Discrete Conformal Maps

a survey

Christian Mercat

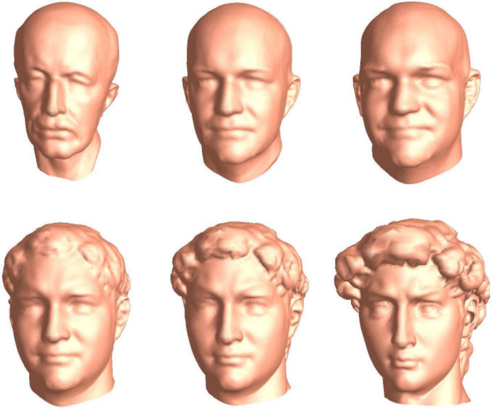
Institut de Mathématiques et de Modélisation de Montpellier (I3M, UM2)

Journées Informatique et Géométrie, INRIA Sophia Antipolis

http://entrelacs.net/home/ mercat@math.univ-montp2.fr

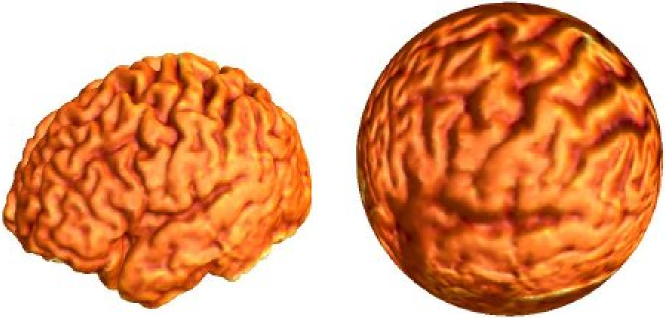
Peter Schröder, CalTech

Global parametrization sending little circles to circles. Texture mapping. Surface matching.



X. Gu & S.-T. Yau, Harvard Uni.

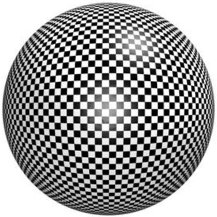
Surface interpolation (especially staying isometric). Remeshing, coarsening, refining.



(a) (b)

Monica K. Hurdal, Florida State University

Commonly parametrize different surfaces to compare functions on it.



Tony Chan, UCLA

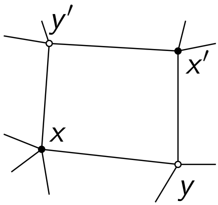
Whether preserving little circles or preserving little squares.

$$f \text{ is conformal } \Rightarrow f(z) = \begin{cases} az + b + o(z), \\ \dfrac{az+b}{cz+d} + o(z^2). \end{cases}$$

Two notions on a "quad-graph":

– Preserve the diagonal ratio (linear),

– or preserve the cross-ratio (Möbius invariant).



Diagonals ratio:

$$\frac{y'-y}{x'-x} = i\,\rho$$

Crossratio:

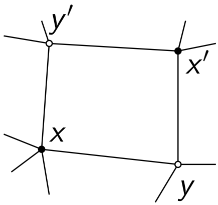
$$\frac{y'-x'}{x'-y}\,\frac{y-x}{x-y'} = q$$

$$f \text{ is conformal } \Rightarrow f(z) = \begin{cases} az + b + o(z), \\ \dfrac{az+b}{cz+d} + o(z^2). \end{cases}$$

Two notions on a "quad-graph":

– Preserve the diagonal ratio (linear),

– or preserve the cross-ratio (Möbius invariant).



Diagonals ratio:

$$\frac{y'-y}{x'-x} = i\,\rho = \frac{f(y')-f(y)}{f(x')-f(x)}$$

Crossratio:

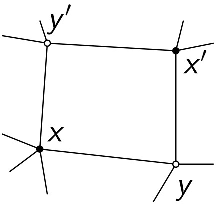
$$\frac{y'-x'}{x'-y}\,\frac{y-x}{x-y'} = q$$

$$f \text{ is conformal } \Rightarrow f(z) = \begin{cases} az + b + o(z), \\ \dfrac{az+b}{cz+d} + o(z^2). \end{cases}$$

Two notions on a "quad-graph":

– Preserve the diagonal ratio (linear),

– or preserve the cross-ratio (Möbius invariant).



Diagonals ratio:

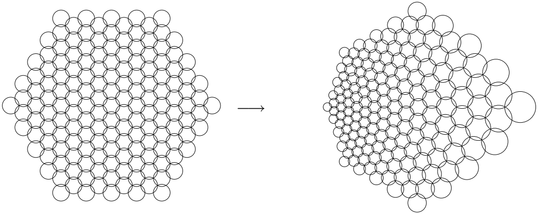
$$\frac{y'-y}{x'-x} = i\,\rho$$

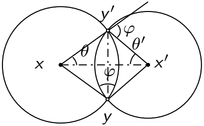
Crossratio:

$$\frac{y'-x'}{x'-y}\,\frac{y-x}{x-y'} = q$$

$$= \frac{f(y')-f(x')}{f(x')-f(y)}\,\frac{f(y)-f(x)}{f(x)-f(y')}$$

Circle patterns are a particular case





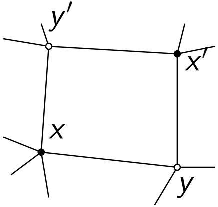
$$q = e^{-2(\theta+\theta')} = e^{-2\varphi}$$

$$\rho = \frac{\cos(\theta-\theta')-\cos(\varphi)}{\sin(\varphi)}$$

A function F preserving the **cross-ratio** can be written in terms of a function f such that

$$F(y) - F(x) = f(x)\, f(y)\, (y - x) = \text{“}F'(z)\, dz\text{”}\,.$$

fulfilling on the face (x, y, x', y'),



$$\begin{aligned} \oint F'(z)\, dz \quad &= \quad f(x)\, f(y)\, (y - x) + f(y)\, f(x')\, (x' - y) + \\ & \qquad f(x')\, f(y')\, (y' - x') + f(y')\, f(x)\, (x - y') = 0 \end{aligned}$$

Circle patterns case: $F(y) - F(x) = r(x)\, e^{i\,\theta(y)}\, (y - x)$.

Understood as Morera equations where function integration is

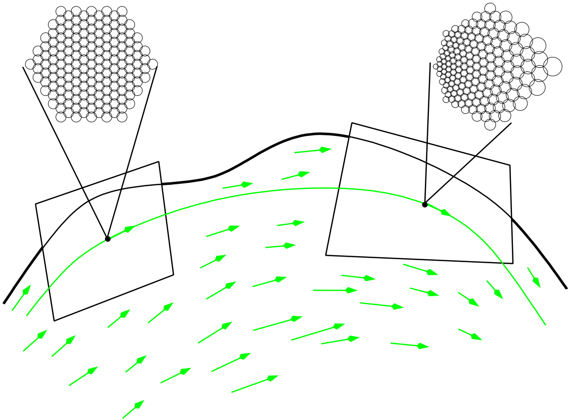
- the *geometric* mean for **cross-ratio** preserving maps

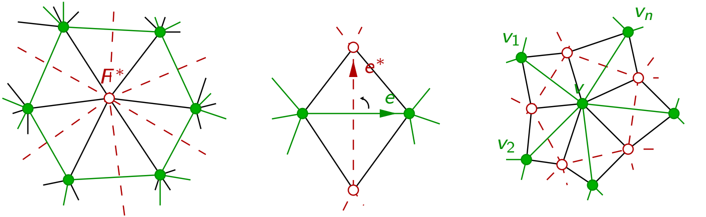
$$\int_{(x,y)} g\, dZ := \sqrt{g(x)\, g(y)}\, (y-x).$$

- the *arithmetic* mean for **diagonal ratio** preserving maps

$$\int_{(x,y)} g\, dZ := \frac{g(x)+g(y)}{2}\, (y-x).$$

When the quadratic case is linearized:





0. Vertex dual to a face. 1. Dual edges. 2. Face dual to a vertex.

The double $\Lambda = \Gamma \oplus \Gamma^*$

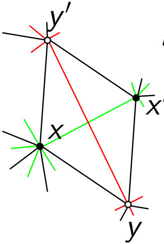
The *chains*-complex $C(\Lambda) = C_0(\Lambda) \oplus C_1(\Lambda) \oplus C_2(\Lambda)$
linear combination of vertices (0), edges (1) and faces (2).
boundary operator $\partial : C_k(\Lambda) \to C_{k-1}(\Lambda)$
Null on vertices, $\partial^2 = 0$.
Its kernel $\ker \partial =: Z_\bullet(\Lambda)$ are the closed chains or *cycles*.
Its image are the *exact* chains.

The space dual to chains form the *cochains*, $C^k(\Lambda) := \mathrm{Hom}(C_k(\Lambda), \mathbb{C})$.
Evaluation is denoted $f(x), \int_{(x,x')} \alpha, \iint_F \omega$.
The *coboundary* is dual to the boundary. $d : C^k(\Lambda) \to C^{k+1}(\Lambda)$, defined by Stokes formula

$$\int\limits_{(x,x')} df := f\left(\partial(x,x')\right) = f(x') - f(x), \qquad \iint\limits_F d\alpha := \oint\limits_{\partial F} \alpha.$$

A *cocycle* is a closed cochain $\alpha \in Z^k(\Lambda)$.

Scalar product weighted by ρ



$$\rho_{(x,x')} = -i\,\frac{y'-y}{x'-x}$$

$$(\alpha,\beta) := \frac{1}{2}\sum_{e\in\Lambda_1}\rho(e)\left(\int_e\alpha\right)\left(\int_e\bar{\beta}\right). \qquad (1)$$

A *Hodge operator* $*$

$$\begin{aligned} * : C^k(\Lambda) &\to C^{2-k}(\Lambda) \\ C^0(\Lambda) \ni f &\mapsto *f : \iint_F *f := f(F^*), \\ C^1(\Lambda) \ni \alpha &\mapsto *\alpha : \int_e *\alpha := -\rho(e^*) \int_{e^*} \alpha, \\ C^2(\Lambda) \ni \omega &\mapsto *\omega : (*\omega)(x) := \iint_{x^*} \omega. \end{aligned} \tag{2}$$

Verifies $*^2 = (-\mathrm{Id}_{C^k})^k$.
The discrete *laplacian* $\Delta = \Delta_\Gamma \oplus \Delta_{\Gamma^*} := -d * d * - * d * d$:

$$(\Delta(f))(x) = \sum_{k=1}^{V} \rho(x, x_k) \left(f(x) - f(x_k)\right).$$

Its kernel are the *harmonic forms*.

$$\alpha \in C^1(\Lambda) \text{ is } \textit{conformal} \text{ iff } d\alpha = 0 \text{ and } *\alpha = -i\alpha, \tag{3}$$

$$\pi_{(1,0)} = \frac{1}{2}(\mathrm{Id} + i\,*) \qquad \pi_{(0,1)} = \frac{1}{2}(\mathrm{Id} - i\,*)$$

$$d' := \pi_{(1,0)} \circ d : C^0(\Lambda) \to C^1(\Lambda), \qquad d' := d \circ \pi_{(1,0)} : C^1(\Lambda) \to C^2(\Lambda)$$

$$f \in \Omega^0(\Lambda) \text{ iff } d'(f) = 0.$$

$$-*\,d\,* = d^* \text{ the adjoint of the coboundary}$$

$$C^k(\Lambda) = \mathrm{Im}\ d \oplus^{\perp} \mathrm{Im}\ d^* \oplus^{\perp} \mathrm{Ker}\ \Delta,$$

$$\mathrm{Ker}\ \Delta = \mathrm{Ker}\ d \cap \mathrm{Ker}\ d^* = \mathrm{Ker}\ d' \oplus^{\perp} \mathrm{Ker}\ d''.$$

$$\wedge : C^k(\diamondsuit) \times C^l(\diamondsuit) \to C^{k+l}(\diamondsuit) \qquad \text{s.t.} \qquad (\alpha, \beta) = \iint \alpha \wedge *\bar{\beta}$$

For $f, g \in C^0(\diamondsuit)$, $\alpha, \beta \in C^1(\diamondsuit)$ et $\omega \in C^2(\diamondsuit)$:

$$(f \cdot g)(x) := f(x) \cdot g(x) \qquad \text{for } x \in \diamondsuit_0,$$

$$\int_{(x,y)} f \cdot \alpha := \frac{f(x) + f(y)}{2} \int_{(x,y)} \alpha \qquad \text{for } (x,y) \in \diamondsuit_1,$$

$$\iint_{(x_1,x_2,x_3,x_4)} \alpha \wedge \beta := \frac{1}{4} \sum_{k=1}^{4} \int_{(x_{k-1},x_k)} \alpha \int_{(x_k,x_{k+1})} \beta - \int_{(x_{k+1},x_k)} \alpha \int_{(x_k,x_{k-1})} \beta,$$

$$\iint_{(x_1,x_2,x_3,x_4)} f \cdot \omega := \frac{f(x_1)+f(x_2)+f(x_3)+f(x_4)}{4} \iint_{(x_1,x_2,x_3,x_4)} \omega$$

$$\text{for } (x_1, x_2, x_3, x_4) \in \diamondsuit_2.$$

$$\wedge : C^k(\diamondsuit) \times C^l(\diamondsuit) \to C^{k+l}(\diamondsuit) \qquad \text{s.t.} \qquad (\alpha,\beta) = \iint \alpha \wedge *\bar{\beta}$$

For $f, g \in C^0(\diamondsuit)$, $\alpha, \beta \in C^1(\diamondsuit)$ et $\omega \in C^2(\diamondsuit)$:

$$(f \cdot g)(x) := f(x)\cdot g(x) \qquad \text{for } x \in \diamondsuit_0,$$

$$\int_{(x,y)} f \cdot \alpha := \frac{f(x)+f(y)}{2} \int_{(x,y)} \alpha \qquad \text{for } (x,y) \in \diamondsuit_1,$$

$$\iint_{(x_1,x_2,x_3,x_4)} \alpha \wedge \beta := \frac{1}{4}\sum_{k=1}^{4} \int_{(x_{k-1},x_k)} \alpha \int_{(x_k,x_{k+1})} \beta - \int_{(x_{k+1},x_k)} \alpha \int_{(x_k,x_{k-1})} \beta,$$

$$\iint_{(x_1,x_2,x_3,x_4)} f \cdot \omega := \frac{f(x_1)+f(x_2)+f(x_3)+f(x_4)}{4} \iint_{(x_1,x_2,x_3,x_4)} \omega$$

$$\text{for } (x_1,x_2,x_3,x_4) \in \diamondsuit_2.$$

A form on $\diamondsuit$ can be *averaged* into a form on Λ:

$$\int_{(x,x')} A(\alpha_\diamondsuit) := \frac{1}{2}\left(\int_{(x,y)} + \int_{(y,x')} + \int_{(x,y')} + \int_{(y',x')}\right)\alpha_\diamondsuit, \tag{4}$$

$$\iint_{x^*} A(\omega_\diamondsuit) := \frac{1}{2}\sum_{k=1}^{d} \iint_{(x_k,y_k,x,y_{k-1})} \omega_\diamondsuit, \tag{5}$$

$\mathrm{Ker}(A) = \mathrm{Span}(d_\diamondsuit \varepsilon)$

- Hodge Star: $* : C^k \to C^{2-k}$, $\int_{(y,y')} * \alpha := \rho_{(x,x')} \int_{(x,x')} \alpha$.
- Discrete Laplacian:
  $d^* := - * d *$, $\Delta := d\, d^* + d^* d$,
  $\Delta f(x) = \sum \rho_{(x,x_k)} (f(x) - f(x_k))$.
- Hodge Decomposition:
  $C^k = \mathrm{Im} d \oplus^{\perp} \mathrm{Im} d^* \oplus^{\perp} \mathrm{Ker}\, \Delta$, $\mathrm{Ker}\, \Delta_1 = C^{(1,0)} \oplus^{\perp} C^{(0,1)}$.
- Weyl Lemma:
  $\Delta f = 0 \iff \iint f * \Delta g = 0$, $\forall g$ compact.
- Green Identity:
  $\iint_D (f * \Delta g - g * \Delta f) = \oint_{\partial D} (f * dg - g * df)$.

$$\alpha \in C^1 \text{ is conformal} \iff \begin{cases} *\alpha = -i\alpha & \text{of type (1,0)} \\ d\alpha = 0 & \text{closed.} \end{cases}$$

- α of type (1,0) (*i.e.* $\int_{(y,y')} \alpha = i\rho_{(x,x')} \int_{(x,x')} \alpha$) has a pole of order 1 in v with a residue $\operatorname{Res}_v \alpha := \frac{1}{2i\pi} \oint_{\partial v^*} \alpha \neq 0$.
- If α is not of type (1,0) on (x, y, x', y'), it has a pole of order > 1.
- Discrete Riemann-Roch theorem: Existence of forms with prescribed poles and holonomies.
- Green Function and Potential, Cauchy Integral Formula: $\oint_{\partial D} f \cdot \nu_{x,y} = 2i\pi \frac{f(x)+f(y)}{2}$.
- Period Matrix, Jacobian, Abel's map, Riemann bilinear relations.
- Continuous limit theorem.

The L^2-norm of the 1-form df is the **Dirichlet** energy

$$E_D(f) := \|df\|^2 = (df,\, df) = \frac{1}{2} \sum_{(x,x')\in\Lambda_1} \rho(x,x') \left| f(x') - f(x) \right|^2$$
$$= \frac{E_D(f|_\Gamma) + E_D(f|_{\Gamma^*})}{2}.$$

The **conformal energy** of the map measures its conformality defect

$$E_C(f) := \tfrac{1}{2}\|df - i * df\|^2.$$

Dirichlet and Conformal enregies are related through

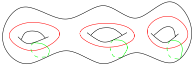
$$\begin{aligned} E_C(f) &= \tfrac{1}{2}\,(df - i * df,\, df - i * df) \\ &= \tfrac{1}{2}\|df\|^2 + \tfrac{1}{2}\|-i * df\|^2 + \operatorname{Re}(df,\, -i * df) \\ &= \|df\|^2 + \operatorname{Im}\iint_{\diamondsuit_2} df \wedge \overline{df} \\ &= E_D(f) - 2\mathcal{A}(f) \end{aligned}$$

with the algebraic area of the image

$$\mathcal{A}(f) := \frac{i}{2}\iint_{\diamondsuit_2} df \wedge \overline{df}$$

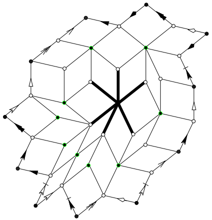
On a face the algebraic algebra of the image reads

$$\begin{aligned} \iint_{(x,y,x',y')} df \wedge \overline{df} &= i\operatorname{Im}\Big((f(x') - f(x))\overline{(f(y') - f(y))}\Big) \\ &= -2i\mathcal{A}\Big(f(x), f(x'), f(y), f(y')\Big) \end{aligned}$$

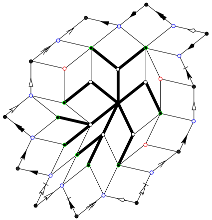


- **Basis of holomorphic forms**(a discrete Riemann surface S)
- find a normalized homotopy basis $\aleph$ of $\diamond(S)$
- **foreach** $\aleph_k$
-
  - compute $\aleph_k^\Gamma$ and $\aleph_k^{\Gamma^*}$
  - compute the real discrete harmonic form ω_k on Γ s.t. $\oint_\gamma \omega_k = \gamma \cdot \aleph_k^\Gamma$
  - compute the form $*\omega_k$ on Γ^*
  - check it is harmonic on on Γ^*
  - compute its holonomies $(\oint_{\aleph_\ell^{\Gamma^*}} *\omega_k)_{k,\ell}$ on the dual graph
- do some linear algebra (R is a rectangular complex matrix) to get the basis of holomorphic forms $(\zeta_k)_k = R(\mathrm{Id} + i\,*)(\omega_k)_k$ s.t. $(\oint_{\aleph_\ell^\Gamma} \zeta_k) = \delta_{k,\ell}$
- define the period matrix $\Pi_{k,\ell} := (\oint_{\aleph_\ell^{\Gamma^*}} \zeta_k)$

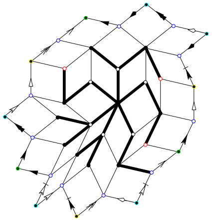
- **spanning tree**(a root vertex)
- $tree \leftarrow sd \leftarrow \{root\}; sdp1 \leftarrow \emptyset; points \leftarrow$ all the vertices;
- **while** $points \neq \emptyset$
- $\quad points \leftarrow points \setminus sd$
- $\quad$ **foreach** $v \in sd$
- - **foreach** $v' \sim v$
  - - **if** $v' \in points$
    - $sdp1 \leftarrow sdp1 \cup \{v'\}$
    - $tree \leftarrow tree \cup \{(v, v')\}$
- $\quad sd \leftarrow sdp1; sdp1 \leftarrow \emptyset$
- **return** $tree$



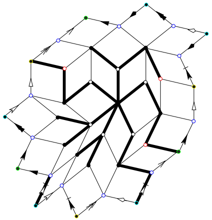
- **spanning tree**(a root vertex)
- $tree \leftarrow sd \leftarrow \{root\}; sdp1 \leftarrow \emptyset; points \leftarrow$ all the vertices;
- **while** $points \neq \emptyset$
- $\quad points \leftarrow points \setminus sd$
- $\quad$ **foreach** $v \in sd$
- - **foreach** $v' \sim v$
  - - **if** $v' \in points$
    - $\quad sdp1 \leftarrow sdp1 \cup \{v'\}$
    - $\quad tree \leftarrow tree \cup \{(v, v')\}$
- $\quad sd \leftarrow sdp1; sdp1 \leftarrow \emptyset$
- **return** $tree$



- **spanning tree**(a root vertex)
- $tree \leftarrow sd \leftarrow \{root\}; sdp1 \leftarrow \emptyset; points \leftarrow$ all the vertices;
- **while** $points \neq \emptyset$
-     $points \leftarrow points \setminus sd$
-     **foreach** $v \in sd$
-     - **foreach** $v' \sim v$
    -     - **if** $v' \in points$
          -     $sdp1 \leftarrow sdp1 \cup \{v'\}$
          -     $tree \leftarrow tree \cup \{(v, v')\}$
-     $sd \leftarrow sdp1; sdp1 \leftarrow \emptyset$
- **return** $tree$

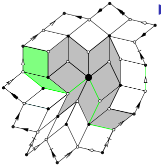


- **spanning tree**(a root vertex)
- $tree \leftarrow sd \leftarrow \{root\}; sdp1 \leftarrow \emptyset; points \leftarrow$ all the vertices;
- **while** $points \neq \emptyset$
- $\quad points \leftarrow points \setminus sd$
- $\quad$ **foreach** $v \in sd$
-
  - **foreach** $v' \sim v$
-
  - **if** $v' \in points$
    - $sdp1 \leftarrow sdp1 \cup \{v'\}$
    - $tree \leftarrow tree \cup \{(v, v')\}$
- $\quad sd \leftarrow sdp1; sdp1 \leftarrow \emptyset$
- **return** *tree*



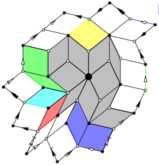
- **fundamental polygon**(a spanning tree)
- $faces \leftarrow$ all the faces; $edges \leftarrow tree \cup \overline{tree}$;
  $toClose : faces \to \mathcal{P}$(all the edges);
- **foreach** $(face \in faces)$ $facesToClose(face) \leftarrow \partial(face) \setminus edges$
- **do**
-
  - $finished \leftarrow true$
  - **foreach** $(face \in faces$ such that $|facesToClose(face)| == 1)$
  -
    - $finished \leftarrow false$; $e = facesToClose(face)$;
    - $edges \leftarrow edges \setminus \{e, \bar{e}\}$; $faces \leftarrow faces \setminus \{face\}$;
      $lface \leftarrow leftFace(\bar{e})$;
    - **if** $(lface)$
    - $facesToClose(lface) \leftarrow facesToClose(lface \setminus \{\bar{e}\})$

    if $(|facesToClose(lface)| == 0)$ $faces \leftarrow faces \setminus \{lface\}$;



- **while** (not($finished$));
- **return** $facesToClose$

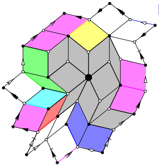
- **fundamental polygon**(a spanning tree)
- $faces \leftarrow$ all the faces; $edges \leftarrow tree \cup \overline{tree}$;
  $toClose : faces \rightarrow \mathcal{P}$(all the edges);
- **foreach** $(face \in faces)$ $facesToClose(face) \leftarrow \partial(face) \setminus edges$
- **do**
-
  - $finished \leftarrow true$
  - **foreach** $(face \in faces$ such that $|facesToClose(face)| == 1)$
  -
    - $finished \leftarrow false$; $e = facesToClose(face)$;
    - $edges \leftarrow edges \setminus \{e, \bar{e}\}$; $faces \leftarrow faces \setminus \{face\}$;
      $lface \leftarrow leftFace(\bar{e})$;
    - **if** $(lface)$
    - $facesToClose(lface) \leftarrow facesToClose(lface \setminus \{\bar{e}\})$
    - **if** $(|facesToClose(lface)| == 0)$ $faces \leftarrow faces \setminus \{lface\}$;



- **while** (not($finished$));
- **return** $facesToClose$

- **fundamental polygon**(a spanning tree)
- $faces \leftarrow$ all the faces; $edges \leftarrow tree \cup \overline{tree}$;
  $toClose : faces \rightarrow \mathcal{P}$(all the edges);
- **foreach** ($face \in faces$) $facesToClose(face) \leftarrow \partial(face) \setminus edges$
- **do**
-
  - $finished \leftarrow true$
  - **foreach** ($face \in faces$ such that $|facesToClose(face)| == 1$)
  -
    - $finished \leftarrow false$; $e = facesToClose(face)$;
    - $edges \leftarrow edges \setminus \{e, \bar{e}\}$; $faces \leftarrow faces \setminus \{face\}$;
      $lface \leftarrow leftFace(\bar{e})$;
    - **if** ($lface$)
    - $facesToClose(lface) \leftarrow facesToClose(lface \setminus \{\bar{e}\})$

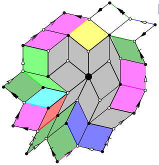
    if ($|facesToClose(lface)| == 0$) $faces \leftarrow faces \setminus \{lface\}$;



- **while** (not($finished$));
- **return** $facesToClose$

- **fundamental polygon**(a spanning tree)
- $\mathit{faces} \leftarrow$ all the faces; $\mathit{edges} \leftarrow \mathit{tree} \cup \overline{\mathit{tree}}$;
  $\mathit{toClose} : \mathit{faces} \to \mathcal{P}$(all the edges);
- **foreach** $(\mathit{face} \in \mathit{faces})$ $\mathit{facesToClose}(\mathit{face}) \leftarrow \partial(\mathit{face}) \setminus \mathit{edges}$
- **do**
-
  - $\mathit{finished} \leftarrow \mathit{true}$
  - **foreach** $(\mathit{face} \in \mathit{faces}$ such that $|\mathit{facesToClose}(\mathit{face})| == 1)$
  -
    - $\mathit{finished} \leftarrow \mathit{false}$; $e = \mathit{facesToClose}(\mathit{face})$;
    - $\mathit{edges} \leftarrow \mathit{edges} \setminus \{e, \bar{e}\}$; $\mathit{faces} \leftarrow \mathit{faces} \setminus \{\mathit{face}\}$;
      $\mathit{lface} \leftarrow \mathit{leftFace}(\bar{e})$;
    - **if** (lface)
    - $\mathit{facesToClose}(\mathit{lface}) \leftarrow \mathit{facesToClose}(\mathit{lface} \setminus \{\bar{e}\})$

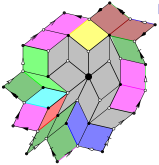
    if $(|\mathit{facesToClose}(\mathit{lface})| == 0)$ $\mathit{faces} \leftarrow \mathit{faces} \setminus \{\mathit{lface}\}$;



- **while** (not($\mathit{finished}$));
- **return** $\mathit{facesToClose}$

- **fundamental polygon**(a spanning tree)
- $faces \leftarrow$ all the faces; $edges \leftarrow tree \cup \overline{tree}$;
  $toClose : faces \rightarrow \mathcal{P}$(all the edges);
- **foreach** $(face \in faces)$ $facesToClose(face) \leftarrow \partial(face) \setminus edges$
- **do**
-
    - $finished \leftarrow true$
    - **foreach** ($face \in faces$ such that $|facesToClose(face)| == 1$)
    -
        - $finished \leftarrow false$; $e = facesToClose(face)$;
        - $edges \leftarrow edges \setminus \{e, \bar{e}\}$; $faces \leftarrow faces \setminus \{face\}$;
          $lface \leftarrow leftFace(\bar{e})$;
        - **if** $(lface)$
        - $facesToClose(lface) \leftarrow facesToClose(lface \setminus \{\bar{e}\})$

        if $(|facesToClose(lface)| == 0)$ $faces \leftarrow faces \setminus \{lface\}$;



- **while** (not($finished$));
- **return** $facesToClose$

∂ and $\bar\partial$

In the continuous case

for $f(z+z_0) = f(z_0) + z \times (\partial f)(z_0) + \bar z \times (\bar\partial f)(z_0) + o(|z|)$,

$$(\partial f)(z_0) = \lim_{\gamma\to z_0} \frac{i}{2\mathcal{A}(\gamma)} \oint_\gamma f d\bar z, \qquad (\bar\partial f)(z_0) = -\lim_{\gamma\to z_0} \frac{i}{2\mathcal{A}(\gamma)} \oint_\gamma f dZ,$$

along a loop γ around z_0.
leading to the discrete definition

$$\begin{array}{rcl} \partial : C^0(\diamondsuit) & \to & C^2(\diamondsuit) \\ f & \mapsto & \partial f = \left[(x,y,x',y') \mapsto -\frac{i}{2\mathcal{A}(x,y,x',y')} \oint\limits_{(x,y,x',y')} f d\bar Z\right] \\ & & = \frac{(f(x')-f(x))(\bar y'-\bar y)-(\bar x'-\bar x)(f(y')-f(y))}{(x'-x)(\bar y'-\bar y)-(\bar x'-\bar x)(y'-y)}, \end{array}$$

$$\begin{array}{rcl} \bar\partial : C^0(\diamondsuit) & \to & C^2(\diamondsuit) \\ f & \mapsto & \bar\partial f = \left[(x,y,x',y') \mapsto -\frac{i}{2\mathcal{A}(x,y,x',y')} \oint\limits_{(x,y,x',y')} f dZ\right] \end{array}$$

A conformal map f fulfills $\bar\partial f \equiv 0$ and (with $Z(u)$ denoted u)

$$\partial f(x,y,x',y') = \frac{f(y')-f(y)}{y'-y} = \frac{f(x')-f(x)}{x'-x}.$$

The jacobian $J = |\partial f|^2 - |\bar\partial f|^2$ compares the areas:

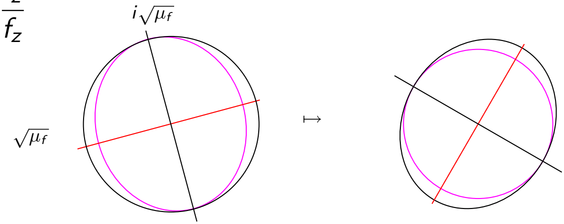
$$\iint_{(x,y,x',y')} df \wedge \overline{df} = J \iint_{(x,y,x',y')} dZ \wedge \overline{dZ}.$$

For a discrete function, one defines the *dilatation* coefficient

$$D_f := \frac{|f_z| + |f_{\bar{z}}|}{|f_z| - |f_{\bar{z}}|}$$

$D_f \geq 1$ for $|f_{\bar{z}}| \leq |f_z|$ (quasi-conformal). Written in terms of the *complex dilatation*:

$$\mu_f = \frac{f_{\bar{z}}}{f_z}$$

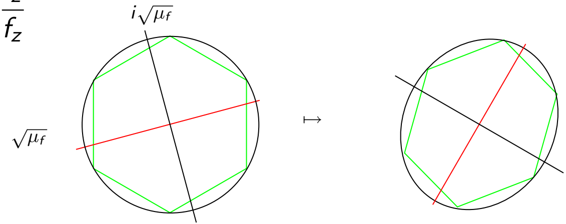


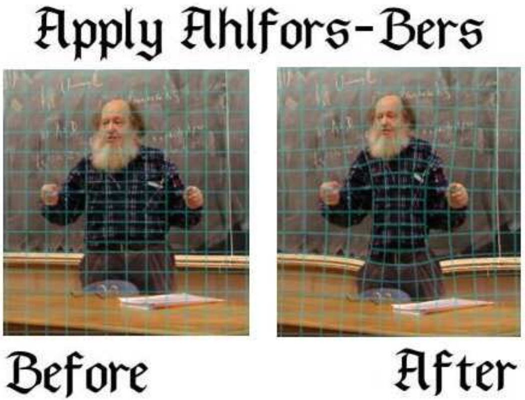
For a discrete function, one defines the *dilatation* coefficient

$$D_f := \frac{|f_z| + |f_{\bar{z}}|}{|f_z| - |f_{\bar{z}}|}$$

$D_f \geq 1$ for $|f_{\bar{z}}| \leq |f_z|$ (quasi-conformal). Written in terms of the *complex dilatation*:

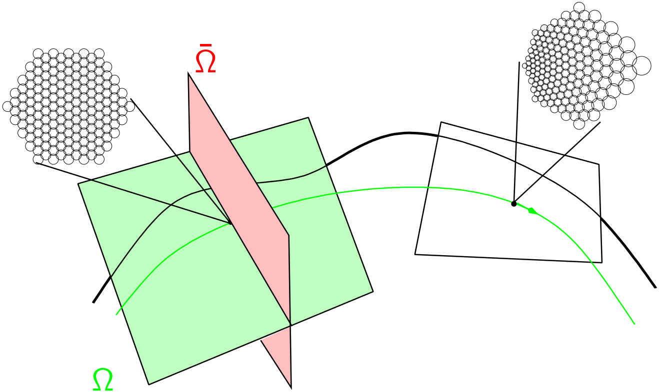
$$\mu_f = \frac{f_{\bar{z}}}{f_z}$$

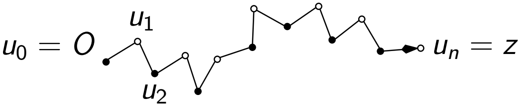




(Arnaud Chéritat and Xavier Buff, L. Émile Picard, Toulouse)

For a given combinatorics, cross-ratio distributions foliates the space of maps. Ω stays on the same complex structure, $\bar{\Omega}$ changes it.





$$\exp(:\lambda: z) = \prod_k \frac{1 + \frac{\lambda}{2}(u_k - u_{k-1})}{1 - \frac{\lambda}{2}(u_k - u_{k-1})}.$$

Generalization of the well known formula

$$\exp(\lambda\, z) = \left(1 + \frac{\lambda\, z}{n}\right)^n + O(\frac{z^2}{n}) = \left(\frac{1 + \frac{\lambda\, z}{2\, n}}{1 - \frac{\lambda\, z}{2\, n}}\right)^n + O(\frac{z^3}{n^2}).$$

The Green function on a lozenges graph is (**Richard Kenyon**)

$$G(O,x) = -\frac{1}{8\,\pi^2\,i}\oint_C \exp(:\lambda:x)\,\frac{\log\frac{\delta}{2}\lambda}{\lambda}\,d\lambda$$

$$\Delta G(O,x) = \delta_{O,x}, \quad G(O,x) \sim_{x\to\infty} \begin{cases} \log|x| & \text{on black vertices,} \\ i\,\arg(x) & \text{on white vertices.} \end{cases}$$

Differentiation of the exponential with respect to λ yields

$$\frac{\partial^k}{\partial\lambda^k}\exp(:\lambda: z) =: Z^{:k:}\exp(:\lambda: z)$$

and $\lambda = 0$ defines the discrete polynomials $Z^{:k:}$ fulfilling $\exp(:\lambda: z) = \sum \frac{Z^{:k:}}{k!}$, absolutely convergent for $\lambda \cdot \delta < 1$.
The primitive of a conformal map is itself conformal:

$$\int_{(x,y)} f\,dZ := \frac{f(x)+f(y)}{2}(y-x).$$

One can solve “differential equations”, and polynomials fulfill

$$Z^{:k+1:} = (k+1)\int Z^{:k:}\,dZ,$$

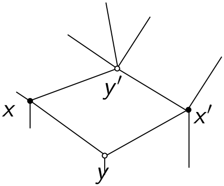
while the exponential fulfills

$$\exp(:\lambda: z) = \frac{1}{\lambda}\int \exp(:\lambda: z)\,dZ.$$

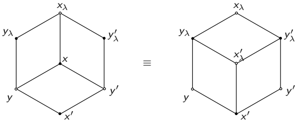
There exists a discrete duality, with $\varepsilon = \pm 1$ on black and white vertices, $f^\dagger = \varepsilon\,\bar{f}$ is conformal.
The derivative f' of a conformal map f is

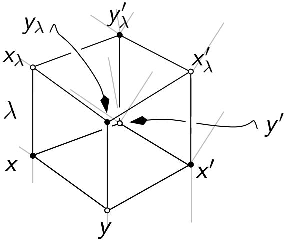
$$f'(z) := \frac{4}{\delta^2}\left(\int_O^z f^\dagger dZ\right)^\dagger + \lambda\,\varepsilon,$$

and verifies $f = \int f' dZ$.

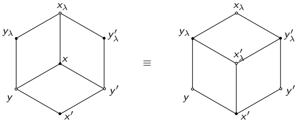


We impose "vertically" the same equation as "horizontally". The $3D$ consistency or **Yang-Baxter** equation yields integrability for rhombic quad-graphs:



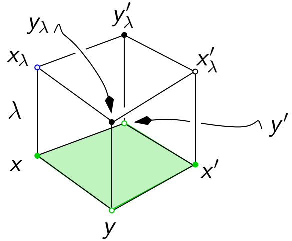


We impose "vertically" the same equation as "horizontally". The $3D$ consistency or **Yang-Baxter** equation yields integrability for rhombic quad-graphs:

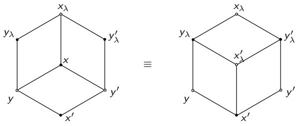


$$i\,\rho = \frac{y-y'}{x-x'}$$

$$q = \frac{(y-x)^2}{(y'-x')^2}$$

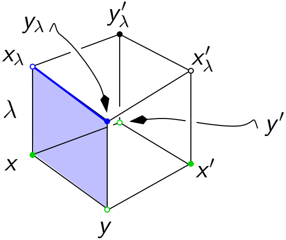


We impose “vertically” the same equation as “horizontally”.
The $3D$ consistency or **Yang-Baxter** equation yields integrability for rhombic quad-graphs:

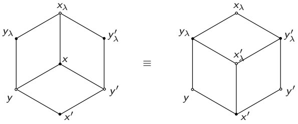


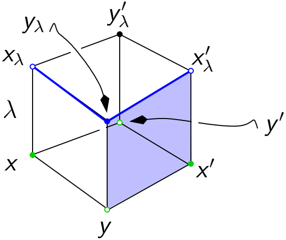
$i\,\rho = \frac{y-x+\lambda}{x-y+\lambda}$

$q = \frac{(y-x)^2}{\lambda^2}$

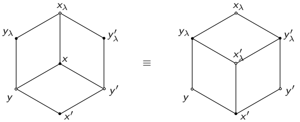


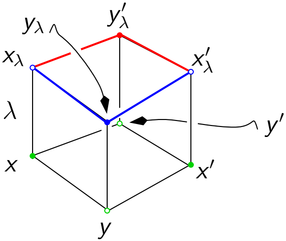
We impose "vertically" the same equation as "horizontally".
The $3D$ consistency or **Yang-Baxter** equation yields integrability for rhombic quad-graphs:



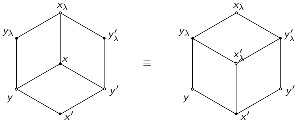


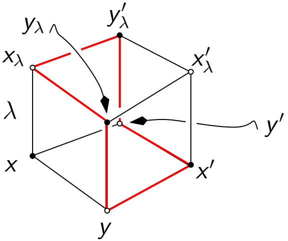
We impose "vertically" the same equation as "horizontally".
The $3D$ consistency or **Yang-Baxter** equation yields integrability for rhombic quad-graphs:



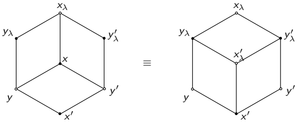


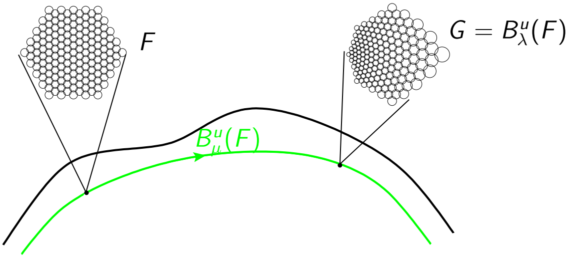
We impose "vertically" the same equation as "horizontally".
The $3D$ consistency or **Yang-Baxter** equation yields integrability for rhombic quad-graphs:



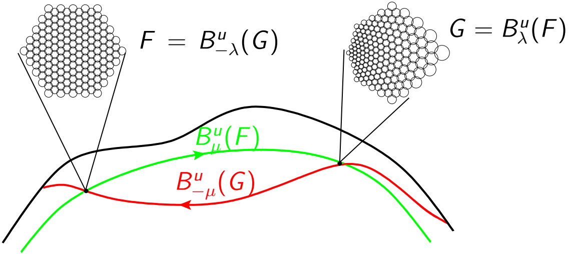


We impose "vertically" the same equation as "horizontally".
The $3D$ consistency or **Yang-Baxter** equation yields integrability for rhombic quad-graphs:

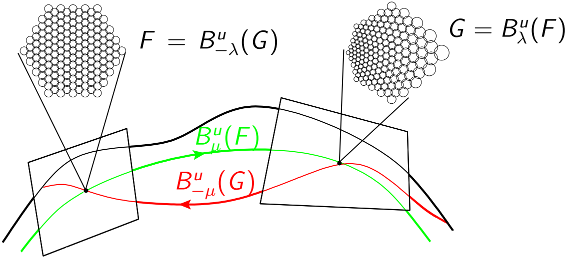




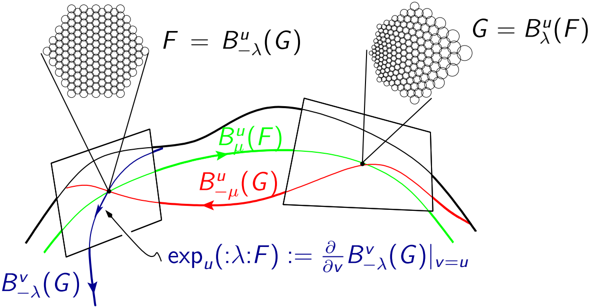
A family of CR-preserving maps, initial condition u, "vertical" parameter λ. Induces a linear map between the tangent spaces. Its kernel: **discrete exponential** $\exp(:\lambda: F)$. Form a **basis**



A family of CR-preserving maps, initial condition u, “vertical” parameter λ. Induces a linear map between the tangent spaces. Its kernel: **discrete exponential** $\exp(:\lambda: F)$. Form a **basis**



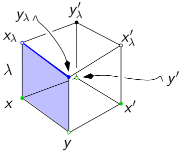
A family of CR-preserving maps, initial condition u, "vertical" parameter λ. Induces a linear map between the tangent spaces. Its kernel: **discrete exponential** $\exp(:\lambda: F)$. Form a **basis**



A family of CR-preserving maps, initial condition u, "vertical" parameter λ. Induces a linear map between the tangent spaces. Its kernel: **discrete exponential** $\exp(:\lambda: F)$. Form a **basis**

$i\,\rho = \frac{y-x+\lambda}{x-y+\lambda}$

$q = \frac{(y-x)^2}{\lambda^2}$



The vertical equation is

$$\frac{\left(f_\lambda(y) - f_\lambda(x)\right)\left(f(y) - f(x)\right)}{\left(f_\lambda(y) - f(y)\right)\left(f_\lambda(x) - f(x)\right)} = \frac{(y-x)^2}{\lambda^2}$$

With the change of variables $g = \ln f$, the continuous limit $(y = x + \epsilon)$ is:

$$\sqrt{g'_\lambda \times g'} = \frac{2}{\lambda} \sinh \frac{g_\lambda - g}{2}.$$

The Bäcklund transformation allows to define a notion of discrete holomorphy in $\mathbb{Z}^d$, for $d > 1$ finite, equipped with rapidities $(\alpha_i)_{1 \leq i \leq d}$.

$$\begin{tikzcd} x+e_j \arrow[r, dash, "\alpha_i"] \arrow[d, dash, "\alpha_j"'] & x+e_i+e_j \arrow[d, dash, "\alpha_j"] \\ x \arrow[r, dash, "\alpha_i"'] & x+e_i \end{tikzcd}$$

$$\rho = \frac{\alpha_i+\alpha_j}{\alpha_i-\alpha_j}$$

$$q = \frac{\alpha_i^2}{\alpha_j^2}$$

$$f(x+e_i+e_j) = L_i(x+e_j)f(x+e_j),$$

$$L_i(x;\lambda) = \begin{pmatrix} \lambda+\alpha_i & -2\alpha_i(f(x+e_i)+f(x)) \\ 0 & \lambda-\alpha_i \end{pmatrix}$$

The *moving frame* $\Psi(\cdot,\lambda) : \mathbb{Z}^d \to GL_2(\mathbb{C})[\lambda]$ for a prescribed $\Psi(0;\lambda)$:

$$\Psi(x+e_i;\lambda) = L_i(x;\lambda)\Psi(x;\lambda).$$

Define $A(\cdot;\lambda):\mathbb{Z}^d \to GL_2(\mathbb{C})[\lambda]$ by $A(x;\lambda) = \frac{d\Psi(x;\lambda)}{d\lambda}\Psi^{-1}(x;\lambda)$.

They satisfy the recurrent relation

$$A(x+e_k;\lambda) = \frac{dL_k(x;\lambda)}{d\lambda}L_k^{-1}(x;\lambda) + L_k(x;\lambda)A(x;\lambda)L_k^{-1}(x;\lambda).$$

A discrete holomorphic function $f:\mathbb{Z}^d \to \mathbb{C}$ is called **isomonodromic**, if, for some choice of $A(0;\lambda)$, the matrices $A(x;\lambda)$ are meromorphic in λ, with poles whose positions and orders do not depend on $x \in \mathbb{Z}^d$. Isomonodromic solutions can be constructed with prescribed boundary conditions. Example: the **Green function**.

Inter2Geo

Interoperable Interactive Geometry

A common file format and a common API for all interactive geometry software

- Cabri
- Cinderella
- Geoplan/Geospace
- Geogebra
- Geonext
- TracEnPoche (Sésamath/MathEnPoche)
- Z.u.L C.a.R
- Kig, DrGeo, GeoProof (Coq geometric prover)
- ...