Sven Weisbrich*, Georgios Malissiovas, and Frank Neitzel

On the fast approximation of point clouds using Chebyshev polynomials

https://doi.org/10.1515/jag-2021-0010 Received February 23, 2021; accepted June 9, 2021

Abstract: Suppose a large and dense point cloud of an object with complex geometry is available that can be approximated by a smooth univariate function. In general, for such point clouds the "best" approximation using the method of least squares is usually hard or sometimes even impossible to compute. In most cases, however, a "nearbest" approximation is just as good as the "best", but usually much easier and faster to calculate. Therefore, a fast approach for the approximation of point clouds using Chebyshev polynomials is described, which is based on an interpolation in the Chebyshev points of the second kind. This allows to calculate the unknown coefficients of the polynomial by means of the Fast Fourier transform (FFT), which can be extremely efficient, especially for high-order polynomials. Thus, the focus of the presented approach is not on sparse point clouds or point clouds which can be approximated by functions with few parameters, but rather on large dense point clouds for whose approximation perhaps even millions of unknown coefficients have to be determined.

Keywords: Chebyshev polynomials, interpolation, least squares approximation, Fast Fourier transform, point cloud

1 Introduction

Modern measuring instruments can record information about the geometry of physical objects and provide the user with discrete points in a Cartesian coordinate system. In geodetic practice, such measurements are mainly carried out with terrestrial laser scanners or photogrammet-

*Corresponding author: Sven Weisbrich, Technische Universität Berlin, Institute of Geodesy and Geoinformation Science, Strasse des 17. Juni 135, 10623 Berlin, Germany, e-mail:

s.weisbrich@tu-berlin.de, ORCID:

https://orcid.org/0000-0001-6786-9008

Georgios Malissiovas, Frank Neitzel, Technische Universität Berlin, Institute of Geodesy and Geoinformation Science, Strasse des 17. Juni 135, 10623 Berlin, Germany, e-mails:

georgios.malissiovas@tu-berlin.de, frank.neitzel@tu-berlin.de, ORCID: https://orcid.org/0000-0002-9160-9019 (G. Malissiovas), https://orcid.org/0000-0001-7241-0656 (F. Neitzel)

ric methods, which are capable of capturing millions of points for the observed object. Often an appropriate curve or surface approximation of the measured object with a continuous function is required, either for further analysis of the measurement results (e.g. deformation analysis) or for the representation of the data in a CAD (computeraided design) model. For objects with complex shapes, freeform curves or surfaces are mainly used for data modelling, usually represented by a linear combination of basis functions, which can be freely chosen by the user depending on the problem at hand. Some sets of basis functions that are widely used and can be easily found in the literature are, for example, B-Splines by Wang [43] or Harmening [21], Chebyshev polynomials by Chang [11], Clenshaw [13] or Smitka [40], Fourier series by Wang [42], radial basis functions by Majdisova [26] or Ohtake [33], spherical harmonics by Gannon [18, 30] or Mousa [30], wavelets by Manson [27] or Zernike polynomials by Ibanez [24]. The variety of different sets of basis functions in combination with the different methods available for calculating the unknown parameters, provides the user with an almost infinite number of possibilities for an approximation of point clouds using freeform curves or surfaces.

In geodetic science, B-Splines are mainly used as basis functions for the approximation of point clouds, given by the recurrence relation introduced by de Boor [15] and Cox [14], while the unknown coefficients of the resulting Spline are usually estimated using the method of least squares. An overview on the approximation of 3D point clouds with freeform curves and surfaces, including Bézier, B-Spline and NURBS is given in the work of Bureick et al. [10] and different mathematical representations of 2D splines and their least squares solution for data approximation are discussed by Ezhov et al. [16]. A general review of point cloud modelling methods for deformation analysis has been presented by Neuner et al. [31] and the quantification of object deformation using B-spline approximation is thoroughly discussed by Harmening [20].

An aspect that has a significant influence on the result of the spline approximation is the optimal knot placement, while different strategies exist. The location can be predefined as proposed in the study of de Boor [15] or Yanagihara and Ohtaki [46], or treated as unknown parameters leading to a nonlinear optimisation problem, which can be solved either directly as a non-linear adjustment problem,

as e.g. shown by Schwetlick and Schütze [39], or by using an additional optimisation algorithm, as e.g. addressed by Galvez et al. [17], Bureick et al. [8, 9] or Schmitt and Neuner [37]. Additionally, several criteria have been presented in the literature for choosing the optimal number of control points, which is known as model selection. In this regard, there are many studies that use the Akaike information criterion [2, 3] or the Bayesian information criterion [38]. A detailed overview of model selection strategies is given by Harmening [20, p. 25 ff.]. Both, the problems of knot placement and model selection can be considered as separate research topics and have a significant impact on the quality of the approximation using B-Splines.

Motivated by these challenges and since these approaches are usually used for small point clouds with only a few unknown parameters to be determined, we present a strategy for the approximation of point clouds using Chebyshev polynomials, which is based on an interpolation in the Chebyshev points of the second kind. Thus, we deliberately avoid the computation of the "best" approximation using the method of least squares in favour of a "near-best" solution, which can be computed much easier and extremely fast using the Fast Fourier transform (FFT). The main focus in this paper is on the curve approximation of large and dense point clouds of objects with complex geometry, consisting of hundreds of millions of points with maybe even millions of unknown parameters to be determined.

Although the basic idea of our presented approach is not new and has been already applied, see e.g. the contributions by Clenshaw [13], Chang [11] or McKinley and Ishihara [29], it has unfortunately received little attention in the current literature. Recent contributions in geodetic science that are known to the authors in this regard are those of Hu et al. [23] and Xi et al. [44], without however exploiting or at least addressing the full potential of this approach. Since we approximate the whole point cloud by a single polynomial, a split of the entire domain, as e.g. required for an approximation with B-splines by choosing an appropriate knot vector, is not needed. In addition, the determination of the optimal polynomial degree is solely based on the unknown coefficients and not, as usually, on the residuals. Therefore, an additional model selection criterion is also not needed.

In this contribution we consider at first only point clouds that represent curves without any kinks, jumps or data gaps, which consequently can be approximated by continuous smooth functions. Since for the determination of the unknown parameters, as well as the optimal polynomial degree, the residuals no longer play an important role, a requirement for orthogonal residuals is not considered or rather explicitly omitted in this paper.

The paper is organized as follows: In Section 2 we present the least squares approximation of functions by a polynomial in the Chebyshev basis and its numerical solution, as it as known in the literature. Afterwards, we adapt the basic methodology of this approach and show in Section 3 how it can be used to efficiently approximate point clouds, which is the main contribution of our work. An example is given in Section 4, while Section 5 briefly summarises the investigations that were carried out and critically reviews the results. An outlook on current research regarding the presented approach concludes this paper.

2 Approximation of functions

At first, we present the approximation of functions by a polynomial in the Chebyshev basis using the method of least squares. Since usually the calculation of the coefficients of the polynomial is very time-consuming and inaccurate, we also present how the coefficients can be calculated fast and efficiently by an interpolation in the Chebyshev points using the Fast Fourier transform (FFT), as it is known in the literature.

2.1 Least squares approximation

The Chebyshev polynomial of the first kind $T_i(x)$ of degree $j \ge 0$ is defined by

$$T_i(x) = \cos(i\arccos(x)), \quad \text{for } x \in [-1, 1],$$

se e. g. Mason [28, p. 2] or Rivlin [36, p. 2]. These polynomials satisfy the three term recurrence relation

$$T_i(x) = 2x T_{i-1}(x) - T_{i-2}(x), \quad \text{for } j \ge 2,$$
 (2)

with the initial conditions

$$T_0(x) = 1$$
 and $T_1(x) = x$. (3)

Let us consider a function $f(x) \in L^2([-1,1])$, the Hilbert space of square integrable functions, which we want to approximate by a polynomial $P_p(x)$ of degree p in the Chebyshev basis. $P_n(x)$ lies in the vector space V spanned by the p + 1 basis functions

$$V = \text{span}\{T_0(x), T_1(x), \dots, T_n(x)\}\tag{4}$$

and is given by the linear combination

$$P_{p}(x) = \sum_{j=0}^{p} c_{j} T_{j}(x)$$

$$= c_{0} T_{0}(x) + c_{1} T_{1}(x) + \dots + c_{p} T_{p}(x),$$
(5)

with the unknown coefficients c_j . In general, f(x) is not within V and is therefore different from $P_n(x)$. The difference between f(x) and P(x) is known as residual function v(x) and can be expressed by

$$f(x) + v(x) = P_n(x). \tag{6}$$

To determine the unknown coefficients c_i of a least squares approximation of f(x), we minimise the length of v(x), given by the L^2 norm

$$\|v(x)\|_2 = \sqrt{\langle v, v \rangle} = \sqrt{\int_{-1}^{1} (v(x))^2 dx}, \tag{7}$$

with $\langle \cdot, \cdot \rangle$ being the inner product of two functions, see e. g. Bronshtein et al. [7, p. 920]. Thus, the objective function reads

$$\Omega(c_0, c_1, \dots, c_n) = \langle v, v \rangle \to \min,$$
 (8)

with

$$v(x) = P_p(x) - f(x) = \sum_{j=0}^{p} c_j T_j(x) - f(x).$$
 (9)

Minimising the length of v(x) is equivalent to demand that v(x) is orthogonal to vector space V, or as Langtangen [25, p. 10] has shown, that v(x) is orthogonal to the p+1 basis functions. This leads to the inner products

$$\langle T_i, \nu \rangle = 0, \tag{10}$$

with i = 0, 1, ..., p. Rearranging (6) and inserting into (10) yields

$$\langle T_i, P_p - f \rangle = 0 \tag{11}$$

and introducing (5) leads to the system of normal equations

$$\langle T_i, \sum_{i=0}^p c_j T_j - f \rangle = 0.$$
 (12)

Since the inner product is bilinear, this equation system results in

$$\sum_{j=0}^{p} c_{j} \langle T_{i}, T_{j} \rangle = \langle T_{i}, f \rangle, \tag{13}$$

for i = 0, 1, ..., p, which can be written in matrix notation as follows

$$\begin{bmatrix}
\langle T_0, T_0 \rangle & \langle T_0, T_1 \rangle & \dots & \langle T_0, T_p \rangle \\
\langle T_1, T_0 \rangle & \langle T_1, T_1 \rangle & \dots & \langle T_1, T_p \rangle \\
\vdots & \vdots & \ddots & \vdots \\
\langle T_p, T_0 \rangle & \langle T_p, T_1 \rangle & \dots & \langle T_p, T_p \rangle
\end{bmatrix}
\begin{bmatrix}
c_0 \\
c_1 \\
\vdots \\
c_p
\end{bmatrix} = \begin{bmatrix}
\langle T_0, f \rangle \\
\langle T_1, f \rangle \\
\vdots \\
\langle T_p, f \rangle
\end{bmatrix}. (14)$$

As Chebyshev polynomials are orthogonal with respect to a weighted inner product

$$\langle T_{i}, T_{j} \rangle = \int_{-1}^{1} \frac{T_{i}(x) \ T_{j}(x)}{\sqrt{1 - x^{2}}} \, dx$$

$$= \begin{cases} \pi, & i = j = 0, \\ \pi/2, & i = j \ge 1, \\ 0, & i \ne j, \end{cases}$$
(15)

the normal equation system (14) simplifies to

$$\begin{bmatrix} \langle T_0, T_0 \rangle & 0 & \dots & 0 \\ 0 & \langle T_1, T_1 \rangle & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & \langle T_p, T_p \rangle \end{bmatrix} \begin{bmatrix} c_0 \\ c_1 \\ \vdots \\ c_p \end{bmatrix} = \begin{bmatrix} \langle T_0, f \rangle \\ \langle T_1, f \rangle \\ \vdots \\ \langle T_p, f \rangle \end{bmatrix}. \quad (16)$$

The normal matrix N is diagonal and the unknown coefficients can be directly determined by

$$c_{j} = \frac{\langle T_{j}, f \rangle}{\langle T_{i}, T_{i} \rangle} \tag{17}$$

with

$$\langle T_j, f \rangle = \int_{-1}^{1} \frac{T_j(x)f(x)}{\sqrt{1-x^2}} \, \mathrm{d}x, \tag{18}$$

for
$$j = 0, 1, ..., p$$
.

The derivation of an analytical solution of the integrals in (17) for arbitrary functions f(x) is hardly feasible, so that a numerical solution is usually used. But, due to singularities of the weighting function in (18) at $x = \pm 1$, the numerical integration of the right hand side is slow and inaccurate for higher polynomial degrees. To derive a very fast and accurate solution for the numerical integration of (18), we present in the following the relationship between the Chebyshev and Fourier series.

2.2 Connection to the Fourier series

Let us introduce the following change of variables

$$x = \cos(\theta) \tag{19}$$

and with

$$\frac{\mathrm{d}x}{\mathrm{d}\theta} = \frac{\mathrm{d}\cos(\theta)}{\mathrm{d}\theta} = -\sin(\theta) \tag{20}$$

we obtain

$$dx = -\sin(\theta) d\theta. \tag{21}$$

Inserting (19) and (21) in the right hand side of (18) and changing the limits according to $\cos(\pi) = -1$ and $\cos(0) = 1$ yields

$$\int_{-1}^{1} \frac{f(x) T_{j}(x)}{\sqrt{1 - x^{2}}} dx = \int_{-1}^{1} \frac{f(x) \cos(j \arccos(x))}{\sqrt{1 - x^{2}}} dx$$

$$= -\int_{\pi}^{0} \frac{f(\cos(\theta)) \cos(j\theta)}{\sqrt{1 - \cos^{2}(\theta)}} \sin(\theta) d\theta$$

$$= -\int_{\pi}^{0} \frac{f(\cos(\theta)) \cos(j\theta)}{\sin(\theta)} \sin(\theta) d\theta$$

$$= \int_{0}^{\pi} f(\cos(\theta)) \cos(j\theta) d\theta$$

$$= \int_{0}^{\pi} f(\cos(\theta)) \cos(j\theta) d\theta.$$
(22)

Hence, the unknown coefficients of a least squares approximation of f(x) in the Chebyshev basis can be determined by

$$c_{j} = \frac{1}{\langle T_{j}, T_{j} \rangle} \int_{0}^{\pi} F(\theta) \cos(j \theta) d\theta.$$
 (23)

The presented change of variables can also be interpreted as a mapping of f(x) onto the upper half of the unit circle, as depicted in Figure 1. An arbitrary point z on the unit circle

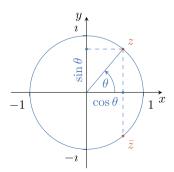


Figure 1: Unit circle in the complex plane with the imaginary unit $\iota = \sqrt{-1}$.

cle is uniquely defined by its coordinates (x, y), satisfying

$$x^2 + y^2 = 1, (24)$$

or by an angle θ , with

Re(z) =
$$x = \cos \theta$$
,
Im(z) = $y = \sin \theta$, (25)

while the upper equation in (25) is the introduced change of variables (19). Due to

$$x = \text{Re}(z) = \text{Re}(\bar{z}) \tag{26}$$

it follows

$$F(\theta) = F(-\theta) \tag{27}$$

and the integral in (23) can also be written as

$$c_{j} = \frac{1}{2} \frac{1}{\langle T_{j}, T_{j} \rangle} \int_{-\pi}^{\pi} F(\theta) \cos(j \theta) d\theta,$$
 (28)

which corresponds to a scaled Fourier cosine transform. While mapping f(x) onto the unit circle we obtain a transplanted function $F(\theta)$ for $\theta \in [-\pi, \pi]$ which is an infinitely differentiable, even and periodic function, as discussed by Orszag [34]. Therefore, $F(\theta)$ has a Fourier cosine series expansion

$$F(\theta) = \frac{1}{2}a_0 + \sum_{j=1}^{\infty} a_j \cos(j\omega\theta), \tag{29}$$

where the coefficients a_i are given by

$$a_{j} = \frac{1}{\pi} \int_{-\pi}^{\pi} F(\theta) \cos(j \, \theta) \, \mathrm{d}\theta \tag{30}$$

and which are equivalent to the coefficients c_j in equation (28) except for a scaling factor, see the handbook by Bronshtein et al. [7, p. 927]. Based on this fact, a fast and accurate numerical solution for the integrals in (28) is presented in the next section and the resulting coefficients of a polynomial approximation in the Chebyshev basis are discussed in more detail on an example.

2.3 Numerical solution by an interpolation in the Chebyshev points

In case $F(\theta)$ in (30) is known at N+1 equally spaced points θ_k , with $k=0,1,2,\ldots,N$, the integral in (30) can be determined approximately by

$$a_j \approx \frac{2}{N} \sum_{k=1}^{N} F(\theta_k) \cos(j \theta_k),$$
 (31)

for j = 0, 1, ..., N and, hence, can be efficiently computed by the Fast Fourier Transform (FFT) or Fast Cosine Transform, see the handbook by Bronshtein et al. [7, p. 928]. According to Trefethen [41, p. 15] this has been probably observed around 1970, see for example the investigations of Ahmed [1] or Orszag [34].

Let us consider some equally spaced points on the upper half of the unit circle, including the boundary points on the intersection with the real axis at ± 1 , as depicted in Figure 2. The projection of these equally spaced points

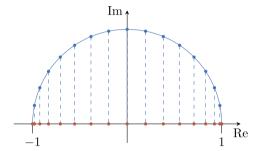


Figure 2: Equispaced points on the upper half on the unit circle in blue and Chebyshev points in red.

onto the real axis are known as *Chebyshev points of the sec*ond kind, *Chebyshev extreme points* or *Chebyshev-Lobatto* points and are given by

$$x_k = \cos\left(\frac{k\pi}{N}\right), \quad 0 \le k \le N.$$
 (32)

In the following we just call them Chebyshev points. These points are the extrema of the Chebyshev polynomials of the first kind, in contrast to the Chebyshev points of the first kind, which are the roots. How the Chebyshev points of the first kind can be used for the approximation of functions will not be considered here any further, but is discussed e. g. by Xu [45].

If we discretize f(x) at the Chebyshev points we obtain

$$f(x_k) = F(\theta_k), \tag{33}$$

which can directly be used to calculate the unknown coefficients c_j using FFT. Thereby the solution of the integrals in equation (18) are only approximated and the calculated coefficients c_j correspond to those of an interpolation in the Chebyshev points.

Although in theory we approximate functions by the method of least squares, this approach is unsuitable in practice as the integrals in equation (17) can be in general solved only by numerical integration, which is in this case

slow and inaccurate. It is rather more efficient to approximate functions by interpolation in the Chebyshev points, where the coefficients calculated by FFT converge to the values of the exact solutions for equation (17) as the number of points increases.

To demonstrate such an approximation, we consider the following function

$$f(x) = \frac{1}{2} \tanh(5x - 0.2) + \frac{1}{2},$$
 (34)

for $x \in [-1, 1]$, which is depicted in Figure 3.

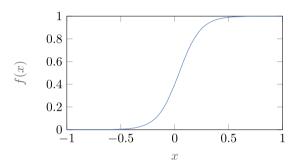


Figure 3: Function $f(x) = \frac{1}{2} \tanh(5x - 0.2) + \frac{1}{2}$.

To show a very important property of an approximation in the Chebyshev basis, we approximate f(x) by a polynomial of degree p=200. The coefficients c_j are depicted in Figure 4 and decrease geometrically until a plateau of rounding errors near level of machine precision is reached for $j \geq 117$ and a floating-point relative accuracy of $\varepsilon = 2.22 \cdot 10^{-16}$ (vertical red line). As long as f(x) is continuous

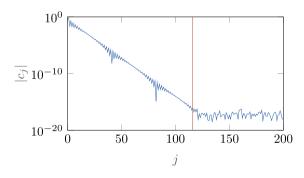


Figure 4: Logarithmic plot of the coefficients $|c_i|$.

and satisfies the Dini-Lipschitz condition, the Chebyshev series expansion will converge to f(x), see the textbook by Mason and Handscomb [28, Chapter 5.3.2]. As a rule of thumb, the truncation error of this approximation is in the same order-of-magnitude as the absolute value of the

last coefficient, as mentioned by Boyd [6, p. 51]. Aurentz [4] designed a chopping algorithm that detects and chops the series at the beginning of this plateau. Consequently, for this example a polynomial of degree p=116 is found as the best approximation of f(x). The resulting residual function v(x) is shown in Figure 5 and reveals, as expected, differences near machine precision. As illustrated by this

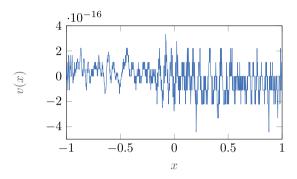


Figure 5: Residual function $v(x) = P_{116}(x) - f(x)$.

example, a Lipschitz continuous function on [-1,1] has a unique Chebyshev series, which is absolutely and uniformly convergent, see Trefethen [41, Theorem 3.1]. It has been shown, that the coefficients c_j in equation (17) can be calculated very fast and accurately by an interpolation in the Chebyshev points using FFT. Furthermore, the optimal polynomial degree of the approximation can be identified by an iterative increase until the coefficients reach the plateau of rounding errors near level of machine precision, which can then be detected and truncated. This provides an efficient and accurate approach for the approximation of functions and is the core of the open-source package *Chebfun* [12], which was introduced in 2004 by Battles and Trefethen [5].

3 Approximation of point clouds

Since we can consider functions as a special case of a point cloud, i. e. a point cloud with an infinite number of errorfree points, we can directly adopt the presented procedure for the approximation of point clouds. Furthermore, we only consider point clouds on the interval [-1,1] since any interval [a,b] can be easily scaled to [-1,1].

At first, let us generate a point cloud by discretising function (34) at n + 1 linearly spaced points (x_i, y_i) for i = 0, 1, ..., n. Now, the actual problem concerning the discretised function arises from the fact, that there are usually no more functional values $f(x_k)$ at the Chebyshev points

 x_k available. Therefore, we have to extract or interpolate the corresponding values for the Chebyshev points from the nearest surrounding points or a cluster of neighboured points.

As long as the true values for $f(x_k)$ at the Chebyshev points are not given any more, the coefficients will not decrease until they reach a plateau of rounding errors near level of machine precision. In this case, a plateau will be formed at a level $\gg \epsilon$ that depends essentially on the density of the point cloud and the amount of noise. Consequently, the task is to identify this plateau and its level and to truncate the coefficients at its beginning using the chopping algorithm of Aurentz and Trefethen [4]. These are in principle the main differences to the approximation of functions using Chebfun [12], whereas the functional values $f(x_k)$ and the level of the plateau are directly given. Hence, Algorithm 1 describes a simple and fast approach for the approximation of point clouds using Chebyshev polynomials.

Algorithm 1 Pseudo code for the calculation of the coefficients c_j for a polynomial approximation of the point cloud x_i, y_j .

```
1: function APPROXPOINTCLOUD(x_i, y_i)
 2:
          m = 6
                                                    \triangleright Initialisation for p.
 3:
          while Plateau is not reached do
 4:
                                                   ⊳ Polynomial degree.
              x_k = \cos\left(\frac{k\pi}{p}\right) \quad \forall k = 0, 1, \dots, p
 5:
              f(x_k) \leftarrow \text{by e. g. linear interpolation}
 6:
 7:
              F(\theta_k) = f(x_k)
               a_i = \text{Re}(\text{fft}(F(\theta_k)))
 8:
               c_i^* \leftarrow \text{by rescaling } a_i
 9:
               if Plateau is reached then
10:
                    c_i \leftarrow \text{by truncating } c_i^*
11:
               else
12:
13:
                   m = m + 1
               end if
14:
15:
          end while
16:
          return c;
17: end function
```

Since we approximate the point cloud by an interpolation in the Chebyshev points x_k , the accuracy of the approximation depends directly on the accuracy of their functional values $f(x_k)$. Besides the shape of the underlying unknown function, the density of the point cloud and the amount of noise mainly influences the accuracy of $f(x_k)$ and consequently of the approximation itself. Both aspects will be addressed in more detail in the following sections.

3.1 Impact of point density

At first, we would like to demonstrate the impact of the point density on the level of the plateau of the coefficients and, thus, on the accuracy of the approximation itself. Therefore, we generate point clouds using function (34) at $n + 1 = \begin{bmatrix} 10^2 & 10^4 & 10^6 \end{bmatrix}$ linearly spaced points and without taking any measurement noise into account. Each point cloud will be approximated by a polynomial of degree p = 200, while the functional values $f(x_k)$ at the Chebyshev points x_k will be determined by a simple linear interpolation. The resulting coefficients for each point cloud are depicted in Figure 6.

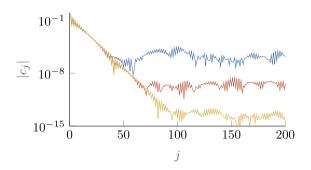


Figure 6: Logarithmic plot of the coefficients $|c_i|$ for 10^2 (blue), 10^4 (red) and 10⁶ (yellow) points without noise.

As already mentioned, and also illustrated by Figure 6, the coefficients decrease until they reach a plateau at a certain level, which is directly related to the point density. This is essential for the presented approach, because the higher the level of the plateau the lower the resulting polynomial degree and consequently, the less accurate the approximation will be.

For this example, the mean level of the plateau for 10² (blue), 10^4 (red) and 10^6 (yellow) points is located at approximately 10^{-6} , 10^{-10} and 10^{-14} , respectively. It can be observed that an increase in the number of points by a factor of 10^2 reduces the level of the plateau by $\approx 10^4$. For a point cloud with 10⁶ points, the resulting coefficients differ to the true values given in Figure 4 by only about $\pm 5 \cdot 10^{-13}$. It should also be noted, that the values for the coefficients forming the plateau do not have the same random characteristics as those in Figure 4, but rather show some systematic variations, which probably result from the calculation of the functional values $f(x_k)$ by a linear interpolation. In general we can conclude that the accuracy of the approximation of point clouds, using the proposed algorithm, increases proportionally with the point density.

3.2 Impact of noise

In this section we illustrate the impact of measurement noise on the level of the plateau. Therefore, we generate a point cloud with 10⁴ linearly spaced points and add normally distributed noise with a standard deviation of $\sigma = [10\% 1\% 0.1\%]$. The resulting coefficients for each point cloud are depicted in Figure 7. First of all, it

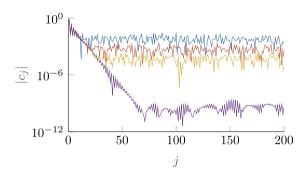


Figure 7: Logarithmic plot of the coefficients $|c_i|$ for 10^4 points with noise of $\sigma = 10\%$ (blue), 1% (red), 0.1% (yellow) and without noise (purple).

is apparent in Figure 7 that the three plateaus are quite close to each other and their levels only differ by a factor of 10 and, therefore, share the same difference as the corresponding standard deviation σ to which they refer. Furthermore, the values of the coefficients forming each plateau have, in this example, a rather random characteristic. In a direct comparison with the coefficients of the point cloud without noise (purple), it is also noticeable that noise has a considerable impact on the level of the plateau. Even if only low noise with $\sigma = 0.1\%$ is present, the level of the plateau already increases from 10⁻¹⁰ to $\approx 10^{-4}$. A part of this increase is certainly due to the choice of calculating the functional values $f(x_k)$ by a simple linear interpolation. Thus, the result could be significantly improved by using better methods than a linear interpolation. However, the measurement noise will always have a considerable impact on the level of the plateau and consequently on the accuracy of the approximation.

As we have shown, the accuracy of an approximation of point clouds by an interpolation in the Chebyshev points is directly related to the point density and measurement noise. In general we can state, that especially for sparse point clouds or point clouds of simple geometries, which can be approximated by low-order polynomials, the presented approach is not the best choice. In such cases it is preferable to estimate the polynomial coefficients using the method of least squares and determine the optimal polynomial degree iteratively using a suitable model selection criterion, as for instance discussed by Harmening and Neuner [22].

However, the presented approach is particularly suitable for the approximation of dense point clouds of complex geometries, where even polynomials of millionth degree can be usually calculated within a few seconds. So the focus is on point clouds where the number of points is much larger than the degree of the polynomial approximation. This will be illustrated in the following section.

4 Example

To demonstrate the suitability of the presented approach we consider the following example of a profile scan of a bust of C. F. Gauss with 100 000 points, as depicted in Figure 8. The profile scan was extracted from a surface scan of the bust and the number of points was increased afterwards. In addition, the resulting point cloud was smoothed in order to reduce the impact of measurement noise as much as possible. The derived profile scan is considered as a point cloud without measurement noise and is used as a reference for the approximation in this example.

To analyse the accuracy of the approximation using our approach, we generate a synthetic point cloud by adding normally distributed noise with a standard deviation of $\sigma_x = \sigma_y = 0.5\,\mathrm{mm}$ to the true coordinates of the points of the reference profile scan. To approximate such a point cloud, we consider the parametric representation of

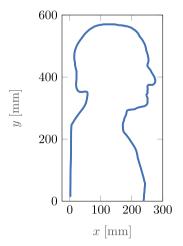


Figure 8: Reference profile scan of a bust of C. F. Gauss with 100 000 points.

the *x* and *y* coordinates in the form

$$x = P_{x,p_x}(u) = \sum_{j=0}^{p_x} c_{x_j} T_j(u),$$

$$y = P_{y,p_y}(u) = \sum_{j=0}^{p_y} c_{y_j} T_j(u),$$
(35)

with the new parameter $u \in [-1, 1]$. In general, there is no corresponding value u_i for each point (x_i, y_i) available. Therefore, three different methods can be found in the literature for calculating u_i for each point, namely "equally spaced", "chord length" and "centripetal", see e.g. the textbook by Piegl and Tiller [35, p. 364 ff.]. In this example, we choose equally spaced values for u_i . Since in this case we determine the functional values at the Chebyshev points by a simple linear interpolation, the stochastic properties of the observations (x_i, y_i) will have no influence on the determination of these values. Therefore, we can assume equally weighted and uncorrelated observations (x_i, y_i) and calculate the unknown parameters c_{x_i} and $c_{\rm v}$ independently. This results in the two point clouds with measurement noise depicted in Figure 9, which will be approximated in the following sections using both our proposed approach and the method of least squares.

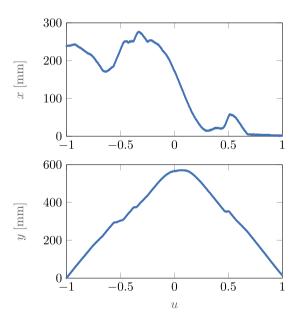


Figure 9: Resulting two point clouds for the x and y coordinates with measurement noise of $\sigma_x = \sigma_y = 0.5$ mm and 100 000 points.

As already mentioned, we have decided, for reasons of simplicity, to determine the functional values at the Chebyshev points by a linear interpolation. To obtain more precise results and also to take the stochastic properties of

the observations (x_i, y_i) into account, the function values at the Chebyshev points could be estimated by e.g. local regression. However, this is out of the scope of this contribution.

4.1 Approximation by interpolation

At first, we approximate the two point clouds from Figure 9 by an interpolation in the Chebyshev points according to the proposed approach described in Algorithm 1. As a solution we obtain a curve which is described by two polynomials of degree $p_x = 528$, $p_y = 422$ and which can not be distinguished visually from the two point clouds in Figure 9. Therefore, we first of all will have a look at the resulting residuals $v_{x_i} = P_{x,528}(u_i) - x_i$ and $v_{v_i} = P_{v,422}(u_i) - y_i$, which in this case are shown as a relative frequency histogram in Figure 10. The two histograms are just slightly

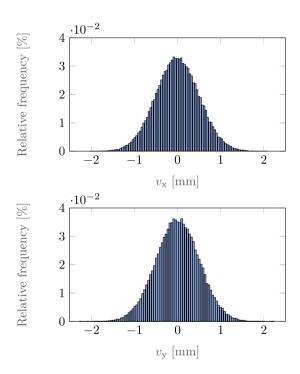


Figure 10: Relative frequency histogram of the resulting residuals v_{x_i} and v_{v_i} for the approximation of the two point clouds with measurement noise of $\sigma_{\rm x}=\sigma_{\rm y}=0.5$ mm and 100 000 points.

different with maximum values of max $|v_{x_i}| \approx 2.07$ mm and $\max |v_{v_s}| \approx 2.26$ mm. The distribution of both sets of residuals correspond very well to a normal distribution with an expectation nearly zero and a standard deviation of $s_x = 0.5021 \,\text{mm}$ and $s_v = 0.5022 \,\text{mm}$ for v_{x_i} and v_{v_i} , respectively. The residuals of both approximations therefore correspond to the previously added normally distributed noise with a standard deviation of $\sigma_x = \sigma_y = 0.5$ mm.

To analyse the accuracy of the approximation in more detail, the deviations Δ_{x_i} and Δ_{v_i} , which result as the difference between the polynomials and the point cloud of the reference profile scan without measurement noise from Figure 8, are shown in Figure 11. Due to the calculation

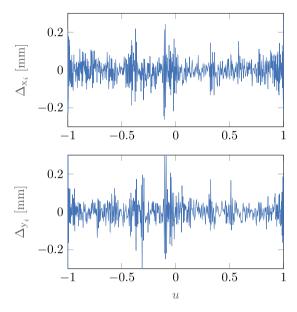


Figure 11: Deviations Δ_{x_i} and Δ_{y_i} between the resulting polynomials and the point cloud of the reference profile scan without measurement noise from Figure 8. For a better visualisation of the deviations, the value range was limited to ± 0.3 mm. Therefore, the largest deviations at the limits at $u \approx \pm 1$ are not visible.

of the functional values $f(x_k)$ at the Chebyshev points by a simple linear interpolation, the largest deviations occur very close to the limits at $u \approx \pm 1$, with maximum values of max $|\Delta_{x_i}| \approx 0.59$ mm and max $|\Delta_{v_i}| \approx 1.21$ mm. However, for larger polynomial degrees, these areas are negligibly small in relation to the total point cloud and can even be removed afterwards if desired. Alternatively, these deviations at the limits can also be reduced by choosing more precise methods for determining the functional values $f(x_k)$. In addition, larger deviations occur in the range of u = -0.1 in both point clouds, which corresponds to a spot on the forehead of the bust. There is a small dent which could not be approximated properly, due to the large measurement noise. Nevertheless, these deviations are still much smaller than the introduced measurement noise of $\sigma_{\rm x} = \sigma_{\rm y} = 0.5\,{\rm mm}$. Apart from some few exceptions, the deviations for the x and y coordinates are mainly below 0.1 mm, with a resulting standard deviation of $s_{\Delta_v} \approx 0.049$ mm and $s_{\Delta_v} \approx 0.051$ mm for Δ_{x_i} and Δ_{v_i} , respectively.

4.2 Least squares approximation

In this section we will derive the basic formulas to compute the solution for a polynomial approximation using the method of least squares, see e. g. the textbooks by Ghilani and Wolf [19, p. 173 ff.] or Niemeier [32, p. 129 ff.]. Since the derivation of the solution for a least squares approximation is identical for x_i and y_i , we will present it only for x_i . The functional model for x_i for the n+1 points according to equation (35) results in

$$x_{i} = P_{x,p_{x}}(u) = \sum_{j=0}^{p_{x}} c_{x_{j}} T_{j}(u_{i})$$

$$= c_{x_{0}} T_{0}(u_{i}) + c_{x_{1}} T_{1}(u_{i}) + \dots + c_{x_{p_{x}}} T_{p_{x}}(u_{i}),$$
(36)

with the unknown parameters c_{x_j} and for i = 0, 1, ..., n. If we consider x_i as observations and u_i as error-free values and introduce a residual v_{x_i} for each observation, we obtain the following observation equations

$$x_i + v_{x_i} = P_{x,p_x}(u) = \sum_{i=0}^{p_x} c_{x_i} T_j(u_i)$$
 (37)

which can be written in matrix notation as

$$\begin{bmatrix}
x_{0} \\
x_{1} \\
\vdots \\
x_{n}
\end{bmatrix} + \begin{bmatrix}
v_{x_{0}} \\
v_{x_{1}} \\
\vdots \\
v_{x_{n}}
\end{bmatrix} = \begin{bmatrix}
T_{0}(u_{0}) \ T_{1}(u_{0}) \dots T_{p_{x}}(u_{0}) \\
T_{0}(u_{1}) \ T_{1}(u_{1}) \dots T_{p_{x}}(u_{1}) \\
\vdots & \vdots & \ddots & \vdots \\
T_{0}(u_{n}) \ T_{1}(u_{n}) \dots T_{p_{x}}(u_{n})
\end{bmatrix} \begin{bmatrix}
c_{0} \\
c_{1} \\
\vdots \\
c_{p_{x}}
\end{bmatrix}.$$

$$\mathbf{A} \qquad \mathbf{C}_{x}$$
(38)

Assuming equally weighted and uncorrelated observations, the weight matrix **P** results in an identity matrix

$$\mathbf{P} = \mathbf{I}_{n+1},\tag{39}$$

which represents the stochastic model. To determine the unknown parameters c_{x_j} of a least squares approximation, we minimise the sum of weighted squared residuals, resulting in the so-called normal equation system

$$\mathbf{A}^{\mathsf{T}}\mathbf{P}\mathbf{A}\mathbf{c}_{\mathsf{x}} = \mathbf{A}^{\mathsf{T}}\mathbf{P}\mathbf{I}_{\mathsf{x}},\tag{40}$$

with the normal matrix

$$\mathbf{N} = \mathbf{A}^{\mathsf{T}} \mathbf{P} \mathbf{A} \tag{41}$$

and the vector of the right hand side

$$\mathbf{n}_{\mathbf{x}} = \mathbf{A}^{\mathsf{T}} \mathbf{P} \mathbf{l}_{\mathbf{x}}.\tag{42}$$

Solving (40) for \mathbf{c}_{x} yields the unknown coefficients $c_{x_{j}}$ of a least squares approximation. In contrast to the resulting normal matrix for the least squares approximation of

functions in equation (16), the normal matrix in (41) is a dense matrix. Therefore, the normal equation system (40) can not be solved as efficiently as (16).

For the two point clouds in Figure 9 we determine the polynomial approximation by solving (40) with a polynomial degree of $p_x=528$ and $p_y=422$. In principle, the same distribution of the residuals results for both polynomials with almost the same properties as already obtained for the approximation by interpolation from Figure 10. Also the resulting deviations hardly differ visually from those in Figure 11, with just slightly smaller values at the limits of max $|\Delta_{\mathbf{x}_i}| \approx 0.44$ mm and max $|\Delta_{\mathbf{y}_i}| \approx 0.97$ mm.

Accordingly, the polynomial approximation by the method of least squares seems to correspond to the one by an interpolation in the Chebyshev points from the previous section, which will be analysed in more detail in the following.

4.3 Comparison

At first, we will compare the solution of the polynomial coefficients obtained by an interpolation in the Chebyshev points from Section 4.1 and by the method of least squares from Section 4.2. The absolute differences between these two sets of coefficients for the polynomials $P_{x,528}(u)$ and $P_{y,422}(u)$ are shown in Figure 12. In general, small differences have been observed for most of the coefficients, that range below 10^{-3} mm, whereby the values for $P_{y,422}(u)$ (red) are a bit smaller than for $P_{x,528}(u)$ (blue). Moreover, both graphs show a similar trend, with an exponential increase of the difference for larger j. Therefore, the maximum absolute difference of $\approx 7 \cdot 10^{-3}$ mm occurs at the last coefficient for both polynomials, which is still nearly 100 times smaller than the added measurement noise of

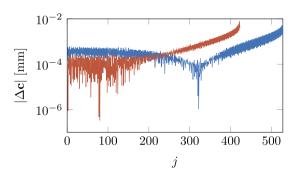


Figure 12: Logarithmic plot of the absolute difference between the polynomial coefficients obtained by an interpolation in the Chebyshev points and by the method of least squares for $P_{x,528}(u)$ in blue and $P_{y,422}(u)$ in red.

 $\sigma_{\rm x}=\sigma_{\rm y}=0.5$ mm. Consequently, the two sets of polynomials only differ slightly from each other, as can be seen in Figure 13.

As in the previous sections, the largest deviations of up to 0.25 mm occur at the limits of the two polynomials. However, since both polynomials approximate the true function rather poorly at the limits, compare Figure 11, no conclusion can be drawn at this point about which approximation is better or worse. Additionally, $\sim 99.95\%$ of all deviations are smaller than $5 \cdot 10^{-2}$ mm and even $\sim 85\%$ are smaller than $5 \cdot 10^{-3}$ mm, thus, they are 100 times smaller than the added measurement noise. Whether these very small deviations play a significant role depends, of course, on the problem itself and has to be answered individually. In the present example, however, these deviations are not relevant.

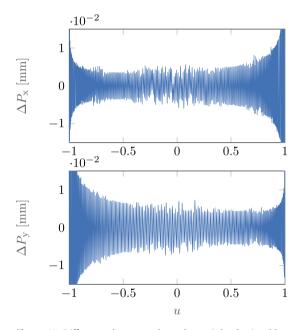


Figure 13: Difference between the polynomials obtained by an interpolation in the Chebyshev points and by the method of least squares. For a better visualisation of the differences, the value range was limited to ± 0.015 mm. Therefore, the largest differences at the limits at $u\approx \pm 1$ are not visible.

Except for the limits, a polynomial approximation of dense point clouds by interpolation in the Chebyshev points or by the method of least squares yields almost the same result. But with the important difference, that the approximation by interpolation can be calculated much faster and also for polynomials of millionth degree, which is hardly possible for an approximation using the method of least squares according to Section 4.2. To clearly emphasise this fact, Table 1 shows the computation time required

Table 1: Computation time required for an approximation by interpolation in the Chebyshev points (IP) and by the method of least squares (LS) of different sized point clouds and polynomial degrees using a budget PC with an Intel Core i5-6400 @ 2.70 GHz. All computation have been performed in Matlab and the normal equation system (40) has been solved using the backslash operator.

			Time	
case	points	polynomial degree	IP	LS
1	10 ³	10 ²	< 1 ms	≈ 1 ms
2	10 ⁴	10 ³	$\approx 2 \text{ ms}$	$\approx 0.5 s$
3	10 ⁵	10 ⁴	\approx 4 ms	\approx 10 min
4	10^{6}	10 ⁵	$\approx 30 ms$	-
5	10 ⁷	10^{6}	$\approx 0.3 \text{ s}$	-
6	10 ⁸	10 ⁷	$\approx 2 \text{ s}$	-

for the approximation of different sized point clouds and polynomial degrees. For small point clouds and low polynomial degrees, there is hardly any significant difference in the computation time between these two approaches, but this changes quickly with increasing number of points and polynomial degrees. For case 3, the computation time using the method of least squares is already ≈ 10 min and, thus, nearly 150 000 times longer than for interpolation. Most of the time is thereby needed to set up the normal matrix (41) and, based on extrapolations, this takes already about 2 weeks for case 4. This is also the reason why in Table 1 the computation time for an approximation using the method of least squares for the last 3 cases could no longer be given. In contrast, even a point cloud with 100 million points can be approximated by a polynomial of 10 millionth degree in less than 2s using the presented approach of an interpolation in the Chebyshev points.

5 Conclusion and outlook

By slightly modifying the general methodology for an approximation of functions by an interpolation in the Chebyshev points, as applied by *Chebfun* [12], the basic idea of an approach was shown that allows even large point clouds with complex geometry to be approximated very fast. The identification of the optimal polynomial degree is based on the determination and truncation of the plateau and is therefore solely based on the computed polynomial coefficients. The impact of the point density and measurement noise on the level of the plateau and consequently on the accuracy of the approximation has been shown in more detail. It has been demonstrated that the measurement noise has the largest impact. The main advantages of the presented approach for the univariate approximation

of point clouds by interpolation in the Chebyshev points in contrast to an approximation using B-Splines can be summarised as follows:

- No knot placement strategy needed.
- No additional model selection criteria needed.
- Extremely fast and efficient.

So far, the main challenge of the presented approach is the determination of the functional values $f(x_k)$ at the Chebyshev points x_k and also the determination of the level of the plateau. Both aspects have a significant impact on the approximation itself. Since we wanted to demonstrate the basic concept of an approximation in the Chebyshev basis by an interpolation in the Chebyshev points in this contribution, we have limited ourselves to the simplest approaches for both aspects. How both aspects can be solved in a smart way is currently being researched.

Moreover, we have neglected the stochastics of the unknown parameters so far, which is, among other aspects, one of the essential result of a least squares approximation. In case the approximation is used e.g. for a geodetic deformation analysis, the stochastic properties of the unknown parameters play an essential role and must be taken into account. Therefore, the determination of the stochastic properties as well as the identification of outliers is currently being investigated in more detail.

Until now, only point clouds that can be described by univariate functions, for example, a trajectory, can be approximated by the presented approach. However, the basic methodology presented in this article, could be directly extended to the approximation of surfaces and is currently being developed.

References

- N. Ahmed and P. S. Fisher. Study of algorithmic properties of Chebyshev coefficients. International Journal of Computer Mathematics, 2(1-4):307-317, 1970.
- [2] H. Akaike. Information theory and an extension of the maximum likelihood principle. In: Proceedings of the Second International Symposium on Information Theory, 2nd-8th September 1971, Tsahkadsor, Armenia, pages 267-281, 1973.
- H. Akaike. A new look at the statistical model identification. IEEE Transactions on Automatic Control, 19(6):716-723, 1974.
- J. L. Aurentz and L. N. Trefethen. Chopping a Chebyshev series. ACM Transactions on Mathematical Software, 43(4):1-21, 2017.
- Z. Battles and L. N. Trefethen. An extension of Matlab to continuous functions and operators. SIAM Journal on Scientific Computing, 25(5):1743-1770, 2004.
- J. P. Boyd. Chebyshev and Fourier Spectral Methods. Dover Books on Mathematics. Dover, Mineola, 2002.

- I. N. Bronshtein, K. A. Semendjajew, G. Musiol, and H. Mühlig. Handbook of Mathematics. Springer, Berlin and New York, 5th ed., 2007.
- [8] J. Bureick, H. Alkhatib, and I. Neumann. Robust spatial approximation of laser scanner point clouds by means of free-form curve approaches in deformation analysis. Journal of Applied Geodesy, 10(1):27-35, 2016.
- J. Bureick, H. Alkhatib, and I. Neumann. Fast converging elitist genetic algorithm for knot adjustment in b-spline curve approximation. Journal of Applied Geodesy, 13(4):317-328,
- [10] J. Bureick, H. Neuner, C. Harmening, and I. Neumann. Curve and surface approximation of 3d point clouds. allaemeine vermessungs-nachrichten (avn), 123:315-327, 2016.
- C. Chang. Discrete-sample curve fitting using Chebyshev polynomials and the approximate determination of optimal trajectories via dynamic programming. IEEE Transactions on Automatic Control, 11(1):116-118, 1966.
- [12] Chebfun. Chebfun - numerical computing with functions. http://www.chebfun.org/, 2021. Last accessed on 16.02.2021.
- [13] C. W. Clenshaw. Curve fitting with a digital computer. The Computer Journal, 2(4):170-173, 1960.
- [14] M. Cox. The numerical evaluation of b-splines. IMA J. Appl. Math., 10(2):134-149, 1972.
- [15] C. de Boor. On calculating with B-splines. J. Approx. Theory, 6(1):50-62, 1972.
- [16] N. Ezhov, F. Neitzel, and S. Petrovic. Spline approximation, part 1: Basic methodology. Journal of Applied Geodesy, 12(2):139-155, 2018.
- [17] A. Gálvez, A. Iglesias, A. Avila, C. Otero, A. Arias, and C. Manchado. Elitist clonal selection algorithm for optimal choice of free knots in b-spline data fitting. Applied Soft Computing, 26:90-106, 2015.
- [18] C. Gannon and R. Liang. Using spherical harmonics to describe large-angle freeform lenses. Applied Optics, 57(28):8143-8147, 2018.
- [19] C. D. Ghilani and P. R. Wolf. Adjustment Computations: Spatial Data Analysis. John Wiley & Sons, Hoboken, N. J., 4th ed., 2006.
- [20] C. Harmening. Spatio-temporal deformation analysis using enhanced B-spline models of laser scanning point clouds. Doctoral thesis, Technische Universität Wien, Vienna, Austria,
- [21] C. Harmening and H. Neuner. A constraint-based parameterization technique for b-spline surfaces. Journal of Applied Geodesy, 9(3):143-161, 2015.
- [22] C. Harmening and H. Neuner. Choosing the optimal number of b-spline control points (part 1: Methodology and approximation of curves). Journal of Applied Geodesy, 10(3):139-157, 2016.
- [23] H. Hu, C. Yuan, and L. Fang. Extrapolation and fitting algorithms for GLONASS satellite orbit. In: 2009 Third International Symposium on Intelligent Information Technology Application, volume 3, pages 282–285, 2009.
- [24] D. R. Ibañez, J. A. Gomez-Pedrero, J. Alonso, and J. A Quiroga. Robust fitting of Zernike polynomials to noisy point clouds defined over connected domains of arbitrary shape. Optics Express, 24(6):5918-5933, 2016.
- [25] H. P. Langtangen. Approximation of functions. http://hplgit. github.io/num-methods-for-PDEs/doc/pub/approx/pdf/ approx-4print-A4-2up.pdf, 2016. Last accessed on 16.02.2021.

- [26] Z. Majdisova and V. Skala. Radial basis function approximations: comparison and applications. Applied Mathematical Modelling, 51:728-743, 2017.
- [27] J. Manson, G. Petrova, and S. Schaefer. Streaming surface reconstruction using wavelets. In Computer Graphics Forum, volume 27, pages 1411-1420. Wiley Online Library, 2008.
- [28] J. C. Mason and D. C. Handscomb. Chebyshev Polynomials. Chapman & Hall/CRC, Boca Raton, FL, 2003.
- [29] J. W. McKinley and T. Ishihara. Chebyshev polynomial curve-fitting technique for strain-gage applications. Experimental Mechanics, 7(6):276-278, 1967.
- [30] M. Mousa, R. Chaine, S. Akkouche, and E. Galin. Efficient spherical harmonics representation of 3d objects. In: 15th Pacific Conference on Computer Graphics and Applications (PG'07), Maui, HI, USA, 29 Oct.-2 Nov. 2007, pages 248-255, IEEE.
- [31] H. Neuner, C. Holst, and H. Kuhlmann. Overview on current modelling strategies of point clouds for deformation analysis. allgemeine vermessungs-nachrichten (avn), 2016.
- [32] W. Niemeier. Ausgleichungsrechnung: Statistische Auswertemethoden. De Gruyter, Berlin et al., 2nd revised and extended edition, 2008.
- [33] Y. Ohtake, A. Belyaev, and H.-P. Seidel. 3d scattered data approximation with adaptive compactly supported radial basis functions. In: Proceedings Shape Modeling Applications, 2004, pages 31-39, Genova, Italy, 7-9 June 2004. IEEE.
- [34] S. A. Orszag. Accurate solution of the Orr-Sommerfeld stability equation. Journal of Fluid Mechanics, 50(04):689, 1971.
- [35] L. Piegl and W. Tiller. The NURBS Book. Springer, Berlin, 2nd ed., 1997.
- [36] T. J. Rivlin. The Chebyshev Polynomials. Pure and Applied Mathematics. Wiley, New York, 1974.

- [37] C. Schmitt and H. Neuner. Knot estimation of b-splines curves. Österreichische Zeitschrift für Vermessung und Geoinformation, 103(2+3):188-197, 2015.
- [38] G. Schwarz. Estimating the dimension of a model. The Annals of Statistics, 6(2):461-464, 1978.
- [39] H. Schwetlick and T Schütze. Least squares approximation by splines with free knots. BIT Numerical Mathematics, 35(3):361-384, 1995.
- [40] V. Smítka and M. Štroner. 3d scanner point cloud denoising by near points surface fitting. In Videometrics, Range Imaging, and Applications XII; and Automated Visual Inspection, volume 8791, page 87910C. International Society for Optics and Photonics, 2013.
- [41] L. N. Trefethen. Approximation Theory and Approximation Practice. SIAM, Philadelphia, PA, 2013.
- [42] D. Wang, V. Kankare, E. Puttonen, M. Hollaus, and N. Pfeifer. Reconstructing stem cross section shapes from terrestrial laser scanning. IEEE Geoscience and Remote Sensing Letters, 14(2):272-276, 2017.
- [43] W. Wang, H. Pottmann, and Y. Liu. Fitting b-spline curves to point clouds by curvature-based squared distance minimization. ACM Transactions on Graphics (ToG), 25(2):214-238, 2006.
- [44] S. Xie, P. Zhang, and L. Liu. Analyzing the precision of Chebyshev polynomial fitting GPS satellite ephemeris. Applied Mechanics and Materials, 353:3410-3413, 2013.
- [45] Kuan Xu. The Chebyshev points of the first kind. Applied Numerical Mathematics, 102:17-30, 2016.
- [46] H. Yanagihara and M. Ohtaki. Knot-placement to avoid over fitting in b-spline scedastic smoothing. Communications in Statistics-Simulation and Computation, 32(3):771-785, 2003.