formulas for the gradient and the Hessian of a spherical polynomial f that allow for fast evaluation at nodes $\boldsymbol{\xi}_j \in \mathbb{S}^2$, $j = 1,\dots,M$, using the NFSFT. Eventually, we arrive at FFT like algorithms for the partial derivatives up to second order of polynomials on the sphere $\mathbb{S}^2$. The following representations of partial derivatives of a spherical polynomial have been originally utilized in [14, Lemma 3.1 and Theorem 3.2].

Lemma 4.1. *Let $f\colon \mathbb{S}^2 \to \mathbb{C}$ be a spherical polynomial of degree $N \in \mathbb{N}$ with Fourier coefficients $\hat{\boldsymbol{f}} := (\hat{f}(n,k))_{(n,k)\in I_N}$. Then its first order partial derivatives can be represented by spherical polynomials*

$$\frac{\partial}{\partial\rho} f(\boldsymbol{\xi}(\theta,\rho)) = \sum_{n=0}^{N} \sum_{k=-n}^{n} \hat{f}_\rho(n,k) Y_n^k(\boldsymbol{\xi}), \qquad \boldsymbol{\xi} \in \mathbb{S}^2$$

and

$$\sin\theta \frac{\partial}{\partial\theta} f(\boldsymbol{\xi}(\theta,\rho)) = \sum_{n=0}^{N+1} \sum_{k=-n}^{n} \hat{f}_\theta(n,k) Y_n^k(\boldsymbol{\xi}), \qquad \boldsymbol{\xi} \in \mathbb{S}^2,$$

of degree N and $N+1$, respectively. Their Fourier coefficients $\hat{\boldsymbol{f}}_\rho := (\hat{f}_\rho(n,k))_{(n,k)\in I_N}$ and $\hat{\boldsymbol{f}}_\theta := (\hat{f}_\theta(n,k))_{(n,k)\in I_{N+1}}$ are related to the Fourier coefficients $\hat{\boldsymbol{f}}$ by the linear mappings

$$\hat{\boldsymbol{f}}_\rho = \boldsymbol{D}_{N,\rho}\hat{\boldsymbol{f}} := (\mathrm{i}k\hat{f}(n,k))_{(n,k)\in I_N} \quad \text{and} \quad \hat{\boldsymbol{f}}_\theta = \boldsymbol{D}_{N,\theta}\hat{\boldsymbol{f}} := (\hat{f}_\theta(n,k))_{(n,k)\in I_{N+1}},$$

respectively, where

$$\hat{f}_\theta(n,k) := a_{n-1,k}\hat{f}(n-1,k) - b_{n+1,k}\hat{f}(n+1,k), \qquad (n,k) \in I_N,$$

with $\hat{f}_{n,k} := 0$ for $|k| > n$ or $n < 0$ and

$$a_{n,k} := n\sqrt{\frac{(n+1)^2-k^2}{(2n+1)(2n+3)}}, \quad b_{n,k} := (n+1)\sqrt{\frac{n^2-k^2}{(2n-1)(2n+1)}}, \qquad (n,k) \in I_N \setminus \{(0,0)\},$$

with $a_{-1,0} := a_{0,0} := b_{0,0} := 0$.

Proof. As a consequence of (4.6) and the recurrence relation of the associated Legendre functions, cf. [2, Eq. 8.5.3 and 8.5.4], we obtain for the partial derivatives of the spherical harmonics Y_n^k, $(n,k) \in I_N$, the expressions

$$\frac{\partial}{\partial\rho} Y_n^k(\boldsymbol{\xi}(\theta,\rho)) = \mathrm{i}k Y_n^k(\boldsymbol{\xi}(\theta,\rho)),$$

and

$$\sin\theta \frac{\partial}{\partial\theta} Y_n^k(\boldsymbol{\xi}(\theta,\rho)) = a_{n,k} Y_{n+1}^k(\boldsymbol{\xi}(\theta,\rho)) - b_{n,k} Y_{n-1}^k(\boldsymbol{\xi}(\theta,\rho)).$$

■

Similarly, we compute the second derivatives of a spherical polynomial by applying the linear mappings $\boldsymbol{D}_{N,\rho}\colon \mathbb{C}^{|I_N|} \to \mathbb{C}^{|I_N|}$, $\boldsymbol{D}_{N,\theta}\colon \mathbb{C}^{|I_N|} \to \mathbb{C}^{|I_{N+1}|}$ multiple times on the vector $\hat{\boldsymbol{f}}$ of Fourier coefficients of f.

Lemma 4.2. *Let $f\colon \mathbb{S}^2 \to \mathbb{C}$ be a spherical polynomial of degree $N \in \mathbb{N}$ with Fourier coefficients $\hat{\boldsymbol{f}} := (\hat{f}(n,k))_{(n,k)\in I_N}$. Then its second order partial derivatives can be represented as spherical polynomials*

$$\frac{\partial^2}{\partial \rho^2} f(\boldsymbol{\xi}(\theta,\rho)) = \sum_{n=0}^{N} \sum_{k=-n}^{n} \hat{f}_{\rho,\rho}(n,k) Y_n^k(\boldsymbol{\xi}), \qquad \boldsymbol{\xi} \in \mathbb{S}^2,$$

$$\sin\theta \frac{\partial^2}{\partial \theta \partial \rho} f(\boldsymbol{\xi}(\theta,\rho)) = \sum_{n=0}^{N+1} \sum_{k=-n}^{n} \hat{f}_{\theta,\rho}(n,k) Y_n^k(\boldsymbol{\xi}), \qquad \boldsymbol{\xi} \in \mathbb{S}^2,$$

and

$$\sin^2\theta \frac{\partial^2}{\partial \theta^2} f(\boldsymbol{\xi}(\theta,\rho)) = \sum_{n=0}^{N+2} \sum_{k=-n}^{n} \hat{f}_{\theta,\theta}(n,k) Y_n^k(\boldsymbol{\xi}(\theta,\rho)) - \cos\theta \sum_{n=0}^{N+1} \sum_{k=-n}^{n} \hat{f}_{\theta}(n,k) Y_n^k(\boldsymbol{\xi}(\theta,\rho))$$

of degree N, $N+1$ and $N+2$, respectively. Their Fourier coefficients are given accordingly to Lemma 4.1 by the linear mappings

$$\begin{aligned} \hat{\boldsymbol{f}}_{\rho,\rho} &:= (\hat{f}_{\rho,\rho}(n,k))_{(n,k)\in I_N} = \boldsymbol{D}_{N,\rho}^2 \hat{\boldsymbol{f}} \in \mathbb{C}^{|I_N|}, \\ \hat{\boldsymbol{f}}_{\theta,\rho} &:= (\hat{f}_{\theta,\rho}(n,k))_{(n,k)\in I_{N+1}} = \boldsymbol{D}_{N,\theta} \boldsymbol{D}_{N,\rho} \hat{\boldsymbol{f}} \in \mathbb{C}^{|I_{N+1}|}. \end{aligned} \tag{4.8}$$

and

$$\begin{aligned} \hat{\boldsymbol{f}}_{\theta,\theta} &:= (\hat{f}_{\theta,\theta}(n,k))_{(n,k)\in I_{N+2}} = \boldsymbol{D}_{N+1,\theta} \boldsymbol{D}_{N,\theta} \hat{\boldsymbol{f}} \in \mathbb{C}^{|I_{N+2}|}, \\ \hat{\boldsymbol{f}}_{\theta} &:= (\hat{f}_{\theta}(n,k))_{(n,k)\in I_{N+1}} = \boldsymbol{D}_{N,\theta} \hat{\boldsymbol{f}} \in \mathbb{C}^{|I_{N+1}|}. \end{aligned} \tag{4.9}$$

Proof. The relation (4.8) follows readily from Lemma 4.1. For formula (4.9) we observe that

$$\sin\theta \frac{\partial}{\partial \theta} \sin\theta \frac{\partial}{\partial \theta} = \sin^2\theta \frac{\partial^2}{\partial \theta^2} + \sin\theta\cos\theta \frac{\partial}{\partial \theta}$$

and apply Lemma 4.1 again. ∎

Combining Lemma 4.1 and 4.2 with the formulas for the spherical gradient (4.3) and the spherical Hessian (4.4) we conclude that the computation of the gradient and the Hessian allow for FFT like algorithms.

Theorem 4.3. *Let $f\colon \mathbb{S}^2 \to \mathbb{R}$ be a real-valued spherical polynomial of degree $N \in \mathbb{N}_0$ and $\boldsymbol{x}_1, \dots, \boldsymbol{x}_M \in \mathbb{S}^2 \setminus \{(0,0,\pm 1)^\top\}$ be some nodes. Then the spherical gradients $\nabla_{\mathbb{S}^2} f(\boldsymbol{x}_j)$ and the spherical Hessian matrices $\mathbf{H}_{\mathbb{S}^2} f(\boldsymbol{x}_j)$, $j = 1, \dots, M$, and therefore, one step of the simultaneous CG method, cf. Algorithm 2, can be computed numerically with error ε and arithmetic complexity $\mathcal{O}(N^2 \log^2 N + M \log^2 \varepsilon^{-1})$.*

Proof. By Lemma 4.1 and Lemma 4.2 we know that the partial derivatives up to second order of f can be represented by spherical harmonics, cf. Lemma 4.1 and Lemma 4.2, and therefore can be evaluated by the NFSFT at any given points $\boldsymbol{x}_1, \dots, \boldsymbol{x}_M \in \mathbb{S}^2 \setminus \{(0,0,\pm 1)^\top\}$ numerically with arithmetic complexity $\mathcal{O}(N^2 \log^2 N + M \log^2 \varepsilon^{-1})$, after computing the corresponding Fourier coefficients in $\mathcal{O}(N^2)$. Hence, it is readily seen that the spherical gradients $\nabla_{\mathbb{S}^2} f$, cf. (4.3), and the Hessian matrices $\mathbf{H}_{\mathbb{S}^2} f$, cf. (4.4), can be evaluated with the same arithmetic complexity at these points.

Note that by (4.2) we can evaluate the geodesics $\gamma_{\boldsymbol{x}_i,\boldsymbol{d}_i}(\alpha_i)$ for any given tangent vectors $\boldsymbol{d}_i \in \mathrm{T}_{\boldsymbol{x}_i}\mathbb{S}^2$ and step sizes $\alpha_i \in \mathbb{R}$, as well as its derivatives $\dot{\gamma}_{\boldsymbol{x}_i,\boldsymbol{d}_i}(\alpha_i)$, $i = 1, \dots, M$, in $\mathcal{O}(M)$. Since any iteration step of Algorithm 2 and Algorithm 1 takes only a bounded number of function and geodesic evaluations, including their derivatives, we arrive at the assertion. ∎

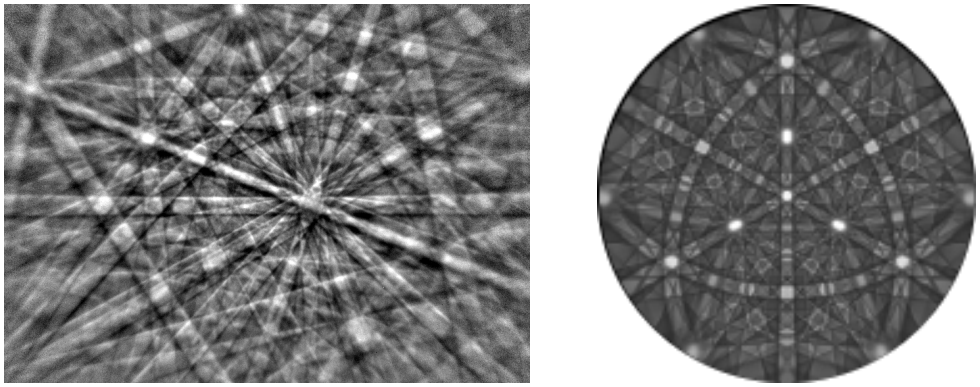

Figure 4.1: A raw EBSD image (left) and a simulated Kikuchi pattern on the sphere (right). Data take from G. Nolze, Federal Institute for Materials Research and Testing, Germany, personal communication.

A real world application – Band detection in Kikuchi pattern. In electron back scatter diffraction a crystalline material is irradiated by an electron beam. The diffraction pattern, the so called Kikuchi pattern, is measured by an area detector and results in an image containing bands intersecting each other at specific angles, cf. Figure 4.1 (left). This pattern can be interpreted best when projected on a two sphere, cf. Figure 4.1 (right). Then any band of the pattern is the intersection of the sphere with a certain crystallographic lattice plane. As the lattice planes are characteristic for each crystal EBSD measurements of Kikuchi pattern are used to identify crystals and there orientation within a specimen [4].

There exist a lot of methods that aim at extracting all the bands in a Kikuchi pattern fast and accurately, e.g., [16]. Most of them consider this problem in the plane as fast algorithms for line detection are well known. On the other hand it is well known that the bands in a Kikuchi pattern are not straight lines but hyperbolics which lowers the accuracy of those methods cf. [27]. Our idea is to deal with the band detection problem directly on the sphere. We propose the following procedure

(1) Project the image data to the sphere and approximate it by a spherical polynomial $f\colon \mathbb{S}^2 \to \mathbb{R}$.

(2) Compute the integrals of f along all geodesics, i.e. compute the Radon transform of f,

$$Rf(\boldsymbol{\xi}) := \int_{\boldsymbol{\eta}\perp\boldsymbol{\xi}} f(\boldsymbol{\eta})\,\mathrm{d}\boldsymbol{\eta}.$$

(3) Find all local extrema of Rf.

For the approximation of the data by a spherical polynomial we use spherical t-designs, i.e., sampling points $\boldsymbol{\xi}_j \in \mathbb{S}^2$, $j = 1, \dots, M$, that allow for exact quadrature up to a polynomial degree t with equal quadrature weights. In our experiment we use a 1000-design consisting of $520,000$ sampling points found on [13], which allows for the computation of the Fourier coefficients of polynomials with degrees at most $N = 500$. In our specific example the data does not cover the whole sphere but only some subregion. Therefore, we continue the data by its mean value to the entire sphere. Figure 4.2 shows the approximation of the data by a spherical polynomial of degree 500.

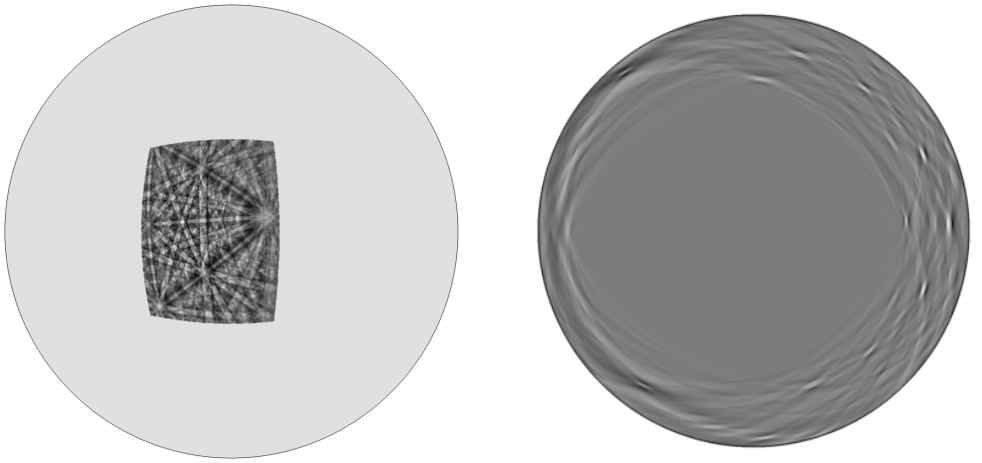

Figure 4.2: The approximation of the raw EBSD data by a spherical polynomial of degree 500 (left) and its Radon transform (right).

For the second step of our procedure, i.e., for the computation of the Radon transform we make use of the fact that the spherical harmonics are eigenfunctions, cf. [10, Lemma 2.1.6], i.e.,

$$\mathcal{R}Y_n^k = P_n(0)Y_n^k, \quad n \in \mathbb{N},\ k = -n, \dots, n,$$

where P_n is the Legendre polynomial defined in (4.5). In particular, the Radon transform of a spherical polynomial is a spherical polynomial of the same degree and we can apply the NFSFT to evaluate it at arbitrary points. The Radon transformed polynomial $\mathcal{R}f$ of the Kikuchy pattern is shown in Figure 4.2.

In the third step we apply the simultaneous CG method, cf. Algorithm 2, to the spherical polynomial $\mathcal{R}f$ with $M = 100,000$ well distributed starting points $\boldsymbol{x}_j := \boldsymbol{\xi}(\theta_j, \rho_j)$, $j = 1, \dots, M$, given in spherical coordinates by

$$\theta_j := \arccos\left(\frac{2j - (M+1)}{M}\right), \qquad \varphi_j := \pi(2j - (M+1))\phi^{-1},$$

where $\phi = \frac{1+\sqrt{5}}{2}$ is the golden ratio. These points lie on a Fibonacci spiral as proposed in [32]. As a result the proposed optimization method detects about 30,000 different local minimzers for the Radon transform Rf, where the most are caused by the constant continuation of the original image together with the polynomial approximation. In Figure 4.3 we plot the 80 smallest one of the upper hemisphere in the Radon transformed image and in Figure 4.4 the corresponding lines in the original image.

5 The rotation group

As a last example we consider the rotation group

$$\mathrm{SO}(3) := \{\boldsymbol{R} \in \mathbb{R}^{3\times 3} \mid \det \boldsymbol{R} = 1,\ \boldsymbol{R}^{-1} = \boldsymbol{R}^\top\}$$

consisting of all orthogonal three-by-three matrices with determinant one. Any matrix $\boldsymbol{R} \in \mathrm{SO}(3)$ has at least one eigenvalue 1. The corresponding eigenvector $\boldsymbol{\xi} \in \mathbb{S}^2$ is called rotation

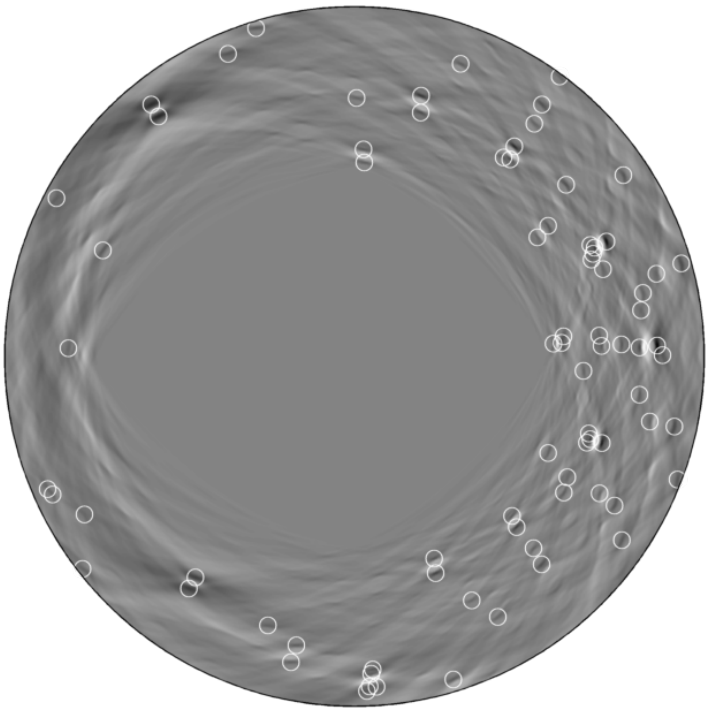

Figure 4.3: The top 80 detected minimizers of the upper hemisphere in the Radon transformed image.

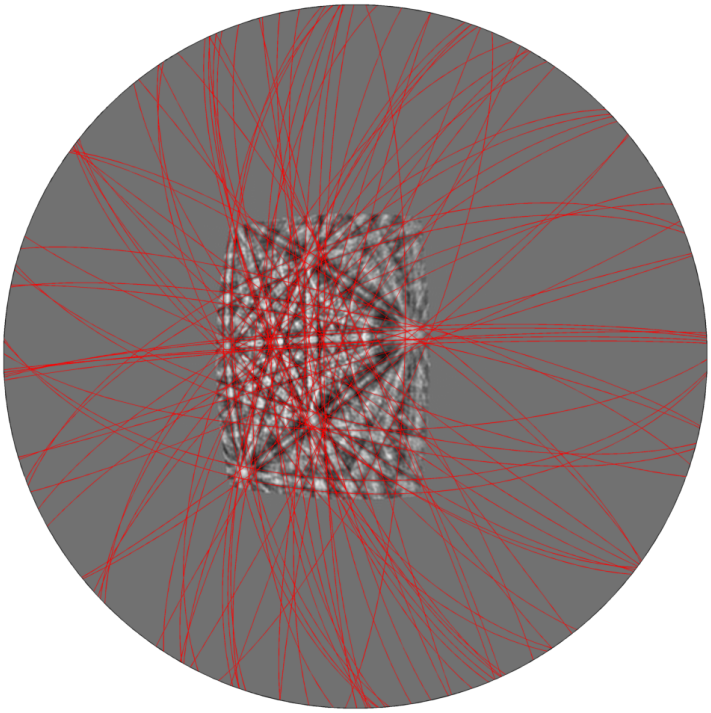

Figure 4.4: The lines corresponding to the top 80 detected minimizers in the Radon transformed image in the original image.

axis of $\boldsymbol{R}$. The rotation angle ω is given by $2\cos\omega = \mathrm{tr}\boldsymbol{R} - 1$, where $\mathrm{tr}\boldsymbol{R}$ denotes the trace of the matrix. Conversely, we denote for any rotation axis $\boldsymbol{\xi} \in \mathbb{S}^2$ and any rotation angle $\omega \in [0, 2\pi)$ the corresponding rotation matrix by $\boldsymbol{R}_{\boldsymbol{\xi},\omega} \in \mathrm{SO}(3)$.

Geometry. As a parameterization of the rotation group we will use Euler angles, i.e., we use the fact that any rotation $\boldsymbol{R} \in \mathrm{SO}(3)$ can be represented as the composition

$$\begin{aligned}\boldsymbol{R}(\varphi_1, \theta, \varphi_2) &:= \boldsymbol{R}_{\boldsymbol{e}_3,\varphi_1}\boldsymbol{R}_{\boldsymbol{e}_2,\theta}\boldsymbol{R}_{\boldsymbol{e}_3,\varphi_2} \\ &= \begin{pmatrix} \cos\varphi_1\cos\theta\cos\varphi_2 - \sin\varphi_1\sin\varphi_2 & -\cos\varphi_1\cos\theta\sin\varphi_2 - \sin\varphi_1\cos\varphi_2 & \cos\varphi_1\sin\theta \\ \sin\varphi_1\cos\theta\cos\varphi_2 + \cos\varphi_1\sin\varphi_2 & -\sin\varphi_1\cos\theta\sin\varphi_2 + \cos\varphi_1\cos\varphi_2 & \sin\varphi_1\sin\theta \\ \sin\theta\cos\varphi_2 & \sin\theta\sin\varphi_2 & \cos\theta \end{pmatrix}\end{aligned} \tag{5.1}$$

of three rotations with fixed axes $\boldsymbol{e}_3 := (0,0,1)^\top$, $\boldsymbol{e}_2 := (0,1,0)^\top$ and suitable chosen angles $(\varphi_1, \theta, \varphi_2) \in [0, 2\pi) \times [0, \pi] \times [0, 2\pi)$.

The tangent space of the rotation group $\mathrm{SO}(3)$ at the identity matrix $\boldsymbol{I} \in \mathrm{SO}(3)$ is given by

$$\mathrm{T}_{\boldsymbol{I}}\mathrm{SO}(3) = \{\boldsymbol{S} \in \mathbb{R}^{3\times3} \mid \boldsymbol{S}^\top = -\boldsymbol{S}\},$$

i.e., it consists of all skew symmetric three-by-three matrices and is also known as the Lie algebra $\mathfrak{so}(3)$. It follows that the tangent space at an arbitrary rotation $\boldsymbol{R} \in \mathrm{SO}(3)$ is obtained by left multiplication with $\boldsymbol{R}$, i.e.,

$$\mathrm{T}_{\boldsymbol{R}}\mathrm{SO}(3) = \boldsymbol{R}\,\mathrm{T}_{\boldsymbol{I}} = \{\boldsymbol{R}\boldsymbol{S} \in \mathbb{R}^{3\times3} \mid \boldsymbol{S}^\top = -\boldsymbol{S}\}.$$

For rotation matrices $\boldsymbol{R}(\varphi_1, \theta, \varphi_2) \in \mathrm{SO}(3)$ with $\theta \notin \{0, \pi\}$ the the partial derivatives of the parameterization with respect to the Euler angles form a basis of the tangent space $\mathrm{T}_{\boldsymbol{R}(\varphi_1,\theta,\varphi_2)}\mathrm{SO}(3)$, which is given by

$$\begin{aligned}\boldsymbol{R}_{\varphi_1}(\varphi_1, \theta, \varphi_2) &= \frac{\partial}{\partial_{\varphi_1}}\boldsymbol{R}(\varphi_1, \theta, \varphi_2) \\ &= \begin{pmatrix} -\sin\varphi_1\cos\theta\cos\varphi_2 - \cos\varphi_1\sin\varphi_2 & \sin\varphi_1\cos\theta\sin\varphi_2 - \cos\varphi_1\cos\varphi_2 & -\sin\varphi_1\sin\theta \\ -\cos\varphi_1\cos\theta\cos\varphi_2 - \sin\varphi_1\sin\varphi_2 & -\cos\varphi_1\cos\theta\sin\varphi_2 - \sin\varphi_1\cos\varphi_2 & \cos\varphi_1\sin\theta \\ 0 & 0 & 0 \end{pmatrix}, \\ \boldsymbol{R}_{\varphi_1}(\varphi_1, \theta, \varphi_2) &= \frac{\partial}{\partial_{\varphi_1}}\boldsymbol{R}(\varphi_1, \theta, \varphi_2) \\ &= \begin{pmatrix} -\cos\varphi_1\sin\theta\cos\varphi_2 & \cos\varphi_1\sin\theta\sin\varphi_2 & \cos\varphi_1\cos\theta \\ -\sin\varphi_1\sin\theta\cos\varphi_2 & \sin\varphi_1\sin\theta\sin\varphi_2 & \sin\varphi_1\cos\theta \\ -\cos\theta\cos\varphi_2 & \cos\theta\sin\varphi_2 & -\sin\theta \end{pmatrix}, \\ \boldsymbol{R}_{\varphi_1}(\varphi_1, \theta, \varphi_2) &= \frac{\partial}{\partial_{\varphi_1}}\boldsymbol{R}(\varphi_1, \theta, \varphi_2) \\ &= \begin{pmatrix} -\cos\varphi_1\cos\theta\sin\varphi_2 - \sin\varphi_1\cos\varphi_2 & -\cos\varphi_1\cos\theta\cos\varphi_2 + \sin\varphi_1\sin\varphi_2 & 0 \\ -\sin\varphi_1\cos\theta\sin\varphi_2 + \cos\varphi_1\cos\varphi_2 & -\sin\varphi_1\cos\theta\cos\varphi_2 - \cos\varphi_1\sin\varphi_2 & 0 \\ \sin\theta\sin\varphi_2 & \sin\theta\cos\varphi_2 & 0 \end{pmatrix}.\end{aligned} \tag{5.2}$$

In contrast to the spherical case these basis vectors are not orthogonal and the corresponding metric tensor reads as

$$\boldsymbol{G}_{\mathrm{SO}(3)}(\varphi_1, \theta, \varphi_2) = \begin{pmatrix} 2 & 0 & 2\cos\theta \\ 0 & 2 & 0 \\ 2\cos\theta & 0 & 2 \end{pmatrix}.$$

For two tangential vectors $\boldsymbol{x}, \boldsymbol{y} \in T_{\boldsymbol{R}}\mathrm{SO}(3)$, $\boldsymbol{x} = x_{\varphi_1}\boldsymbol{R}_{\varphi_1} + x_\theta \boldsymbol{R}_\theta + x_{\varphi_2}\boldsymbol{R}_{\varphi_2}$ and $\boldsymbol{y} = y_{\varphi_1}\boldsymbol{R}_{\varphi_1} + y_\theta \boldsymbol{R}_\theta + y_{\varphi_2}\boldsymbol{R}_{\varphi_2}$ their inner product is

$$\boldsymbol{x} \cdot \boldsymbol{y} = (x_{\varphi_1}, x_\theta, x_{\varphi_2})\boldsymbol{G}(\varphi_1, \theta, \varphi_2)(x_{\varphi_1}, x_\theta, x_{\varphi_2})^\top.$$

Let $\boldsymbol{D} \in T_{\boldsymbol{I}}\mathrm{SO}(3)$ be a tangential vector at the identity. Then the geodesic $\gamma_{\boldsymbol{I},\boldsymbol{D}}$ with $\gamma_{\boldsymbol{I},\boldsymbol{D}}(0) = \boldsymbol{I}$ and derivative $\dot\gamma_{\boldsymbol{I},\boldsymbol{D}}(0) = \boldsymbol{D}$ is given by the exponential map

$$\gamma_{\boldsymbol{I},\boldsymbol{D}}(t) = \mathrm{e}^{t\boldsymbol{D}} := \sum_{k=0}^{\infty} \frac{(t\boldsymbol{D})^k}{k!}. \tag{5.3}$$

Accordingly, the geodesic $\gamma_{\boldsymbol{R},\boldsymbol{D}}$ with $\gamma_{\boldsymbol{R},\boldsymbol{D}}(0) = \boldsymbol{R} \in \mathrm{SO}(3)$ and derivative $\dot\gamma_{\boldsymbol{R},\boldsymbol{D}}(0) = \boldsymbol{D} \in T_{\boldsymbol{R}}\mathrm{SO}(3)$ is obtained by

$$\gamma_{\boldsymbol{R},\boldsymbol{D}}(t) = \boldsymbol{R}\mathrm{e}^{t\boldsymbol{R}^T\boldsymbol{D}}.$$

Gradient and Hessian. With respect to the canonical basis the gradient of a function $f \in C(\mathrm{SO}(3))$ takes the form

$$\nabla_{\mathrm{SO}(3)} f(\boldsymbol{R}(\varphi_1, \theta, \varphi_2)) = (\boldsymbol{R}_{\varphi_1}, \boldsymbol{R}_\theta, \boldsymbol{R}_{\varphi_2})\, \boldsymbol{G}^{-1}_{\mathrm{SO}(3)}(\varphi_1, \theta, \varphi_2) \begin{pmatrix} f_{\varphi_1} \\ f_\theta \\ f_{\varphi_2} \end{pmatrix}$$

where $f_{\varphi_1} := \frac{\partial}{\partial \varphi_1} f(\boldsymbol{R}(\varphi_1, \theta, \varphi_2)$, $f_\theta := \frac{\partial}{\partial \theta} f(\boldsymbol{R}(\varphi_1, \theta, \varphi_2)$ and $f_{\varphi_2} := \frac{\partial}{\partial \varphi_2} f(\boldsymbol{R}(\varphi_1, \theta, \varphi_2)$ are the first order partial derivatives of f with respect to the Euler angles $\varphi_1, \theta, \varphi_2$. Similarly, the Hessian $Hf(\boldsymbol{R}(\varphi_1, \theta, \varphi_2)) \in \mathbb{C}^{3\times 3}$ reads in matrix form as

$$\mathbf{H}_{\mathrm{SO}(3)} f(\boldsymbol{R}(\varphi_1, \theta, \varphi_2)) = \begin{pmatrix} f_{\varphi_1,\varphi_1} & f_{\varphi_1,\theta} & f_{\varphi_1,\varphi_2} \\ f_{\theta,\varphi_1} & f_{\theta,\theta} & f_{\theta,\varphi_2} \\ f_{\varphi_2,\varphi_1} & f_{\varphi_2,\theta} & f_{\varphi_2,\varphi_2} \end{pmatrix} - f_{\varphi_1}\boldsymbol{\Gamma}^{\varphi_1} - f_\theta \boldsymbol{\Gamma}^\theta - f_{\varphi_2}\boldsymbol{\Gamma}^{\varphi_2},$$

with Christoffel symbols $\Gamma^{\varphi_1}_{ij}$, Γ^{θ}_{ij}, $\Gamma^{\varphi_1}_{ij}$, $i, j = 1, 2, 3$, given for the Euler angle parameterization by

$$\boldsymbol{\Gamma}^{\varphi_1} = \frac{1}{2}\begin{pmatrix} 0 & \cot\theta & 0 \\ \cot\theta & 0 & -\sin^{-1}\theta \\ 0 & -\sin^{-1}\theta & 0 \end{pmatrix}, \quad \boldsymbol{\Gamma}^{\theta} = \frac{1}{2}\begin{pmatrix} 0 & 0 & \sin\theta \\ 0 & 0 & 0 \\ \sin\theta & 0 & 0 \end{pmatrix}, \quad \boldsymbol{\Gamma}^{\varphi_1} = \frac{1}{2}\begin{pmatrix} 0 & -\sin^{-1}\theta & 0 \\ -\sin^{-1}\theta & 0 & \cot\theta \\ 0 & \cot\theta & 0 \end{pmatrix}.$$

Wigner–D functions. As basis functions on the rotation group we consider Wigner-D functions, cf. [35], which are defined in terms of Euler angles as

$$D^{k,k'}_n(\boldsymbol{R}(\varphi_1, \theta, \varphi_2)) = \mathrm{e}^{-\mathrm{i}k\varphi_1}\mathrm{e}^{-\mathrm{i}k'\varphi_2} d^{k,k'}_n(\cos\theta), \quad n \in \mathbb{N},\ k, k' = -n, \ldots, n, \tag{5.4}$$

with Wigner-d functions $d^{k,k'}_n : [-1, 1] \to \mathbb{R}$, $n \in \mathbb{N}$, $k, k' = -n, \ldots, n$,

$$d^{k,k'}_n(t) = \frac{(-1)^{n-k}}{2^n}\sqrt{\frac{(n+k)!}{(n-k')!(n+k')!(n-k)!}}\sqrt{\frac{(1-t)^{k'-k}}{(1+t)^{k+k'}}}\frac{\mathrm{d}^{n-k}}{\mathrm{d}t^{n-k}}\frac{(1+t)^{k'+n}}{(1-t)^{k'-n}}. \tag{5.5}$$

The Wigner-D functions form an orthogonal basis in the space $L^2(\mathrm{SO}(3))$ with respect to the Haar measure of the Lie group SO(3). A polynomial of degree N on the rotation group is

a function $f\colon \mathrm{SO}(3) \to \mathbb{C}$ that permits for a series expansion into Wigner-D functions up to order N, i.e.,

$$f(\boldsymbol{R}) := \sum_{(n,k,k')\in J_N} \hat{f}(n,k,k') D_n^{k,k'}(\boldsymbol{R}), \qquad \boldsymbol{R} \in \mathrm{SO}(3), \tag{5.6}$$

where we use the index set

$$J_N := \{(n,k,k') \in \mathbb{N} \times \mathbb{Z}^2 \ : \ n = 0,\dots,N;\ k,k' = -n,\dots,n\}.$$

Similar to the spherical case we can write the evaluation of a polynomial f on the rotation group at arbitrary nodes $\boldsymbol{R}_1,\dots,\boldsymbol{R}_M \in \mathrm{SO}(3)$ as the product of a Fourier matrix

$$\boldsymbol{F}_{N,M} := \left(D_n^{k,k'}(\boldsymbol{R}_j)\right)_{j=1,\dots,M;(n,k,k')\in J_N} \in \mathbb{C}^{M\times|J_N|}$$

with the vector of Fourier coefficients $\hat{\boldsymbol{f}} = (\hat{f}(n,k,k'))_{(n,k,k')\in J_N}$. This matrix vector product can be computed approximately by a nonequispaced fast SO(3) Fourier transform (NFSOFT) with arithmetic complexity $\mathcal{O}(N^3 \log^2 N + M \log^2(\varepsilon^{-1}))$, cf. [29, 24].

For the partial derivatives of a polynomial on the rotation group we have the following representations.

Lemma 5.1. *Let $f\colon \mathrm{SO}(3) \to \mathbb{C}$ be a polynomial on the rotation group of degree $N \in \mathbb{N}$ with Fourier coefficients $\hat{\boldsymbol{f}} := (\hat{f}(n,k,k'))_{(n,k,k')\in J_N}$. Then its first order partial derivatives can be represented as polynomials on the rotation group*

$$\frac{\partial}{\partial\varphi_i} f(\boldsymbol{R}(\varphi_1,\theta,\varphi_2)) = \sum_{(n,k,k')\in J_N} \hat{f}_{\varphi_i}(n,k,k') D_n^{k,k'}(\boldsymbol{R}), \qquad \boldsymbol{R} \in \mathrm{SO}(3), \qquad i = 1,2,$$

and

$$\sin\theta \frac{\partial}{\partial\theta} f(\boldsymbol{R}(\varphi_1,\theta,\varphi_2)) = \sum_{(n,k,k')\in J_N} \hat{f}_{\theta}(n,k,k') D_n^{k,k'}(\boldsymbol{R}), \qquad \boldsymbol{R} \in \mathrm{SO}(3),$$

of degree N and $N+1$, respectively. Their Fourier coefficients $\hat{\boldsymbol{f}}_{\varphi_i} := (\hat{f}_{\varphi_i}(n,k,k'))_{(n,k,k')\in J_N}$, $i = 1,2$, and $\hat{\boldsymbol{f}}_{\theta} := (\hat{f}_{\theta}(n,k,k'))_{(n,k,k')\in J_{N+1}}$ are related to the Fourier coefficients $\hat{\boldsymbol{f}}$ by the linear mappings

$$\begin{aligned}
\hat{\boldsymbol{f}}_{\varphi_1} &= \boldsymbol{D}_{N,\varphi_1}\hat{\boldsymbol{f}} := (-\mathrm{i}k\hat{f}(n,k,k'))_{(n,k,k')\in J_N},\\
\hat{\boldsymbol{f}}_{\varphi_2} &= \boldsymbol{D}_{N,\varphi_2}\hat{\boldsymbol{f}} := (-\mathrm{i}k'\hat{f}(n,k,k'))_{(n,k,k')\in J_N},\\
\hat{\boldsymbol{f}}_{\theta} &= \boldsymbol{D}_{N,\theta}\hat{\boldsymbol{f}} := (\hat{f}_{\theta}(n,k,k'))_{(n,k)\in J_{N+1}},
\end{aligned}$$

where

$$\hat{f}_{\theta}(n,k,k') := a_{n-1}^{k,k'}\hat{f}(n-1,k,k') - b_n^{k,k'}\hat{f}(n,k,k') - c_{n+1}^{k,k'}\hat{f}(n+1,k,k'), \qquad (n,k,k') \in J_{N+1},$$

with $\hat{f}_{n,k,k'} := 0$ for $|k| > n$ or $|k'| > n$ or $n < 0$ and

$$\begin{aligned}
a_n^{k,k'} &:= \frac{n}{(n+1)(2n+1)}\sqrt{((n+1)^2-k^2)((n+1)^2-k'^2)}, \quad (n,k,k') \in J_N,\\
b_n^{k,k'} &:= \frac{kk'}{n(n+1)}, \quad c_n^{k,k'} = \frac{n}{n(2n+1)}\sqrt{(n^2-k^2)(n^2-k'^2)}, \quad (n,k,k') \in J_N \setminus \{(0,0,0)\}
\end{aligned}$$

with $a_{-1,0,0} := b_{0,0,0} := c_{0,0,0} := 0$.

Proof. As in Lemma 4.2 the assertion follows from recurrence relations of the Wigner-D functions, cf. [35, Eq. (1), (8) and (9), p. 94],

$$\begin{aligned}
\frac{\partial}{\partial\varphi_1} D_n^{k,k'}(\boldsymbol{R}(\varphi_1,\theta,\varphi_2)) &= -ikD_n^{k,k'}(\boldsymbol{R}(\varphi_1,\theta,\varphi_2)) \\
\frac{\partial}{\partial\varphi_2} D_{k,k'}^{n}(\boldsymbol{R}(\varphi_1,\theta,\varphi_2)) &= -ik'D_n^{k,k'}(\boldsymbol{R}(\varphi_2,\theta,\varphi_2)) \\
\sin\theta\frac{\partial}{\partial\theta} D_n^{k,k'}(\boldsymbol{R}(\varphi_1,\theta,\varphi_2)) &= \\
a_n^{k,k'} D_{n+1}^{k,k'}(\boldsymbol{R}(\varphi_2,\theta,\varphi_2)) - b_n^{k,k'} D_n^{k,k'}(\boldsymbol{R}(\varphi_2,\theta,\varphi_2)) &- c_n^{k,k'} D_{n-1}^{k,k'}(\boldsymbol{R}(\varphi_2,\theta,\varphi_2)).
\end{aligned}$$

■

Lemma 5.2. *Let $f\colon \mathrm{SO}(3)\to\mathbb{C}$ be a polynomial on the rotation group of degree $N\in\mathbb{N}$ with Fourier coefficients $\hat{\boldsymbol{f}} := (\hat{f}(n,k,k'))_{(n,k,k')\in J_N}$. Then its second order partial derivatives can be represented as polynomials*

$$\frac{\partial^2}{\partial\varphi_i\partial\varphi_j} f(\boldsymbol{R}(\varphi_1,\theta,\varphi_2)) = \sum_{(n,k,k')\in J_N} f_{\varphi_i,\varphi_j}(n,k,k') D_n^{k,k'}(\boldsymbol{R}), \quad \boldsymbol{R}\in\mathrm{SO}(3), \qquad i,j=1,2,$$

$$\sin\theta\frac{\partial^2}{\partial\theta\partial\varphi_i} f(\boldsymbol{R}(\varphi_1,\theta,\varphi_2)) = \sum_{(n,k,k')\in J_N} f_{\theta,\varphi_i}(n,k,k') D_n^{k,k'}(\boldsymbol{R}), \quad \boldsymbol{R}\in\mathrm{SO}(3), \qquad i=1,2,$$

and

$$\begin{aligned}
\sin^2\theta\frac{\partial^2}{\partial\theta\partial\theta} f(\boldsymbol{R}(\varphi_1,\theta,\varphi_2)) = &\sum_{(n,k,k')\in J_{N+2}} f_{\theta,\theta}(n,k,k') D_n^{k,k'}(\boldsymbol{R}) \\
&- \cos\theta \sum_{(n,k,k')\in J_{N+1}} f_{\theta}(n,k,k') D_n^{k,k'}(\boldsymbol{R}), \quad \boldsymbol{R}\in\mathrm{SO}(3),
\end{aligned}$$

are polynomials of degree N, $N+1$ and $N+2$, respectively. Their Fourier coefficients are given accordingly to Lemma 5.1 by

$$\begin{aligned}
\hat{\boldsymbol{f}}_{\varphi_i,\varphi_j} &:= (\hat{f}_{\varphi_i,\varphi_j}(n,k,k'))_{(n,k,k')\in J_N} = \boldsymbol{D}_{N,\varphi_i}\boldsymbol{D}_{N,\varphi_j}\hat{\boldsymbol{f}} \in \mathbb{C}^{|J_N|}, \qquad i,j=1,2, \\
\hat{\boldsymbol{f}}_{\theta,\varphi_i} &:= (\hat{f}_{\theta,\varphi_i}(n,k,k'))_{(n,k,k')\in J_{N+1}} = \boldsymbol{D}_{N,\theta}\boldsymbol{D}_{N,\varphi_i}\hat{\boldsymbol{f}} \in \mathbb{C}^{|J_{N+1}|} \qquad i=1,2,
\end{aligned}$$

and

$$\begin{aligned}
\hat{\boldsymbol{f}}_{\theta,\theta} &:= (\hat{f}_{\theta,\theta}(n,k,k'))_{(n,k,k')\in J_{N+2}} = \boldsymbol{D}_{N+1,\theta}\boldsymbol{D}_{N,\theta}\hat{\boldsymbol{f}} \in \mathbb{C}^{|J_{N+2}|}, \\
\hat{\boldsymbol{f}}_{\theta} &:= (\hat{f}_{\theta}(n,k,k'))_{(n,k,k')\in J_{N+1}} = \boldsymbol{D}_{N,\theta}\hat{\boldsymbol{f}} \in \mathbb{C}^{|J_{N+1}|}.
\end{aligned}$$

Proof. The assertions follow from Lemma 5.1 similarly as in Lemma 4.2. ■

Theorem 5.3. *Let $f\colon \mathrm{SO}(3) \to \mathbb{R}$ be a real-valued polynomial on the rotation group of degree $N \in \mathbb{N}_0$ and let $\boldsymbol{x}_1, \dots, \boldsymbol{x}_M \in \mathrm{SO}(3) \setminus \{\boldsymbol{R}(\varphi_1, \theta, \varphi_2) \in \mathrm{SO}(3) \ : \ \theta = 0, \pi;\ \varphi_1, \varphi_2 \in [0, 2\pi)\}$ be some nodes. Then the gradients $\nabla_{\mathrm{SO}(3)} f(\boldsymbol{\xi}_i)$ and the Hessian matrices $\mathbf{H}_{\mathrm{SO}(3)} f(\boldsymbol{x}_j)$, $j = 1, \dots, M$, , and therefore, one step of the simultaneous CG method, cf. Algorithm 2, can be computed numerically with error ε and arithmetic complexity $\mathcal{O}(N^3 \log^2 N + M \log^2 \varepsilon^{-1})$.*

Proof. The proof is similar to that of Theorem 4.3 on the sphere $\mathbb{S}^2$, where we note that the evaluation of the exponential map in (5.3) for the description of the geodesics can be evaluated in $\mathcal{O}(M)$ arithmetic operations by performing e.g. a singular value decomposition. ■

A real world application - Detecting preferred orientations in polycrystalline materials. Polynomials on the rotation group are frequently applied in fabric analysis of polycrystalline materials. In this setting a function $f\colon \mathrm{SO}(3) \to \mathbb{R}$, the so called orientation distribution function (ODF), is used to model the relative volume of crystal orientations within a specimen [8]. There are two methods for determining this function experimentally.

The first method uses X–ray, neutron or synchrotron diffraction to measure pole figures of the specimen which are essentially given by the SO(3) Radon transform of the ODF f. Thus the determination of f requires the solution of an inverse problem [26]. As the Radon transform on the rotation group allows for a similar singular value decomposition as the spherical Radon transform [20] there are many algorithms for the solution of the inverse problem, cf.[9, 20, 5, 19], that uses polynomials on the rotation group as approximation of the ODF.

The second method for the determination of the ODF of a specimen uses electron back scatter diffraction (EBSD) to measure the orientation of grains at certain locations at the surface of the specimen directly. These orientations can be interpreted as a random sample of the underlying orientation distribution function. In order to recover the ODF from such data kernel density estimation is used, cf. [18]. The corresponding algorithms utilize the NSOFT and, hence, end up with polynomials on the rotation group.

Figure 5.1 shows a typical example of an ODF. It was computed by kernel density estimation from an EBSD data set of Forsterite crystals [1] as an polynomial of degree $N = 32$. Note that the space of polynomials of degree 32 on the rotation group has dimension $47,905$. Applying the simultaneous CG method, cf. Algorithm 2, with $M = 50,000$ randomly chosen starting points to the negated ODF we obtain more than 500 different local maximizers. In Figure 5.1 we plot the largest 21 local maximizers, which have function values greater than one.

References

[1] A. T. A., A. Vauchez, and D. Ionov. Deformation, static recrystallization, and reactive melt transport in shallow subcontinental mantle xenoliths. *Earth Planet. Sci. Lett*, 272:65–77, 2008.

[2] M. Abramowitz and I. A. Stegun, editors. *Handbook of Mathematical Functions*. National Bureau of Standards, Washington, DC, USA, 1972.

[3] P.-A. Absil, C. G. Baker, and K. A. Gallivan. Trust-region methods on Riemannian manifolds. *Found. Comput. Math.*, 7:303–330, 2007.

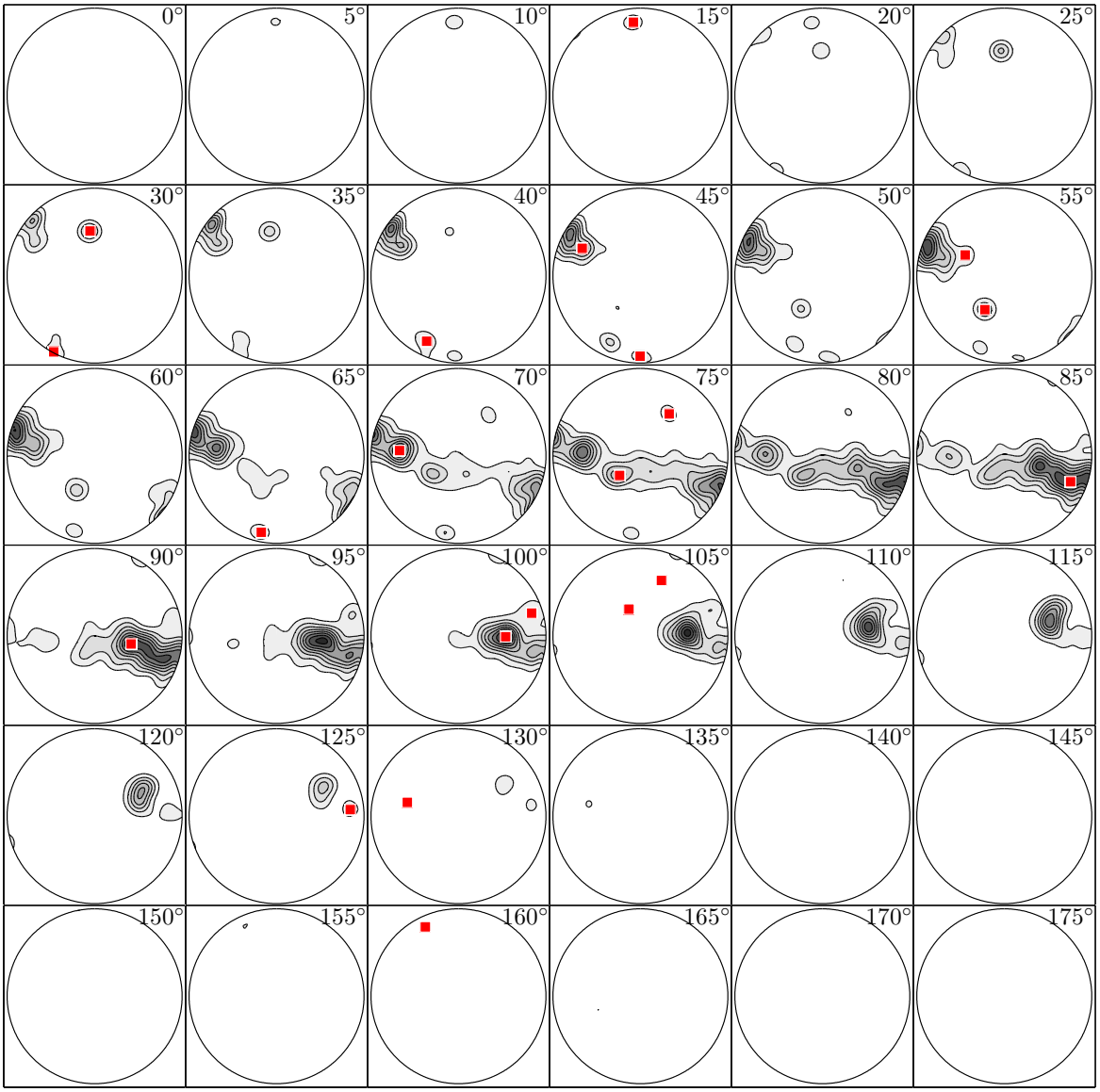

Figure 5.1: An orientation distribution function (ODF) $f\colon \mathrm{SO}(3) \to \mathbb{R}$ plotted as sections with constant difference between the first and the second Euler angle $\varphi_1 - \varphi_2 = 0°, 7.5°, \ldots, 172.5°$ and with $(\theta, \varphi_1 - \varphi_2)$ as polar coordinates. The 21 largest local minimizers detected by the simultaneous CG method are marked as red squares. Note that not every local extrema in a section is a local extrema of the function.

[4] B. L. Adams, S. I. Wright, and K. Kunze. Orientation imaging: The emergence of a new microscopy. *Journal Metallurgical and Materials Transactions A*, 24:819 – 831, 1993.

[5] F. Bachmann, R. Hielscher, and H. Schaeben. Texture analysis with MTEX- free and open source software toolbox. *Diffusion and Defect Data Pt.B: Solid State Phenomena*, 160:63 – 68, 2010.

[6] C. Bajaj, B. Bauer, R. Bettadapura, and A. Vollrath. Nonuniform Fourier transforms for rigid-body and multidimensional rotational correlations. *SIAM J. Sci. Comput.*, 35(4):B821 – B845, 2013.

[7] F. Bornemann, D. Laurie, S. Wagon, and J. Waldvogel. *The SIAM 100-digit challeng: A study in high-accuracy numerical computinge*. SIAM, Philadelphia, 2004.

[8] H. J. Bunge. *Texture Analysis in Material Science*. Butterworths, 1982.

[9] H. J. Bunge and C. Esling. The harmonic method. In H. R. Wenk, editor, *Prefered Orientations in Deformed Metals and Rocks - An Introduction to Modern Texture Analysis*, pages 109 – 122. Academic Press, Orlando, 1985.

[10] F. Dai and Y. Xu. *Approximation Theory and Harmonic Analysis on Spheres and Balls*. Springer Link, 2013.

[11] J. W. Daniel. The conjugate gradient method for linear and nonlinear operator equations. *SIAM J. Numer. Anal.*, 4:10 – 26, 1967.

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

[13] M. Gräf. Quadrature rules on manifolds. `http://www.tu-chemnitz.de/~potts/workgroup/graef/quadrature`, 2013.

[14] 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.

[15] W. W. Hager and H. Zhang. A survey of nonlinear conjugate gradient methods. *Pac. J. Optim.*, 2:35 – 58, 2006.

[16] P. Heilmann, W. Clark, and D. Rigney. Computerized method to determine crystal orientations from Kikuchi patterns. *Ultramicroscopy*, 9:365 – 371, 1982.

[17] S. Helgason. *Differential Geometry, Lie Groups, and Symmetric Spaces*, volume 80 of *Pure and Applied Mathematics*. Academic Press, Boston, 6th edition, 1993.

[18] R. Hielscher. Kernel density estimation on the rotation group and its application to crystallographic texture analysis. *J. Multivariate Anal.*, 119:119 – 143, 2013.

[19] R. Hielscher. Numerical inversion of the Funk transform on the rotation group. *Inverse Problems*, 29:125014, 2013.

[20] R. Hielscher, D. Potts, J. Prestin, H. Schaeben, and M. Schmalz. The Radon transform on $SO(3)$: A Fourier slice theorem and numerical inversion. *Inverse Problems*, 24:025011, 2008.

[21] R. Hielscher and H. Schaeben. A novel pole figure inversion method: specification of the *MTEX* algorithm. *Journal of Applied Crystallography*, 41(6):1024–1037, 2008.

[22] 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.

[23] 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.

[24] J. Keiner and A. Vollrath. A New Algorithm for the Nonequispaced Fast Fourier Transform on the Rotation Group. *SIAM J. Sci. Comput.*, 2012. to appear.

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

[26] S. Matthies, G. Vinel, and K. Helmig. *Standard Distributions in Texture Analysis*, volume 1. Akademie-Verlag Berlin, 1987.

[27] C. Maurice and R. Fortunier. A 3D Hough transform for indexing EBSD and Kossel patterns. *J. Microscopy*, 230:520–529, 2008.

[28] J. Nocedal and S. J. Wright. *Numerical optimization*. Springer Series in Operations Research and Financial Engineering. Springer, New York, second edition, 2006.

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

[30] D. Potts, G. Steidl, and M. Tasche. Fast Fourier transforms for nonequispaced data: A tutorial. In J. J. Benedetto and P. J. S. G. Ferreira, editors, *Modern Sampling Theory: Mathematics and Applications*, pages 247 – 270, Boston, MA, USA, 2001. Birkhäuser.

[31] S. T. Smith. Optimization techniques on Riemannian manifolds. In *Hamiltonian and gradient flows, algorithms and control*, volume 3 of *Fields Inst. Commun.*, pages 113 – 136. Amer. Math. Soc., Providence, RI, 1994.

[32] R. Swinbank and R. J. Purser. Fibonacci grids: A novel approach to global modelling. *Quarterly Journal of the Royal Meteorological Society*, 132:1769 – 1793, 2006.

[33] A. Townsend and L. N. Trefethen. An extension of chebfun to two dimensions. *SIAM J. Sci. Comp.*, to appear.

[34] C. Udrişte. *Convex functions and optimization methods on Riemannian manifolds*, volume 297 of *Mathematics and its Applications*. Kluwer Academic Publishers Group, Dordrecht, 1994.

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