Automatic Generation of Riemann Surface Meshes

Matthias Nieser, Konstantin Poelke, and Konrad Polthier*

Freie Universität Berlin, Germany
{matthias.nieser,konstantin.poelke,konrad.polthier}@fu-berlin.de

Abstract. Riemann surfaces naturally appear in the analysis of complex functions that are branched over the complex plane. However, they usually possess a complicated topology and are thus hard to understand. We present an algorithm for constructing Riemann surfaces as meshes in $\mathbb{R}^3$ from explicitly given branch points with corresponding branch indices. The constructed surfaces cover the complex plane by the canonical projection onto $\mathbb{R}^2$ and can therefore be considered as multivalued graphs over the plane – hence they provide a comprehensible visualization of the topological structure.

Complex functions are elegantly visualized using domain coloring on a subset of $\mathbb{C}$. By applying domain coloring to the automatically constructed Riemann surface models, we generalize this approach to deal with functions which cannot be entirely visualized in the complex plane.

1 Introduction

Riemann surfaces are a fundamental concept in modern complex analysis, topology and algebraic geometry. Bernhard Riemann himself introduced them 1851 in his dissertation "Grundlagen für eine allgemeine Theorie der Functionen einer veränderlichen complexen Größe", but it was Felix Klein and Hermann Weyl who caused his idea to become known among the mathematicians in the beginning 20th century. Since then, among other things, Riemann surfaces serve as generalized domains for complex functions because multi-valued complex functions can be turned into single-valued functions when defined on such a surface instead of the complex plane. However, these surfaces might possess a complicated topological structure since multi-valued functions give rise to ramifications determined by characteristic points - the so-called branch points.

Riemann surfaces on manifolds are used for several methods in computer graphics. For example, [1] uses a universal covering for the computation of shortest cycles in each homotopy class of a surface. The surface parameterization method [2] computes a 4-sheeted covering in order to represent the (multivalued) parameter function on higher genus surfaces. The notion of covering spaces provides a nice theoretical foundation of this parameterization approach.

* Supported by DFG Research Center Matheon "Mathematics for key technologies".

B. Mourrain, S. Schaefer, and G. Xu (Eds.): GMP 2010, LNCS 6130, pp. 161–178, 2010.
© Springer-Verlag Berlin Heidelberg 2010

1.1 Related Work

Surprisingly there is very little about the computer-aided visualization of Riemann surfaces. The most important contribution might be the work by Trott [3,4] for Wolfram Research. He uses the symbolic derivation and nonlinear equations solver provided by Mathematica and computes 3D plots based on an explicit function definition. To the best of our knowledge it is the only work that automatically generates visualizations of Riemann surfaces – most available images of Riemann surfaces usually show explicitly parametrized surfaces.

Important achievements concerning the technique of *domain coloring* are done by Farris [5], who also introduced this term. He uses simple color gradients without further features. The colorings of Pergler [6] and Lima da Silva [7] are of better quality and Lundmark [8] gives a detailed introduction and uses a color scheme revealing several important aspects of complex functions. Hlavacek [9] provides a gallery of complex function plots, using color schemes similar to Lundmark's and Pergler's. Further advancement and additional indicators are provided in [10].

1.2 Contribution

In this paper we present an algorithm for computing 3D models for Riemann surfaces based on given branch points and branch indices. Usually these surfaces possess a non-trivial topology and are thus hard to capture. The models we construct are a visualization technique for an easier understanding of the topology of these surfaces. This process consists of two parts:

1. Cutting the surface open, either by a user given cut graph or by computing the shortest cut graph as in [11], and computing multiple surface layers.
2. Computing a height function for each of the layers. The resulting surface is then interpreted as a graph over the complex plane. Since this embedding is used for visualization, there is possibly more than one height function that could be used in order to produce reasonable results. We use harmonic height functions as a natural choice as they produce almost smooth surfaces.

Whereas Trott's method [3] provides 3D plots of Riemann surfaces, we generate triangulated meshes which can be used as 3D models for further processing. As our meshes will be patched together from single sheets, we can easily extract parts of the model in order to focus on the important features of the surface.

We also apply domain coloring to these Riemann meshes – a method traditionally used for visualizing complex functions by a color map on the complex plane. However, complex functions usually define a covering of the complex plane, which can be explicitly constructed as a triangle mesh with our method. Using this triangle mesh as new domains for domain coloring, we can visualize analytical extensions of functions that are discontinuous when defined on the complex plane which helps for an overall understanding of those functions.

The paper is organized as follows: Sect. 2 introduces Riemann surfaces and gives the theoretical background and general setting of our approach. Sect. 3

explains the main concept of our visualization approach. The algorithm itself is explained in Sect. 4, and in Sect. 5 results are presented and discussed.

2 Riemann Surfaces and Complex Analysis

2.1 Problem Statement

The problem of visualizing covering surfaces has arisen to us in the analysis of complex functions. Given a holomorphic map $\eta : U \subset \mathbb{C} \to V \subset \mathbb{C}$ between two simply connected domains in the complex plane, its geometrical structure can be visualized by domain coloring as demonstrated in [10]. Domain coloring uses a reference image which is defined over V and transfers it to U via η. The resulting image in U yields many information about η.

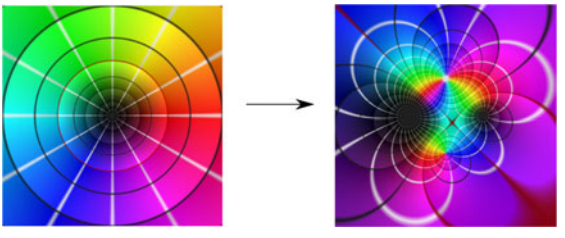


Fig. 1. Domain coloring (cf. section 5.2) is a technique that makes use of a color scheme to visualize complex valued functions. The left image shows a reference color scheme that represents the complex plane. The right image shows a typical color plot of a meromorphic function. The black spots denote zeros, here with multiplicity 2 (the bigger one on the left) and 1 (the one on the right). The white spot in the upper third is a double pole whereas the punctual spot at the bottom is a simple pole. This can also be seen from the multiplicity and order of the colors around these points – the double zeros and poles are surrounded by the complete color circle twice in contrast to the simple zeros and poles.

A problem arises if η is non-injective and one ones to color its inverse η^{-1} which is now a multivalued function. In this case, there are points in U which obtain more than one color and we would obtain sort of a *multivalued* image. We present a method for visualizing even those functions. The main idea is to use a covering surface X of V and a bijective map $f : U \to X$ that encodes the same geometrical information as η.

2.2 Theoretical Background

This section gives a short introduction into Riemann Surfaces and covering spaces. Good introductions to the general theory are e.g. [12, 13, 14, 15].

Definition 1 (Riemann Surface). *A* Riemann Surface *is a Hausdorff space together with a holomorphic structure, i.e. with an atlas of charts* $\{(U_i, h_i)\}, h_i : U_i \to \mathbb{C}$ *whose transition maps* $h_j \circ h_i^{-1}$ *are* biholomorphic *complex functions.*

In our setting, we use the notion of coverings from complex analysis. They are special cases of topological coverings equipped with a complex structure and can be easily described using local winding maps:

Definition 2 (Winding Map, Covering, Branch Point, Branch Index). *A* winding map $\eta : U \to V$ *between two disks is a map which is isomorphic to the function* $z \mapsto z^n$ *on a unit disk in* $\mathbb{C}$. n *is called the winding number of* η.

A covering (X, η) *of a Riemann Surface* Y *is defined by a map* $\eta : X \to Y$, *such that each point* $y \in Y$ *has a neighbourhood* V *whose inverse image is a union of countably many disks (layers) and the restriction of* η *to each of these disks is a winding map (with winding number* $n(x)$, $\forall x \in \eta^{-1}(y)$*).*

If there is a point x *with* $n(x) \geq 2$, *the covering is* branched *and* $y = \eta(x)$ *is a* branch point *with* branch index $n(x)$. *If* $n(x) = 1$ *for all* $x \in X$, *the covering is* unbranched.

The preimage of a point $y \in Y$ is called the *fibre* of y. One can show, that if the fibre is finite then the sum $gr(\eta) := \sum_{x \in \eta^{-1}(y)} n(x)$ is independant of the choice of y and is therefore called the *grade* of η.

Theoretically, our approach can handle finite or infinite coverings. However, for simplicity we restrict to coverings with finite fibres. For instance, they arise naturally as coverings induced by implicit functions that are defined by algebraic equations. In particular, if one assumes η to be proper (i.e. preimages of compact sets are compact) and to have only finite fibres, η is called a *finite map* and we have the following theorem:

Theorem 1. *Every finite holomorphic map between Riemann surfaces defines a covering.*

The finite holomorphic maps $\mathbb{C} \to \mathbb{C}$ are exactly the non-constant polynomials, hence the powers $z \mapsto z^n$ and all finite concatenations of them are coverings with finite fibres. An example of an infinite covering is the exponential map $\exp : \mathbb{C} \to \mathbb{C}^*$.

For a given covering X, a *Deck map* is an automorphism on X which leaves all fibres invariant, i.e. $D(\eta) := \{g \in \mathrm{Aut}(X) : \eta \circ g = \eta\}$. A Deck map of a covering η is uniquely defined by a given point $a \in X$ and its image $\eta(a)$. The set of all Deck maps form a group and this *Deck group* uniquely defines the topology of the covering.

We consider coverings which are *normal* and *cyclic* meaning that the Deck group is isomorphic to the modulo group $(\mathbb{Z}/n\mathbb{Z}, +)$.

3 Approach

3.1 Visualization of Holomorphic Functions

We construct a topologically correct triangle mesh X for a given covering $\eta : U \to V$ on given simply connected sets $U, V \subset \mathbb{C}$. For simplicity, let η be finite (i.e. η has finitely many layers). The surface X covers V in the same way as U

does, i.e. there is a covering map $\pi : X \to V$ and an isomorphism $f : U \to X$, such that $\eta = f \circ \pi$. Thus, both coverings π and η are isomorphic (Fig. 2, left).

We realize the surface X as a triangle mesh, which admits the same combinatorics as V and whose vertices live in $\mathbb{C} \times \mathbb{R}$. The projection operator π is just the Euclidean projection $(z, r) \mapsto z$. The covering can therefore be seen as a (multivalued) graph over the complex plane. However, in general X cannot be embedded in $\mathbb{R}^3$ and we usually obtain self-intersections.

As an additional visualization, one can now apply the domain coloring technique to these Riemann surfaces (Fig. 2, right). Instead of V, the surface X gets colored by transfering the domain image via f onto X. This will produce a continuous pattern, since f is bijective (in contrast to η). Since π is just a projection along the real axis, all the metric information about the function is still contained in f and can intuitively be captured by the viewer.

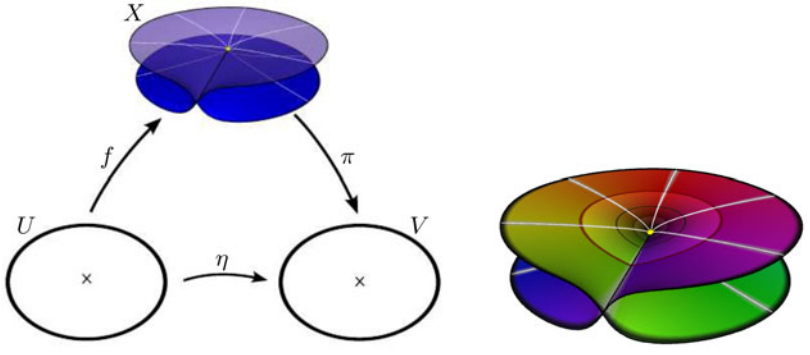


Fig. 2. Left: Given covering η will be visualized by X. Right: Domain coloring of $\eta(z) = z^2$.

This visualization helps to recognize the different types of extraordinary points. Branch points are characterized as center points of a helix, whereas singularities can be recognized as special points in the texture on the covering.

3.2 Branch Points and Branch Graph

Given the position of all branch points $B = \{b_0, \ldots, b_{d-1}\} \subset V$ with corresponding ramification indices $(r_0, \ldots, r_{d-1})$, this uniquely defines a covering over the complex plane which is normal and cyclic up to isomorphism.

A neighborhood of a branch point $p \in \eta^{-1}(B)$ on the covering looks like the union of one or more helices with r_i layers. Away from branch points, $V \setminus B$ is covered by an unbranched surface – the fibre of every open disc is isomorphic to just $N \in \mathbb{N}$ copies of the disk. For a globally consistent covering, we need at least $N := \mathrm{lcm}(r_0, \ldots, r_{d-1})$ many layers (the covering is of degree N). Thus, the fibre over each branch point b_i consists of N/r_i many helices with r_i layers each.

For the construction we need to enumerate the different layers of X. In general, there is no globally consistent enumeration since the covering is a connected surface and the layers exchange their role in different regions.

Given an arbitrary point $v_0 \in V \setminus B$ as root point, its fibre consists of N disjoint points which will be enumerated by $x_0, \ldots, x_{d-1}$. For each pair of points (x_0, x_i), $i \in \{0, \ldots, d-1\}$, there is a unique Deck map (defined on the whole covering) which maps x_0 to x_i. This Deck map transfers the cyclic order of the layers to any other fibre in V (the Deck map is a permutation in each fibre). Thus, all fibres in the covering are global consistently ordered in the same cyclic manner.

We can therefore enumerate the d layers as follows: Let G be a *cut graph* of $V \setminus B$, i.e. the union of paths $\{\gamma_k\}$, which cut the surface open into a simply connected disk (Fig. 3). Each path γ_k must start and end either at a branch point or at the boundary of V. The covering $\eta^{-1}(V \setminus G)$ of this slotted surface then decomposes into d disjoint connected components X_i.

The preimage $\eta^{-1}(\gamma_k)$ of a cut path on the covering consists of d paths in X. Each of these (lifted) paths separate two layers: X_i on the left side and X_j on the right side of the directed path. Because of the cyclic order, the difference $\mathrm{sh}(\gamma_k) = j - i$ is the same for all these lifted paths and is called the *shift* of γ_k. The paths γ_k together with their shifts $\mathrm{sh}(\gamma_k)$ form a *branch graph* of V.

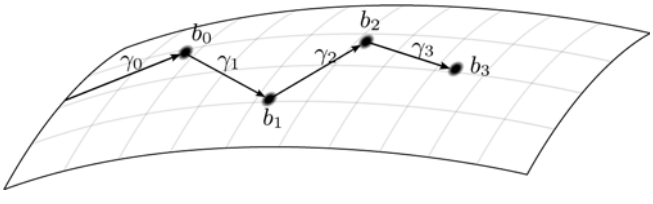


Fig. 3. A branch graph on a surface

The branch graph just connects different layers X_i to a closed covering surface. When a given point $p \in X_i$ in an arbitrary layer of the covering is continuously moved around, it still stays on X_i until a cut path is crossed. In this case, the layer changes to $X_{(i \pm \mathrm{sh}(\gamma_k)) \bmod d}$ (with a + sign if it crosses the path from left to right).

The labeling X_i therefore depends on the choice of the cut graph. However, the covering itself is independant from this choice. Our algorithm will construct an arbitrary branch graph from given branch points and uses it for the generation of the covering. The topology of the covering surface will not depend on G, but only on the position and indices of branch points.

3.3 Shifts

The shifts of a cut path define, how the different layers on the left and right side of the path are connected. Similarly, we introduce the shift at a branch point:

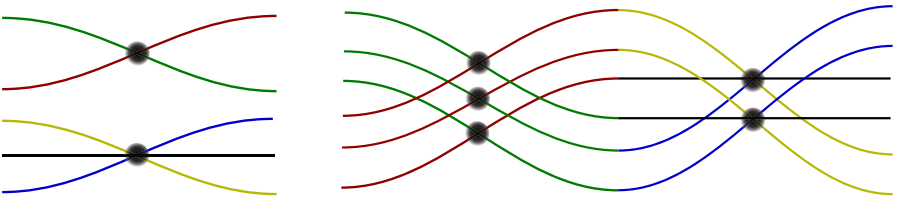


Fig. 4. 2D cut through the layers of different coverings. *Top left*: Branch point with $r_i = 2$. *Bottom left*: branch point with $r_i = 3$. *Right*: Covering with $N = 6$ sheets and two branch points. The left one has $r_i = 2$ and a shift of 3, the right one has $r_i = 3$ and a shift of 2.

Definition 3 (Lifting, Layer Shift). *Given a point $p \in V$ and an infinitesimal small loop $\delta : [0, 1] \to V$ around p (counterclockwise). Let δ' be a* lifting *of δ, i.e. a (not necessarily closed) path in X with $\pi(\delta') = \delta$. Denote X_i, X_j the layers of X, such that $\delta'(0) \in X_i$ and $\delta'(1) \in X_j$. Then $sh(p) := (j - i)$ mod N is called the* layer shift *of the point p.*

The layer shift just measures how many layers are being crossed when walking around p once. The shift is 0 for all regular points and $\neq 0$ at branch points. The shifts of branch points and the shifts of cut paths are related by the following equation:

Let b_i be a branch point and γ_k the set of paths starting or ending in b_i. Then the *shift* of b_i is the sum

$$\mathrm{sh}(b_i) = \sum_{\text{starting } \gamma_k} \mathrm{sh}(\gamma_k) - \sum_{\text{ending } \gamma_k} \mathrm{sh}(\gamma_k). \tag{1}$$

This defines a linear relation between shifts of the d branch points and shifts of the d cut paths (since V has a boundary, which is also connected by the branch graph G). The sum $-\sum_{b_i} \mathrm{sh}(b_i)$ mod N is called the *shift* of the boundary. If it is 0, then the boundary of V lifts to a closed loop on X, otherwise walking along the boundary once will end on another layer on the covering.

The shift at a branch point depends on the ramification index as follows: The neighborhood of each branch point consists of N/r_i helices whose layers are entwinded. Thus, the shift is an arbitrary number $s_i N/r_i$, $s_i \in \mathbb{N}$, but s_i must be coprime to r_i in order to produce the correct number of helices, e.g. set $\mathrm{sh}(b_i) := N/r_i$.

The algorithm only needs the branch graph and the shifts of all cut paths. If only the branch points and their ramification indices are given, then their shifts can be chosen as described above and the shifts of the cut paths uniquely follow from Eqn. 1 (assuming that the shift of the boundary is also prescribed, e.g. to 0).

For the application of domain coloring, it is necessary to explicitly prescribe the shifts in addition to the ramification indices (see Sect. 5.2). This would be an optional input for the algorithm.

4 Algorithmic Generation of Riemann Surface Models

The algorithm for generating the surface models can be separated into several parts. As an input it takes a set of branch points $\{b_0, \ldots, b_{d-1}\}$ which are located on vertices on a simply connected planar geometry M_h, i.e. a triangulated planar mesh.

Additionally, the number of layers N of the covering and the shift of the branch points $\mathrm{sh}(b_i)$ are given. One could alternatively prescribe the local ramification indices r_i of all branch points and then set $N := \mathrm{lcm}(r_0, \ldots, r_{d-1})$ and $\mathrm{sh}(b_i) := s_i N/r_i$ for any arbitrary integer s_i which is coprime to r_i. The shift of the boundary is set to $-\sum_{b_i} \mathrm{sh}(b_i)$.

The following subsections describe the individual steps of the algorithm:

Algorithm 1. Compute Riemann Surface

Input: Triangulated domain M_h, Branch points b_i, Shifts $\mathrm{sh}(b_i)$

1 Generate a branch graph (Sect. 4.1)
2 Cut domain geometry along the branch graph (Sect. 4.2)
3 Compute height function on branch graph in all sheets (Sect. 4.3)
4 Extend height function smoothly to the inner (Sect. 4.4)

4.1 Building the Cut Graph

The cut graph of the surface M_h consists of paths γ_k along edges of M_h, whose union cut the surface open into a topological disk. Branch points are thereby considered as infinitesimal holes in the surface, thus they must be connected by the branch graph.

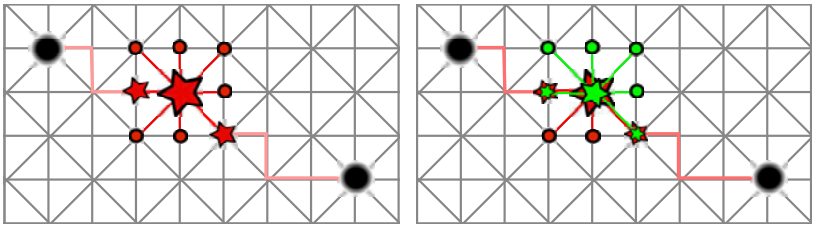
The choice of a special cut graph to given singularities is not unique. There may be more than one graph and these paths might be topologically different and the choice of another cut graph alters the embedding of our constructed surface into $\mathbb{C} \times \mathbb{R}$. However the topology of the embedding does only depend on the branch points and not on the cut graph.

Hence, we allow any user given cut graph as an input. If such a graph is not specified explicitly, it can be automatically generated – a canonical choice is the shortest cut graph of the surface. [11] describes an algorithm on computing this shortest cut graph for surfaces without boundary. A generalization to surfaces with boundary can be found in [16].

Given the cut paths γ_i, their shifts are computed as explained in Sect. 3.3. They are completely determined by the shifts of the branch points.

4.2 Cutting the Base Geometry

The next step is cutting the plane M_h along all cut paths. Each cut path γ_k is given as a path on edges of the planar domain geometry, i.e. it can be described by a list of vertices $v_1, \ldots, v_k$. Cutting M_h along γ_k means that every vertex



(a) Every vertex v (red star) has neighbors (red dots) which are its adjacent vertices in M_h and neighbors on the branch cut γ_k (small stars).

(b) Duplicating v yields a vertex v' (green star). Update the neighbor information such that v (resp. v')is connected with its adjacent vertices lying on the left hand side (resp. right hand side) of γ_k. Note that by cutting the plane, every path becomes part of the boundary.

Fig. 5. Cutting M_h along γ_k

v_j, $j \in \{2, \ldots, k-1\}$ has to be duplicated and its neighborhood has to be updated. The original vertex v_j is connected with vertices on the left hand side of γ_k, whereas the copy v'_j is connected with vertices on the right hand side only. Figure 5 demonstrates this procedure.

The resulting geometry then represents one sheet of the future covering surface. Since all sheets are of the same topology, the slotted domain surface is copied $N-1$ times and we obtain a total of N geometries X_i all cut in exactly the same manner.

4.3 Boundary Constraints for the Height Function

After having cut the domain, we now have N triangle meshes X_i representing the different sheets of the covering, which still live in the complex plane. This section and Sect. 4.4 deals on lifting them into $\mathbb{C} \times \mathbb{R}$ by computing a height function. We start by prescribing height values on the boundary of X_i, i.e. on the cut path and the outer surface boundary.

The sheets in the resulting geometry should be stacked according to their cyclic order, i.e. X_0 is at the bottom and X_{N-1} is the top most layer. The height difference between two consecutive sheets is defined by a constant $\Delta > 0$, thus layer X_i is assigned a height value of $i\Delta$.

Given a sheet X_i and a cut path γ_k between two branch points b_m and b_n, we will now describe, how the height values for the vertices on the left side of γ_k are computed. The vertices on the right side are then handled in the same manner, but using the inverted path with a negative shift value of $-\mathrm{sh}(\gamma_k)$.

The vertices in X_i on γ_k (on the left side) should be connected to vertices of the layer $X_{(i+\mathrm{sh}(\gamma_k)) \bmod N}$. The height function must therefore change smoothly from $i\Delta$ (in X_i) to $(i + \mathrm{sh}(\gamma_k)) \bmod N \cdot \Delta$ (in the neighboring sheet). The path

γ_k is somewhere between these two layers, thus let is give it a height of the average $h_{\text{left}}(\gamma_k) := (i + (i + \text{sh}(\gamma_k)) \bmod N)/2$ for a moment.

If we just assigned this constant height value to all vertices on γ_k, it would cause problems in the vicinity of the start and end points of γ_k (branch points b_m, b_n), since several layers with different height values are glued together there. Hence we interpolate smoothly between these height values.

As described in Sect. 3.2, the branch point b_m occures exactly $N/r_m = \text{sh}(b_m)$ times in the covering. Denote them by $b'_{m,0}, \ldots, b'_{m,N/r_j-1}$. Each $b'_{m,i}$ connects all the sheets $X_i, X_{i+\text{sh}(b_i)}, \ldots, X_{N-\text{sh}(b_j)}$ (Fig. 6, left). We define the height of $b'_{m,i}$ as the average value $((N - \text{sh}(b_j))/2 + i)\Delta$.

We can now define the height function on γ_k with the following properties:

- Height in start point b_m is $((N - \text{sh}(b_m))/2 + (i \bmod \text{sh}(b_m)))\Delta$.
- Height in end point b_n is $((N - \text{sh}(b_n))/2 + (i \bmod \text{sh}(b_n)))\Delta$.
- Height of vertices between b_n and b_m is $(i + ((i + \text{sh}(\gamma_k)) \bmod N))/2$.

We let the height function be constant on the middle third of the branch cut and interpolate on the other parts with a clamped cubic spline, i.e. a polynomial of the form

$$s(x) = a + bx + cx^2 + dx^3 \tag{2}$$

with given boundary values and first derivatives at the end points (Fig. 6, right).

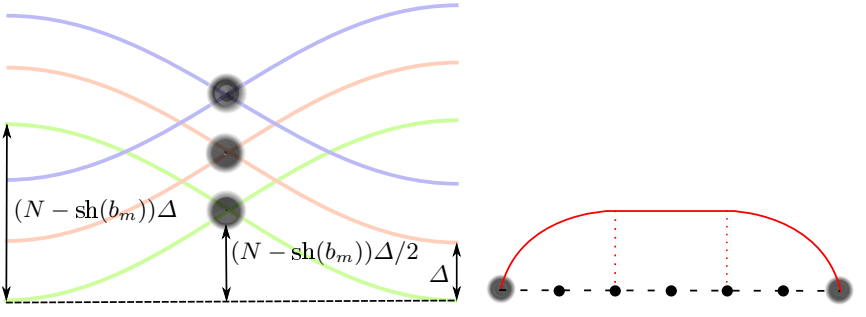


Fig. 6. Computing the height function. Left: Vicinity of a branch point b_m, where different layers are connected. Right: Height function along a cut path γ_k.

It remains to define a height function on the outer boundary of X_i. All our examples are constructed such that the shift of the outer boundary is 0 (i.e. the shifts of all branch points sum to 0). In this case, the height on the outer boundary is just the constant $i\Delta$. If the shift of the boundary is not 0, then the height function must interpolate smoothly between the different layers when walking around the boundary once. It does not matter how this interpolation is done as long as it defines a smooth height function along the boundary.

4.4 Computing the Height Function on the Inside

Having defined a height function on the boundary of each layer X_i, we now solve the Dirichlet problem with given boundary values:

$$\begin{aligned} \Delta f &= 0 \text{ in } X_i \\ f &= \text{prescribed height values on } \partial X_i \end{aligned} \tag{3}$$

More precisely, we solve the corresponding discrete variational formulation

$$Av = b \text{ with } v, b \in \mathbb{R}^{d\times 1}, A \in \mathbb{R}^{d\times d} \tag{4}$$

where d is the number of vertices of X_i, A is the Laplacian matrix, containing the well-known cotan weights of the underlying triangle mesh, and b is the right side, which is almost zero except for the boundary values f.

The solution is a harmonic function and produces a smooth surface in the interior of all sheets. However, along the cuts the surface is in general not C^1 continuous.

It should be mentioned that this height function is just one option and of course one is free to choose any arbitrary height function. Heuristically, harmonic functions often produce elegant surfaces and they are closely related to holomorphic functions (in fact, real and imaginary part of a holomorphic function are harmonic functions). The only thing one has to care about is that the values on the boundary fit together, such that the "gluing" is correctly reflected by the model.

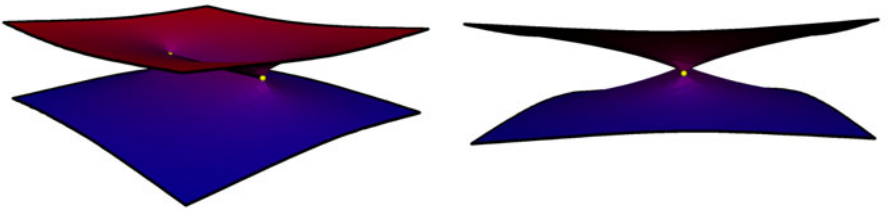
5 Results

5.1 Riemann Surface Models

The following models are generated by our algorithm. We applied further domain coloring techniques on some models, which are then shown in Sect. 5.2. Branch points are highlighted as small yellow balls. The transition between adjacent layers as well as the surface boundary is marked by black lines.

Figures 7 and 8 show a two-sheeted Riemann surface with two branch points of shift 1 (i.e. index is 2). We generated two embeddings using different cut graphs. The cut graph in Fig. 7 directly connects both branch points, whereas in Fig. 8, the two branch cuts emanate from a singularity and both meet at infinity. Thus, the situation is indeed the same and both surfaces are therefore topologically equivalent. Since we can only compute a compact subset of the infinite Riemann surface, we have to take care, that the height function respects the layer shift at the patch boundary, whenever the cut graph meets the boundary.

Figure 9 shows a Riemann surface with three sheets and two branch points of index 3, the shifts are 1 and -1. When crossing the branch cut between the two singularities from left to right one ends up the next lower sheet, or on the top most sheet when starting from the lowest one.



(a) In this case the branch cut coincides with the line of self intersection, which is in general not the case.

(b) A side view shows the similarity to the schematic pictures in Fig. 4

Fig. 7. A surface with two sheets and two branch points of index 2

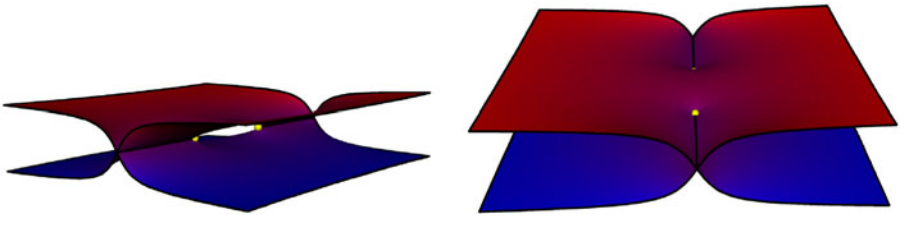
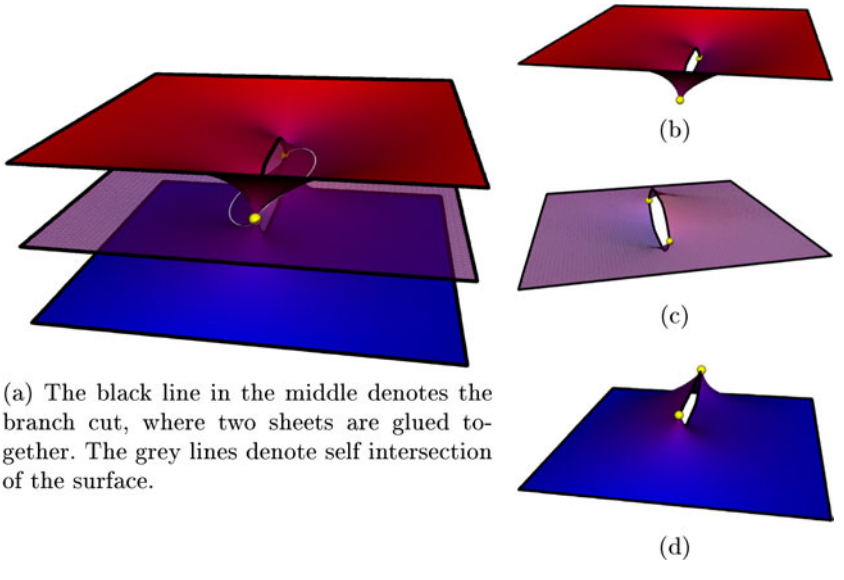


Fig. 8. Another embedding of the Riemann surface from Fig. 7



(a) The black line in the middle denotes the branch cut, where two sheets are glued together. The grey lines denote self intersection of the surface.

Fig. 9. A surface with three sheets and two branch points. (a) shows the whole surface. (b)-(d) show the three sheets separately.

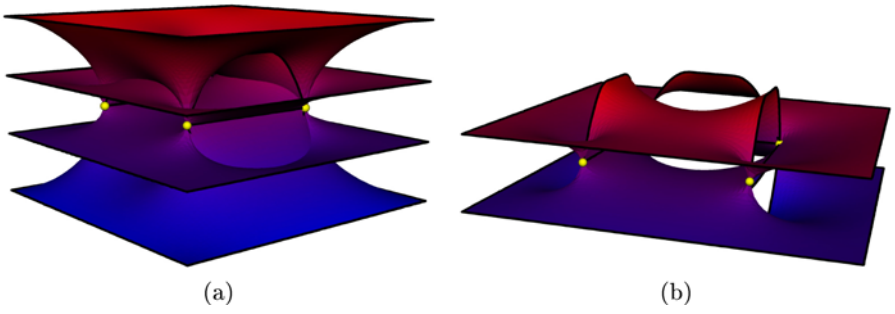


Fig. 10. Riemann surface with four layers and four branch points. (a) Whole surface. (b) Second and third sheet slightly rotated.

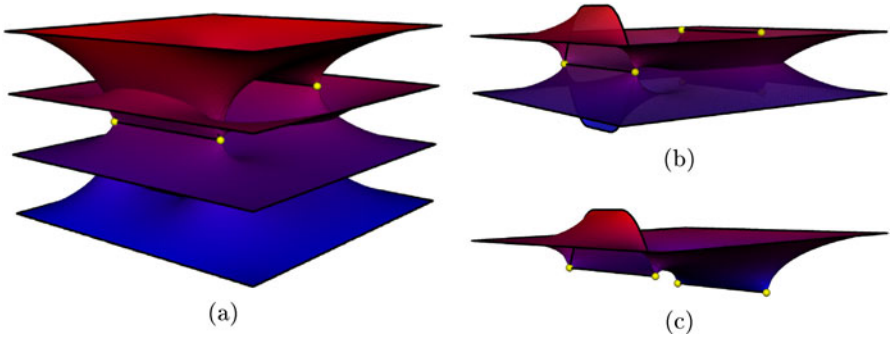


Fig. 11. Riemann surface with four sheets and four branch points. (a) Whole surface. (b) Second and third sheet. (c) Displaying only the third sheet reveals the hidden branch points.

A more complex model is shown in Fig. 10. It has four sheets and four branch points A, B, C, D of shift 1, 2, 2 and 3, respectively. The branch points form a square (the branch point on the corner in the back is hidden) and are connected by branch cuts between A and B, B and C and C and D. The two leftmost branch points (A and D) in (a) are not connected. When starting from the right on the upper sheet and crossing the branch cuts between C and D and A and B, one does not change the layer. This corresponds to the fact that the branch shifts along these cuts sum up to four which is zero modulo four.

Figure 11 again shows the same situation with the same branch points as in Fig. 10, but with a different branch graph. This time, A and C are connected as well as B and D, so the branch graph is disconnected. Note that this is also a valid configuration since the branch shifts along each cut sum up to zero modulo four. However, we obtain a different embedding in $\mathbb{R}^3$, although all branch shifts are the same as in the previous picture. In (b) one can clearly see the layer shift of one when crossing the cut in the front of the picture. Note that the second branch cut in the back connects two branch points of shift 2 and thus the model

locally decomposes into two connected components around this cut. That is why a second pair of branch points is introduced which can be slightly seen.

5.2 Domain Coloring on Riemann Surfaces

As mentioned in Sec. 2.2, Riemann surfaces are naturally induced by holomorphic covering maps. However the visualization of complex functions as graphs over a domain is not possible, since their graphs live in $\mathbb{C}^2 \cong \mathbb{R}^4$. That is why one often makes use of an elegant technique called *domain coloring* (see [5] or [10]), which encodes the range of a complex function as a color scheme that is plotted directly onto the domain. We use this technique to visualize complex functions whose proper domains are Riemann surfaces. In particular, for a given covering map η as in Fig. 2, we can define its inverse mapping as a function living on the Riemann surface. Those functions are naturally multivalued when defined over the complex plane and hence cannot be properly defined as holomorphic functions on $\mathbb{C}$ – in fact they are not even continuous and one has to cut the plane in order to define at least a so called *holomorphic branch of a function*. With the notation of Fig. 2, these branches can be considered as restrictions of f^{-1} to a sheet $X_i \subset X$ and $\pi\,|_{X_i}$ becomes an isomorphism between the cut complex plane and X_i.

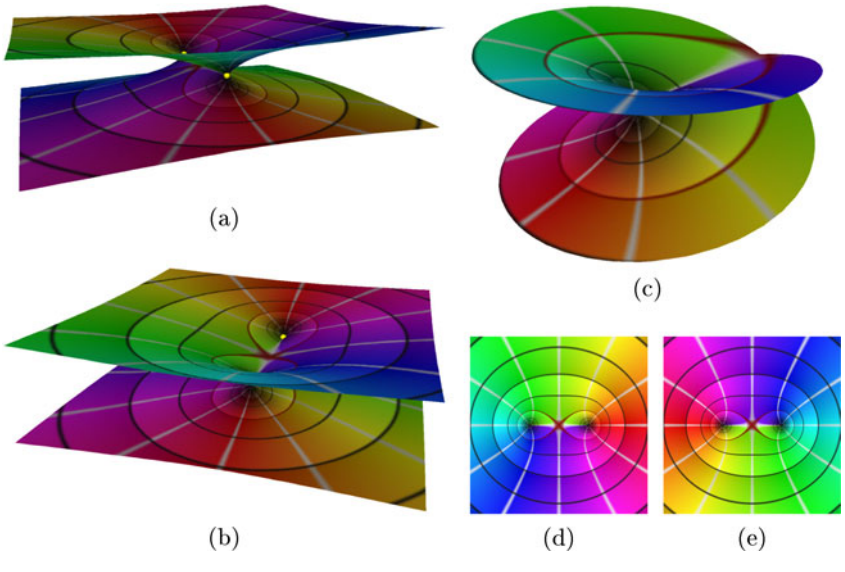


Fig. 12. A function having two branches and a branch cut between the branch points -1 and $+1$. (a) and (b) show the model from Fig. 7 with the coloring for $f^{-1}(z) := (z-1)^{1/2}(z+1)^{1/2}$. Whereas the restriction to a single sheet results in a color–discontinuous planar plot (see (d) and (e)), the coloring on the surface is continuous. (c) shows the local topology of our model close to a branch point. The typical two-sheeted helix is a picture often shown as an explicitly parametrized surface for the Riemann surface induced by $z \mapsto z^2$.

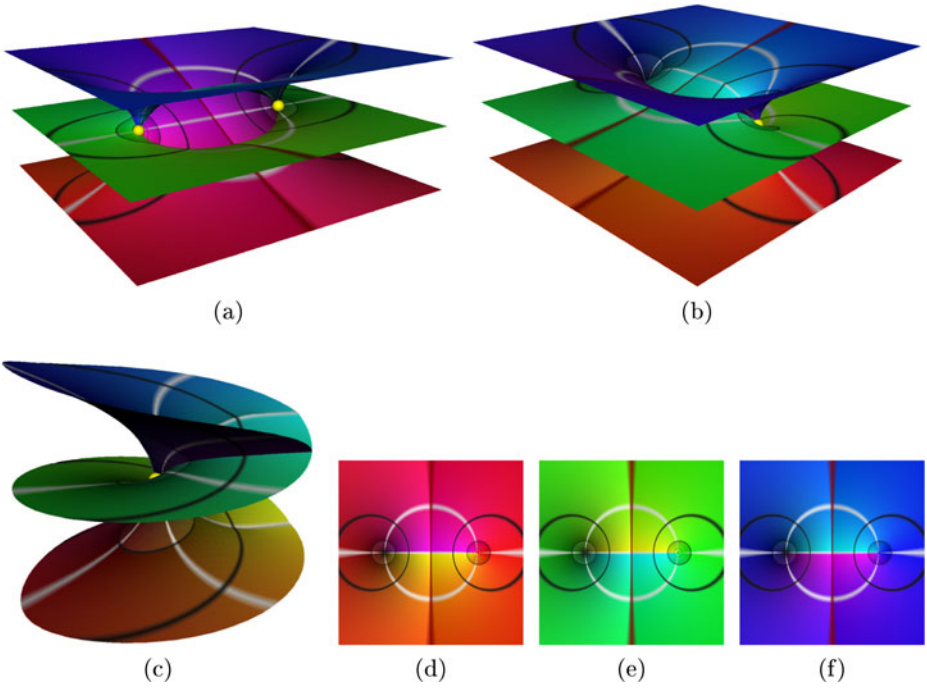


Fig. 13. Domain coloring of a function with three branches. (c) Local model in a neighborhood of a branch point The planar color plots (d)–(f) are discontinuous between the branch points which is resolved on the Riemann surface model.

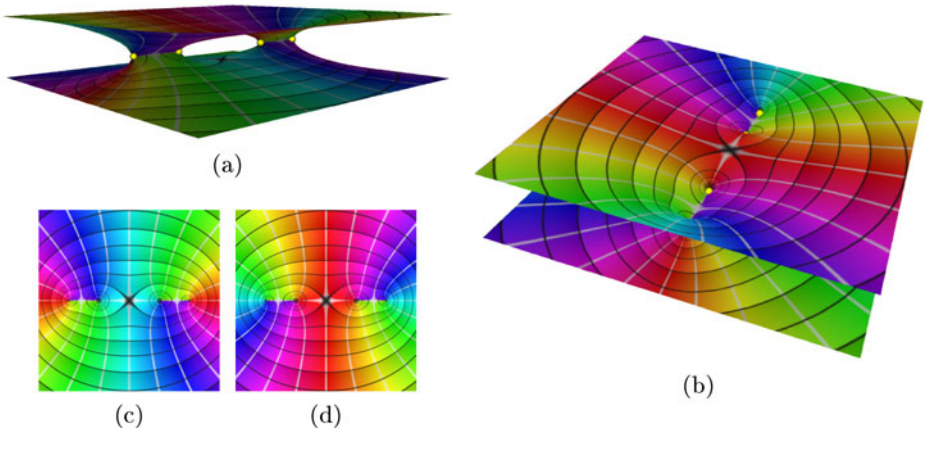


Fig. 14. (a) and (b) Domain colored Riemann surface with two disconnected branch cuts. (c) and (d) The planar domain coloring of the function is discontinuous at the branch cut.

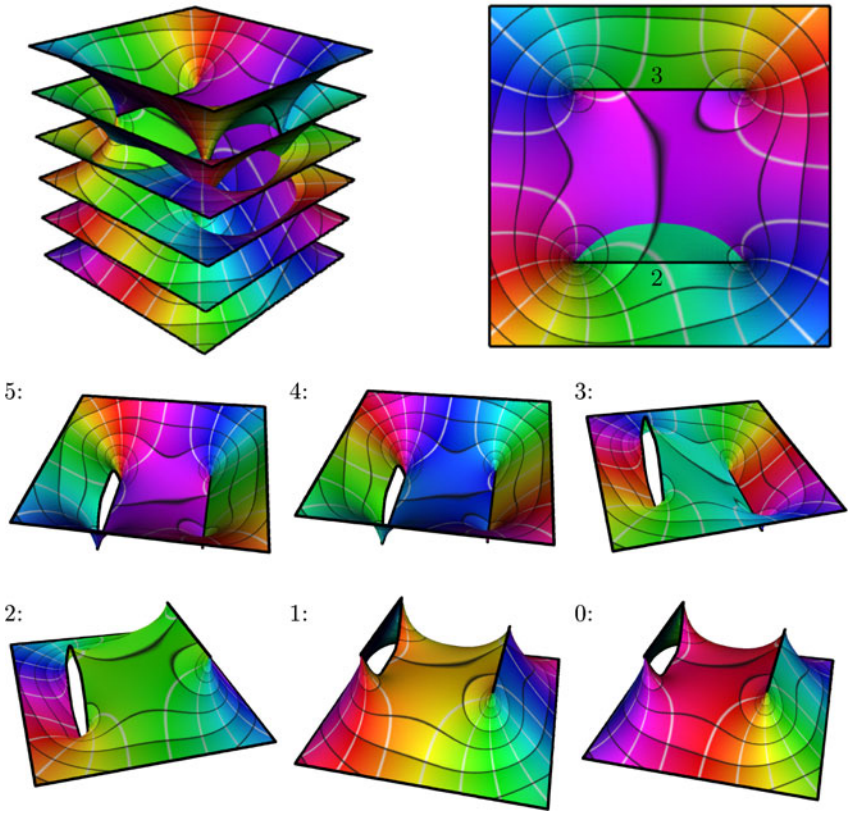


Fig. 15. A 6-sheeted covering of a function having two branch cuts with different shift and thus with branch points of different branch indices – two of index 3 and two of index 2. Top left: whole covering. Top right: 2d view on top most layer. The black lines are the cut paths with not vanishing shift (shift 2 and 3). The 6 pictures below show the layers separately.

We apply domain coloring to Riemann surfaces and obtain globally color-continuous plots corresponding to globally holomorphic mappings on the surfaces. These surfaces are computed by using our method with the branch points and the shifts of η which can be derived from the function definition. The coloring is induced by f^{-1} and the colored surfaces reveal the behaviour of a function as well as the topology of the induced Riemann surface.

Note that the composition $f^{-1} \circ (\pi \mid_{X_i})^{-1}$ maps from the cut complex plane into $\mathbb{C}$ and can be visualized using planar domain coloring, but the unrestricted map $(\pi \circ f)^{-1}$ is in general multivalued. Figures 9 – 12 show some colored surfaces together with planar domain colorings obtained by restriction to a single sheet. A two-sheeted Riemann surface with domain coloring is shown in Fig. 12.

Figure 13 shows the domain coloring of a function having three sheets and a two branch points at $+1$ and -1. The Riemann surface is the same one as

in Fig. 9, this time equipped with domain coloring of a holomorphic function. One can see the smooth transition between the first and the third sheet in (a) whereas (b) shows the transition between the second and third sheet.

Figure 14 shows a Riemann surface with two branch cuts which are not connected, or, equivalently, as being connected by a branch cut with shift 0, which does not cause any sheet transition. As in the example in Fig. 12, the branch cuts connect two sheets and the restriction to a single sheet results in a discontinuouities of the branched function, see (c), (d). Note that the function is continous on every sheet between the branch cuts.

6 Outlook

Although our algorithm is capable to deal with a large class of functions, there are some issues left to solve.

Having realized the generation of Riemann surface models in $\mathbb{R}^3$ it is tempting to switch to the projective setting and consider compact surfaces over the compactified complex plane $\mathbb{C} \cup \{\infty\}$, i.e. the Riemann sphere. A closely related advancement is the consideration of covering surfaces over arbitrary manifolds.

Another challenging task is the extension of domain coloring to maps between arbitrary Riemann surfaces, i.e. maps which are multivalued and non-injective when defined on the complex plane. This implies that neither the map itself nor its inverse can be continuously defined on $\mathbb{C}$.

References

1. Yin, X., Jin, M., Gu, X.: Computing shortest cycles using universal covering space. Vis. Comput. 23(12), 999–1004 (2007)
2. Kälberer, F., Nieser, M., Polthier, K.: Quadcover - surface parameterization using branched coverings. Comput. Graph. Forum. 26(3), 375–384 (2007)
3. Trott, M.: Visualization of Riemann surfaces (2009), `http://library.wolfram.com/examples/riemannsurface/` (retrieved December 8, 2009)
4. Trott, M.: Visualization of Riemann surfaces of algebraic functions. Mathematica in Education and Research 6, 15–36 (1997)
5. Farris, F.A.: Visualizing complex-valued functions in the plane, `http://www.maa.org/pubs/amm_complements/complex.html` (retrieved December 8, 2009)
6. Pergler, M.: Newton's method, Julia and Mandelbrot sets, and complex coloring, `http://users.arczip.com/pergler/mp/documents/ptr/` (retrieved December 8, 2009)
7. da Silva, E.L.: Reviews of functions of one complex variable graphical representation from software development for learning support, `http://sorzal-df.fc.unesp.br/~edvaldo/en/index.htm` (retrieved on December 8, 2009)
8. Lundmark, H.: Visualizing complex analytic functions using domain coloring (2004), `http://www.mai.liu.se/~halun/complex/domain_coloring-unicode.html` (retrieved on December 8, 2009)

9. Hlavacek, J.: Complex domain coloring, http://www6.svsu.edu/~jhlavace/Complex_Domain_Coloring/index.html (retrieved on December 8, 2009)
10. Poelke, K., Polthier, K.: Lifted domain coloring. Computer Graphics Forum 28(3), 735–742 (2009)
11. Erickson, J., Whittlesey, K.: Greedy optimal homotopy and homology generators. In: SODA, pp. 1038–1046 (2005)
12. Farkas, H.: Riemann Surfaces. Springer, New York (1980)
13. Lamotke, K.: Riemannsche Flächen. Springer, Berlin (2005)
14. Forster, O.: Lectures on Riemann Surfaces, 4th edn. Graduate Texts in Mathematics. Springer, Heidelberg (1999)
15. Needham, T.: Visual Complex Analysis. Oxford University Press, Oxford (2000)
16. Kälberer, F., Nieser, M., Polthier, K.: Stripe parameterization of tubular surfaces. In: Topology-Based Methods in Visualization III. Mathematics and Visualization. Springer, Heidelberg (to appear 2010)
17. Riemann, B.: Grundlagen für eine allgemeine theorie der functionen einer veränderlichen complexen größe (1851)