ALGORITHMS FOR CONSTRAINED MINIMIZATION OF QUADRATIC FUNCTIONS – GEOMETRY AND CONVERGENCE ANALYSIS

HELMUT POTTMANN AND MICHAEL HOFER

1. Introduction

The motivation for the present work comes from our recently published paper [2] on the design of motions constrained by a contacting surface pair. A central part of that paper is an extension of variational subdivision in the unrestricted case [4, 5] to the design of curves on surfaces. These surfaces may lie in a higher dimensional space $\mathbb{R}^d$, and can have arbitrary dimension $2 < k < d$. Since [2] did just use, but not study the algorithm and its convergence, we will now do it, but in a much more general setting. In this way, the algorithm is useful for the solution of a variety of geometric optimization problems.

In the following, we present and analyze an algorithm for the computation of the point $\mathbf{p}^*$ in an m-dimensional surface $\Phi \subset \mathbb{R}^n$, which is closest to a given point $\mathbf{p} \in \mathbb{R}^n$. Of course, this point $\mathbf{p}^*$ is the footpoint of a normal from $\mathbf{p}$ onto $\Phi \subset \mathbb{R}^n$.

This problem arises in optimization in the context of minimization of a quadratic function

$$F : \mathbb{R}^n \to \mathbb{R},\ F(\mathbf{x}) = \mathbf{x}^T \cdot Q \cdot \mathbf{x} + 2\mathbf{q}^T \cdot \mathbf{x} + q,$$

with a symmetric positive definite matrix Q, under a set of constraints $c_k(\mathbf{x}) = 0, k = 1, \ldots, n-m$. The solution set of the constraint equations is a surface $\Phi \subset \mathbb{R}^n$. We assume that Φ has dimension m and that it is smooth in a neighborhood of the minimum $\mathbf{p}^*$. The geometric interpretation of this problem views the matrix Q as matrix of the inner product to a Euclidean metric in $\mathbb{R}^n$. With $\mathbf{p} = -Q^{-1} \cdot \mathbf{q}$ as minimizer of F in $\mathbb{R}^n$, the function is, up to an unimportant additive constant, equivalent to

$$F(\mathbf{x}) = (\mathbf{x} - \mathbf{p})^T \cdot Q \cdot (\mathbf{x} - \mathbf{p}),$$

i.e., the squared distance $\|\mathbf{x} - \mathbf{p}\|^2$ between points $\mathbf{p}$ and $\mathbf{x}$ in the mentioned Euclidean norm. Therefore, the minimizer of F, restricted to Φ, is the closest point $\mathbf{p}^* \in \Phi$ to $\mathbf{p}$, and thus a normal footpoint.

For the following analysis, it is convenient to introduce in $\mathbb{R}^n$ a Cartesian coordinate system with respect to the Euclidean norm given by F. This is the

Key words and phrases. optimization, shortest distance computation, orthogonal projection onto a manifold.

same as assuming for the moment that Q is the identity matrix. The inner product $\mathbf{x}^T \cdot \mathbf{y}$ of two vectors will then also be written as $\mathbf{x} \cdot \mathbf{y}$.

2. Normal footpoint via Newton algorithm

The standard way to solve the present footpoint problem is the use of Newton's algorithm. Let us briefly review this approach, since it shows why we want to modify it for certain practical problems.

It is sufficient here to assume that we are given a parametrization $\mathbf{c}(u_1, \ldots, u_m) = \mathbf{c}(\mathsf{u})$ of the surface Φ. A normal footpoint $\mathbf{p}^* = \mathbf{c}(\mathsf{u}^*)$ from $\mathbf{p}$ onto Φ must solve the following set of equations,

$$f_k(\mathsf{u}) := (\mathbf{c}(\mathsf{u}) - \mathbf{p}) \cdot \mathbf{c}_{,k}(\mathsf{u}) = 0, \quad k = 1, \ldots, m. \tag{1}$$

Here, $\mathbf{c}_{,k} = \partial \mathbf{c} / \partial u_k$. Solving the nonlinear system of equations $\mathsf{f} = (f_1, \ldots, f_m) = 0$ in a Newton iteration requires linearization at the current iterate u_c,

$$\mathsf{f}(\mathsf{u}) = \mathsf{f}(\mathsf{u}_c) + D\mathsf{f}(\mathsf{u}_c) \cdot (\mathsf{u} - \mathsf{u}_c) = 0.$$

With the first derivative matrix $D\mathsf{f}$, assumed to be regular, the Newton algorithm obtains the next iterate u_+ as

$$\mathsf{u}_+ = \mathsf{u}_c - (D\mathsf{f})^{-1}(\mathsf{u}_c) \cdot \mathsf{f}(\mathsf{u}_c). \tag{2}$$

The first derivative matrix is

$$D\mathsf{f} = (d_{ij}), \quad d_{ij} = (\mathbf{c} - \mathbf{p}) \cdot \mathbf{c}_{,ij} + \mathbf{c}_{,i} \cdot \mathbf{c}_{,j}. \tag{3}$$

Note that $\mathbf{c}_{,i} \cdot \mathbf{c}_{,j} = g_{ij}$ are the elements of the first fundamental form of the surface representation $\mathbf{c}$. If $D\mathsf{f}$ is Lipschitz continuous in a neighborhood of u^*, a Newton iteration converges quadratically. This means that the current error $\mathsf{e}_c = \mathsf{u}_c - \mathsf{u}^*$ is related to the error after the iteration via

$$\|\mathsf{e}_+\| \leq C \|\mathsf{e}_c\|^2. \tag{4}$$

Geometric interpretation of cases, where one has only linear (or superlinear) convergence, is still missing. An example for linear convergence is in case of a planar curve, where $\mathbf{p}$ is the curvature center of the curve at the footpoint.

3. Footpoint computation with a projected gradient algorithm

We see that Newton's method for footpoint computation involves second order derivatives of Φ. There are a number of applications where this is too time consuming or where Φ is given in a form such that second order information is not available and – by the nature of the data – hard to estimate. Therefore, we will now describe a projected gradient algorithm together with a convergence analysis of it. The geometric interpretation of the error estimates leads us to a steplength control strategy of the resulting linearly convergent algorithm. It even offers us the chance to an optimized steplength choice which results in nearly quadratic convergence.

The iterative algorithm which we propose and investigate consists in each iteration of the following two steps (see Fig. 1 (**a**)).

(1) Compute the tangent space T^m of Φ at the current approximation $\mathbf{x}_c$ of the footpoint and project the point $\mathbf{p}$ orthogonally into T^m to get the point $\mathbf{p}_T$.
(2) Compute an appropriate steplength s and project $\mathbf{x}_s := \mathbf{x}_c + s(\mathbf{p}_T - \mathbf{x}_c)$ in a direction transversal to Φ onto Φ ; this yields the next approximation $\mathbf{x}_+$ of the footpoint.

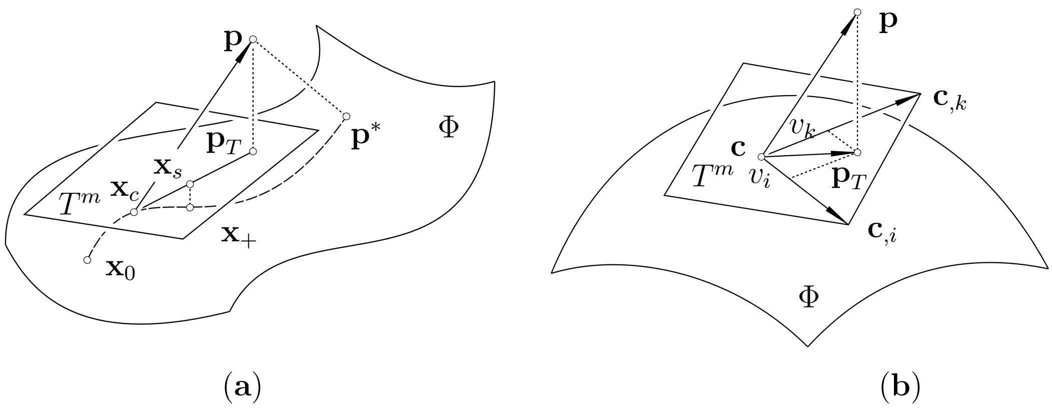

FIGURE 1. (**a**) Footpoint computation with a projected gradient algorithm. (**b**) The tangentspace T^m at $\mathbf{c}$.

Since $\mathbf{p}-\mathbf{x}_c$ is the negative gradient of the squared distance function $\frac{1}{2}\|\mathbf{x}-\mathbf{p}\|^2$ at $\mathbf{x}_c$, the method is in fact a familiar projected gradient algorithm [1]. However, we do not require convexity of Φ, which is often done in standard texts on optimization. We will carefully investigate this scheme from a geometric viewpoint and then apply it to the numerical solution of several geometric optimization problems.

3.1. **Computation for a parametrized surface Φ and convergence analysis.** We will first study the convergence behaviour. For that, it is sufficient to work with a parametrically given surface $\mathbf{c}(\mathsf{u})$.
Step 1. The tangent space T^m at $\mathbf{c}(\mathsf{u}_c)$ is spanned by the m vectors $\mathbf{c}_{,i}(\mathsf{u}_c)$, $i = 1, \ldots, m$. The unknown coordinates v_i of the projection $\mathbf{p}_T$ of $\mathbf{p}$ in this basis (see Fig. 1 (**b**)) follow from the linear system

$$[\mathbf{c} - \mathbf{p} + \sum_{i=1}^{m} v_i \mathbf{c}_{,i}] \cdot \mathbf{c}_{,k} = 0, \quad k = 1, \ldots, m. \tag{5}$$

Evaluation is at $\mathsf{u} = \mathsf{u}_c$. With the first derivative matrix $D\mathbf{c}$ and the matrix $G = (g_{ik})$ of the first fundamental form, the coordinate vector $\mathsf{v} = (v_1, \ldots, v_m)$ is given by

$$\mathsf{v} = G^{-1}(\mathsf{u}_c) \cdot D\mathbf{c}(\mathsf{u}_c) \cdot (\mathbf{p} - \mathbf{c}(\mathsf{u}_c)). \tag{6}$$

Step 2. The second step of the algorithm, namely projection onto Φ shall be performed by linearization of $\mathbf{c}$ at u_c. This means that the new parameter value u_+ is computed with an appropriate steplength s as

$$\mathsf{u}_+ = \mathsf{u}_c + s\mathsf{v}. \tag{7}$$

For the *convergence analysis* we assume that the footpoint occurs to $\mathsf{u}^* = 0$ and lies at the origin, $\mathbf{c}(\mathsf{u}^*) = 0$. Hence, $\mathbf{p} \cdot \mathbf{c}_{,i}(0) = 0$. A Taylor expansion of the surface $\mathbf{c}(\mathsf{u}) = (c_1(\mathsf{u}), \ldots, c_m(\mathsf{u}))$ at the footpoint $\mathbf{c}(0) = 0$ is given by

$$c_i(\mathsf{u}) = \nabla c_i(0) \cdot \mathsf{u} + \frac{1}{2}\mathsf{u}^T \cdot \nabla^2 c_i(0) \cdot \mathsf{u} + R_{i,3}. \tag{8}$$

Here, ∇ and ∇^2 denote gradient and Hessian, respectively, and $R_{i,3}$ is the cubic remainder term in which third order derivatives appear. ¿From this we see that the m coordinates of the vector $D\mathbf{c}(\mathsf{u}) \cdot (\mathbf{p} - \mathbf{c}(\mathsf{u}))$ possess the Taylor expansion

$$\mathbf{c}_{,k} \cdot (\mathbf{p} - \mathbf{c}) = \sum_{i=1}^{m} (\mathbf{p} \cdot \mathbf{c}_{,ik} - g_{ik}) u_i + R_{k,2}.$$

The quadratic remainder term contains derivatives of $\mathbf{c}$ up to third order. Note that $\mathbf{p}$ equals $d\mathbf{n}$, with $d \geq 0$ as distance to the footpoint $\mathbf{c}(0) = 0$ and $\mathbf{n}$ as a unit surface normal vector there. The matrix

$$L = (l_{ik}) = (\mathbf{n} \cdot \mathbf{c}_{,ik}),$$

is the matrix of the second fundamental form in case of a hypersurface Φ ($m = n-1$). We will interpret it in the other cases later. With this notation, the vector v of (6) is

$$\mathsf{v} = G^{-1} \cdot (d\,L - G) \cdot \mathsf{u}_c = d(G^{-1} \cdot L) \cdot \mathsf{u}_c - \mathsf{u}_c. \tag{9}$$

$W := G^{-1} \cdot L$ is the matrix of the Weingarten map at the footpoint in the hypersurface case. The updated parameter point u_+ for the footpoint becomes

$$\mathsf{u}_+ = \mathsf{u}_c + s\mathsf{v} = [(1-s)I + sdW)] \cdot \mathsf{u}_c + R_2. \tag{10}$$

Assuming bounded derivatives of $\mathbf{c}$ up to third order in a neighborhood of the footpoint, the remainder term R_2 can be bounded by $C\|\mathsf{u}_c\|^2$. Since the minimum is at $\mathsf{u}^* = 0$, error and parameter vectors agree, and we find the error estimate

$$\|\mathsf{e}_+\| \leq \|(1-s)I + sdW)\|_2 \, \|\mathsf{e}_c\| + C\, \|\mathsf{e}_c\|^2. \tag{11}$$

Before we give an interpretation of the spectral norm $\|(1-s)I + sdW)\|_2$, we need a geometric meaning of the matrices L and W. To obtain such an interpretation, we consider the $m+1$-dimensional affine space N^{m+1} spanned by $\mathbf{p}$ and the tangent space T^m at the footpoint $\mathbf{p}^*$. Now, the surface Φ is projected orthogonally into N^{m+1}. The resulting surface Φ^N is a hypersurface in N^{m+1} and has a parameterization of the form

$$\mathbf{c}^N(\mathsf{u}) = \mathbf{c}(\mathsf{u}) + \mathbf{c}^*(\mathsf{u}), \tag{12}$$

where $\mathbf{c}^*$ lies in the orthogonal complement of N^{m+1}. Note that $\mathbf{p}^*$ is also footpoint of $\mathbf{p}$ to Φ^N within N^{m+1}. Because of $\mathbf{n} \cdot \mathbf{c}_{,ik}(0) = \mathbf{n} \cdot \mathbf{c}^N_{,ik}(0)$, the matrix L is the matrix of the second fundamental form of the projected surface at the footpoint $\mathbf{p}^* = \mathbf{c}(0) = \mathbf{c}^N(0)$. By the agreement of the tangent spaces of Φ and Φ^N at $\mathbf{p}^*$, we have $\mathbf{c}^*_{,i}(0) = 0$, and thus the first fundamental forms of Φ and Φ^N at 0 agree. Hence, the matrix $W = G^{-1} \cdot L$ is the matrix of the Weingarten map of Φ^N at $\mathbf{p}^*$. Its eigenvalues are the principal curvatures κ_i of Φ^N at the footpoint $\mathbf{p}^*$.

Remark 1. For a visualization of the involved curvatures at the footpoint, we consider the simplest case, namely a curve $\mathbf{c}(u)$ in $\mathbb{R}^3$, see Fig. 2. The arising curvature at the footpoint $\mathbf{p}^*$ of a point $\mathbf{p}$ is obtained by projection of the curve into the plane N^2 spanned by $\mathbf{p}$ and the tangent T of $\mathbf{c}$ at $\mathbf{p}^*$. This is also a tangent plane of the connecting cone Γ of $\mathbf{p}$ and $\mathbf{c}$. It is well-known that the curvature of the orthogonal projection $\mathbf{c}^N$ of $\mathbf{c}$ onto N^2 is the geodesic curvature of the curve $\mathbf{c} \subset \Gamma$ at $\mathbf{p}^*$. Moreover, it is the curvature of the curve $\mathbf{c}^0$ which is obtained by development (isometric mapping) of the cone Γ into a Euclidean plane. Since this development shows all steps of the algorithm without distortion, it is not surprising that the convergence analysis of the case of a space curve (in fact, a curve in a space of arbitrary dimension) is reducible to the case of a planar curve.

Let us now study the case, where we work with a full step, i.e., $s = 1$. Then, the error estimate (11) becomes

$$\|\mathsf{e}_+\| \leq d \, \|W\|_2 \, \|\mathsf{e}_c\| + C \, \|\mathsf{e}_c\|^2 \leq d \, |\kappa_{max}| \, \|\mathsf{e}_c\| + C \, \|\mathsf{e}_c\|^2. \tag{13}$$

Here, $|\kappa_{max}|$ denotes the maximum absolute value of the principal curvatures of Φ^N at the footpoint. Now $\rho_{min} = 1/|\kappa_{max}|$ is the minimal principal curvature radius of Φ^N at $\mathbf{p}^*$. The error estimate (13) shows that we have linear convergence for $d \, |\kappa_{max}| < 1$.

Theorem 1. *If the distance d of $\mathbf{p}$ to Φ is less than the minimal principal curvature radius of Φ^N at the footpoint $\mathbf{p}^*$, the proposed projected gradient algorithm converges linearly, even if we take a full step, $s = 1$. Here, Φ^N denotes the orthogonal projection of the surface Φ into the affine space spanned by $\mathbf{p}$ and the tangent space of Φ at the normal footpoint $\mathbf{p}^*$.*

For a complete proof of the theorem, we still need to show that another kind of projection in step 2 than the one via linearization of the parametric representation does not change the convergence rate. This will be done in subsection 3.2.

Clearly, the projected gradient algorithm works very well for points $\mathbf{p}$ sufficiently close to the surface. For $d = 0$, we even have quadratic convergence. This

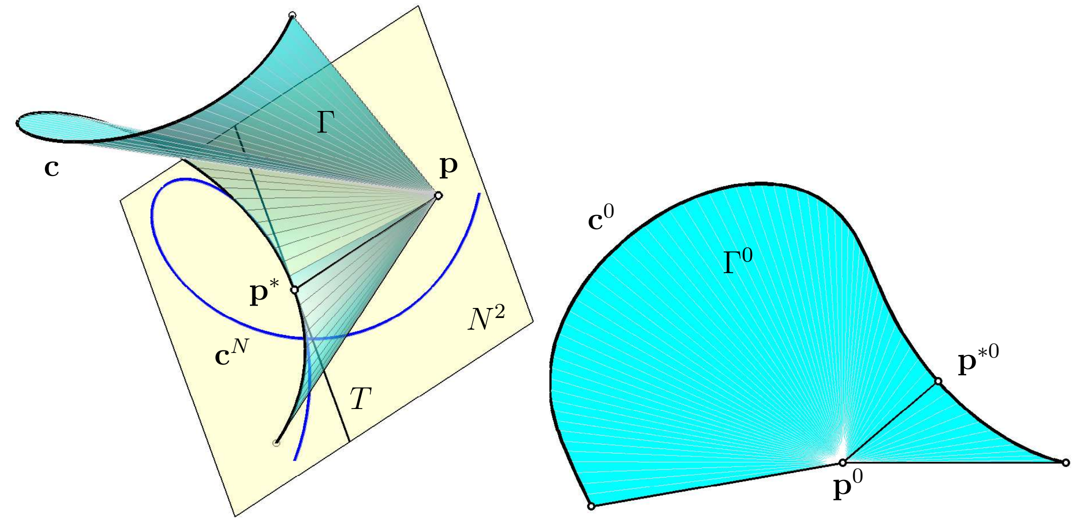

FIGURE 2. (**a**) Connecting cone Γ of $\mathbf{p}$ and the space curve $\mathbf{c}$. (**b**) Development Γ^0 of Γ and $\mathbf{c}^0$ of $\mathbf{c}$ in $\mathbb{R}^2$.

may seem a strange situation, but an example for an application is the following: We are given a parametric surface $\mathbf{c}(\mathsf{u})$ and a point $\mathbf{p}$, which is known to lie on $\mathbf{c}$, but its parameter values are not yet known. The present algorithm can solve this inversion problem with quadratic convergence, even if we do not use a Newton algorithm.

For larger distances, we cannot work with the full step $s = 1$. We have to choose s such that the constant $C_1 := \|(1-s)I + sdW)\|_2$ in the error estimate (11) is less than 1. If κ is an eigenvalue of W to the eigenvector w, we have $[(1-s)I + sdW] \cdot \mathsf{w} = (1-s+sd\kappa)\mathsf{w}$, and thus $1-s+sd\kappa$ is an eigenvalue of the matrix $M = (1-s)I + sdW$ to the same eigenvector. Hence, $\|M\|_2 < 1$ requires

$$-1 < 1 - s + sd\kappa_i < 1,$$

for all principal curvatures κ_i of Φ^N at $\mathbf{p}^*$. We have chosen the normal vector $\mathbf{n}$ such that $d > 0$, but we cannot assume positive values of κ_i. For the following, it is convenient to distinguish between two cases.

(**a**) Assume $d\kappa_{max} > 1$. Now $1 - s + sd\kappa_{max} > 1 - s + s = 1$ and no choice of s would guarantee convergence. Fortunately, this case does not arise in our setting: We argue in N^{m+1}. Here, $\mathbf{p}$ would be on the same side of the tangent hyperplane T^m of Φ^N as the principal curvature center $\mathbf{k}_{max}$ to the largest principal curvature κ_{max}. However, $\mathbf{p}$ would be at a larger distance to $\mathbf{p}^*$ than $\mathbf{k}_{max}$. This means that d cannot be the globally smallest distance. In other words, $\mathbf{p}^*$ is a footpoint, but not the one to the smallest distance. We are not interested in such a footpoint here.

(b) Let $d\kappa_{max} < 1$. We set $d\kappa_{max} - 1 = -K$ with $K > 0$. Now, any choice of s with

$$0 < s < \frac{2}{K} \tag{14}$$

gives a constant $C_1 < 1$, i.e., linear convergence.

Theorem 2. *With the notations from Theorem 1, and a choice of the steplength s, which satisfies*

$$0 < s < \frac{2}{|d\kappa_{max} - 1|}, \tag{15}$$

the proposed projected gradient algorithm exhibits linear convergence.

Before we address the question, whether we can actually achieve quadratic convergence with a clever choice of s, we focus on the influence of step 2 of the algorithm.

3.2. **The projection in step 2 does not change the convergence rate.** Whereas the first step in our algorithm is purely geometric, the version of the second step which has been presented so far is just applicable if the surface Φ is given in parametric representation. This may be very undesirable, for example, if we are minimizing a quadratic function under equality constraints; those define Φ as an intersection of implicit surfaces (level sets).

Let us now describe what we consider an *admissible projection for step 2*: We assume that a differentiable m-parametric set P of affine spaces of dimension $n - m$ has been defined. These affine spaces shall be transversal to Φ and form a fibration of a neighborhood of Φ. We could call P a family of *pseudonormal spaces*. Now projection is performed with these pseudonormal spaces: To project $\mathbf{x}$ onto Φ, intersect the unique element $S \in P$, which contains $\mathbf{x}$, with the surface Φ.

Before we proceed with the discussion, let us mention a few examples for an admissible projection:

(1) For $m = n - 1$, i.e., a hypersurface $\Phi \subset \mathbb{R}^n$, we project parallel to a fixed line L; the only requirement is that L is transversal to all tangent hyperplanes of Φ in the neighborhood of the footpoint which is considered by the algorithm.
(2) For a curve Φ, take P as a family of parallel hyperplanes, which are not tangent to the curve in the required neighborhood of the footpoint.
(3) For an implicitly defined hypersurface $c(\mathbf{x}) = 0$, project parallel to the gradient ∇c. Since we need regularity of the surface in a neighborhood of the footpoint anyway, it can be expected that the required gradients do not vanish there. Note that the algorithm would only use gradients at the points $\mathbf{p}_T$. These can be viewed as samples of the gradient field along some hypersurface Ψ close to Φ, and thus it fulfils the requirements. It

might be sufficient to choose a fixed direction, given by the gradient at some initial point $\mathbf{p}_0$ sufficiently close to the footpoint.

(4) For a surface Φ, given as intersection of implicit surfaces $c_k(\mathbf{x}) = 0,\ k = 1, \dots, n-m$, we may use a projection space spanned by the $n-m$ gradients ∇c_k.

We would like to show that the projection of a tangent vector v differs from the projection via linearization only in the second order. This means that the difference between the two kinds of projections is $O(\|\mathsf{v}\|^2)$.

For a proof, we use a Taylor expansion of $\mathbf{c}(\mathsf{u})$ at some point, say $\mathsf{u} = 0$,

$$\mathbf{c}(\mathsf{u}) = \mathbf{c}(0) + \sum_i \mathbf{c}_{,i}(0) u_i + \frac{1}{2} \sum_{j,k} \mathbf{c}_{,jk}(0) u_j u_k + R_3(\mathsf{u}).$$

A point $\mathbf{p}_T$ in the tangent plane of $\mathbf{c}(0)$, namely $\mathbf{c}(0) + \sum_i \mathbf{c}_{,i}(0) v_i$ is projected to some point $\mathbf{x}_+$ of the surface,

$$\mathbf{x}_+ = \mathbf{c}(0) + \sum_i \mathbf{c}_{,i}(0) v_i + \mathbf{t}(v_1, \dots, v_m).$$

If we vary v, the vector $\mathbf{t}$ depends smoothly on v and for sufficiently small $\|\mathsf{v}\|$, it is transversal to the tangent space at $\mathbf{c}(0)$, which is spanned by the vectors $\mathbf{c}_{,i}(0)$. The point $\mathbf{x}_+$ is some surface point, so there exists u such that

$$\mathbf{c}(0) + \sum_i \mathbf{c}_{,i}(0) u_i + \frac{1}{2} \sum_{j,k} \mathbf{c}_{,jk}(0) u_j u_k + R_3(\mathsf{u}) = \mathbf{c}(0) + \sum_i \mathbf{c}_{,i}(0) v_i + \mathbf{t}(\mathbf{v}).$$

The second derivative vectors $\mathbf{c}_{,jk}(0)$ have a tangential component $\sum \lambda_{ijk} \mathbf{c}_{,i}(0)$; the rest is transversal. Hence, the tangential components in the vector equation above are

$$u_i - v_i + \frac{1}{2} \sum_{j,k} \lambda_{ijk} u_j u_k + R_{i,3},$$

with some cubic remainder term $R_{i,3}$. This shows that the difference between linearization and projection satisfies

$$\|\mathsf{u} - \mathsf{v}\| = O(\|\mathsf{u}\|^2). \tag{16}$$

Hence, it does not influence the error estimate for our optimization algorithm.

3.3. **Stepsize selection for nearly quadratic convergence.** Let us start with the case of a planar curve, with κ being the curvature at the footpoint. Then the choice of s with

$$s = \frac{1}{|d\kappa - 1|} \tag{17}$$

removes the linear part in the error estimate and we get *quadratic convergence*. Of course, in practice, we do not know d and κ, so that we have to estimate these values from the known data at the previous iterates. Since we want to avoid second order derivatives, we will introduce errors in the estimates of $d\kappa$. Although this will not give quadratic convergence from a theoretical viewpoint, it can be expected to yield a very good performance of the algorithm. Note that quadratic convergence in Newton's algorithm is rooted in the curvature computation which is involved in the linear system for each step (see (3)).

As soon as we have three consecutive points $\mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3$ on our way towards the footpoint, we may estimate $d\kappa$ with help of a circle through the points $\mathbf{x}_k$. In Algorithm 1 we use tangents of the developed curve, which follow with help of the footpoints $\mathbf{p}_T$, to estimate $d\kappa$. Then s can be adapted to yield fast convergence.

For a general surface Φ, error estimates of the type (11), which are based on the norm of the matrix $(1-s)I + sdW$, are worst case estimates. They do not consider the *direction, in which the footpoint is approached.* If we consider the direction, we can do better.

Let us first look at the algorithm used with very small stepsize s. It can be seen as a projected gradient flow: At each point $\mathbf{x} \in \Phi$ in a neighborhood of the footpoint $\mathbf{p}^*$, compute the tangent vector $\mathbf{v} = \mathbf{p}_T - \mathbf{x}$. The path of the algorithm is then a solution of the differential equation $\mathbf{x}' = \mathbf{v}(\mathbf{x})$. The behaviour of these curves near the footpoint is governed by the Jacobian of $\mathbf{v}$ there. It is easy to show that its eigenvectors are the principal curvature directions of the surface Φ^N, and the eigenvalues are $d\kappa_i - 1 < 0$. Therefore, almost all curves are approaching the footpoint in direction of the smallest value of $|d\kappa_i - 1|$. Thus, this algorithm adapts itself to a direction, which would allow the largest steps according to equation (17).

Our goal now is to have the iterates arranged along a curve, but to use optimal steplengths to achieve nearly quadratic convergence. For this, we consider a curve $\mathbf{c}_f$ on Φ, containing the individual iterates $\mathbf{x}_i$ for the footpoint in our projected gradient algorithm with sufficiently small stepsize s. The footpoint $\mathbf{p}^*$ is also a footpoint on the curve $\mathbf{c}_f$. Moroever, we can view the algorithm as the footpoint algorithm applied to curve $\mathbf{c}_f$, although the latter is not known. In view of Remark 1, we can look at the planar development of the cone Γ_f, which connects $\mathbf{p}$ with $\mathbf{c}_f$. An approximation of this development in a neighborhood of the current iterate $\mathbf{x}_i$ can be computed with the following algorithm.

Algorithm 1. *Estimates d and κ via a planar development of a part of the cone Γ_f (see Fig. 3).*

(1) *In the first step of the algorithm we compute the distances*

$$a_i = ||\mathbf{p}_{T,i} - \mathbf{x}_i||, \quad b_i = ||\mathbf{p}_{T,i} - \mathbf{p}||, \tag{18}$$

$$c_i = ||\mathbf{x}_{i-1} - \mathbf{p}||, \quad f_i = ||\mathbf{x}_{i-1} - \mathbf{x}_i|| \tag{19}$$

*With these distances we can immediately develop the triangle with vertices $\mathbf{x}_i, \mathbf{p}_{T,i}, \mathbf{p}$ to get (in a planar coordinate system with axes ξ, η, see Fig. 3(**b**))*

$$\mathbf{x}_i^0 = (0,0)^T, \quad \mathbf{p}_{T,i}^0 = (a_i, 0)^T, \quad \mathbf{p}^0 = (a_i, b_i)^T \tag{20}$$

(2) *In order to develop the triangle with vertices* $\mathbf{x}_i, \mathbf{p}, \mathbf{x}_{i-1}$ *into the same planar coordinate system, we compute the two intersection points* $\mathbf{s}_{i,1}, \mathbf{s}_{i,2}$ *of the two circles* $k_{i,1}$ *(midpoint* $\mathbf{p}^0$*, radius* c_i*) and* $k_{i,2}$ *(midpoint* $\mathbf{x}_i^0$*, radius* f_i*). Then we compute the distances*

$$e_{i,j} := ||\mathbf{p}_{T,i}^0 - \mathbf{s}_{i,j}||, \quad j = 1, 2. \tag{21}$$

We compare the distances $e_{i,j}$ *to the distance* $e_i = ||\mathbf{p}_{T,i} - \mathbf{x}_{i-1}||$ *and set*

$$\mathbf{x}_{i-1}^0 := \mathbf{s}_{i,j} := (x_i, y_i)^T \quad \textit{with } j \textit{ corresponding to} \quad \min_{j\in\{1,2\}} |e_i - e_{i,j}|.$$

(3) *Now the circle* k_i *through the points* $\mathbf{x}_{i-1}^0$ *and* $\mathbf{x}_i^0$ *tangent to* $[\mathbf{x}_i^0, \mathbf{p}_{T,i}^0]$ *(the* x*-axis) has midpoint* $\mathbf{m}_i$ *and radius* ϱ_i *given by*

$$\mathbf{m}_i = (0, \varrho_i)^T, \quad \varrho_i = \frac{x_i^2 + y_i^2}{2y_i}. \tag{22}$$

This gives us $\kappa_i := 1/\varrho_i$*, an estimate of the curvature. An estimate of the distance of* $\mathbf{p}$ *to the current footpoint* $\mathbf{x}_i$ *is given by* $d_i := |\ ||\mathbf{p}^0 - \mathbf{m}_i|| - |\varrho_i|\ |$*. The stepsize* s_i *at the current step of the projected gradient algorithm is then given by*

$$s_i = \frac{1}{|d_i\kappa_i - 1|}. \tag{23}$$

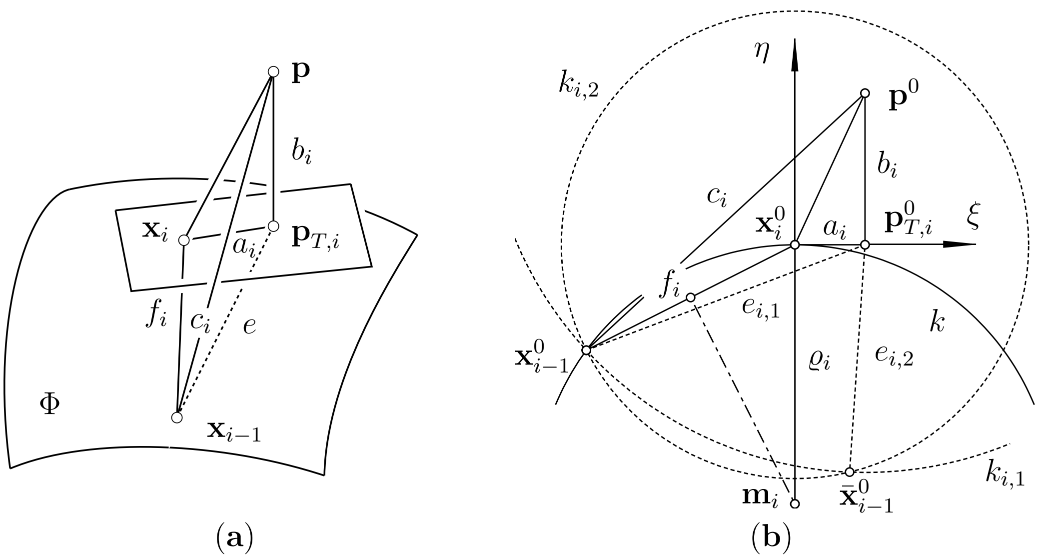

FIGURE 3. (**a**) Distances in $\mathbb{R}^n$. (**b**) Development of a part of Γ_f in $\mathbb{R}^2$.

3.4. **Summary of the algorithm for a general quadratic function.** Let us return to the originally addressed minimization of a quadratic function

$$F : \mathbb{R}^n \to \mathbb{R},\ F(\mathbf{x}) = \mathbf{x}^T \cdot Q \cdot \mathbf{x} + 2\mathbf{q}^T \cdot \mathbf{x} + q,$$

with a symmetric, positive definite matrix Q. The minimizer $\mathbf{p}$ of F in $\mathbb{R}^n$ is given by $\mathbf{p} = -Q^{-1} \cdot \mathbf{q}$. The Euclidean metric introduced via Q has an inner product, defined by

$$\langle \mathbf{v}, \mathbf{w} \rangle := \mathbf{v}^T \cdot Q \cdot \mathbf{w}. \tag{24}$$

The minimizer $\mathbf{p}^*$, which is constrained to a given surface $\Phi \subset \mathbb{R}^n$ is found by the following algorithm, which follows immediately from the previous considerations. We just have to carefully check where the inner product is involved.

Algorithm 2. *Computes the minimum $\mathbf{p}^*$ of a quadratic function with positive definite matrix Q (and unconstrained minimum $\mathbf{p}$) under the constraint that $\mathbf{p}^*$ lies on a given m-dimensional surface $\Phi \subset \mathbb{R}^n$. Starting with an initial guess $\mathbf{x}_0$ for the minimum, iteratively apply the following two steps.*

(1) *At the current iterate $\mathbf{x}_c \in \Phi$, compute a basis $\{\mathbf{c}_1, \ldots, \mathbf{c}_m\}$ in the tangent space of Φ, its Gramian matrix $G = (g_{ij}) = (\langle \mathbf{c}_i, \mathbf{c}_j \rangle)$ for the inner product (24), and the vector $\mathsf{r} = (r_j) = (\langle \mathbf{p} - \mathbf{x}_c, \mathbf{c}_j \rangle)$. Solve the linear system $G \cdot \mathsf{v} = \mathsf{r}$. With help of the solution $\mathsf{v} = (v_1, \ldots, v_m)$, define the tangent vector $\mathbf{t} = \sum_i v_i \mathbf{c}_i$.*

(2) *With distance computations based on the norm $\|\mathbf{x} - \mathbf{y}\|^2 = \langle \mathbf{x} - \mathbf{y}, \mathbf{x} - \mathbf{y} \rangle$ and the stepsize strategy of subsection 3.3 (performed in the Euclidean plane), compute a steplength s and the point $\mathbf{p}_T = \mathbf{x}_c + s\mathbf{t}$. With an admissible projection according to subsection 3.2, project $\mathbf{p}_T$ onto Φ to obtain the next iterate $\mathbf{x}_+$.*

References

[1] Geiger, C., Kanzow, C., 2002. *Theorie und Numerik restringierter Optimierungsaufgaben*, Springer, Berlin/Heidelberg/New York.

[2] Hofer, M., Pottmann, H., Ravani, B., 2003. Geometric design of motions constrained by a contacting surface pair, Comp. Aided Geom. Design 20, 523–547.

[3] Kelley, C. T., 1999. *Iterative Methods for Optimization* SIAM, Philadelphia.

[4] Kobbelt, L., 1996. A variational approach to subdivision, Comp. Aided Geom. Design 13, 743–761.

[5] Kobbelt, L., Schröder, P., 1998. A multiresolution framework for variational subdivision, ACM Trans. Graphics 17, 209–237.

[6] Patrikalakis, N. M., Maekawa, T., 2002. *Shape Interrogation for Computer Aided Design and Manufacturing*, Springer, Berlin/Heidelberg/New York.

Institute of Discrete Mathematics and Geometry, Vienna University of Technology, Wiedner Hauptstr. 8-10, A-1040 Wien, Austria., E-mail: pottmann, hofer@geometrie.tuwien.ac.at

Geometry Preprint Series No. 121, Vienna Univ. of Technology