PREPRINT 446

FEBRUARY 2016

Analytical investigations for the design of fast approximation methods for fitting curves and surfaces to scattered data

Karl-Heinz Brakhage

Institut für Geometrie und Praktische Mathematik
Templergraben 55, 52062 Aachen, Germany

Email address: brakhage@igpm.rwth-aachen.de (Karl-Heinz Brakhage)

Analytical investigations for the design of fast approximation methods for fitting curves and surfaces to scattered data

Karl-Heinz Brakhage

Institut für Geometrie und Praktische Mathematik, RWTH Aachen, Templergraben 55, 52056 Aachen, Germany

Abstract

We present an analytical framework for linear and nonlinear least squares methods and adopt it to the construction of fast iterative methods for this purpose. The results are directly applicable to curves and surfaces that have a representation as a linear combination of smooth basis functions associated with the control points. Standard Bézier and B-spline curves / surfaces as well as subdivision schemes have this property. In the global approximation step for the control points our approach couples the standard linear approximation part with the reparameterization to heavily reduce the number of overall steps in the iteration process. This can be formulated in such a way that we have a standard least squares problem in each step. For the local nonlinear parameter corrections our results allow for an optimal choice of the methods used in different stages of the process. Furthermore, regularization terms that express the fairness of the intermediate and / or final result can be added. Adaptivity is easily integrated in our concept. Moreover our approach is well suited for reparameterization occurring in grid generation.

Keywords: Splines, Multivariate Approximation, Least Squares, Fairing, Numerical Analysis, Numerical Linear Algebra

1. Introduction

Fitting curves and surfaces to unorganized point clouds or sample points from given curves / surfaces (aka reparameterization) is an often occurring problem in engineering CAD / CAGD and computer graphics. It has a big history in literature. For a long period surface fitting methods used the distance between a sample point and the corresponding foot point on the fitting surface for minimizing the objective function. We call these methods point distance minimization (PDM). A further, different approach to solve the approximation problem of curves and surfaces are active contour models. The origin of this technique is a paper by Kass et al. [8]. They use a variational formulation of parametric curves for detecting contours in images.

It is well known that the parameterization problem is a fundamental one for the whole approximation process and the final result. Therefore, parameter correction procedures have to be used to improve the quality of the final approximation. Unfortunately the decoupling of the overall fitting procedure into the two independent steps parameter correction and solving a linear least squares problem with fixed parameters leads to very slow convergence. To overcome this Pottmann et al. [11] introduced an approach based on the minimization of a quadratic approximant of the squared distance function. An additional aim was to avoid the parameterization problem and to construct algorithms of second order convergence. Methods based on this idea are called SDM (squared distance minimization). Unfortunately in general the second order Taylor approximant does not lead to symmetric positive definite system matrices and for this reason the existence and uniqueness of the minimum cannot be guaranteed. Thus the second order Taylor approximation was modified to ensure positive definiteness. However this modification destroys the second order and thus the

Email address: brakhage@igpm.rwth-aachen.de (Karl-Heinz Brakhage)

claimed quadratic convergence of those methods. Nevertheless such approaches need less iterations. On the other hand the main drawback is a large computational overhead. The curvature computation and the setup of a more complex SDM error function lead to computational inefficiency of SDM. A comparison in [6] shows that the time to attain comparable results used by SDM on iterative optimization is about 30% to 50% more than PDM. Furthermore the parameterization is not really totally avoided.

In [5] the goal of a new development is the avoidance of the curvature computation together with the construction of a cheap error function that accelerates the parameter correction. The algorithm therein can be used for the construction of smooth surfaces from point clouds as well as for the reparameterization of given surfaces. The methods are not restricted to be applied to Bézier or B-spline surfaces. They can be used for subdivision surfaces as well. The focus in that paper is on an analytical understanding of the coupling of parameter correction with the linear least squares approximation step and the transformation of the mathematical model to a form that can be used efficiently for fast solvers. In comparison with the standard PDM approach with decoupling of the linear solver and the parameter correction that method has a tremendous speed up. It needs much less iterations without the drawback of the computational overhead of SDM. Furthermore that method can still be written in the form of a standard least squares problem which is not the case for SDM. Thus the normal equations and iterative solvers for them can be used as well as orthogonal transformations that have a better condition number than the normal equations. To reduce the number of parameter corrections (the outer loop) a combination of the PDM and the distances of the data points to the linear approximation of the target surface at the projection point is used. The latter one coincides with the squared distance function for points on the surface. The method has superlinear convergence with only a small computational overhead for surface normals. It can be applied to the approximation with standard Bézier- or B-splines as well as with subdivision surfaces.

In this paper we will focus on the necessary parameter correction steps. We will see that different methods have to be used in the different stages consisting of: Initial computation of parameter values, parameter correction during the first steps and parameter correction when the residuals are already small. Furthermore we will see that curvature plays an important role. To get a deeper understanding for the choice of optimal methods for parameter correction steps the possible algorithms are analyzed in detail regarding convergence and error estimation.

The rest of this paper is organized as follows. First we give some basic notations and properties of Bézier, B-spline and subdivision surfaces and the needed basics on approximation with them. Then we briefly summarize the above-mentioned algorithm from [5]. Next the main topic is the analytical investigation on the nonlinear approximation algorithms for parameter correction. It will be shown that the theoretically derived behavior is directly reflected in our implementation and leads to an enormous speedup. Finally a summary and an evaluation regarding computation time and accuracy of the algorithms usually applied for these purposes will be given.

2. Basics on surfaces and approximation

Bézier curves of order n are given by

$$\mathbf{x}(u) = \sum_{i=0}^{n} B_i^n(u)\,\mathbf{p}_i = \sum_{i=0}^{n} \binom{n}{i} u^i (1-u)^{n-i}\,\mathbf{p}_i \tag{1}$$

with $n+1$ control points $\mathbf{p}_i \in \mathbf{R}^d,\ i = 0, \ldots, n$ (here $d = 2, 3$). The $B_i^n(u)$ are the Bernstein polynomials. Derivatives of Bézier curves of order n are Bézier curves of order $n-1$. They can be computed from the control points automatically. Another important advantage for interpolation and approximation with curves like (1) is the linearity in the control points $\mathbf{p}_i$. B-spline curves have these properties, too. They are given by

$$\mathbf{x}(u) = \sum_{i=0}^{n} N_i(u)\,\mathbf{p}_i \tag{2}$$

where $N_i(u)$ is the i-th normalized B-spline function. Surfaces are represented by tensor products of the form

$$\mathbf{x}(u,v) = \sum_{i=0}^{n}\sum_{j=0}^{m} B_i^n(u)\, B_j^m(v)\, \mathbf{p}_{ij} \qquad \text{or} \tag{3}$$

$$\mathbf{x}(u,v) = \sum_{i=0}^{n}\sum_{j=0}^{m} N_{i,p}(u)\, N_{j,q}(v)\, \mathbf{p}_{ij} \tag{4}$$

with control points $\mathbf{p}_{ij}$ for Bézier and B-spline surfaces, respectively. Again these representations are linear in the control points. The same applies for stationary subdivision curves and surfaces. The algorithms presented below can be used for all these curve and surface classes. We rewrite all the above representations in the form

$$\mathbf{x}(u,v) = \sum_{j=1}^{N} B_j(u,v)\, \mathbf{p}_j \tag{5}$$

with general basis functions $B_j(u,v)$.

More details and a more precise formulation on this topic with respect to approximation can be found in [5] and the literature referred therein, e.g. [7, 10, 14, 15].

To be more precise with our least squares formulation we have to introduce some technical notations. For an optimal approximation of a given surface $\mathbf{y}(s,t),\ (s,t) \in [s_{min}, s_{max}] \times [t_{min}, t_{max}] =: D \subset \mathbf{R}^2$ by a parametric surface we have to determine the control points $\mathbf{p}_j$ associated to (5) in such a way that

$$\max_{(s,t)\in D}\ \min_{(u,v)\in[0,1]^2} \|\mathbf{y}(s,t) - \mathbf{x}(u,v)\|_2 \tag{6}$$

is minimized. Since in practice this problem is too complex to solve we switch to a discrete approximation problem. In each (sub-)domain a set of approximation points $\mathbf{y}_i = \mathbf{y}(s_i,t_i)$ is chosen. In this paper we assume that an initial simple mesh corresponding to the correct topology of the target surface and allowing the computation of the (u_i,v_i) is already given. We further use the error estimator

$$\delta_i = \|\mathbf{y}_i - \mathbf{x}\,(u_i,v_i))\|_2 \quad \text{and} \quad \delta = \max_i \{\delta_i\}\ . \tag{7}$$

We want the error to become small measured in the ∞-norm but we are solving the minimization problem for the 2-norm. Scaling the different equations depending on the δ_i (for large δ_i we use large scaling values and vice versa) leads to an improvement for the optimization (see [3] for instance). In this paper we will not report on these effects. If the error is too large, we (recursively) subdivide the parameter domain. For subdivision surfaces this is a normal subdivision step. For B-splines we use knot insertion and for Béziers we split each subdomain into four equal parts. The corresponding parameter values (u_i,v_i) are recomputed for the new domains. This step is not necessary for B-splines. In all cases we have good starting values after the first overall iteration.

The basic approximation principle, explained for surfaces, is as follows. Let $\mathbf{y}_i,\ i \in \{1,\ldots,M\} =: I$ be given data points or samples on a given target surface. We want to compute a good approximating parametric fitting surface with a representation of the form (5). For the B-spline case we further assume that the knot vectors U and V are already determined. In a first step we have to find (at least approximately) the nearest points

$$\mathbf{y}_i \approx \mathbf{x}_i = \mathbf{x}(u_i,v_i) = \sum_{j=1}^{N} B_j(u_i,v_i)\, \mathbf{p}_j =: \sum_{j=1}^{N} a_{ij}\, \mathbf{p}_j \tag{8}$$

on the fitting surface. A good estimation of the parameter values (u_i,v_i) might be a difficult task. Furthermore it is well known that the parameterization problem is a fundamental one for the whole approximation process and the final result. Therefore, parameter correction procedures have to be used to improve the quality of the final approximation. The decoupling of the overall fitting procedure into the two independent steps parameter correction and solving a linear least squares problem with fixed parameters leads to very

slow convergence. SDM needs much less iterations of the outer loop. The necessary curvature computation and the setup of a more complex SDM error function leads to computational inefficiency of SDM. The goal of fast algorithms is the avoidance of the curvature computation together with the construction of a cheap error function that accelerates the parameter correction.

A modification of the second order approximation of the SDM error function F_d is necessary because there are points for which F_d is **not** a positive definite quadratic form. That leads to a system matrix that is not symmetric positive definite such that we cannot guarantee the existence and uniqueness of the minimum. At those points the iso value surfaces of the error functions are hyperboloids. The standard modification from [11] changes the sign for the negative eigenvalues resulting in F_d^+. Now we end up with a positive definite system matrix but the second order of the approximant is destroyed and thus the claimed quadratic convergence of that method. In [5] a scaled combination of the standard PDM minimization function $(\mathbf{x}-\mathbf{y})^2$ and $(\mathbf{n}^T(\mathbf{x}-\mathbf{y}))^2$ is used. The latter one coincides with F_d^+ if $d=0$, which is the case if $\mathbf{x}_i$ lies on the target surface. From this we conclude that it is a good choice to make the scaling dependent of d:

$$F_d^{new}(\mathbf{x}) = (\mathbf{x}-\mathbf{y})^2 + \lambda(d)\left(\mathbf{n}^T(\mathbf{x}-\mathbf{y})\right)^2 . \tag{9}$$

For using (9) in the minimization concept one only needs a normal $\mathbf{n}_i$ for each sample point $\mathbf{y}_i$. Since the normal of the fitting surface $\mathbf{x}(u,v)$ approximates the normal of the target surface, one can even use the normal of $\mathbf{x}(u,v)$. The high cost for the curvature computation is avoided, too.

For the sample points $\mathbf{y}_i, i = 1,2,\dots M$ with associated parameter values (u_i, v_i) of the fitting surface we use the following setup. According to (8) the $\mathbf{x}_i$ are given as $\mathbf{x}_i = \mathbf{x}(u_i, v_i)$. We collect the control points $\mathbf{p}_j$ in a vector (of 3d vectors) $\mathbf{p}$, the $\mathbf{y}_i$ in a vector $\mathbf{y}$ and all coefficients $a_{ij} := B_j(u_i, v_i)$ in a matrix $A \in \mathbf{R}^{M\times N}$. With this notation PDM is simply

$$\mathbf{p}^\star = \underset{\mathbf{p}}{\operatorname{argmin}} \|A\,\mathbf{p} - \mathbf{y}\|_2 . \tag{10}$$

For the solution of (10) orthogonal transformations or the normal equations can be used:

$$A^T A\,\mathbf{p}^\star = A^T\,\mathbf{y} . \tag{11}$$

Using $\mathbf{x}_i = \sum_j a_{ij}\,\mathbf{p}_j$ we can write the distance to the tangent plane as

$$\|\mathbf{n}_i^T\,(\mathbf{x}_i - \mathbf{y}_i)\|_2 = \left\|\mathbf{n}_i^T\,\left(\sum_j a_{ij}\mathbf{p}_j - \mathbf{y}_i\right)\right\|_2 = \left\|\sum_j a_{ij}\,\mathbf{n}_i^T\,\mathbf{p}_j - \mathbf{n}_i^T\,\mathbf{y}_i\right\|_2 . \tag{12}$$

Notice that (12) leads to a minimization over the control points $\mathbf{p}_j$. To use matrix notations we have to separate the x, y and z components. We define $N_x = \operatorname{diag}\{\mathbf{n}_{i,x}\}$, $N_y = \operatorname{diag}\{\mathbf{n}_{i,y}\}, N_z = \operatorname{diag}\{\mathbf{n}_{i,z}\}$ and collect the terms $\mathbf{n}_i^T\,\mathbf{y}_i$ in a vector $\mathbf{d}$. We also have to split $\mathbf{p}$ into its x-components $\mathbf{p}_x$, y-components $\mathbf{p}_y$ and z-components $\mathbf{p}_z$. Now we can write this part as

$$\|N_x\,A\,\mathbf{p}_x + N_y\,A\,\mathbf{p}_y + N_z\,A\,\mathbf{p}_z - \mathbf{d}\|_2 \to \min. \tag{13}$$

For our final minimization problem according to (9) we use (10) and a scaled portion of (13). Analogously to $\mathbf{p}$ we split $\mathbf{y}$ into $\mathbf{y}_x$, $\mathbf{y}_y$ and $\mathbf{y}_z$. With

$$\hat{A} = \begin{pmatrix} A & 0 & 0 \\ 0 & A & 0 \\ 0 & 0 & A \\ \lambda N_x A & \lambda N_y A & \lambda N_z A \end{pmatrix},\ \hat{\mathbf{p}} = \begin{pmatrix} \mathbf{p}_x \\ \mathbf{p}_y \\ \mathbf{p}_z \end{pmatrix} \text{ and } \hat{\mathbf{y}} = \begin{pmatrix} \mathbf{y}_x \\ \mathbf{y}_y \\ \mathbf{y}_z \\ \lambda\,\mathbf{d} \end{pmatrix} \tag{14}$$

(compare (9)) our minimization problem now reads

$$\hat{\mathbf{p}}^\star = \underset{\hat{\mathbf{p}}}{\operatorname{argmin}} \|\hat{A}\,\hat{\mathbf{p}} - \hat{\mathbf{y}}\|_2. \tag{15}$$

Thus again we can use orthogonal transformations, the normal equations or iterative methods. Especially for recursive approximation of surfaces the iterative methods are much faster. The parameter $\lambda \geq 0$ is chosen depending on the error estimator. For small errors we use a large λ. Details for implementing solvers for this problem and the reduction of the number of outer loops can be found in [5]. The part on the optimization of the parameter corrections was left out in that paper. For the initial setup and in each subdivision step the paramter correction has do be done for each sample point a couple of times. We will now focus on this topic in the next section.

3. Parameter corrections: accelerating nonlinear least squares

As stated in the previous section we have to repeatedly solve the following problem: Find (at least approximately) the nearest points $\mathbf{x}_i = \mathbf{x}(u_i, v_i) \approx \mathbf{y}_i$ for the sample points on the fitting surface. Since the time spent for these approximations is a significant part of the overall run time it is worth thinking about optimal algorithms for this subtask. We will now focus on an analytical understanding of the arising nonlinear least squares problems.

This main topic of the paper is organized as follows: We start with the necessary notations for the nonlinear least squares problem. Next we briefly introduce the relevant algorithms for our problem class. Then some theorems on convergence behavior will be proven. These results are used for designing our algorithms and the achievements will be shown.

For a parametrically given surface $\mathbf{x} : \mathbf{R}^2 \to \mathbf{R}^3$ and a given point $\mathbf{p} \in \mathbf{R}^3$ we (locally) search for the parameters $(u^\star, v^\star)$ in a subset $\mathcal{M} \subset \mathbf{R}^2$ such that

$$\|\mathbf{x}(u^\star, v^\star) - \mathbf{p}\|_2 = \min_{(u,v)\in\mathcal{M}} \|\mathbf{x}(u, v) - \mathbf{p}\|_2 \,. \tag{16}$$

In general the nonlinear least squares problem can be formally considered as follows. Find $\mathbf{x}^\star \in \mathcal{M} \subset \mathbf{R}^n$, a local minimizer for $\|F(\mathbf{x})\|_2$ with a function $F : \mathbf{R}^n \to \mathbf{R}^m$. We will write this as

$$\mathbf{x}^\star = \operatorname*{argmin}_{\mathbf{x}\in\mathcal{M}} \|F(\mathbf{x})\|_2 \,. \tag{17}$$

Note that $\mathbf{x}^\star$ also (locally) minimizes the function $\phi(\mathbf{x})$ given by

$$\phi(\mathbf{x}) = \frac{1}{2} \|F(\mathbf{x})\|_2^2 = \frac{1}{2} F(\mathbf{x})^T F(\mathbf{x}) \,. \tag{18}$$

That is $\phi(\mathbf{x}^\star) = \min_{\mathbf{x}\in\mathcal{M}} \phi(\mathbf{x})$. Let us denote the l times continuously differentiable functions from $\mathcal{M}$ to $\mathcal{N}$ by $C^l(\mathcal{M}, \mathcal{N})$. For $F \in C^1(\mathcal{M}, \mathbf{R}^m)$ a necessary condition for a local minimizer $\mathbf{x}_s \in \mathcal{M}$ is

$$\phi'(\mathbf{x}_s) = \nabla\phi(\mathbf{x}_s) = F'(\mathbf{x}_s)^T F(\mathbf{x}_s) = 0 \,. \tag{19}$$

A point $\mathbf{x}_s$ with $\phi'(\mathbf{x}_s) = 0$ is called stationary point. If in addition

$$\phi''(\mathbf{x}) = F'(\mathbf{x})^T F'(\mathbf{x}) + \sum_{i=1}^{m} F_i(\mathbf{x})\, F_i''(\mathbf{x}) =: F'(\mathbf{x})^T F'(\mathbf{x}) + F(\mathbf{x})^T \otimes F''(\mathbf{x}) \tag{20}$$

is a symmetric positive definite matrix (spd) for $\mathbf{x} = \mathbf{x}_s$ then we have a local minimum at $\mathbf{x}^\star = \mathbf{x}_s$.

In our case the time consumption is due to the high number of minimization problems we have to solve but not to the high dimension of the individual problems. When we start our approximation process we have poor estimations of the parameter values and eventually large residuals. After some iterations and subdivisions we have good estimations of the parameter values and small residuals. This gives rise to use different iteration methods during the overall process.

In many cases iterative methods converge towards the solutions $\mathbf{x}^\star$ in at least two clearly different stages. When $\mathbf{x}_0$ is a poor approximation for the solution we are satisfied if the method produces iteration vectors that move steadily towards $\mathbf{x}^\star$ at all. Later on in the approximation process we want the method to move

faster towards the solution. To classify the convergence quality we define the error vector $\mathbf{e}^k$ for the k-th vector in our iteration by

$$\mathbf{e}^k := \mathbf{x}^k - \mathbf{x}^\star \,. \tag{21}$$

Furthermore we distinguish between ***linear convergence***

$$\exists K_0 \wedge \alpha < 1 \text{ such that } \forall k > K_0 \quad : \quad \|\mathbf{e}^{k+1}\| \leq \alpha \, \|\mathbf{e}^k\| \quad \text{ with } \alpha < 1 \,, \tag{22}$$

quadratic convergence

$$\exists K_0 \wedge \alpha \text{ such that } \forall k > K_0 \quad : \quad \|\mathbf{e}^{k+1}\| \leq \alpha \, \|\mathbf{e}^k\|^2 \,, \tag{23}$$

and ***superlinear convergence***

$$\lim_{k \to \infty} \frac{\|\mathbf{e}^{k+1}\|}{\|\mathbf{e}^k\|} = 0 \,. \tag{24}$$

Quadratic convergence implies superlinear convergence. Both imply linear convergence. To get a better understanding for the choice of the best method in the different stages next we study the practicability, robustness and efficiency of several algorithms for our minimization problems.

The Nelder-Mead optimization algorithm [9] is a Downhill-Simplex method for finding a local minimum of a function of several variables. The term simplex is used as generalized triangle in n dimensions. In the literature it is sometimes nicknamed "Amoeba". It is simple to implement. In addition, it is very robust and does not need any derivatives. Thus it can easily be adapted to arbitrary surfaces. Especially for highly curved surfaces this is the method of choice at least for the initial steps. For further information on implementation issues in the context of surface reparameterization see [13]. For the case of equally sided simplices the algorithm can be described as follows: In the first stage move downhill by reflecting the worst point (largest ϕ-value) or the second worst point at the hyperplane given by the remaining points. If neither of the just mentioned steps leads to a smaller ϕ-value shrink the current simplex towards the best point with a factor of $1/2$. The shrinking steps belong to the second stage of the iteration. Thus the best convergence we can expect from this method is linear convergence with $\alpha = 1/2$. There exist several attempts to optimize the downhill step and the shrinkage. All modifications result in methods with linear convergence and $\alpha \approx 1/2$. We only use the Nelder-Mead algorithm with a minor number of sample points for determining the parameter values for the first time if we do not have good initial guesses. Thus it does not devour a relevant part of time.

In the literature the mostly used method for parameter corrections is the repeated projection onto the tangent plane

$$\mathbf{t}(u,v) = \mathbf{x}(u,v) + \Delta u \, \mathbf{x}_u(u,v) + \Delta v \, \mathbf{x}_v(u,v) \tag{25}$$

and computing $(\Delta u, \Delta v)$ as an update for the (u,v) values. This is equivalent to the well-known Gauß-Newton method that can be formulated as follows. For a given starting point $\mathbf{x}^0$ we iterate due to

$$\begin{array}{rcl} \mathbf{s}^k & = & \displaystyle\operatorname*{argmin}_{\mathbf{s} \in \mathbf{R}^n} \|F'(\mathbf{x}^k)\,\mathbf{s} + F(\mathbf{x}^k)\|_2 \\ \mathbf{x}^{k+1} & = & \mathbf{x}^k + \mathbf{s}^k \end{array} \,. \tag{26}$$

Assuming that F' has full rank for the iteration and using the normal equations for solving (26) we can explicitly give the fix point function $\Phi(\mathbf{x})$ for further investigations. To clarify the difference to the Newton method for the computation of stationary points (compare (19)), we give the iteration function $\Phi_N(\mathbf{x})$ for this method, too.

$$\begin{array}{rcl} \Phi(\mathbf{x}) & = & \mathbf{x} - \left(F'(\mathbf{x})^T\, F'(\mathbf{x})\right)^{-1}\, F'(\mathbf{x})^T\, F(\mathbf{x}) \\ \Phi_N(\mathbf{x}) & = & \mathbf{x} - \left(F'(\mathbf{x})^T\, F'(\mathbf{x}) + F(\mathbf{x})^T \otimes F''(\mathbf{x})\right)^{-1}\, F'(\mathbf{x})^T\, F(\mathbf{x}) \end{array} \tag{27}$$

In general Gauß-Newton is not a method of second order convergence. For that reason we can not force convergence with starting values *close enough* to the fix point. From the convergence theorem of Ostrowski

we know, that beside the technical condition $\Phi \in C^1(\mathcal{M}, \mathcal{M})$ for an open set $\mathcal{M} \subset \mathbf{R}^n$ a sufficient one for local convergence towards the fix point $\mathbf{x}^\star \in \mathcal{M}$ is

$$\rho_{\max}\left(\Phi'(\mathbf{x}^\star)\right) < 1\,, \tag{28}$$

where $\rho_{\max}(B)$ is according to absolute value the largest eigenvalue of B – the spectral radius. From the **Contraction Mapping Theorem** we have the ***a-priori*** error estimate

$$\|\mathbf{x}^\star - \mathbf{x}^k\| \leq \frac{q^k}{1-q}\,\|\mathbf{x}^1 - \mathbf{x}^0\| \tag{29}$$

and the ***a-posteriori*** error estimate

$$\|\mathbf{x}^\star - \mathbf{x}^k\| \leq \frac{q}{1-q}\,\|\mathbf{x}^k - \mathbf{x}^{k-1}\|\,, \tag{30}$$

where q is the Lipschitz contraction number regarding Φ. Close to the fixed point $\mathbf{x}^\star$ we expect the contraction number $q \approx \rho_{\max}\left(\Phi'(\mathbf{x}^\star)\right)$. Thus the error is reduced by a factor of $\approx \rho_{\max}\left(\Phi'(\mathbf{x}^\star)\right)$ in each step.

For $F \in C^2(\mathcal{M}, \mathcal{M})$ using (27) and the abbreviation $A(\mathbf{x}) := \left(F'(\mathbf{x})^T\,F'(\mathbf{x})\right)^{-1}$ we get

$$\Phi'(\mathbf{x}) = I - A'(\mathbf{x})\phi'(\mathbf{x}) - A(\mathbf{x})\left(F(\mathbf{x})^T \otimes F''(\mathbf{x}) + \underbrace{F'(\mathbf{x})^T\,F'(\mathbf{x})}_{A^{-1}(\mathbf{x})}\right). \tag{31}$$

Simplifying the last equation and using $\phi'(\mathbf{x}^\star) = 0$ yields

$$\Phi'(\mathbf{x}^\star) = -\left(F'(\mathbf{x}^\star)^T\,F'(\mathbf{x}^\star)\right)^{-1}\left(F(\mathbf{x}^\star)^T \otimes F''(\mathbf{x}^\star)\right)\,. \tag{32}$$

From this we can see that large residuals and high curvature are counterproductive for (good) convergence. On the other hand it is clear that for moderate curvature and small residuals the convergence is good. In our special case for surfaces we have

$$\Phi'(u,v) = -\begin{pmatrix} \mathbf{x}_u^2(u,v) & \mathbf{x}_u^T(u,v)\mathbf{x}_v(u,v) \\ \mathbf{x}_u^T(u,v)\mathbf{x}_v(u,v) & \mathbf{x}_v^2(u,v) \end{pmatrix}^{-1} (\mathbf{x}(u,v) - \mathbf{p})^T \otimes \begin{pmatrix} \mathbf{x}_{uu}(u,v) & \mathbf{x}_{uv}(u,v) \\ \mathbf{x}_{uv}(u,v) & \mathbf{x}_{vv}(u,v) \end{pmatrix}. \tag{33}$$

For the reparameterization process with subdivision the residuals $\|\mathbf{y}_i - \mathbf{x}(u_i, v_i)\|_2$ tend to zero. Thus in the final stages the Gauß-Newton iteration behaves like a superlinearly convergent method.

To compare the complexity with the Newton method we give Φ_N from (27) for the case of surfaces

$$\Phi_N(u,v) = \begin{pmatrix} u \\ v \end{pmatrix} - \left(\begin{pmatrix} \mathbf{x}_u^2 & \mathbf{x}_u^T\mathbf{x}_v \\ \mathbf{x}_u^T\mathbf{x}_v & \mathbf{x}_u^2 \end{pmatrix} + (\mathbf{x} - \mathbf{p})^T \otimes \begin{pmatrix} \mathbf{x}_{uu} & \mathbf{x}_{uv} \\ \mathbf{x}_{uv} & \mathbf{x}_{vv} \end{pmatrix}\right)^{-1} \begin{pmatrix} \mathbf{x}_u^T \\ \mathbf{x}_v^T \end{pmatrix} (\mathbf{x} - \mathbf{p})\,. \tag{34}$$

Here we have omitted the arguments (u, v) for the function $\mathbf{x}$ and its derivatives. The (main) overhead to the Gauß-Newton method is the computation of three partial derivatives and four scalar products.

Next we want to proof a convergence theorem for the Gauß-Newton method. For this purpose we need some more notations and some Lemmata to split the proof. We start with two important theorems which we state in the context of full rank Jacobians. For a matrix $A \in \mathbf{R}^{m\times n}$ with $m \geq n$ and full rank n we define the pseudoinverse A^+ of A as

$$A^+ := \left(A^T\,A\right)^{-1}\,A^T\,. \tag{35}$$

For full rank matrices this definition matches the more general definition using the singular value decomposition. The following perturbation theorem including proof can for instance be found in [1] as Theorem 2.2.4.

Theorem 1. ***(Perturbation Theorem)*** *Let A, $\tilde{A} \in \mathbf{R}^{m\times n}$ with rank(A) = rank$(\tilde{A}) = n$. Furthermore let $\eta := \|A^+\|_2\,\|A - \tilde{A}\|_2 < 1$. Then*

$$\|\tilde{A}^+\|_2 \leq \frac{1}{1-\eta}\,\|A^+\|_2\,. \tag{36}$$

Theorem 2. ***(Mean Value Theorem)*** *Let $\mathcal{M} \subset \mathbf{R}^n$ be convex and closed and $F \in C^1(\mathcal{M}, \mathbf{R}^n)$. Then* $\forall \mathbf{x}, \mathbf{y} \in \mathcal{M}$

$$F(\mathbf{y}) - F(\mathbf{x}) = \int_0^1 F'(\mathbf{x} + s\,(\mathbf{y} - \mathbf{x}))\,(\mathbf{y} - \mathbf{x}))\,ds\,. \tag{37}$$

For the following results we need some technical presuppositions. We will refer to these assumptions with (GNA).

(1) $\mathcal{M} \subset \mathbf{R}^n$ is a convex and closed set,

(2) $F \in C^1(\mathcal{M}, \mathbf{R}^m)$ with $m > n$,

(3) F' is Lipschitz-continuous: $\forall \mathbf{x}, \mathbf{y} \in \mathcal{M} : \|F'(\mathbf{x}) - F'(\mathbf{y})\|_2 \leq L\,\|\mathbf{x} - \mathbf{y}\|_2$,

(4) $\mathbf{x}^\star \in \mathcal{M} \backslash \partial\mathcal{M}$ is a stationary point of ϕ,

(5) $F'(\mathbf{x}^\star)$ has full rank.

Lemma 3. *Under the above assumptions (GNA) we define $T(\mathbf{x}) := F(\mathbf{x}^\star) - F(\mathbf{x}) + F'(\mathbf{x})\,(\mathbf{x} - \mathbf{x}^\star)$. Then*

$$\|T(\mathbf{x})\|_2 \leq \frac{L}{2}\|\mathbf{x} - \mathbf{x}^\star\|_2^2\,. \tag{38}$$

Proof. Due to Theorem 2 we have

$$\begin{aligned} T(\mathbf{x}) &= \textstyle\int_0^1 F'(\mathbf{x} + s\,(\mathbf{x}^\star - \mathbf{x}))\,(\mathbf{x}^\star - \mathbf{x})\,ds + F'(\mathbf{x})\,(\mathbf{x} - \mathbf{x}^\star) \\ &= \textstyle\int_0^1 F'(\mathbf{x} + s\,(\mathbf{x}^\star - \mathbf{x}))\,(\mathbf{x}^\star - \mathbf{x}) - F'(\mathbf{x})\,(\mathbf{x}^\star - \mathbf{x})\,ds \\ &= \textstyle\int_0^1 \left(F'(\mathbf{x} + s\,(\mathbf{x}^\star - \mathbf{x})) - F'(\mathbf{x})\right)\,(\mathbf{x}^\star - \mathbf{x})\,ds \end{aligned}$$

Taking norms and using assumption (GNA)(3) we get

$$\|T(\mathbf{x})\|_2 \leq \int_0^1 L\,\|s\,(\mathbf{x}^\star - \mathbf{x})\|_2\,\|\mathbf{x}^\star - \mathbf{x}\|_2\,ds = L\,\|\mathbf{x}^\star - \mathbf{x}\|_2^2 \int_0^1 s\,ds = \frac{L}{2}\|\mathbf{x} - \mathbf{x}^\star\|_2^2\,.$$

□

Lemma 4. *Let the assumptions (GNA) be fulfilled and $r := \dfrac{1}{L}\min\left\{\dfrac{1}{3},\ \dfrac{1}{2}\,\dfrac{1}{\frac{1}{3} + 2\,\|F'(\mathbf{x}^\star)\|_2}\right\}$ and $U_r(\mathbf{x}^\star) := \left\{\mathbf{x} \in \mathbf{R}^n \;\middle|\; \|\mathbf{x} - \mathbf{x}^\star\|_2 \leq r\right\} \subset \mathcal{M}$. Then*

$$\|F'(\mathbf{x})^T\,F'(\mathbf{x}) - F'(\mathbf{x}^\star)^T\,F'(\mathbf{x}^\star)\|_2 \leq \frac{1}{2} \tag{39}$$

and

$$\|(F'(\mathbf{x})^T\,F'(\mathbf{x}))^{-1}\|_2 \leq 2\,\|(F'(\mathbf{x}^\star)^T\,F'(\mathbf{x}^\star))^{-1}\|_2\,. \tag{40}$$

Proof. Using the Lipschitz continuity (GNA)(3) and the restrictions on r we get

$$\begin{aligned} & \|F'(\mathbf{x})^T\,F'(\mathbf{x}) - F'(\mathbf{x}^\star)^T\,F'(\mathbf{x}^\star)\|_2 \\ = \;& \|F'(\mathbf{x})^T\,F'(\mathbf{x}) - F'(\mathbf{x})^T\,F'(\mathbf{x}^\star) + F'(\mathbf{x})^T\,F'(\mathbf{x}^\star) - F'(\mathbf{x}^\star)^T\,F'(\mathbf{x}^\star)\|_2 \\ = \;& \|F'(\mathbf{x})^T\,(F'(\mathbf{x}) - F'(\mathbf{x}^\star)) + \left(F'(\mathbf{x})^T - F'(\mathbf{x}^\star)^T\right)\,F'(\mathbf{x}^\star)\|_2 \\ \leq \;& \|F'(\mathbf{x})\|_2\,L\,\|\mathbf{x} - \mathbf{x}^\star\|_2 + L\,\|\mathbf{x} - \mathbf{x}^\star\|_2\,\|F'(\mathbf{x}^\star)\|_2 \\ = \;& L\,\|\mathbf{x} - \mathbf{x}^\star\|_2\,(\|F'(\mathbf{x}) - F'(\mathbf{x}^\star) + F'(\mathbf{x}^\star)\|_2 + \|F'(\mathbf{x}^\star)\|_2) \\ \leq \;& L\,\|\mathbf{x} - \mathbf{x}^\star\|_2\,(L\,\|\mathbf{x} - \mathbf{x}^\star\|_2 + 2\,\|F'(\mathbf{x}^\star)\|_2) \\ \leq \;& L\,\|\mathbf{x} - \mathbf{x}^\star\|_2\,\left(\tfrac{1}{3} + 2\,\|F'(\mathbf{x}^\star)\|_2\right) \leq \tfrac{1}{2} \end{aligned}$$

Finally (40) follows from Theorem 1. □

Theorem 5. *Under the assumptions (GNA) there exist constants $r > 0$ and $\tilde{L} = \tilde{L}\left(L, \left\|\left(F'(\mathbf{x}^\star)^T\, F'(\mathbf{x}^\star)\right)^{-1}\right\|_2\right)$ such that for $\|\mathbf{x}^\star - \mathbf{x}^k\|_2 \leq r$*

$$\|\mathbf{x}^\star - \mathbf{x}^{k+1}\|_2 \leq \tilde{L}\left(\|F'(\mathbf{x}^\star)\|_2\, \|\mathbf{x}^\star - \mathbf{x}^k\|_2^2 + \|F(\mathbf{x}^\star)\|_2\, \|\mathbf{x}^\star - \mathbf{x}^k\|_2\right)\,. \tag{41}$$

Proof. We use the pseudoinverse property $A^+\, A = I$ and the necessary condition (19) for $\mathbf{x}^\star$:

$$\begin{aligned}
\mathbf{x}^\star - \mathbf{x}^{k+1} &= \mathbf{x}^\star - \mathbf{x}^k + F'(\mathbf{x}^k)^+\, F(\mathbf{x}^k) \\
&= F'(\mathbf{x}^k)^+\left(F'(\mathbf{x}^k)\,(\mathbf{x}^\star - \mathbf{x}^k) + F(\mathbf{x}^k)\right) \\
&= F'(\mathbf{x}^k)^+(\underbrace{F'(\mathbf{x}^k)\,(\mathbf{x}^\star - \mathbf{x}^k) + F(\mathbf{x}^k) - F(\mathbf{x}^\star)}_{-T(\mathbf{x}^k)} + F(\mathbf{x}^\star)) \\
&\overset{(19)}{=} -F'(\mathbf{x}^k)^+ T(\mathbf{x}^k) + (F'(\mathbf{x}^k)^T\, F'(\mathbf{x}^k))^{-1}(F'(\mathbf{x}^k)^T - F'(\mathbf{x}^\star)^T)\, F(\mathbf{x}^\star)
\end{aligned}$$

Using the abbreviation $L_A^\star := \|(F'(\mathbf{x}^\star)^T\, F'(\mathbf{x}^\star))^{-1}\|_2$ and the restrictions on r from Lemma 4 and Lemma 3 for the first term of the right hand side we get the estimation

$$\begin{aligned}
\|F'(\mathbf{x}^k)^+\, T(\mathbf{x}^k)\|_2 &\leq 2\,\|(F'(\mathbf{x}^\star)^T\, F'(\mathbf{x}^\star))^{-1}\|_2\, 2\,\|F'(\mathbf{x}^\star)\|_2\, \tfrac{L}{2}\, \|\mathbf{x}^k - \mathbf{x}^\star\|_2^2 \\
&= 2\, L_A^\star\, L\, \|F'(\mathbf{x}^\star)\|_2\, \|\mathbf{x}^k - \mathbf{x}^\star\|_2^2\,.
\end{aligned} \tag{42}$$

And for the second term we get

$$\|(F'(\mathbf{x}^k)^T\, F'(\mathbf{x}^k))^{-1}(F'(\mathbf{x}^k)^T - F'(\mathbf{x}^\star)^T)\, F(\mathbf{x}^\star)\|_2 \leq 2\, L_A^\star\, L\, \|\mathbf{x}^k - \mathbf{x}^\star\|_2\, \|F(\mathbf{x}^\star)\|_2\,.$$

Thus we can estimate $\|\mathbf{e}^{k+1}\|_2$ as

$$\|\mathbf{x}^\star - \mathbf{x}^{k+1}\|_2 \leq 2\, L_A^\star\, L\, \left(\|F'(\mathbf{x}^\star)\|_2\, \|\mathbf{x}^k - \mathbf{x}^\star\|_2^2 + \|F(\mathbf{x}^\star)\|_2\, \|\mathbf{x}^k - \mathbf{x}^\star\|_2\right)\,.$$

□

Corollary 6. *Under the assumptions (GNA) there exist constants $r > 0$ and $L_A^\star$ such that for $\|\mathbf{x}^\star - \mathbf{x}^k\|_2 \leq r$*

$$\|\mathbf{x}^\star - \mathbf{x}^{k+1}\|_2 \leq L\, \|F'(\mathbf{x}^\star)^+\|_2\, \|\mathbf{x}^\star - \mathbf{x}^k\|_2^2 + 2\, L\, L_A^\star \|F(\mathbf{x}^\star)\|_2\, \|\mathbf{x}^\star - \mathbf{x}^k\|_2\,. \tag{43}$$

Proof. The proof is very similar to that of Theorem 5. We use the same abbreviation $L_A^\star$ and directly apply Theorem 1 to replace (42) with

$$\|F'(\mathbf{x}^k)^+\, T(\mathbf{x}^k)\|_2 \leq 2\,\|(F'(\mathbf{x}^\star)^+\|_2\, \frac{L}{2}\, \|\mathbf{x}^k - \mathbf{x}^\star\|_2^2 = L\, \|F'(\mathbf{x}^\star)^+\|_2\, \|\mathbf{x}^k - \mathbf{x}^\star\|_2^2\,. \tag{44}$$

□

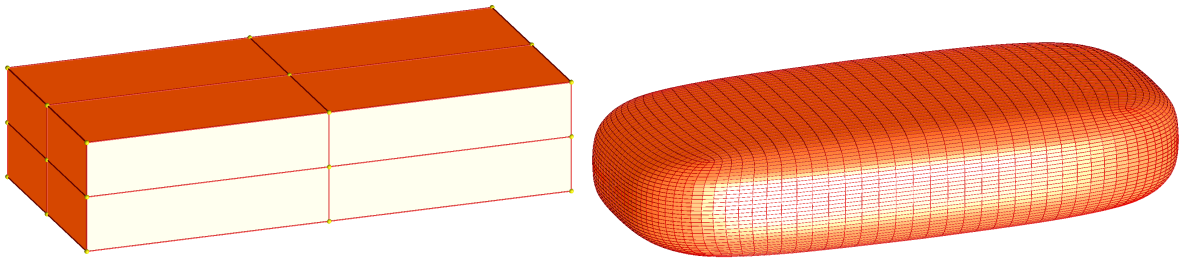

Figure 1: Recursive approximation with Catmull Clark Subdivision: Starting polyhedron (initial guess) and according limit surface.

As an example for the typical convergence behavior we use the approximation of an ellipsoid by a Catmull Clark Subdivision surface. Let us briefly describe the approach before we go into details. For the reparameterization of surfaces we project the limit points (surface points) at preselected parameter positions onto the target surface. This has the big advantage that we can pre-compute the $B_j(u_i, v_i)$ (see (5)). These coefficients are called masks. Then it is very easy to set up the system matrix for the overall linear problem. For more details on this topic see [2, 4, 13]. Figure 1 shows on the left hand side the initial 26 control points and 48 edges of the Catmull Clark Subdivision surface building the control polyhedron consisting of 24 faces. On the right we see the limit surface belonging to these control points. We use 98 sample points – 26 corresponding to the vertices, 24 to the face centers and 48 to the edge midpoints – for this approximation. Only six different masks are necessary for this strategy. For this poor approximation on the initial level there are several sample points for which the standard Gauß-Newton iteration does not converge – even for good starting values.

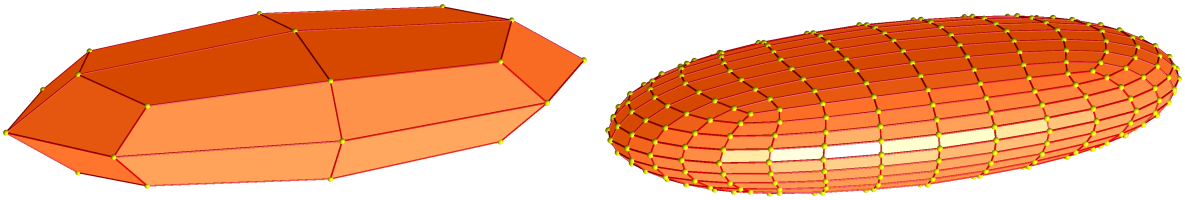

Figure 2: Recursive approximation with Catmull Clark Subdivision: The plots show the vertices and edges of the control polyhedron for the initial level and after two subdivision steps.

We used the Newton method to overcome this in the initial step. Already after the first re-computation of the control points due to the parameter corrections it converges for this example. On the first level of subdivision the convergence of the Gauß-Newton method is good. From the next level on it behaves like a quadratically convergent method. Only one step suffices for parameter correction. In Figure 2 we have plotted the control polyhedron for the initial level (after the approximation process on that level) and that after two subdivision steps. Finally in Figure 3 we see the surface for these control points.

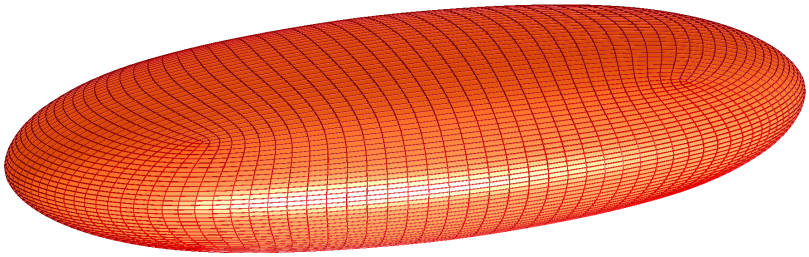

Figure 3: Recursive approximation with Catmull Clark Subdivision: Limit surface after two subdivision steps.

Let us be more precise on this example. The ellipsoid is given by

$$\mathbf{x} : (u, v) \mapsto \begin{pmatrix} a\,\cos(u)\,\cos(v) \\ b\,\sin(u)\,\cos(v) \\ c\,\sin(v) \end{pmatrix} \quad \text{with} \quad \begin{pmatrix} a \\ b \\ c \end{pmatrix} = \begin{pmatrix} 5 \\ 2 \\ 1 \end{pmatrix} . \tag{45}$$

The starting polyhedron in Figure 1 is the cuboid $[-5, 5] \times [-2, 2] \times [-1, 1]$. With this choice the six points on the three principle axes are interpolated for the limit surface. Determining the parameter values for the above-mentioned 98 sample points there are a lot of residuals larger than 0.1, some are even larger than 0.5.

Already on the first level the maximum relative error according to (7) (this is δ divided by the bounding box diagonal of the object: here $\delta/10.95$) is reduced by four iterations with the algorithm presented in

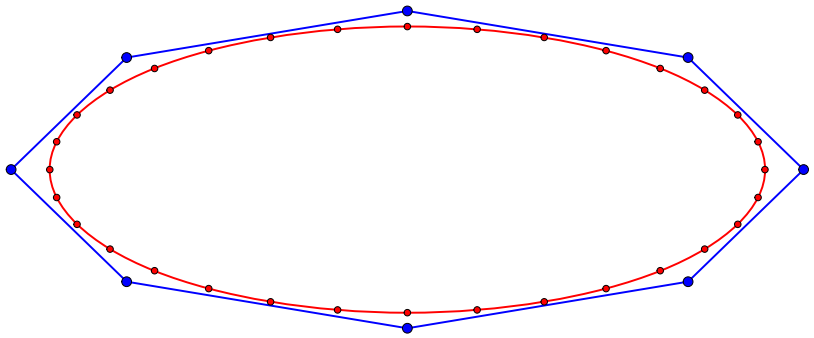

Figure 4: Recursive approximation with Catmull Clark Subdivision: Cut with $z = 0$ on the first level. Sample points for approximation and additional points for error estimation are shown.

[5] to $4e-4$ and the mean relative error ($\sum_{i=1}^{M} \delta_i / M$) to $1.6e-4$. This effect is visualized by a cut in Figure 4. To manage the control of the needed subdivision level we use a modified error function. We use the above mentioned points for minimization. Then we additionally use the approximation points from the next subdivision level and compute the δ_i for the extra points, too. All these points are shown in Figure 4. Normally the errors at the extra points are a little bit larger than at the positions we used for approximation. We stop the subdivision if all errors have fallen below the requested tolerance. Notice that in case of further subdivision with the supplementary points we already have all approximation points for the next level. For most applications coming from computer graphics we are allowed to use local refinements to reduce the number of control points in the final representation. Because the local strategy implicates (further) extraordinary vertices this is not allowed in the case of block structured grid generation. Further information on this topic can be found in [13, 12].

Exemplified we study the convergence behavior for a parameter correction near

$$\begin{pmatrix} u^\star \\ v^\star \end{pmatrix} = \begin{pmatrix} 0.1 \\ 0.1 \end{pmatrix} \quad \text{implying} \quad \mathbf{x}^\star = \mathbf{x}(u^\star, v^\star) = \begin{pmatrix} 4.950 \\ 0.1987 \\ 0.09983 \end{pmatrix} \quad \text{and} \quad \mathbf{n}^\star = \mathbf{n}(u^\star, v^\star) = \begin{pmatrix} 0.8713 \\ 0.2186 \\ 0.4393 \end{pmatrix}, \tag{46}$$

where $\mathbf{n}$ denotes the normal of the surface. We use $\mathbf{p} = \mathbf{x}^\star + \delta_i \, \mathbf{n}^\star$ with different δ_i's as sample point. Due to $\mathbf{x}^\star - \mathbf{p} = \delta_i \, \mathbf{n}^\star$ the residual for this point is δ_i. For $\delta_i = 0.25$ we get

$$\begin{pmatrix} \mathbf{x}_u^{\star\,2} & \mathbf{x}_u^{\star\,T} \mathbf{x}_v^\star \\ \mathbf{x}_u^{\star\,T} \mathbf{x}_v^\star & \mathbf{x}_v^{\star\,2} \end{pmatrix} = \begin{pmatrix} 4.167 & 0.2072 \\ 0.2072 & 1.237 \end{pmatrix} \tag{47}$$

$$(\mathbf{x}^\star - \mathbf{p})^T \otimes \begin{pmatrix} \mathbf{x}_{uu}^\star & \mathbf{x}_{uv}^\star \\ \mathbf{x}_{uv}^\star & \mathbf{x}_{vv}^\star \end{pmatrix} = \begin{pmatrix} 1.089 & 0 \\ 0 & 1.100 \end{pmatrix} \tag{48}$$

Thus for this residual δ_i we compute $\rho_{\max}(\Phi'(u^\star, v^\star)) = 0.8998 =: \tilde{q}$ and the Gauß-Newton method slowly converges. From the error estimations (29) and (30) of the **Contraction Mapping Theorem** we conclude with good starting values for $(u^\star, v^\star)$ the need of roughly 87 iterations to gain 3 more significant digits. Using Catmull Clark or cubic B-spline surfaces for approximation the residual δ_i at the current subdivision level is expected to be reduced by a factor of 16 in each next subdivision step. This is due to the fact that these surfaces have approximation order 4. Since the residual directly scales down the spectral radius this has significant influence on the convergence speed of the Gauß-Newton method. After one subdivision step we have $\rho_{\max}(\Phi'(u^\star, v^\star)) = 0.05624$ and we need only 5 iterations to get 6 more significant digits for $(u^\star, v^\star)$. One more subdivision later 3 iterations are sufficient for 7 more significant digits. In this example $\delta_i = 0.2778$ is the largest residual for which the Gauß-Newton method converges. These theoretical results are reflected in practice as shown in Table 1.

In all cases we use $(u^0, v^0) = (0, 0)$ as starting value and choose the sample point $\mathbf{p}$ according to (46) and the description thereafter. Thus δ_i is the residual for the minimizer $(u^\star, v^\star) = (0.1, 0.1)$. The tables show the

$\delta_i = 0.25$, $(u^0, v^0) = (0, 0)$			
	Gauß-Newton		**Newton**
k	r^k	$\tilde{q}^k$	r^k
0	1		1
1	0.7980	0.798	1.040e-01
2	0.3168	0.397	2.592e-03
3	0.3343	1.06	1.532e-06
4	0.2306	0.690	5.284e-13
5	0.2349	1.025	7.211e-17
6	0.1771	0.753	
7	0.1762	0.995	
10	0.1112	0.817	
40	4.651e-03	0.897	
80	6.836e-05	0.8998	
200	2.160e-10	0.8998	

$\delta_i = 0.25/16$, $(u^0, v^0) = (0, 0)$			
	Gauß-Newton		**Newton**
k	r^k	$\tilde{q}^k$	r^k
0	1		1
1	4.793e-02	0.0479	2.181e-01
2	2.257e-03	0.0471	1.918e-02
3	1.277e-04	0.0566	1.447e-04
4	7.178e-06	0.0562	8.174e-09
5	4.037e-07	0.0562	1.044e-16
6	2.270e-08	0.0562	
7	1.277e-09	0.0562	
8	7.181e-11	0.0562	
9	4.038e-12	0.0562	
10	2.270e-13	0.0562	
20	2.550e-17		

$\delta_i = 0.25/256$, $(u^0, v^0) = (0, 0)$			
	Gauß-Newton		**Newton**
k	r^k	$\tilde{q}^k$	r^k
0	1		1
1	0.00436	0.00436	2.356e-01
2	6.814e-06	0.00156	2.323e-02
3	1.799e-08	0.00264	2.224e-04
4	6.116e-11	0.00340	2.026e-08
5	2.143e-13	0.00350	8.246e-17
6	9.195e-16	0.00429	

$\delta_i = 1$, $(u^0, v^0) = (0, 0)$			
	Gauß-Newton		**Newton**
k	r^k	$\tilde{q}^k$	r^k
0	1		1
1	3.199	3.20	4.443e-02
2	0.8136	0.254	1.958e-04
3	3.46	4.26	3.786e-09
4	0.9301	0.269	1.000e-16
100	0.7648	1.10	
200	1.768	1.42	

Table 1: Convergence of the Gauß-Newton and the Newton method for different residuals.

relative error $r^k = \|(u^k, v^k) - (u^\star, v^\star)\|_2 / \|(u^\star, v^\star)\|_2$ for the iterations with the Newton and the Gauß-Newton method. Furthermore for the Gauß-Newton method we show the local contraction $\tilde{q}^k = \|\mathbf{e}^k\|_2 / \|\mathbf{e}^{k-1}\|_2$. Our choice of the starting value causes the relative error to be 1 for it. Hence we can easily read the number of attained significant digits from the relative error r^k.

We start with a residual $\delta_i = 0.25$ (upper left sub-table of Table 1). Our theoretical considerations above let us expect slow convergence of the Gauß-Newton method for this case. This is clearly affirmed. As expected $\tilde{q}^k$ coincides with $\tilde{q} = \rho_{\max}(\Phi'(u^\star, v^\star))$ in the final state of the iteration.

If we assume the previous δ_i is the residual after parameter corrections and recomputations of the control points, we expect the residual $\delta_i = 0.25/2^4$ on the next subdivision level. These results are shown in the next sub-table. At this level the Gauß-Newton method already gains more than one significant digit in each step. In practice only for the initial guess the starting values for the (u_i, v_i) values are poor. Already on the initial level the changes due to the recalculation of the control points are moderate. Those on the next level (due to subdivision) are small in low levels and tiny in high levels. Thus we start with approximations that qualitatively match the first iterate of the Gauß-Newton method from the sub-tables for $\delta_i \in \{0.25/16, 0.25/256\}$.

During the iterations with methods of quadratic or superlinear convergence we can use the error estimate $\|\mathbf{x}^\star - \mathbf{x}^k\| \approx \|\mathbf{x}^{k+1} - \mathbf{x}^k\|$ in the final state for $\mathbf{x}^k$. This for instance is implied by (23) for quadratic convergence. When $\alpha \|\mathbf{e}^k\|_2 < 10^{-2}$ the next iteration vector has at least two more significant digits. Thus $\|\mathbf{x}^{k+1} - \mathbf{x}^k\|$ matches $\|\mathbf{e}^k\|$ with two significant digits. That is more than enough for an error estimator. In Table 1 the behavior is reflected not only for the Newton Method but additionally for the Gauß-Newton method if the contraction number is small. In fact, if on the actual subdivision level the Gauß-Newton

method converges at a point with parameters $(u^\star, v^\star)$ then the contraction number q must be less than 1. On the next subdivision level we expect a contraction number less than $1/16$ for surface classes with convergence order 4. (This is only true if the surface we approximate is in C^4 in that area.) Thus we gain at least one more significant digit with each iteration. If we tag the points where the Gauß-Newton method converges on the current level we can use the just mentioned error estimates on the next level for these points with the Gauß-Newton method, too.

We have designed our algorithms for parameter corrections applying the knowledge from the above analysis. In all test examples we made the following observations with adaptive subdivision. During the initial state almost always the Newton method (with optional damping) works fine. Only for very complex problems with high curvature we had to switch to the Nelder-Mead algorithm. After the first subdivision step only for very few sample points the Gauß-Newton method does not converge. Due to the very good starting values from the previous level (we use interpolation for the points in between) and the smaller residuals on the following subdivision levels one Gauß-Newton step is enough to get a sufficient accuracy for the parameters.

Let us conclude this section with some remarks on the assumptions for our analysis in the special context of parameter corrections. For regular parameterizations (here $\mathbf{x} = (u, v)$) $F'(\mathbf{x})$ has full rank everywhere. Furthermore for orthogonal parameterization $F'(\mathbf{x})^T F'(\mathbf{x})$ is a diagonal matrix. In the proofs we chose the constants for the sake of simplicity. They can be relaxed in some cases. The main observation for the convergence behavior of the Gauß-Newton method is the proportionality to curvature and the norm of the current residual. The latter one is reduced on each next subdivision level. The first one keeps nearly constant locally.

4. Conclusion and future research

We have presented and analyzed a novel and fast adaptive approximation approach for curves and surfaces. It can be used for the construction of smooth surfaces from point clouds as well as for the reparameterization of given surfaces. The methods are not restricted to be applied to Bézier or B-spline surfaces. They can be used for subdivision surfaces as well. In this paper the focus has been set on an analytical understanding of the nonlinear least squares methods for parameter correction. It turned out that in relation to efficiency different methods have to be used in different stages. The choice has to be dependent on the local curvature and the current residual thereabouts. Thus even on each subdivision level we might use different methods in different areas.

There is still a large amount of work left for future research. Line search methods have the potential to force convergence for the Gauß-Newton method and can assist the choice for it in an earlier state. We want to further improve the overall iteration by a better adaptation of the error functional to the respective residuals. Furthermore, we will test our algorithms with other surface classes.

References

[1] Å. Björck, Numerical Methods in Matrix Computations, ISBN 978-3-319-05089-8, Springer, 2015.

[2] K.-H. Brakhage, Modified Catmull-Clark methods for modelling, reparameterization and grid generation, in: S. Harms, N. Wolpert (Eds), *Proceedings of the 2. Internationales Symposium Geometrisches Modellieren Visualisieren und Bildverarbeitung, 2007*, ISBN 3-9808066-9-3, pp109-114.

[3] K.-H. Brakhage, Ph. Lamby, Application of B-spline techniques to the modeling of airplane wings and numerical grid generation, *CAGD 25(9)* (2008) 738-750.

[4] K.-H. Brakhage, Grid generation and grid conversion by subdivision schemes in: B. K. Soni, F. Guibalt, R. Cameraro (Eds), *Proceedings of the 11th International Conference on Numerical Grid Generation in Computational Field Simulations, 2009*, May 24-28, 2009, Montreal, Canada.

[5] K.-H. Brakhage, Fast approximation methods for fitting surfaces to unorganized point clouds, *IGPM Preprint (2015)*, https://www.igpm.rwth-aachen.de/forschung/preprints/438, accepted for publication in: MASCOT15 Proceedings - IMACS Series in Computational and Applied Mathematics.

[6] K. Cheng, W. Wang, H. Qin, K.-Y. Wong, H. Yang, Y. Liu, Fitting subdivision surfaces to unorganized point data using SDM, in: *12th Pacific Conference on Computer Graphics and Applications, 2004*, IEEE, pp16-24.

[7] G. Farin, *Curves and Surfaces for CAGD, A Practical Guide*, The Morgan Kaufmann Series in Computer Graphics and Geometric Modeling, 5th Edition, 2002.

[8] M. Kass, A. Witkin, D. Terzopoulos, Snakes: active contour models, *Int. J. Comput. Vis. 1(4)* (1988) 321-332.
[9] J.A. Nelder, R. Mead, A simplex method for function minimization, *Comput. J. 7* (1965) 308-313.
[10] J. Peters, U. Reif, *Subdivision Surfaces*, Series: Geometry and Computing, Vol. 3, Springer, 2008.
[11] H. Pottmann, M. Hofer, Geometry of the squared distance function to curves and surfaces, in: H.C. Hege, K. Polthier (Eds), *Visualization and Mathematics III*, 2003, pp223-244.
[12] M. Rom, K.-H. Brakhage, Reparametrization and volume mesh generation for Computational Fluid Dynamics using modified Catmull-Clark methods, in: M. Floater, T. Lyche, M.-L. Mazure, K. Mørken, L.L. Schumaker (Eds), *Mathematical Methods for Curves and Surfaces*, Lecture Notes in Computer Science 8177, Springer 2014, pp425-441.
[13] M. Rom, B-Spline Volume Meshing for CFD Simulations Using Modified Catmull-Clark Methods (Ph.D. Thesis), RWTH Aachen, ISBN 978-3-8440-3421-9, Shaker Verlag Aachen, 2015.
[14] J. Stam, Exact evaluation of Catmull-Clark Subdivision surfaces at arbitrary parameter values, in: *Proc. SIGGRAPH 98*, pp395-404.
[15] J. Stam, Exact evaluation of Loop Subdivision surfaces, in: *SIGGRAPH CDROM Proceedings 98*.