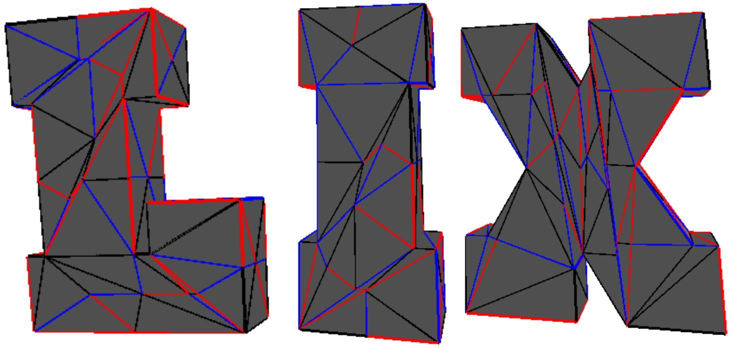
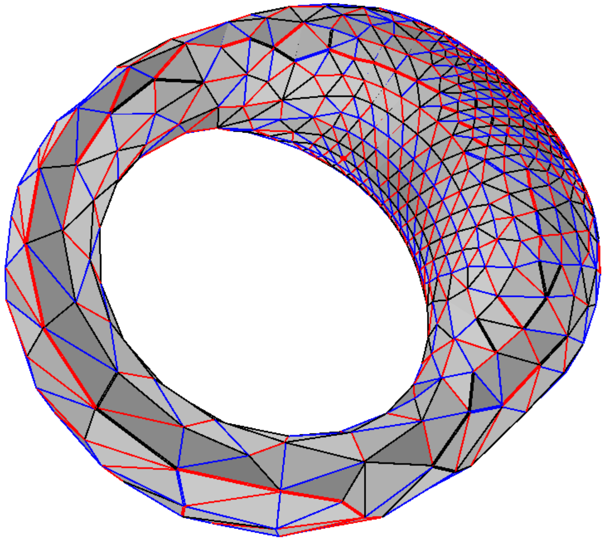
Habilitation à diriger des recherches
(Université Paris Cité)

Algorithms for graphs on surfaces: from graph drawing to graph encoding

june 27th, 2022

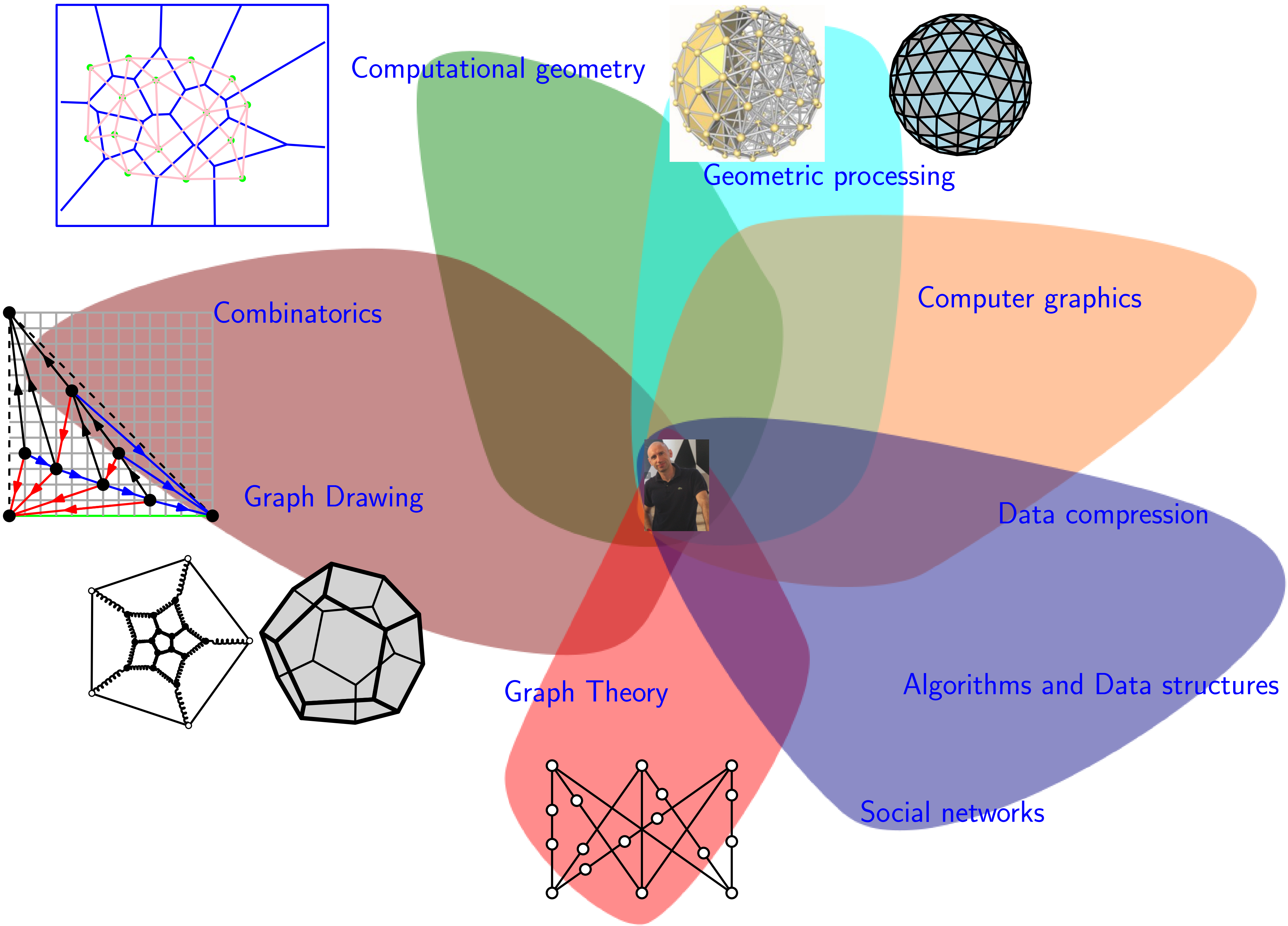
Luca Castelli Aleardi





At the frontier of several domains (where graphs play an important role)

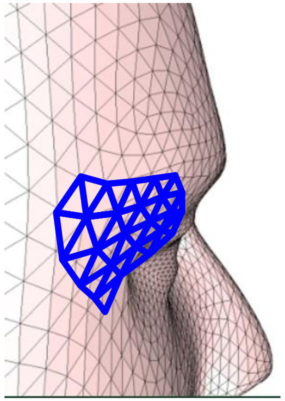
(from computational geometry to geometric processing, combinatorics, graph algorithms, ...)

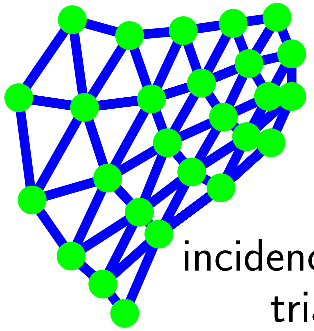


3D surface meshes, graphs on surfaces

What is a surface mesh?

informally: a set of vertices, edges and faces (polygons) defining a polyhedral surface embedded in 3D (discrete approximation of a shape)





incidence relations between triangles, vertices

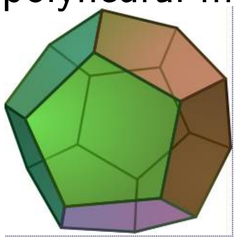
for us this is equivalent to

combinatorial map: a graph + a combinatorial embedding (on a surface)

$+$

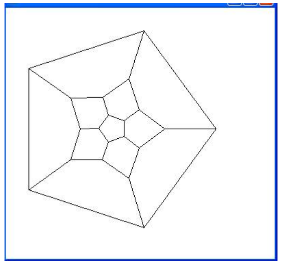
geometric realization in 2D or 3D

genus 0 polyhedral mesh



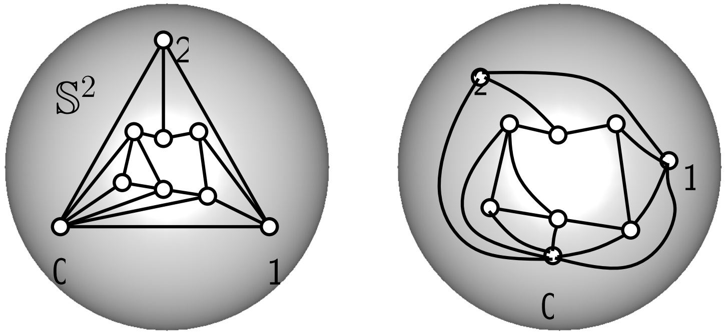
$\approx$

planar map



a Cellular graph embedding is

a 1-to-1 continuous map of G into S^2 s.t:
edges are represented as paths (curves) with no crossings (their interiors are disjoints)
faces are homemorphic to topological disks



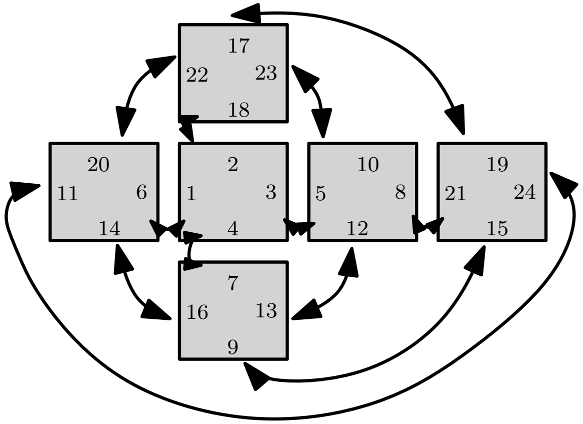
two cellular embeddings defining the same planar graph

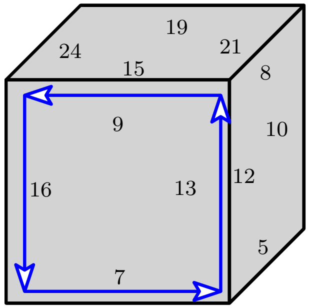
Combinatorial map

2 permutations on the set H of the $2m$ **darts**

(i) α involution without fixed point;

(ii) ϕ gives the cyclic ordering of the darts (edges) around each face





$\phi = (1,2,3,4)(17,23,18,22)(5,10,8,12)\ldots$

$\alpha = (2,18)(3,5)(4,7)(12,13)(9,15)\ldots$

Some facts about planar graphs

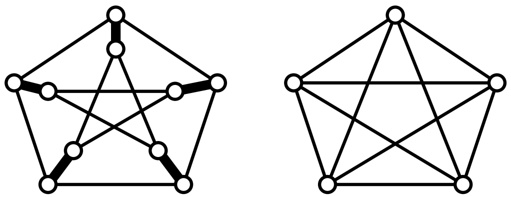
(genus 0 meshes)

("As I have known them")

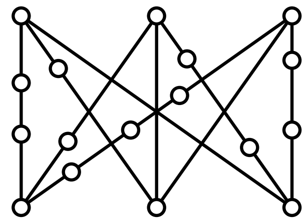
Major results on planar graphs

Kuratowski theorem (1930) (cfr Wagner's theorem, 1937)

- G contains neither K_5 nor $K_{3,3}$ as minors (or no subdivisions of K_5 nor $K_{3,3}$)



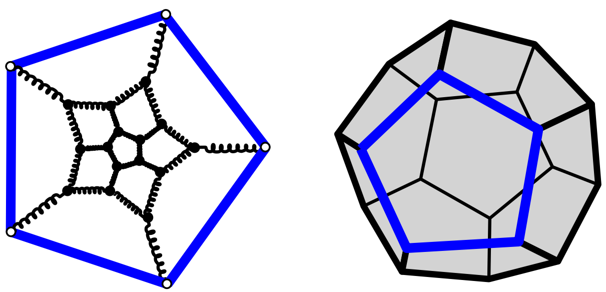
K_5 is a minor of the Petersen graph



subdivision of $K_{3,3}$

Thm (Tutte barycentric method, 1963)

Every 3-connected planar graph G admits a convex representation in R^2.



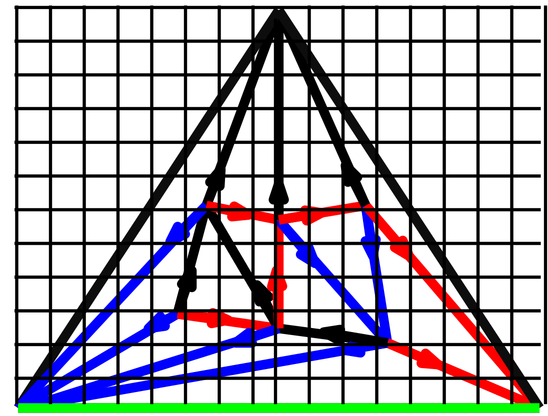
Thm (Colin de Verdière, 1990) Colin de Verdiere invariant (multiplicity of λ_2 eigenvalue of a generalized laplacian)

- $\mu(G) \leq 3$

$$\begin{bmatrix} 4 & -1 & \dots & \dots & 0 \\ -1 & 5 & \dots & & \\ \dots & & \dots & & \\ \dots & & & & \dots \\ 0 & \dots & & & 3 \end{bmatrix} \qquad L_G[i,k] = \begin{cases} deg(v_i) \\ -A_G[i,j] \end{cases}$$

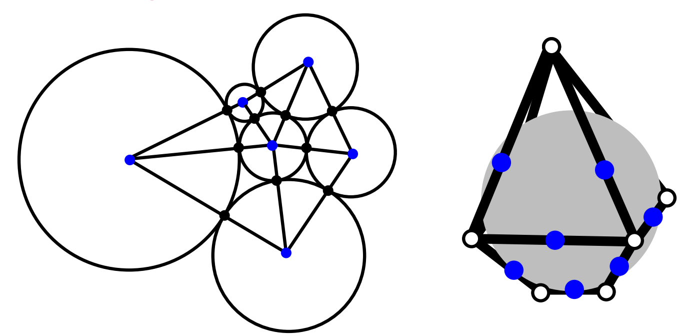
Schnyder woods ('89)

- planarity criterion via dimension of partial orders: $dim(G) \leq 3$
- linear-time grid drawing, with $O(n) \times O(n)$ resolution



Thm (Koebe-Andreev-Thurston)

Every planar graph with n vertices is isomorphic to the intersection graph of n disks in the plane.



Planar straight-line drawings

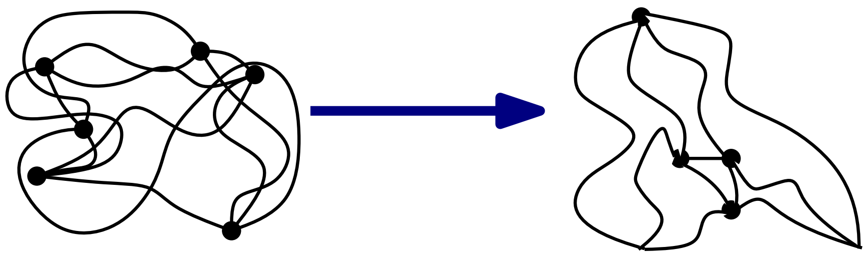
(of planar graphs)

Straight-line planar drawings of planar graphs

Problem definition (Planarity testing, Embedding a planar graph)

Input: a planar graph

Output: the planar map (cellularly embedded graph)



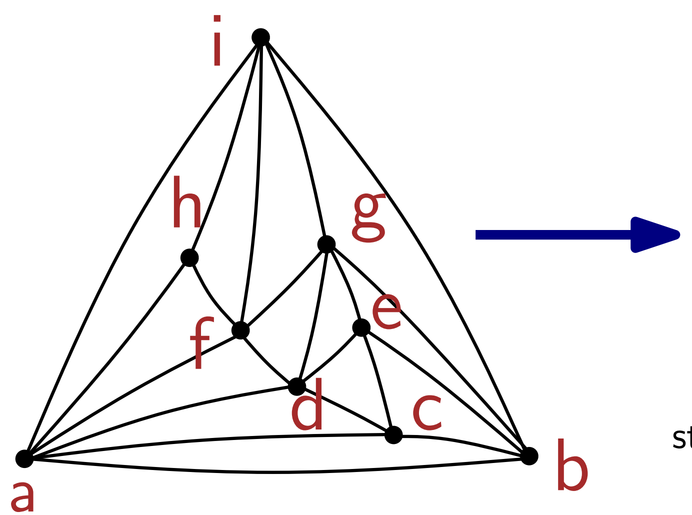
Problem definition (drawing graphs in the plane)

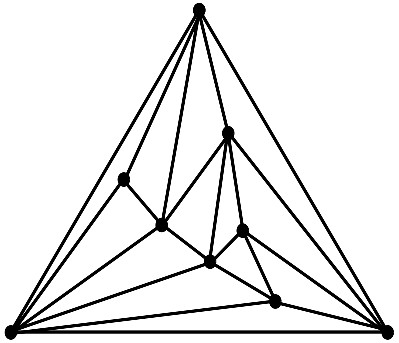
Input: a (planar) map

Output: a straight-line planar drawing (crossing-free)

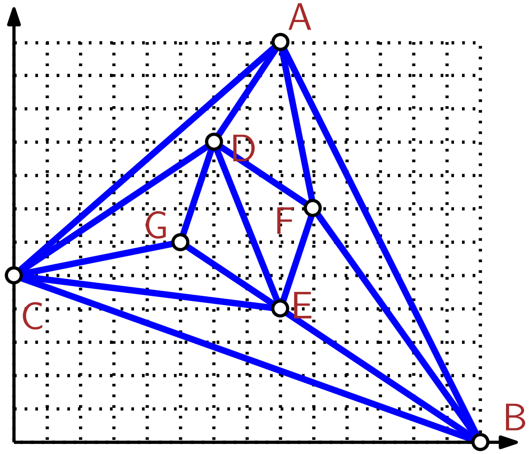
Input of the problem: planar map

(a, b, c) (d, e, g)
(a, c, d) (e, b, g)
(d, c, e) (a, f, h)
(c, b, e) (a, h, i)
(a, d, f) (i, h, f)
(f, d, g) (i, f, g)
(i, g, b) (i, b, a)



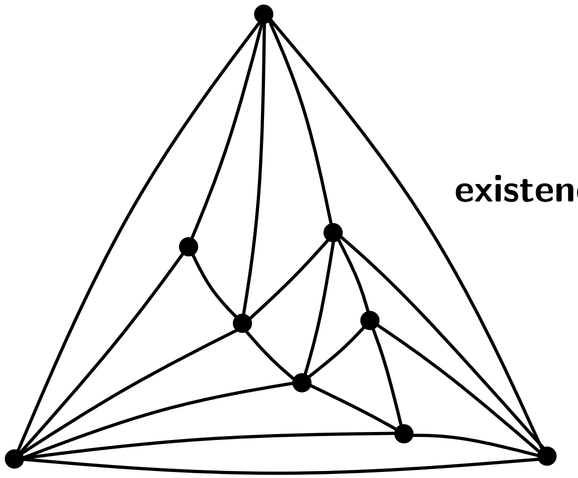


straight-line crossing-free drawing



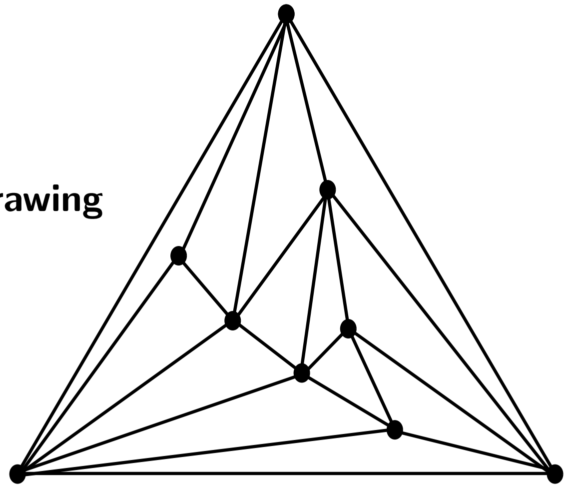
straight-line grid drawing
(bounded area and resolution)

Planar straight-line drawings



existence of straight-line drawing

$\Rightarrow$

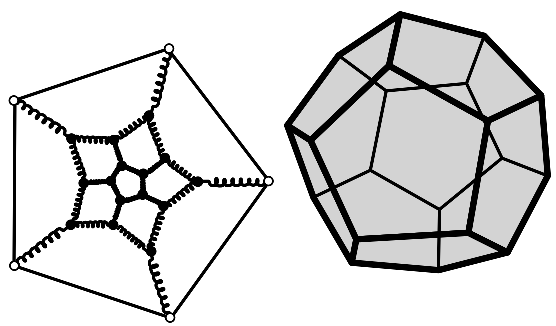


[Wagner'36]

[Fary'48]

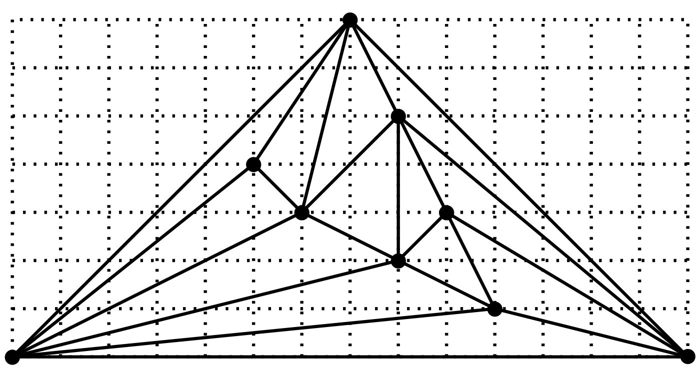
[Stein'51]

Classical algorithms:



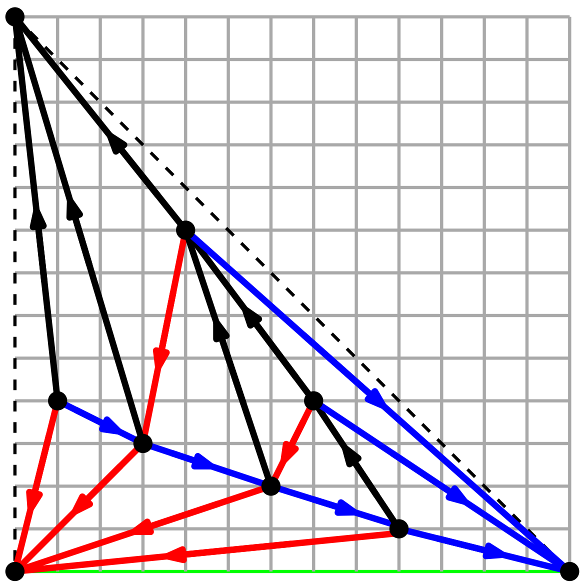
[Tutte'63]

spring-embedding



[De Fraysseix, Pach, Pollack 89]

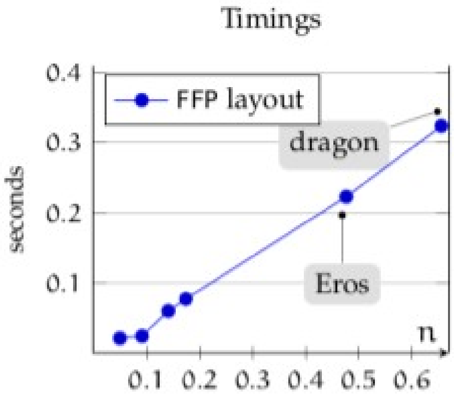
incremental (**Shift-algorithm**)

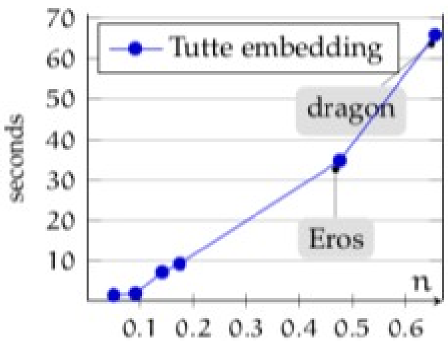


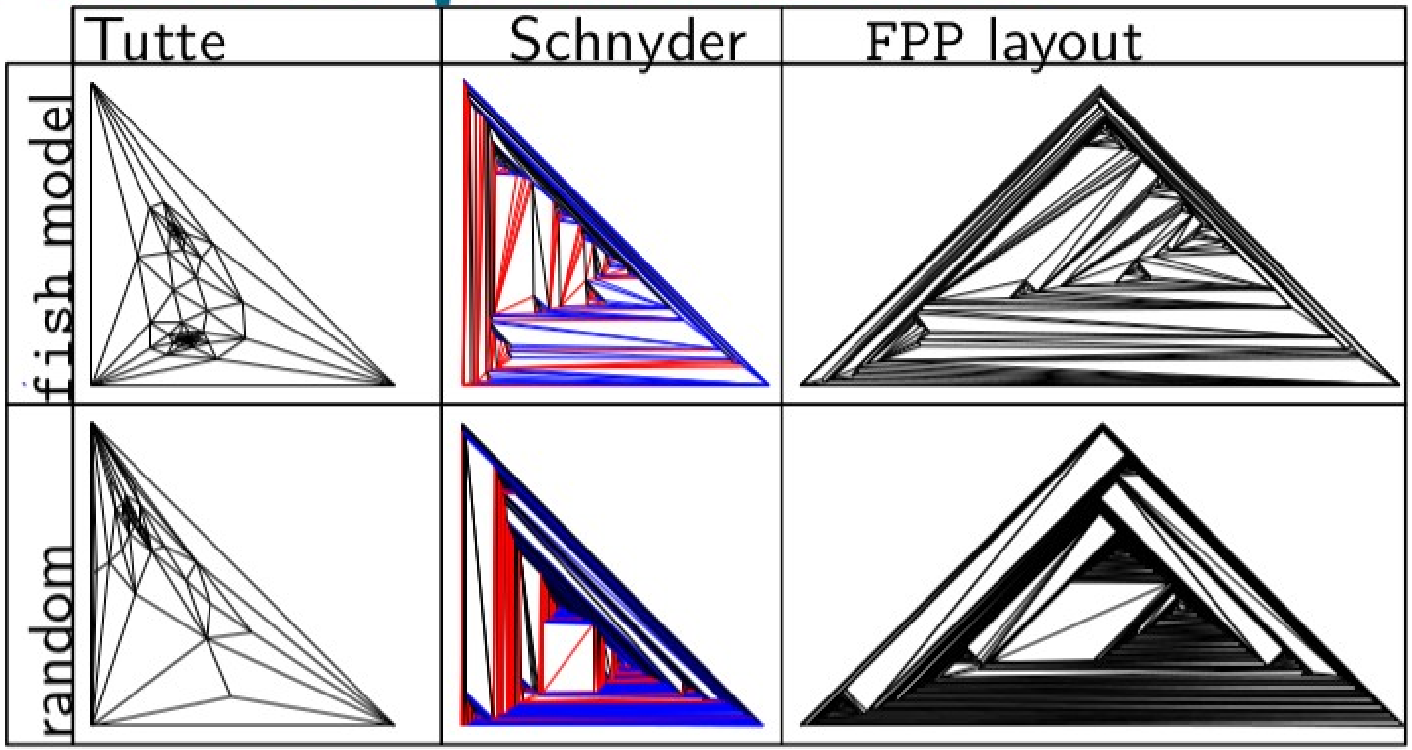
[Schnyder'90]

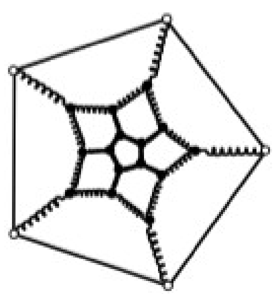
face-counting principle

Practical performances









[Tutte'63]

minimize the spring energy

$$E(\rho) := \sum_{(i,j)\in E} |\mathbf{x}(v_i) - \mathbf{x}(v_j)|^2 = \sum_{(i,j)\in E} (x_i - x_j)^2 + (y_i - y_j)^2$$

solve large sparse linear systems

$$\mathbf{x}(v_i) = \sum_{j \in \mathcal{N}(i)} \frac{1}{deg(v_i)} \mathbf{x}(v_j)$$

(Fruchterman and Reingold, 1991)

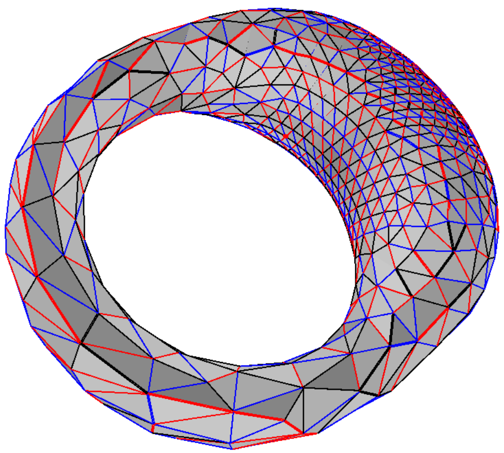
force-directed paradigm

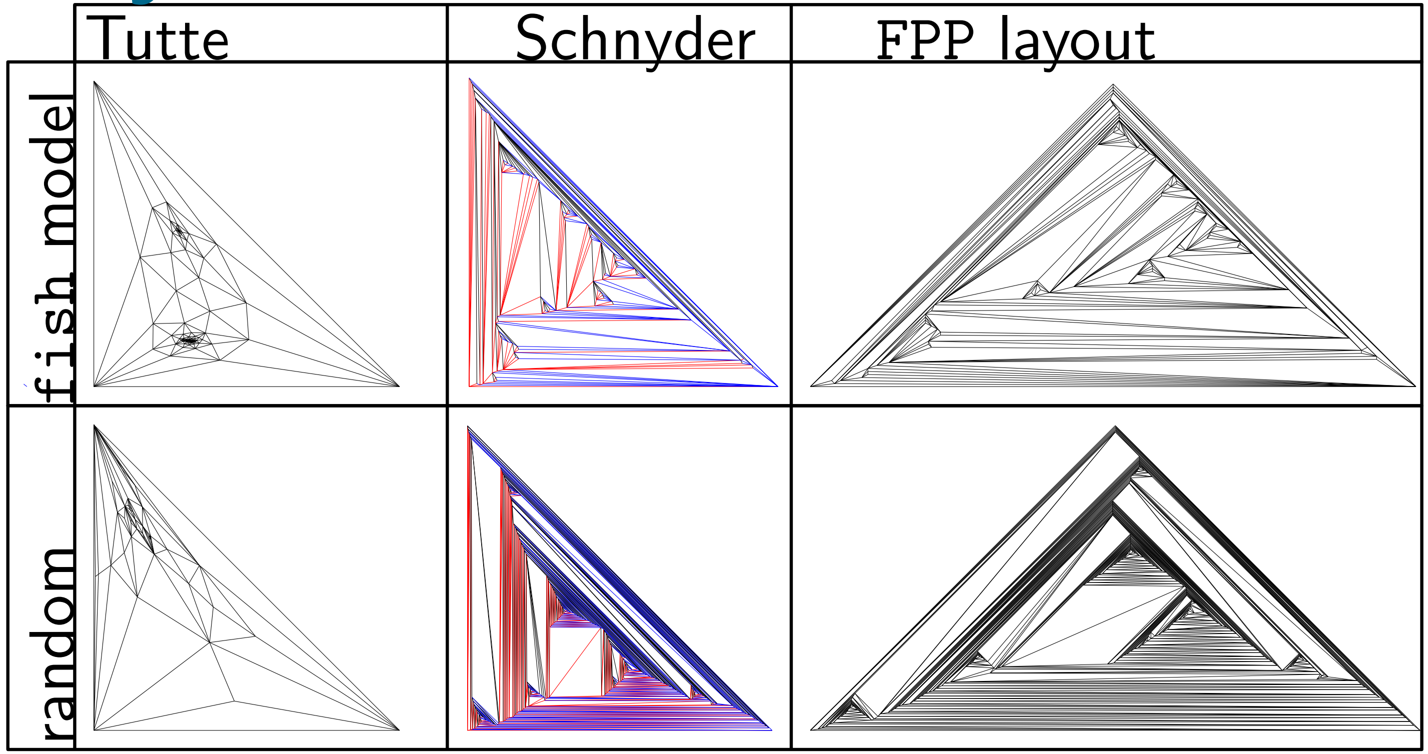
$$\mathbf{F}_a(v) = c_1 \cdot \sum_{(u,v)\in E} \log(dist(u,v)/c_2)$$

$$\mathbf{F}_r(v) = c_3 \cdot \sum_{u \in V} \frac{1}{\sqrt{dist(u,v)}}$$

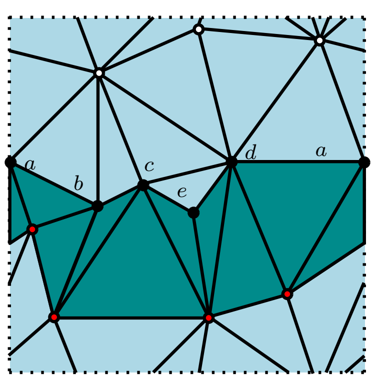
My contributions

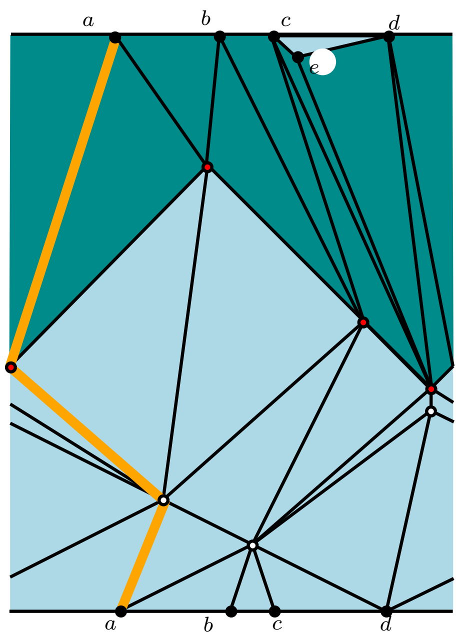
Schnyder woods and canonical orderings for higher genus surfaces



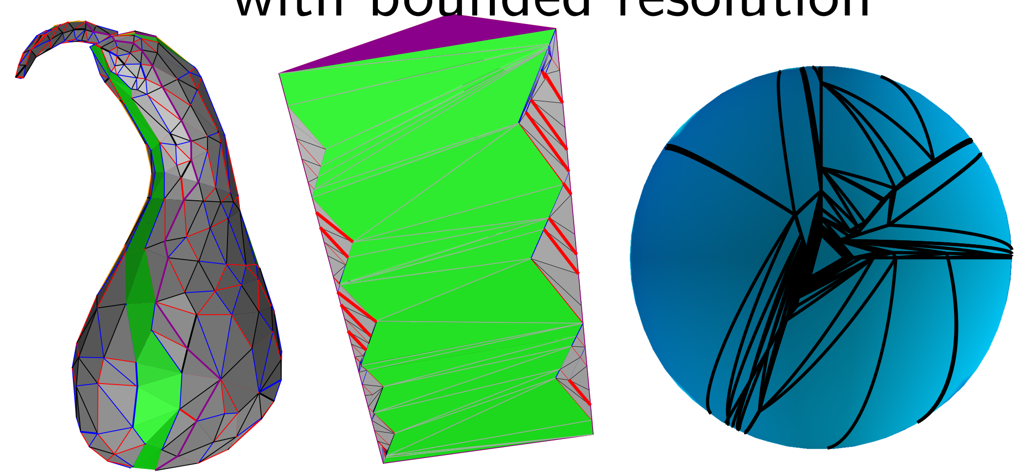


periodic toroidal drawings





Spherical drawings with bounded resolution

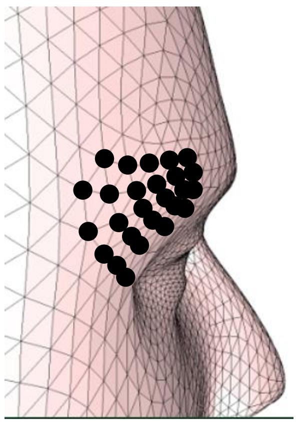


Graph encoding

Graph encoding problem: motivation

Geometric v.s combinatorial information

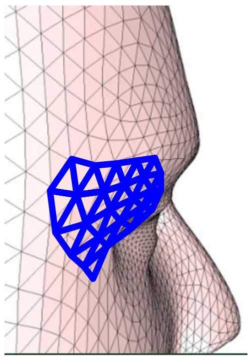
Geometry

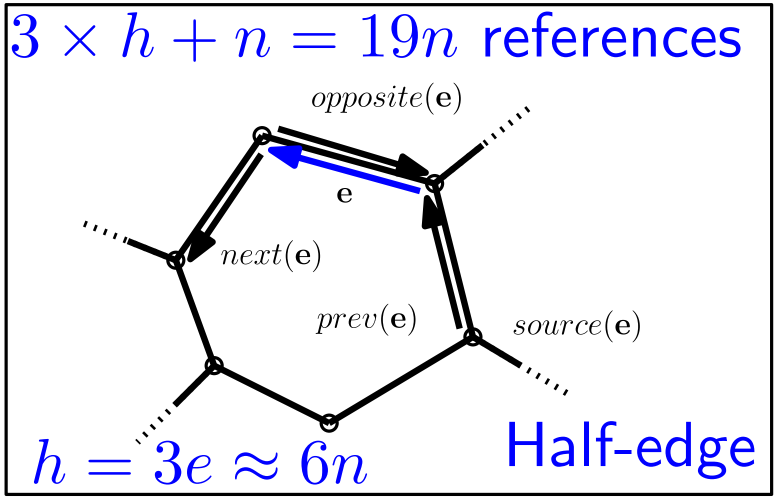


vertex coordinates

between 30 et 96 bits/vertex

"Connectivity": combinatorial information underlying triangulation (incidence relations between triangles, vertices, edges)





David statue (Stanford's Digital Michelangelo Project, 2000)

2 billions polygons

32 Giga bytes (without compression)

$19n \log n$ or $608n$ bits

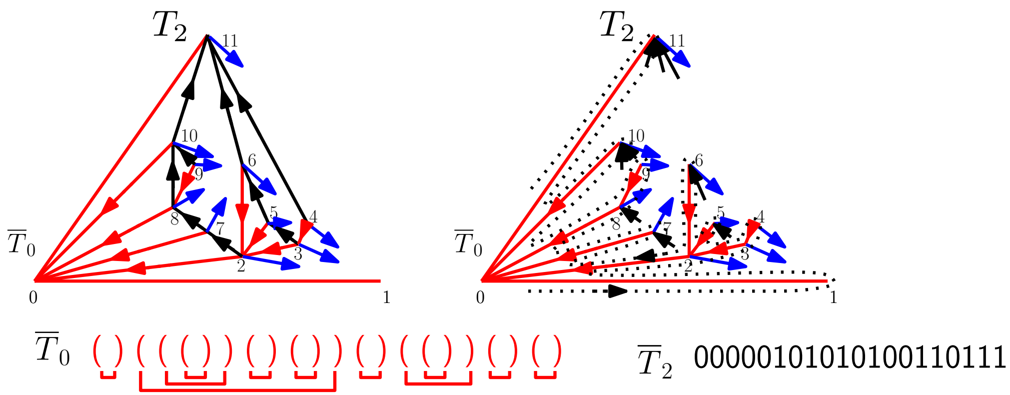
$$\#\{\text{triangulations}\} = \frac{2(4n+1)!}{(3n+2)!(n+1)!} \approx \frac{16}{27}\sqrt{\frac{3}{2\pi}} n^{-5/2} \left(\frac{256}{27}\right)^n$$

$$\Rightarrow \quad \text{entropy} = \log_2 \frac{256}{27} \approx 3.24 \text{ bit/vertex.}$$

Mesh encoding (worst case analysis)

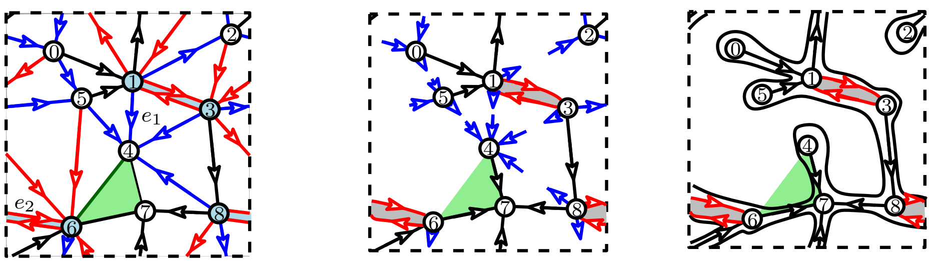
Mesh compression (no support of fast queries)

Canonical orderings - Schnyder woods (He, Kao, Lu '99)



$\overline{T}_2$ 0000010101010100110111

higher genus Schnyder woods (Castelli Aleardi Fusy Lewiner SoCG'08)



(standard) data structures

triangle based DS (CGAL)

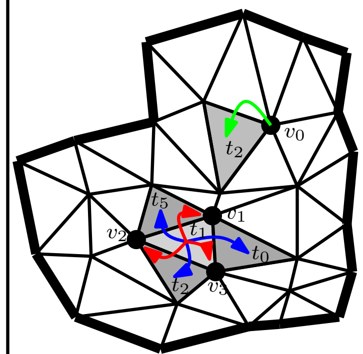
```
class Triangle{
  Triangle t1, t2, t
  Vertex v1, v2, v3;
}
 class Vertex{
  Triangle  root;
  Point p;
}
```

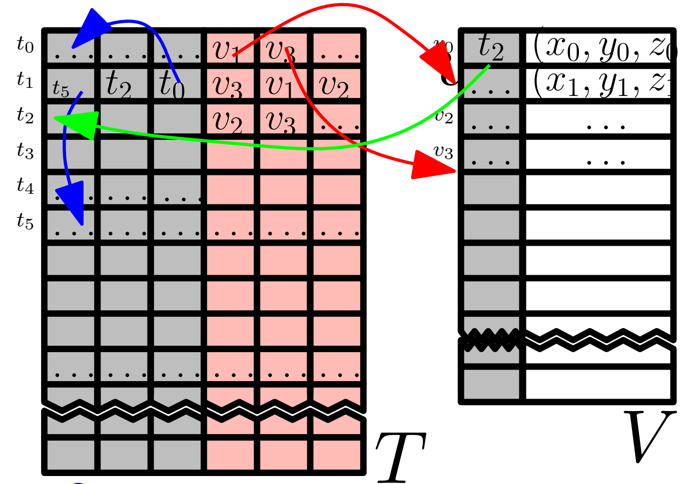
for each triangle, store:

- 3 references to neighboring faces
- 3 references to incident vertices

for each vertex, store:

- 1 reference to an incident face





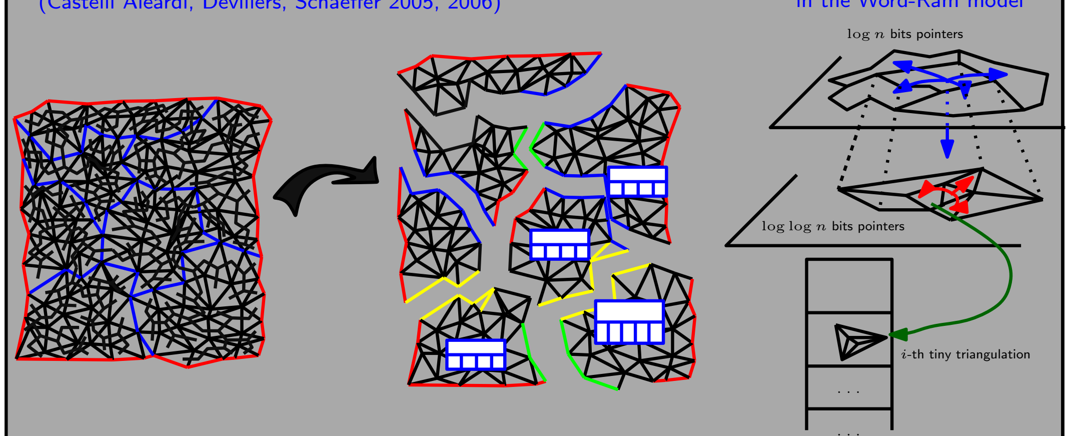
$(3+3) \times f + n$

$6 \times 2n + n = 13n$ references

Succinct representations (supporting queries in $O(1)$ time)

(Castelli Aleardi, Devillers, Schaeffer 2005, 2006)

in the Word-Ram model



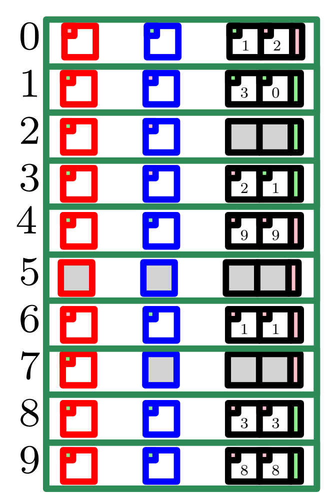
$$3.2451n + O\left(\frac{\log\log n}{\log n}\right) = 3.2451n + o(n) \text{ bits/vertex}$$

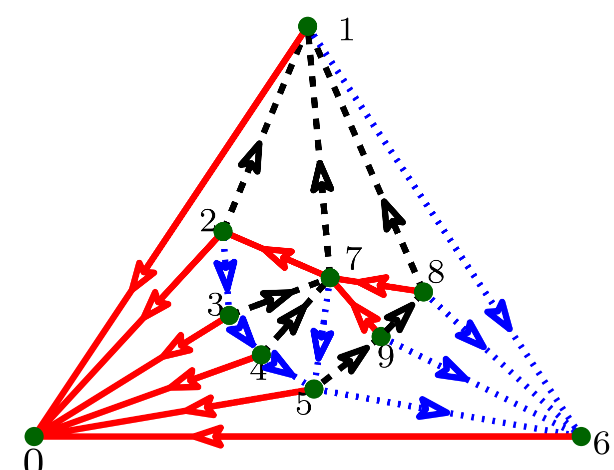
asymptotic optimal bound

Practical compact data structures (fast implementations)

(Castelli Aleardi, Devillers 2011)

1.2 - 1.9 times slower than standard data structures



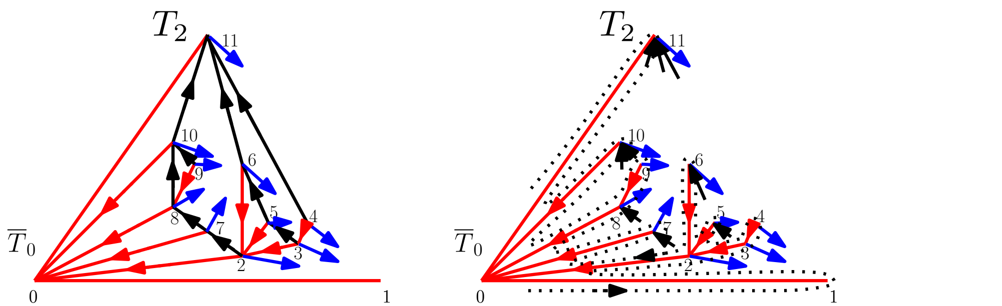


$4n$ references

Mesh encoding (worst case analysis)

Mesh compression (no support of fast queries)

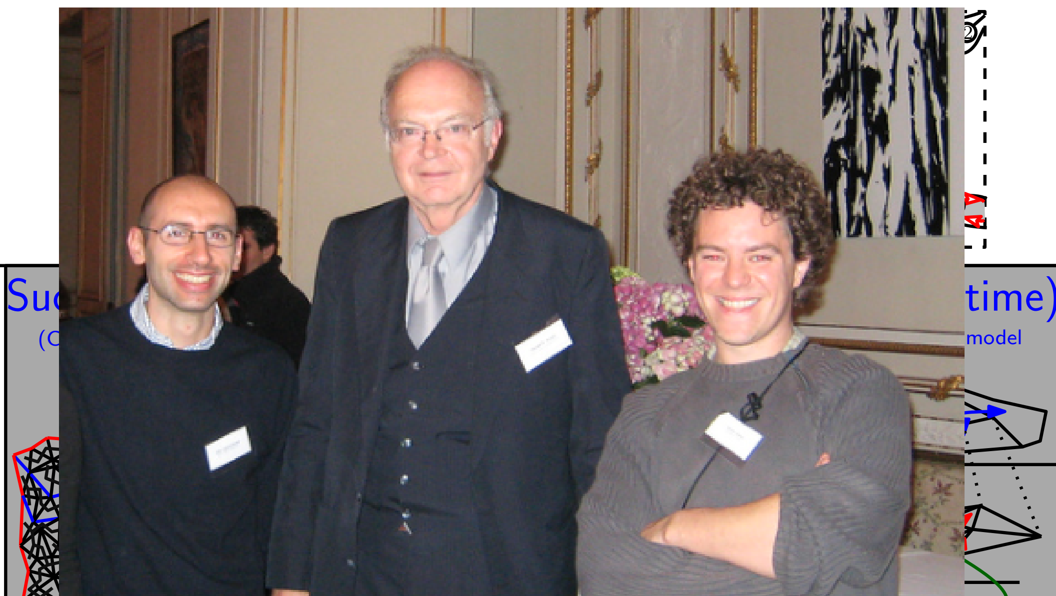
Canonical orderings - Schnyder woods (He, Kao, Lu '99)



$\overline{T}_0$ ( ) ((( )) ( ) ( )) ( ) (( )) ( ) ( )

$\overline{T}_2$ 0000010101010100110111

higher genus Schnyder woods (Castelli Aleardi Fusy Lewiner SoCG'08)



D. Knuth (Bordeaux, dec. 2007)

"Dear Luca and Jeremy, if you want you that your algorithm and data structures will appear in my books, first please provide an implementation and check its performance."

asymptotic optimal bound

(standard) data structures

triangle based DS (CGAL)

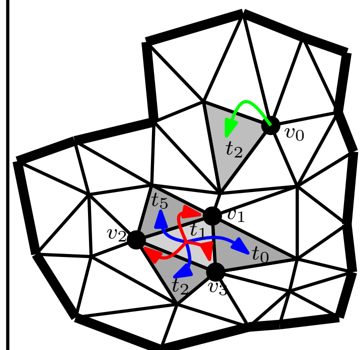
```
class Triangle{
  Triangle t1, t2, t
  Vertex v1, v2, v3;
}
 class Vertex{
  Triangle root;
  Point p;
}
```

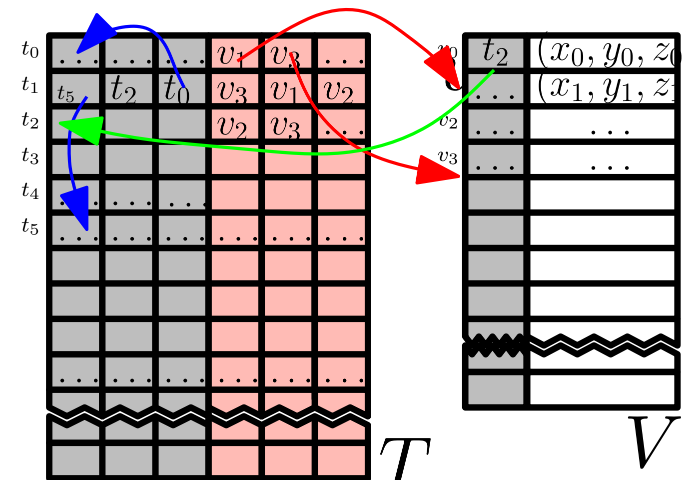
for each triangle, store:

- 3 references to neighboring faces
- 3 references to incident vertices

for each vertex, store:

- 1 reference to an incident face





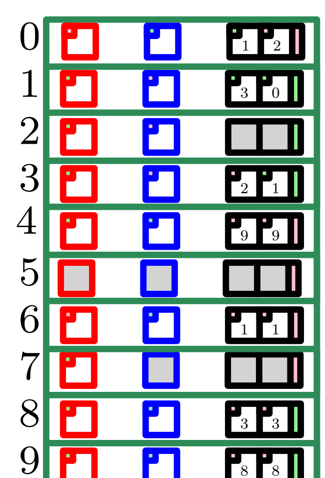
$(3+3) \times f + n$

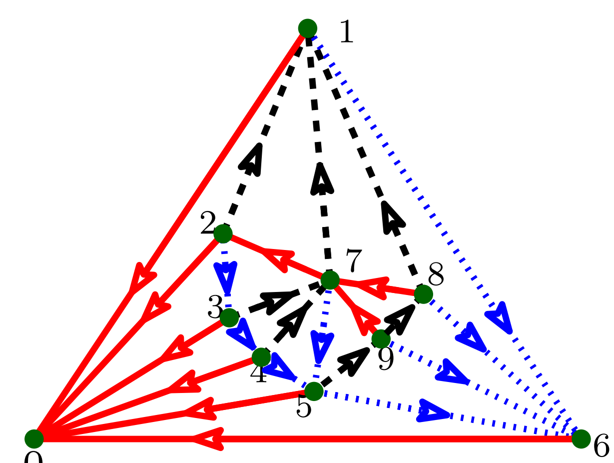
$6 \times 2n + n = 13n$ references

Practical compact data structures (fast implementations)

(Castelli Aleardi, Devillers 2011)

1.2 - 1.9 times slower than standard data structures





$4n$ references

Schnyder woods

(and related structures)

Some (classical) applications

[Felsner, Bonichon et al. '10, ...]

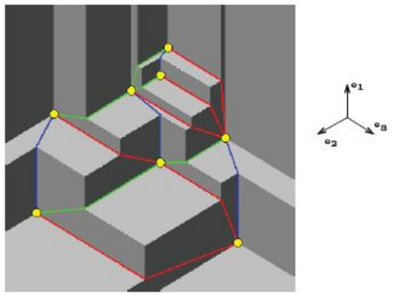


Figure 2: A coplanar orthogonal surface with its geodesic emb

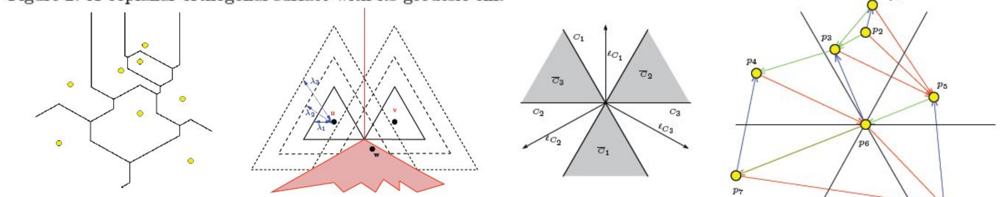
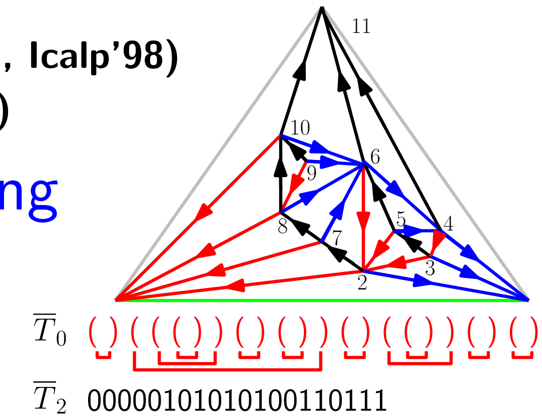


Figure 3: (a) TD-Voronoi diagram. (b) $\lambda_1 < \lambda_2 < \lambda_3$ stand for three triangular distances. Set $\{u, v\}$ is an ambiguous point set, however $\{u, v, w\}$ is non-ambiguous.

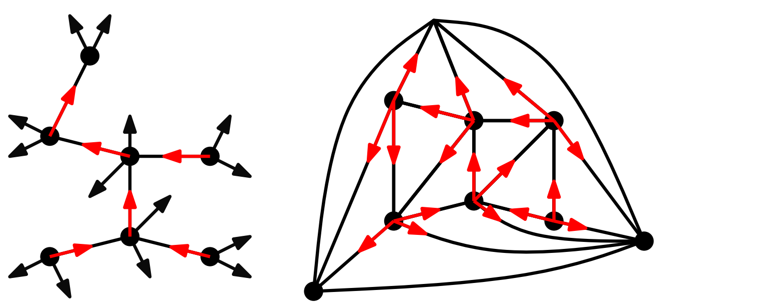
Schnyder woods, TD-Delaunay graphs, orthogonal surfaces and Half-Θ_6-graphs

(Chuang, Garg, He, Kao, Lu, Icalp'98)
(He, Kao, Lu, 1999)

Graph encoding



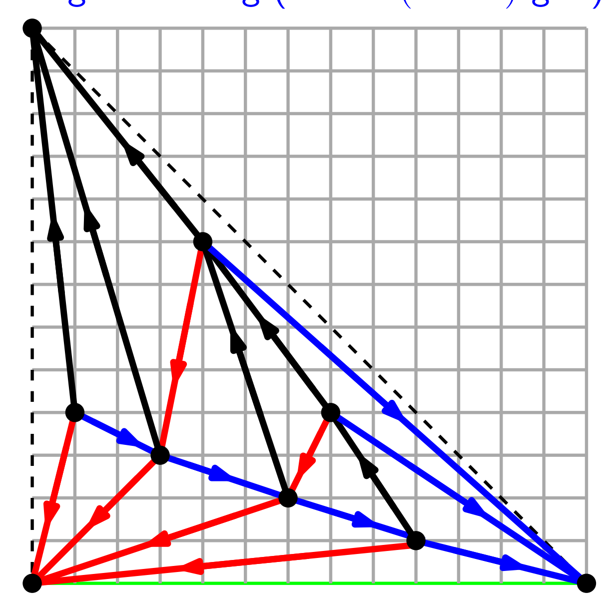
(Poulalhon-Schaeffer, Icalp 03)
bijective counting, random generation



$$c_n = \frac{2(4n+1)!}{(3n+2)!(n+1)!}$$

$\Rightarrow$ optimal encoding ≈ 3.24 bits/vertex

(Schnyder '90)
Planar straight-line grid drawing (on a $O(n \times n)$ grid)

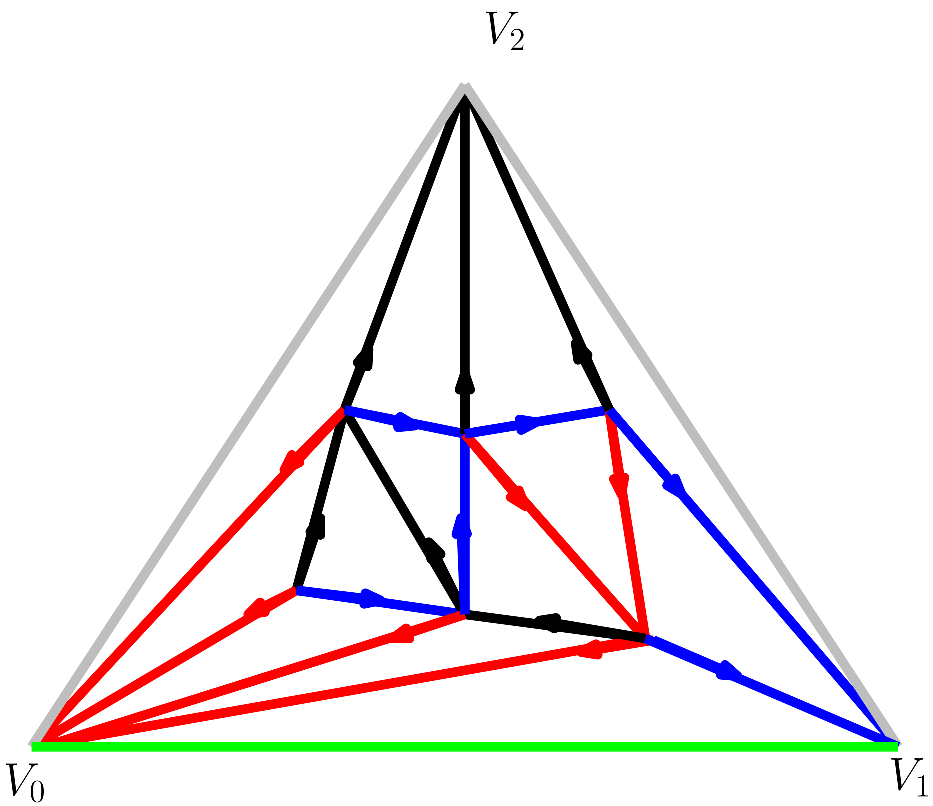


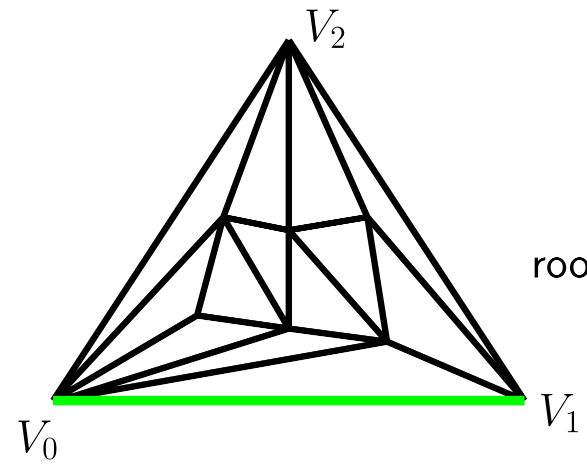
Schnyder woods

(definitions)

Schnyder woods (for plane triangulations): definition

[Schnyder '90]

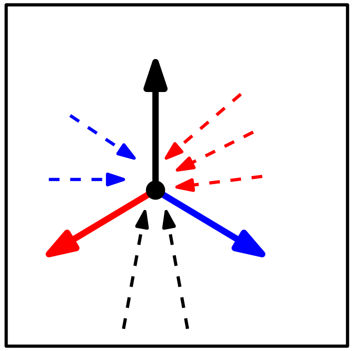




rooted triangulation on n nodes

A Schnyder wood of a (rooted) planar triangulation is partition of all inner edges into three sets T_0, T_1 and T_2 such that

i) edge are colored and oriented in such a way that each inner node has exaclty one outgoing edge of each color

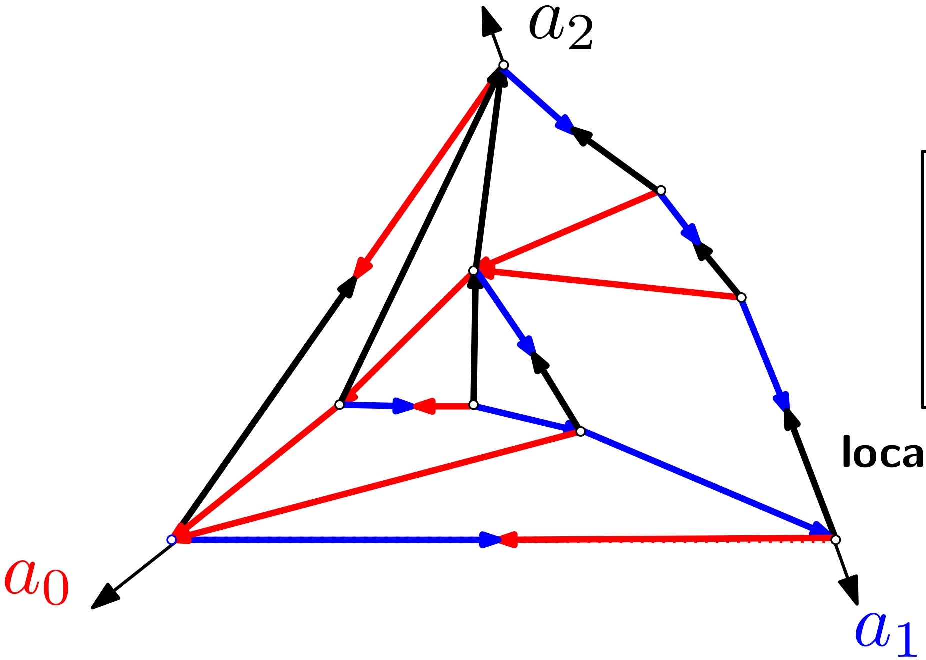


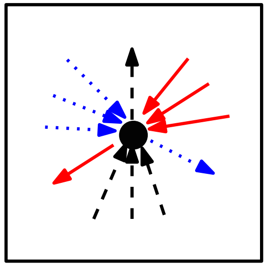
ii) colors and orientations around each inner node must respect the local Schnyder condition

iii) inner edges incident to V_i are of color i and oriented toward V_i

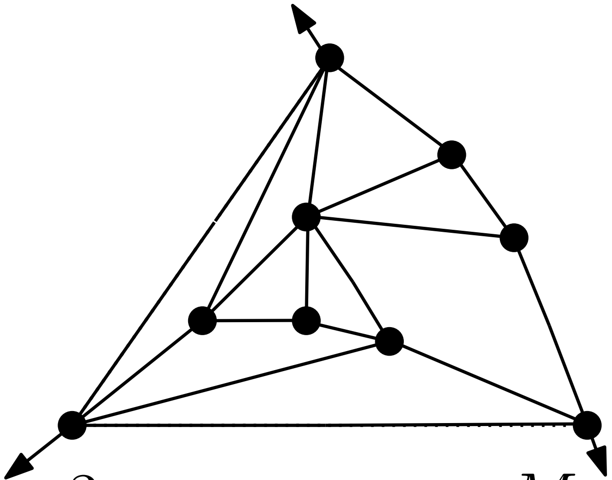
Schnyder woods (3-connected maps): definition

[Di Battista Tamassia Vismara]
[Felsner]





local Schnyder rule



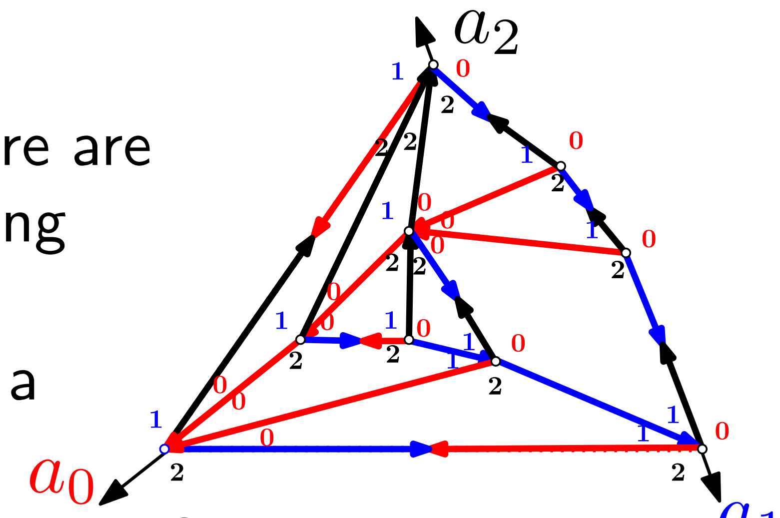
3-connect. map M

W1) edges have one or two (opposite) orientations. If an edge 3 is bo-oriented than the two direction have distinct colors

W2) the edges at a_i are outgoing of color i

W3) **local rule for vertices:** at each vertex there are three outgoing edges (one in each color) satisfying the local Schnyder rule

W4) there is no interior face whose boundary is a directed cycle in one color

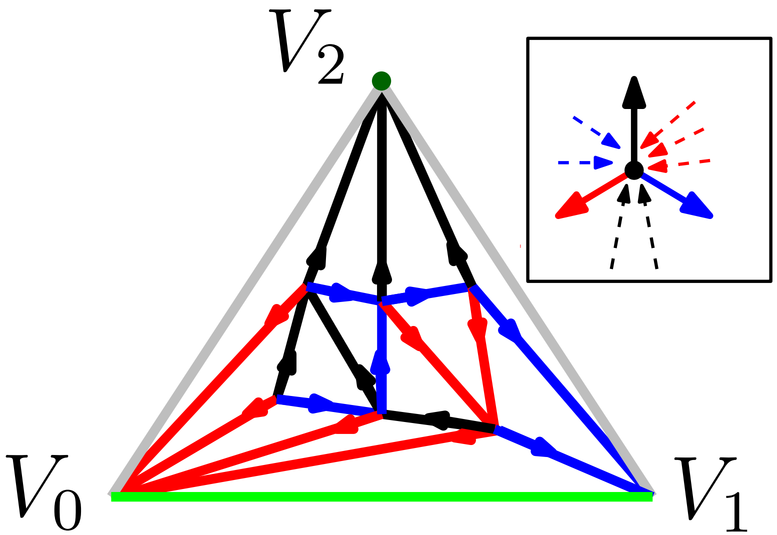


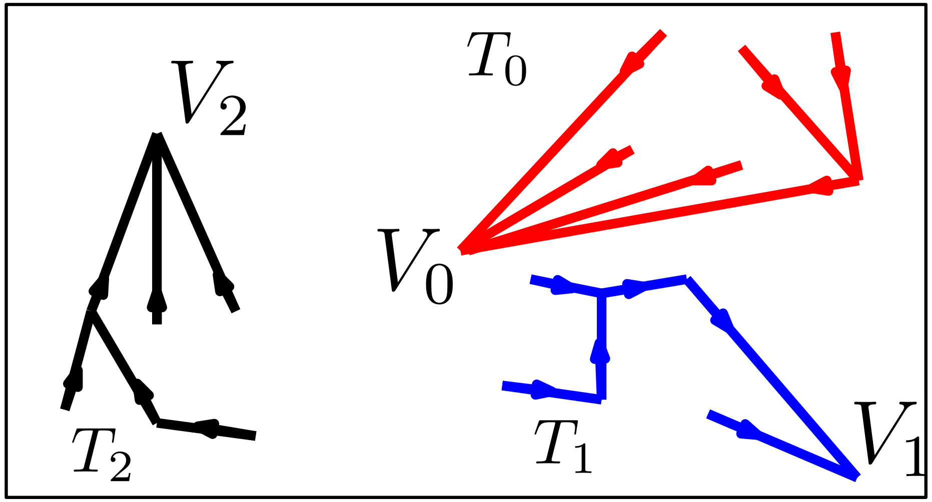
Schnyder labeling

Schnyder woods: global spanning property

Theorem **[Schnyder '90]**

The three sets T_0, T_1, T_2 are spanning trees of the inner vertices of $\mathcal{T}$ (each rooted at vertex V_i)





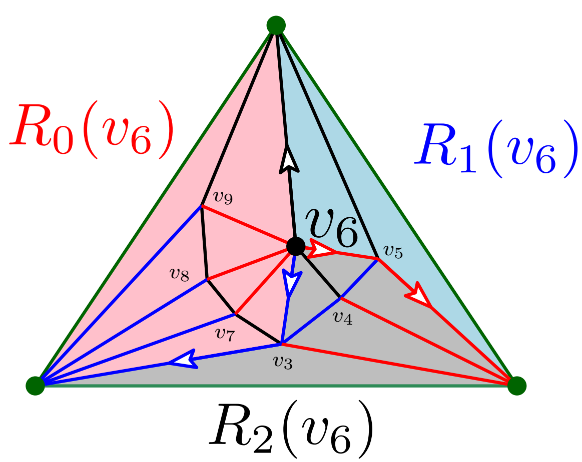
Corollary

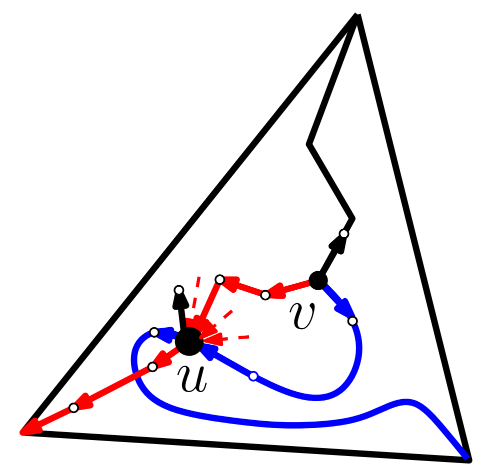
For each inner vertex v the three monochromatic paths P_0, P_1, P_2 directed from v toward each vertex V_i are vertex disjoint (except at v) and partition the inner faces into three sets $R_0(v), R_1(v), R_2(v)$

$$P_0(v_6) = \{(v_6, v_3), (v_3, V_0)\}$$

$$P_1(v_6) = \{(v_6, v_5), (v_5, V_1)\}$$

$$P_2(v_6) = \{(v_6, V_2)\}$$



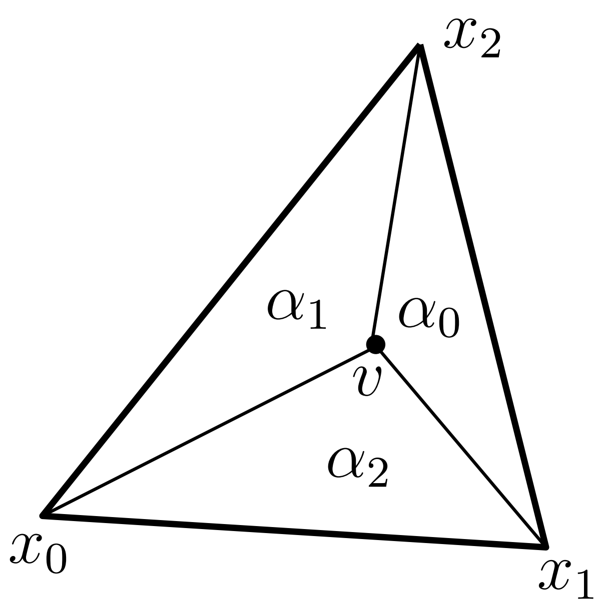


Face counting algorithm

(Schnyder algorithm, 1990)

Face counting algorithm

Geometric interpretation

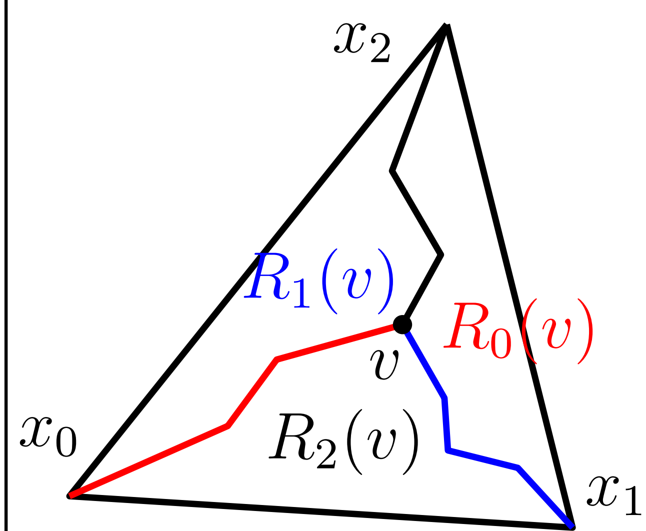


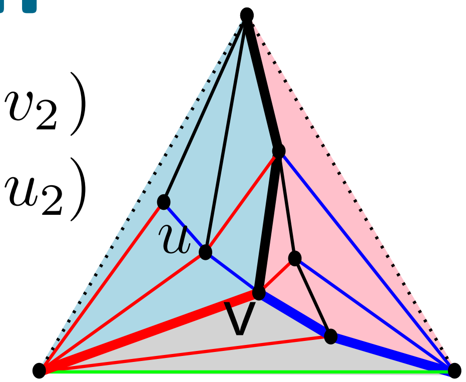
$v = \alpha_0 x_0 + \alpha_1 x_1 + \alpha_2 x_2$

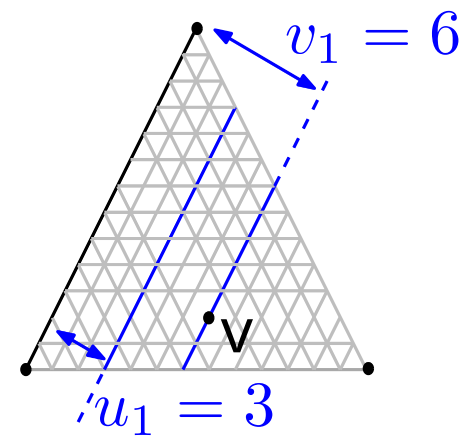
where α_i is the normalized area

v $\rightarrow (5, 6, 2) := (v_0, v_1, v_2)$

u $\rightarrow (7, 3, 3) := (u_0, u_1, u_2)$







$v = \frac{|R_0(v)|}{|F|-1} x_0 + \frac{|R_1(v)|}{|F|-1} x_1 + \frac{|R_2(v)|}{|F|-1} x_2$

$|R_i(v)|$ is the number of triangles in $R_i(v)$

Theorem

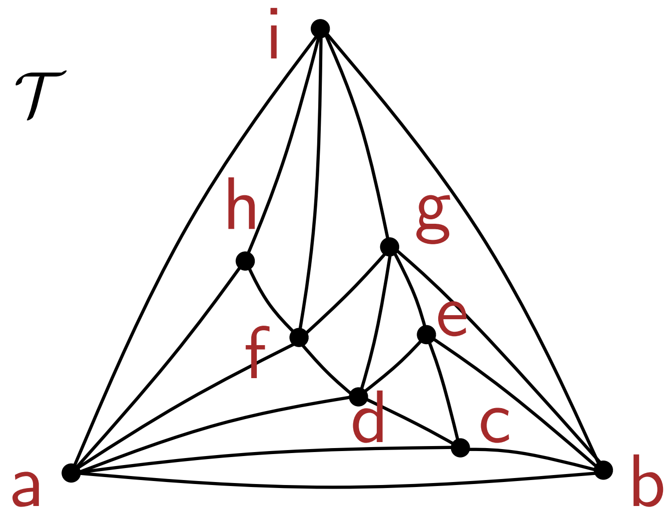
For a 3-connected planar map $\mathcal{M}$ having f vertices, there is drawing on a grid of size $(f-1) \times (f-1)$

Theorem (Schnyder, Soda '90)

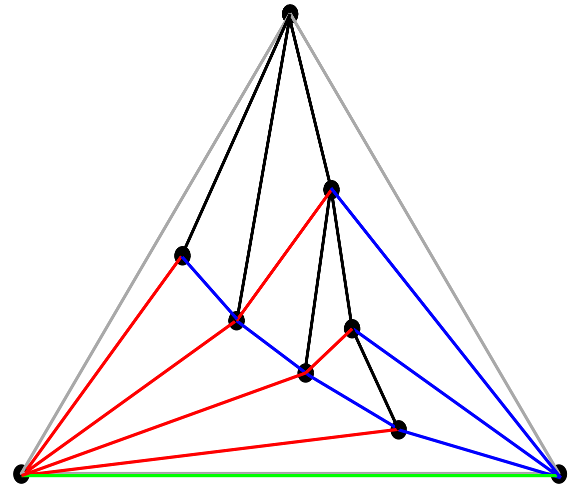
For a triangulation $\mathcal{T}$ having n vertices, we can draw it on a grid of size $(2n-5) \times (2n-5)$, by setting $x_0 = (2n-5, 0)$, $x_1 = (0,0)$ and $x_2 = (0, 2n-5)$.

Face counting algorithm: example

Input: $\mathcal{T}$



$\Rightarrow$



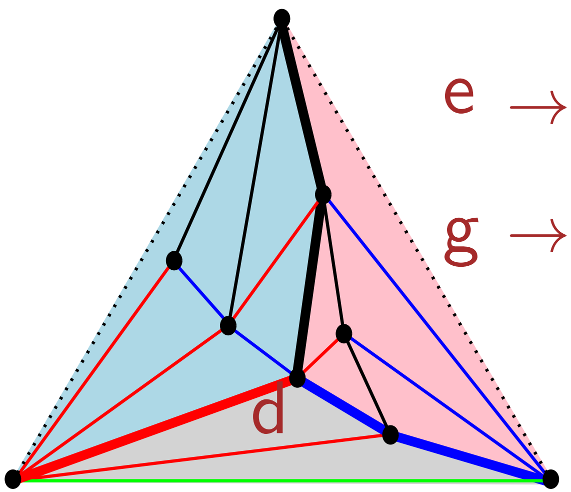
$\mathcal{T}$ endowed with a Schnyder wood

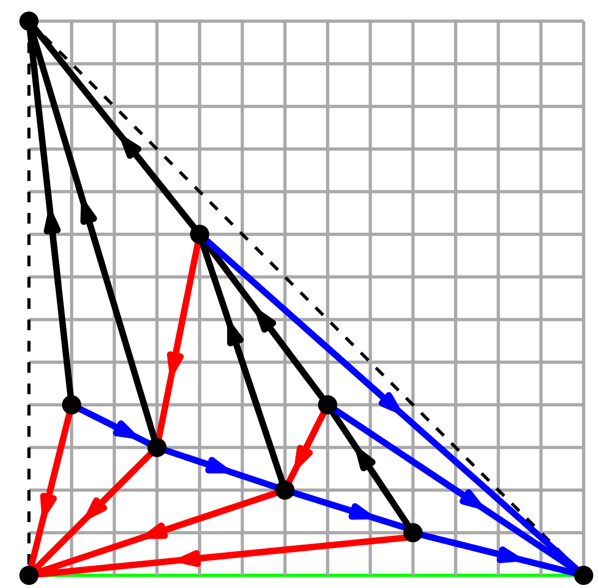
a $\rightarrow (0,0)$ b $\rightarrow (0,1)$ i $\rightarrow (1,0)$

c $\rightarrow (\frac{9}{13}, \frac{1}{13})$ **d** $\rightarrow (\mathbf{\frac{5}{13}}, \mathbf{\frac{6}{13}})$

e $\rightarrow (\frac{7}{13}, \frac{4}{13})$ f $\rightarrow (\frac{3}{13}, \frac{3}{13})$

g $\rightarrow (\frac{4}{13}, \frac{8}{13})$ h $\rightarrow (\frac{1}{13}, \frac{4}{13})$





Canonical orderings

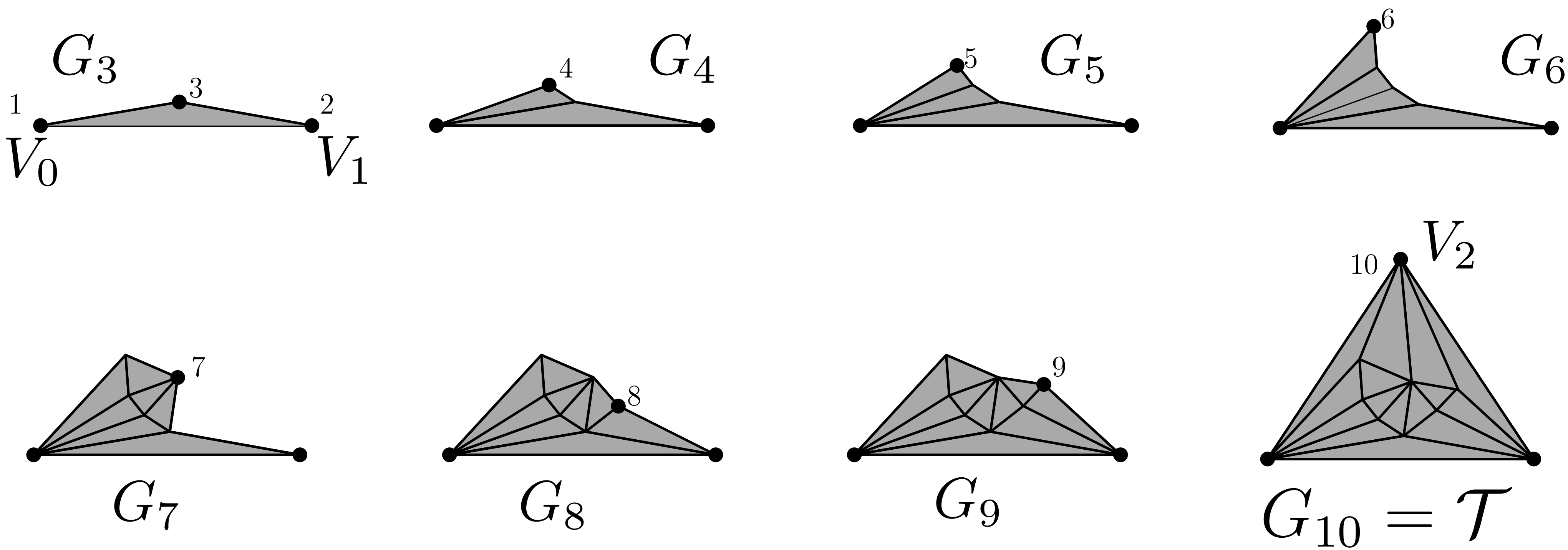
(for planar triangulations)

Canonical orderings: definition

[de Fraysseix Pach Pollack]

Definition 2.6 ([FPP90]) *Let $\mathcal{T}$ be a plane triangulation, whose vertices on the outer (root) face are denoted V_0, V_1, V_2. An ordering $\pi = \{v_1, v_2, \ldots, v_n\}$ of the n vertices of $\mathcal{T}$ is called a* canonical ordering *if the subgraphs G_k ($3 \leqslant k \leqslant n$) induced by the vertices $v_1, \ldots, v_k$ satisfy the following conditions (where we denote by B_k the cycle bounding the outer face of G_k):*

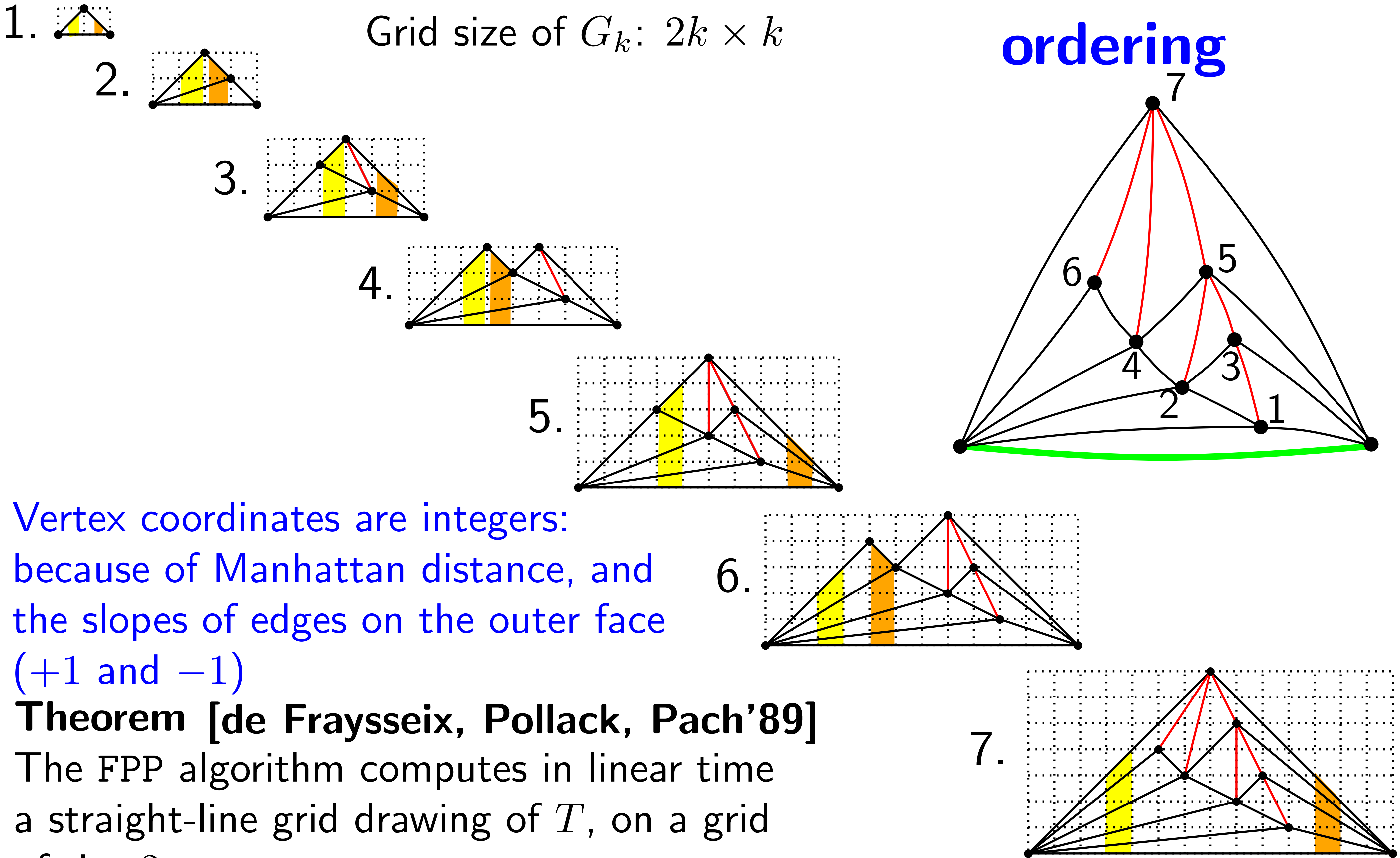
- *G_k is 2-connected and internally triangulated, and $G_n = \mathcal{T}$;*
- *v_1 and v_2 belong to the outer face (V_0, V_1, V_2);*
- *for each $k \geqslant 3$ the vertex v_k is on the B_k and its neighbors in G_{k-1} are consecutive on B_{k-1}.*



incremental shift algorithm (original FPP)

use the canonical ordering

Grid size of G_k: $2k \times k$



Vertex coordinates are integers: because of Manhattan distance, and the slopes of edges on the outer face ($+1$ and -1)

Theorem [de Fraysseix, Pollack, Pach'89]

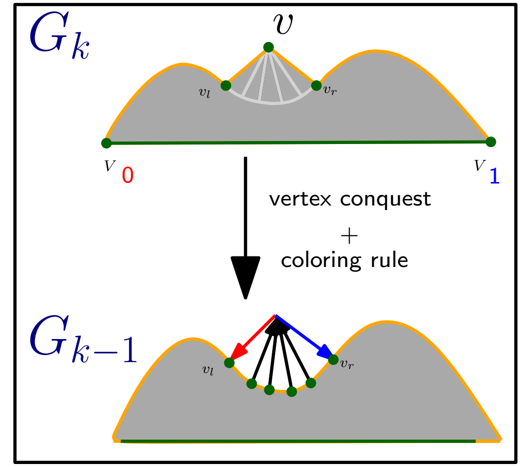
The FPP algorithm computes in linear time a straight-line grid drawing of T, on a grid of size $2n \times n$

Schnyder woods (and canonical orderings): existence

Theorem [Brehm '00]

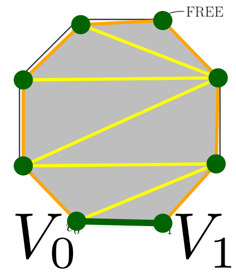
Every planar triangulation admits a Schnyder wood (and a canonical ordering), which can be computed in linear time, via vertex shellings.

Start the traversal from the root face (V_0, V_1, V_2) and remove **free** vertices (without chordal edges)

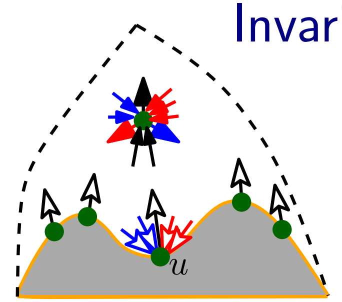


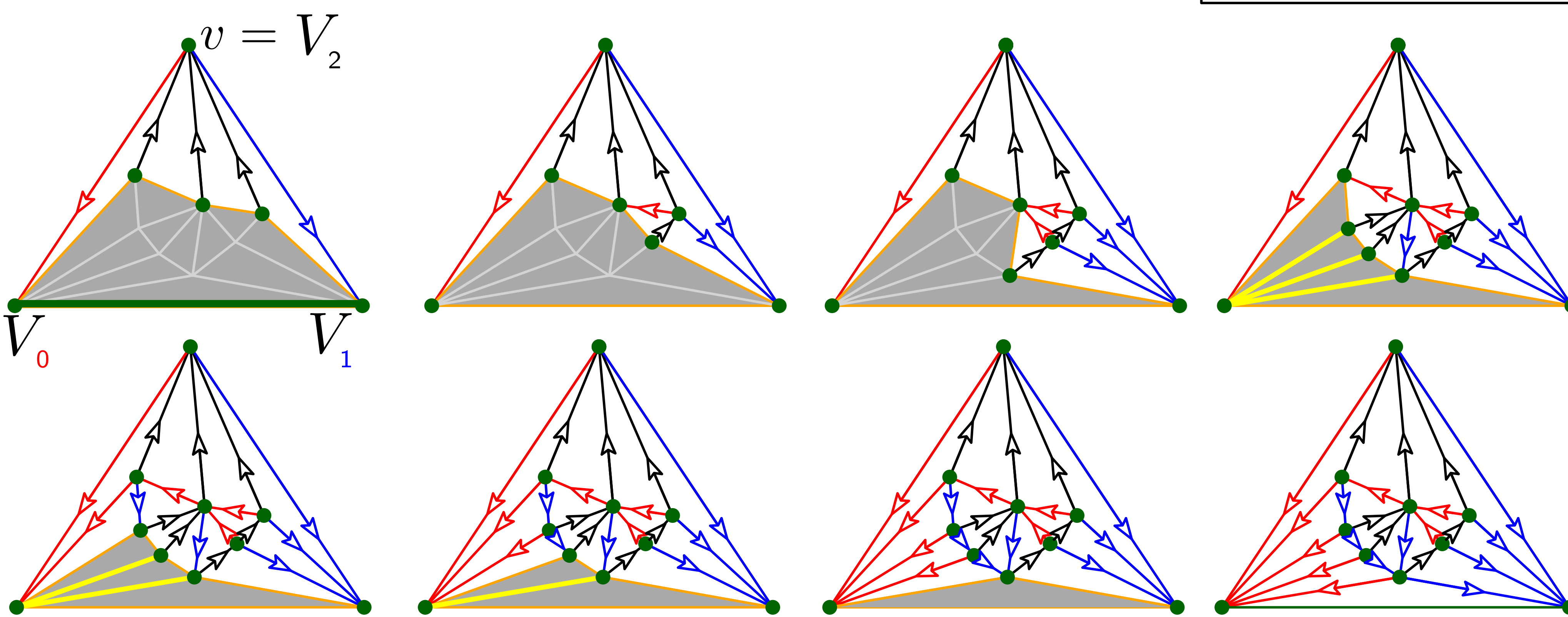
Correctness

There must be a free vertex v (not v_0 nor v_1) without chords



Invariant

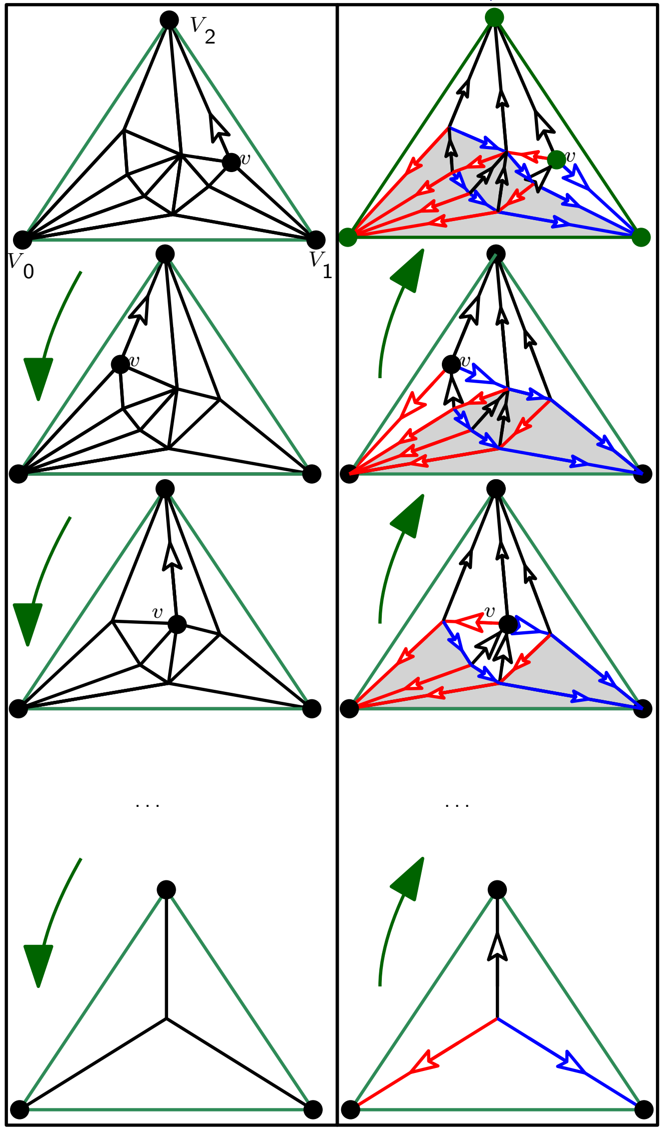




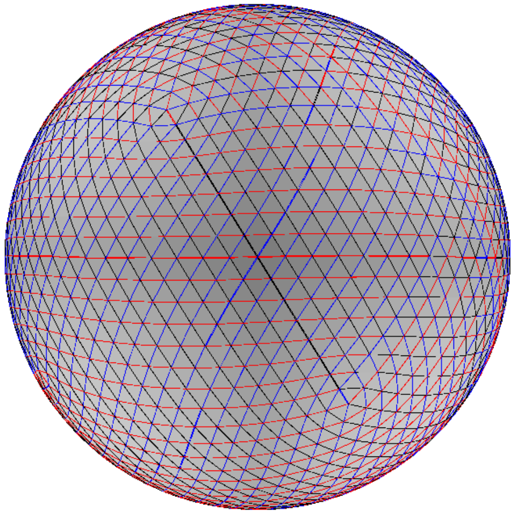
Schnyder woods (and canonical orderings): existence

Theorem [Brehm '00]
Every planar triangulation admits a Schnyder wood (and a canonical ordering), which can be computed in linear time, via vertex shellings.

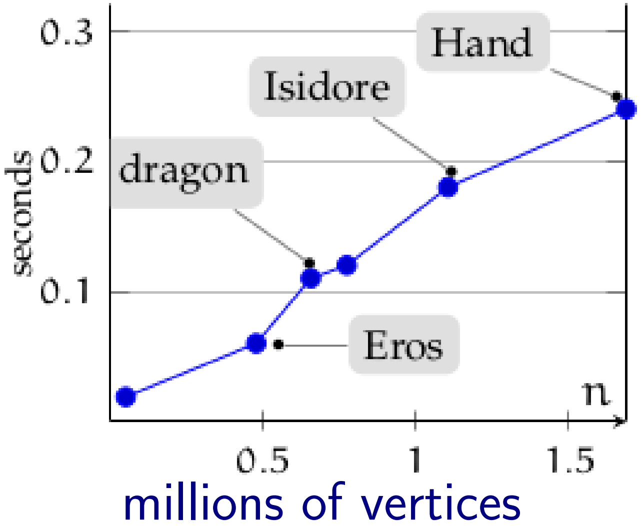
alternative way
via edge contractions/expansions



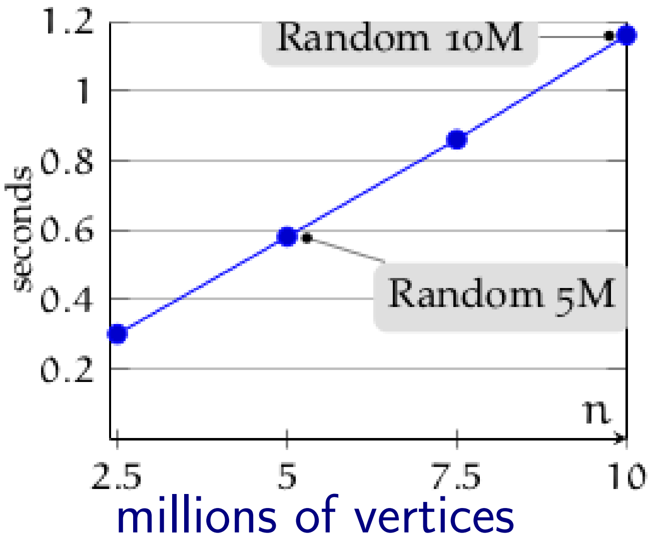
(very fast implementation)



(real-world graphs)



(random triangulations)

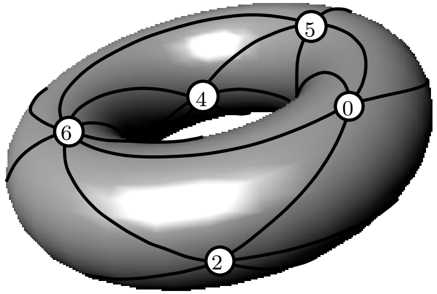


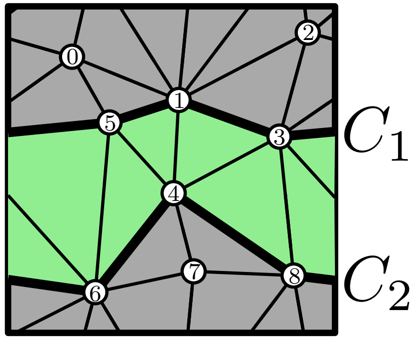
Schnyder woods and higher genus surfaces

(several possible generalizations)

(pioneeristic) toroidal tree decomposition

[Bonichon Gavoille Labourel, 2005]

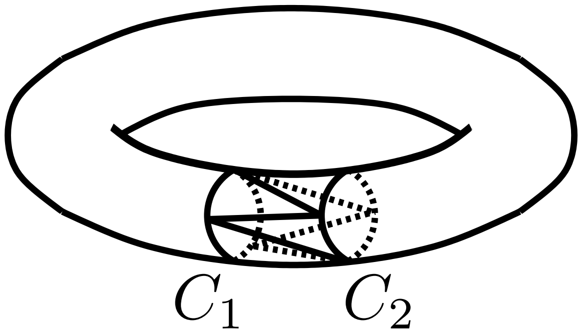




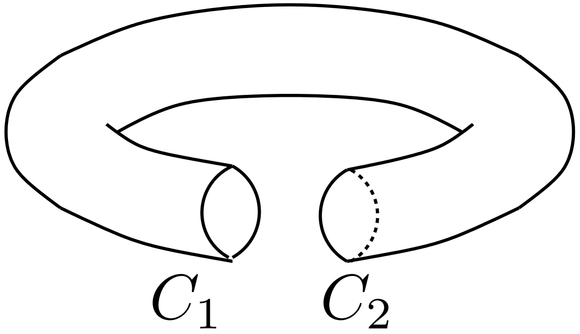
The **tambourine** solution

Compute a pair of adjacent non contractible cycles

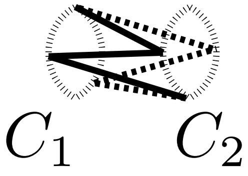
Graph G



Graph H



Tambourine T

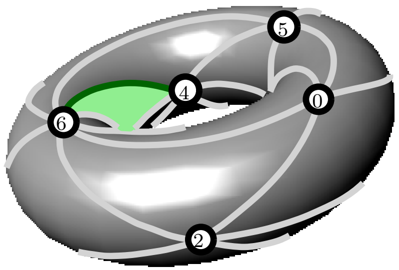


Inconvenients:

- valid only for toroidal triangulations (genus 1)
- potentially large (non constant) number of vertices on C_1 and C_2 not satisfying the local condition
- shortest non contractible cycles are not trivial to compute

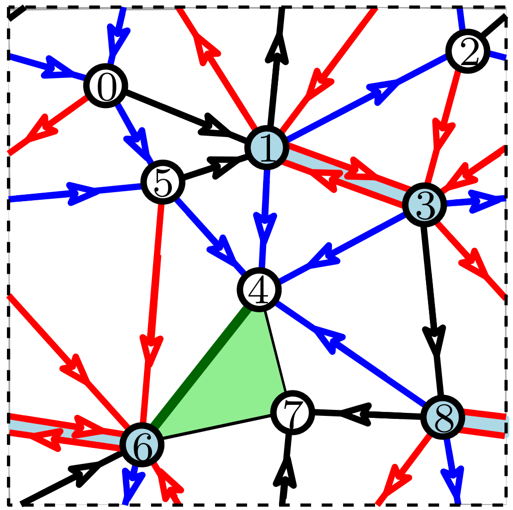
Definition I: genus g Schnyder woods

[Castelli-Aleardi Fusy Lewiner, SoCG'08]



$E^s = \{(1,3),(6,8)\}$

$V_0 = v_6, V_1 = v_4, V_2 = v_7$



Def: partition of all "inner" edges into four sets

T_0, T_1, T_2 and E^s

such that

almost all (non inner) vertices have outgoing degree 3

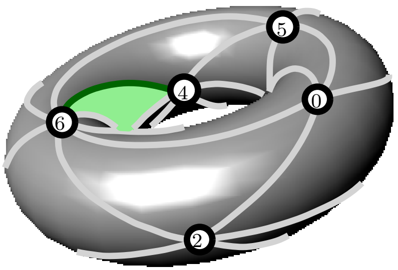
all edges in T_0, T_1 and T_2 have one color/orientation

at most $4g$ special vertices (outdegree > 3)

the set E^s contains at most $2g$ edges (multiple edges)

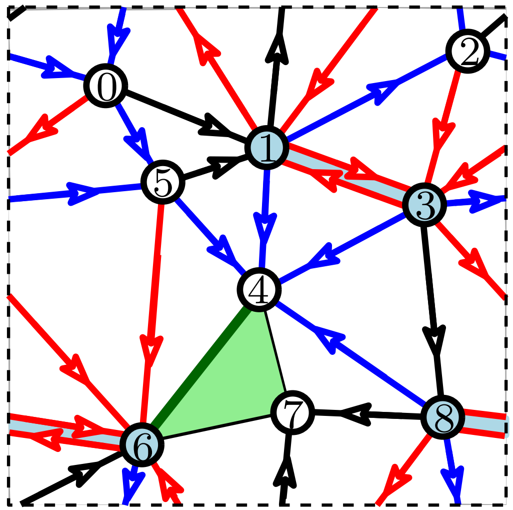
Definition I: genus g Schnyder woods

[Castelli-Aleardi Fusy Lewiner, SoCG'08]

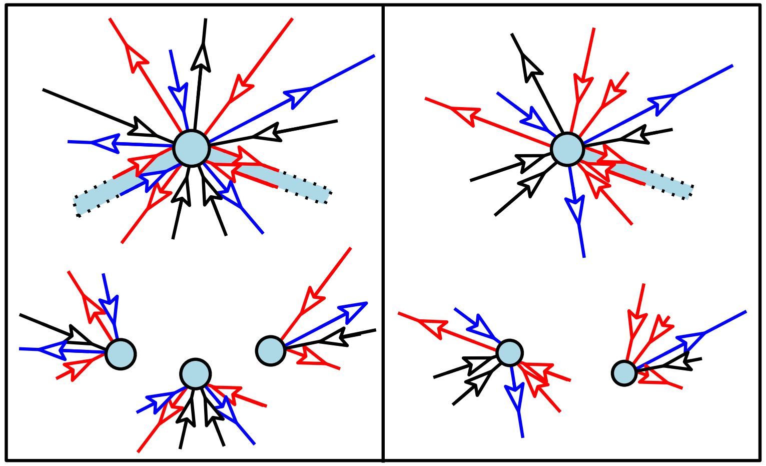


$E^s = \{(1,3),(6,8)\}$

$V_0 = v_6, V_1 = v_4, V_2 = v_7$



local condition for special vertices



Def: partition of all "inner" edges into four sets

T_0, T_1, T_2 and E^s

such that

almost all (non inner) vertices have outgoing degree 3

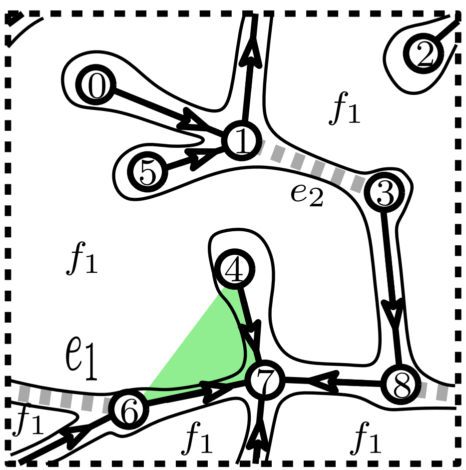
all edges in T_0, T_1 and T_2 have one color/orientation

at most $4g$ special vertices (outdegree > 3)

the set E^s contains at most $2g$ edges (multiple edges)

new local conditions around special vertices

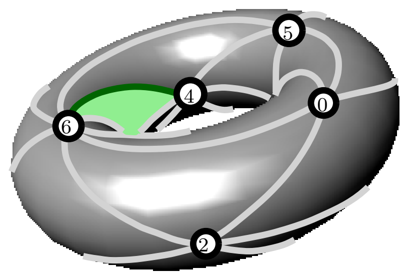
The graph $G_2 = \overline{T}_2 \cup \{e_1, e_2\}$ is a cut-graph

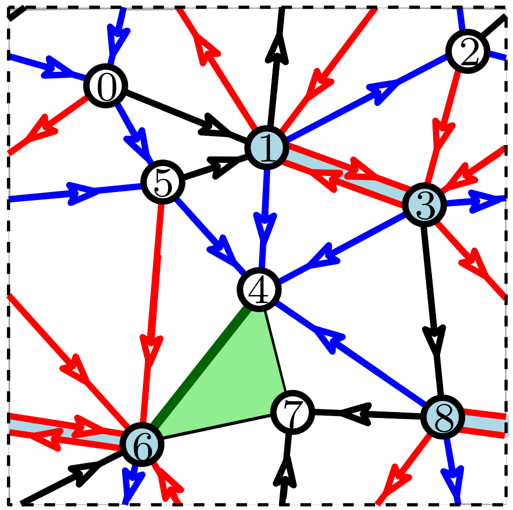


$G_2 = \overline{T}_2 \cup \{e_1, e_2\}$

Genus g Schnyder woods: spanning property

[Castelli-Aleardi Fusy Lewiner, SoCG'08]





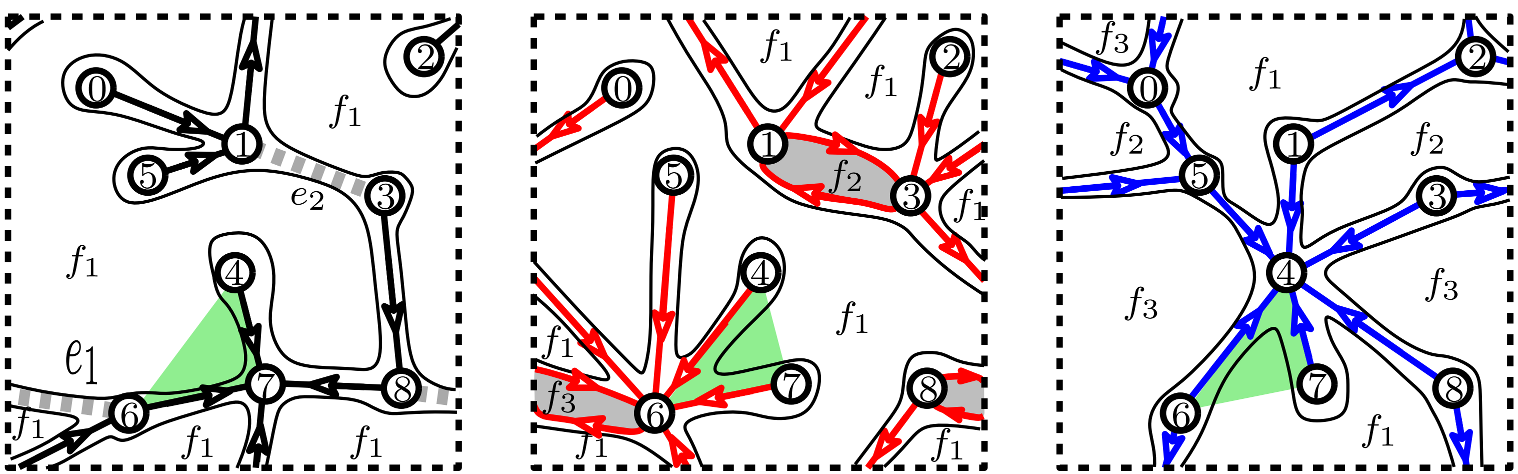
$E^s = \{(1,3),(6,8)\}$

$V_0 = v_6, V_1 = v_4, V_2 = v_7$

Theorem

The set of (possibly multiple) edges of color 0, 1 and 2 lead to maps of genus g satisfying:

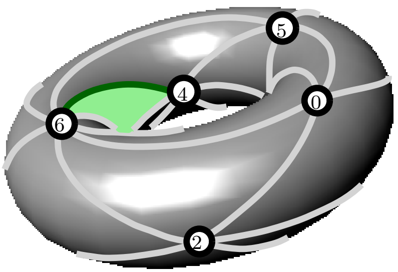
- G_0, G_1 are cellularly spanning subgraphs with $1+2g$ faces (possibly degenerated);
- G_2 is a 1 face map (a g-tree)



$G_2 = \overline{T}_2 \cup \{e_1, e_2\}$ G_0 G_1

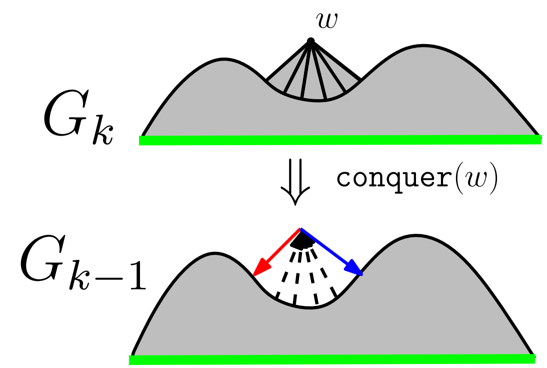
Genus g Schnyder woods: existence

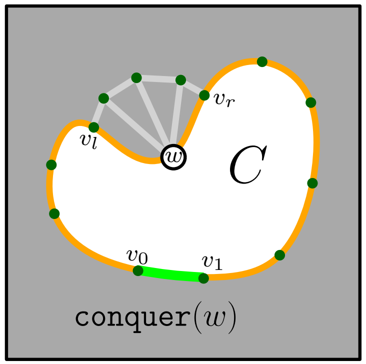
[Castelli-Aleardi Fusy Lewiner, SoCG'08]

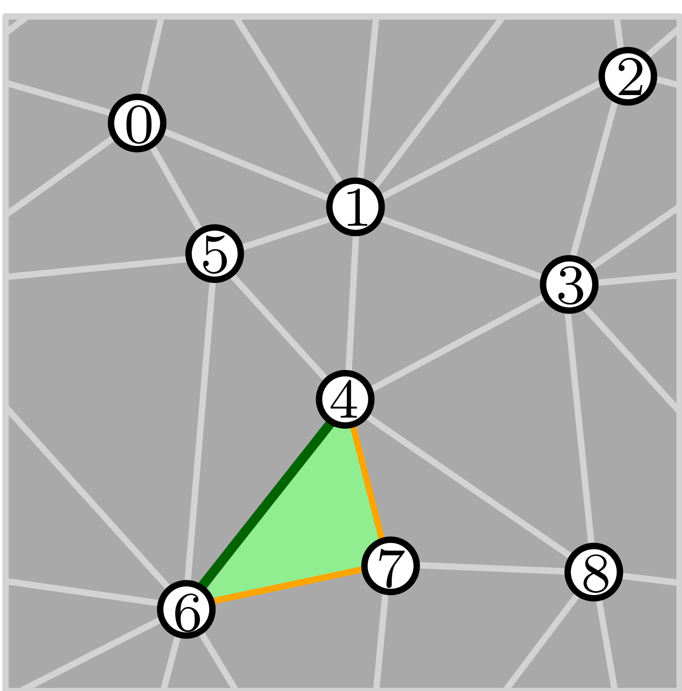


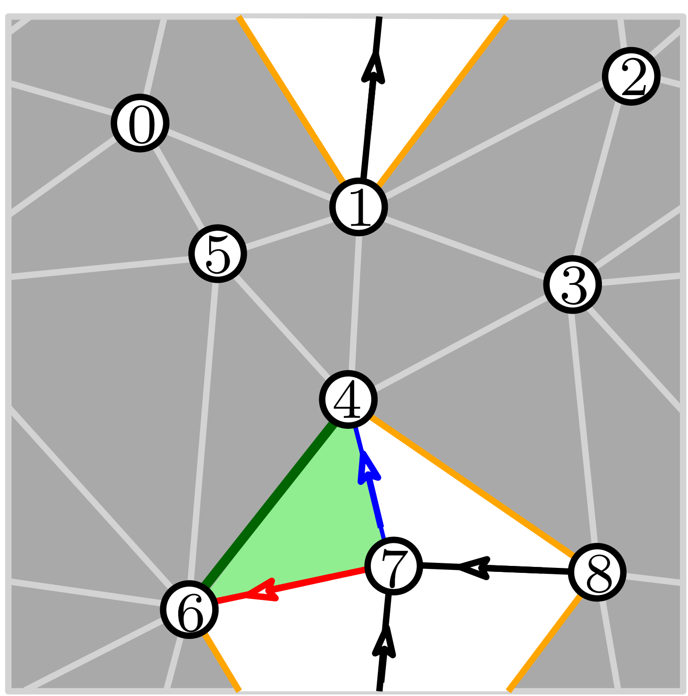
Incremental vertex shelling algorithm

Perform a vertex conquest (as far as you can) until you get stuck





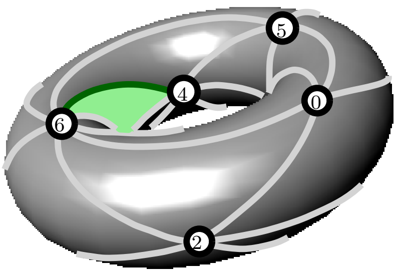




conquer(7)

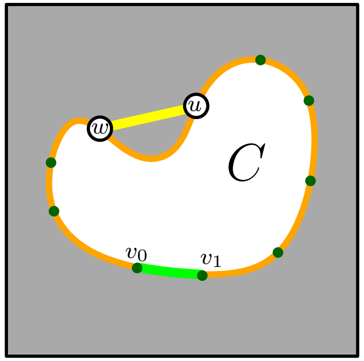
Genus g Schnyder woods: existence

[Castelli-Aleardi Fusy Lewiner, SoCG'08]



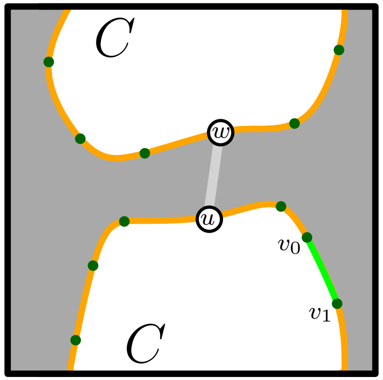
Incremental vertex shelling algorithm
Perform a vertex conquest (as far as you can)
until you get stuck

No more free vertices

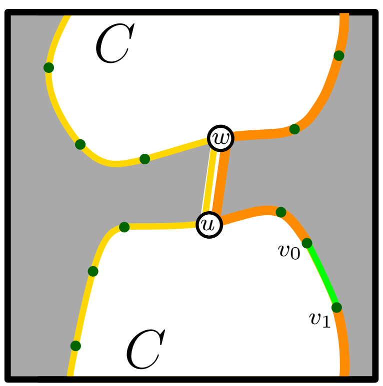


all boundary vertices are
incident to chordal edges

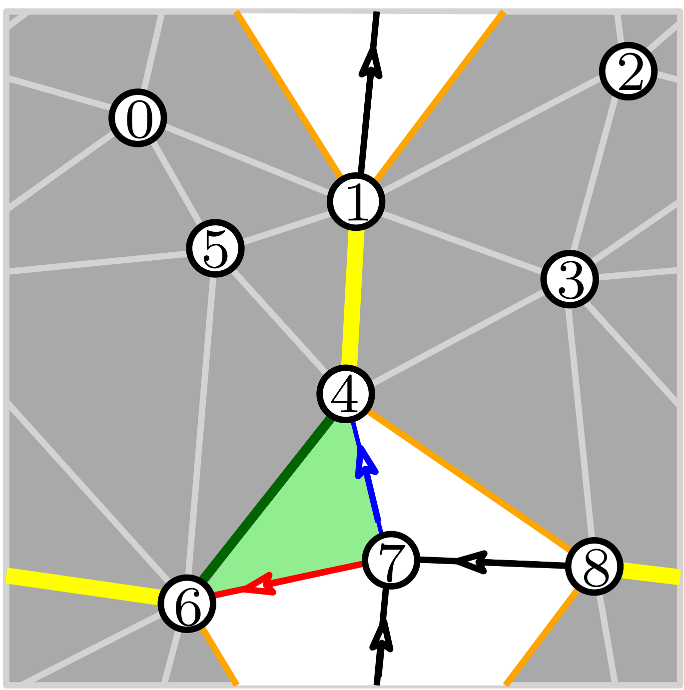
one boundary



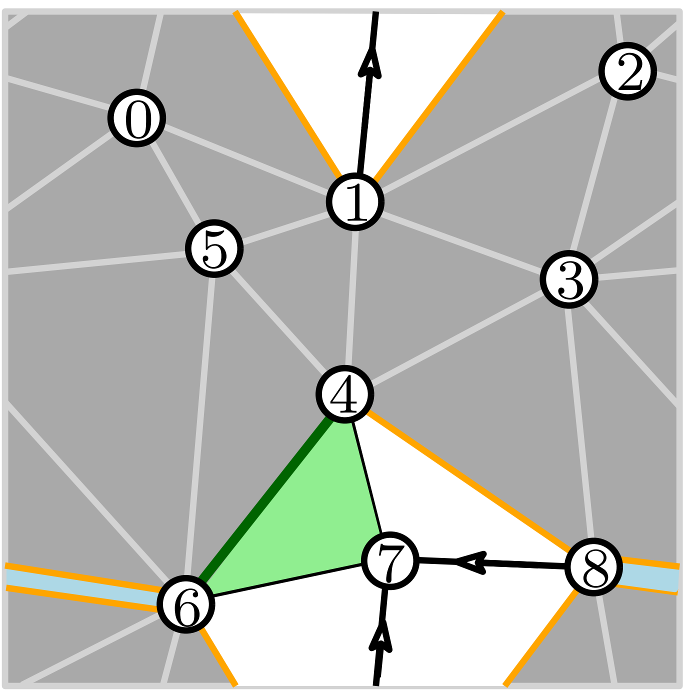
$\Downarrow$ $\texttt{split}(u, w)$



two boundaries



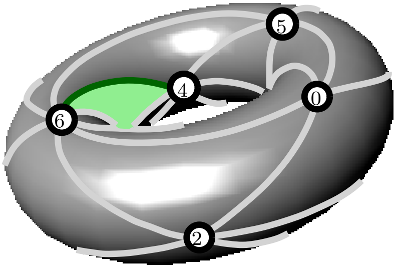
C is a topological disk



$\texttt{split}(6, 8)$

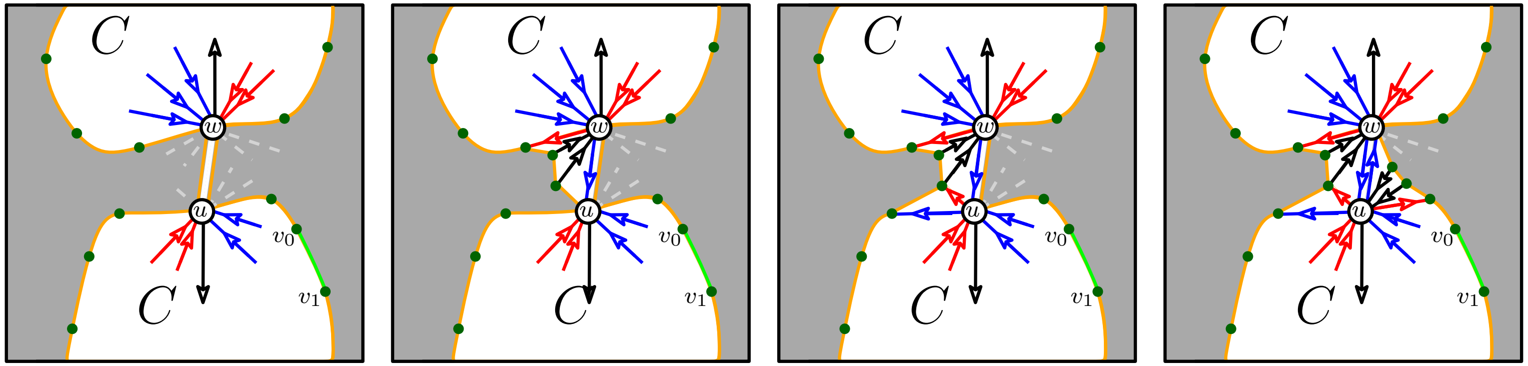
Genus g Schnyder woods: existence

[Castelli-Aleardi Fusy Lewiner, SoCG'08]



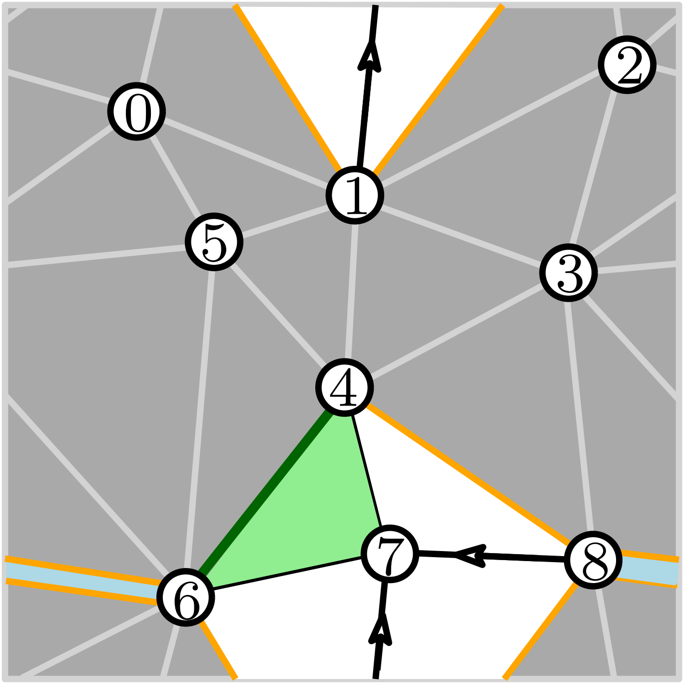
Incremental vertex shelling algorithm

Perform a vertex conquest (as far as you can)
until you get stuck

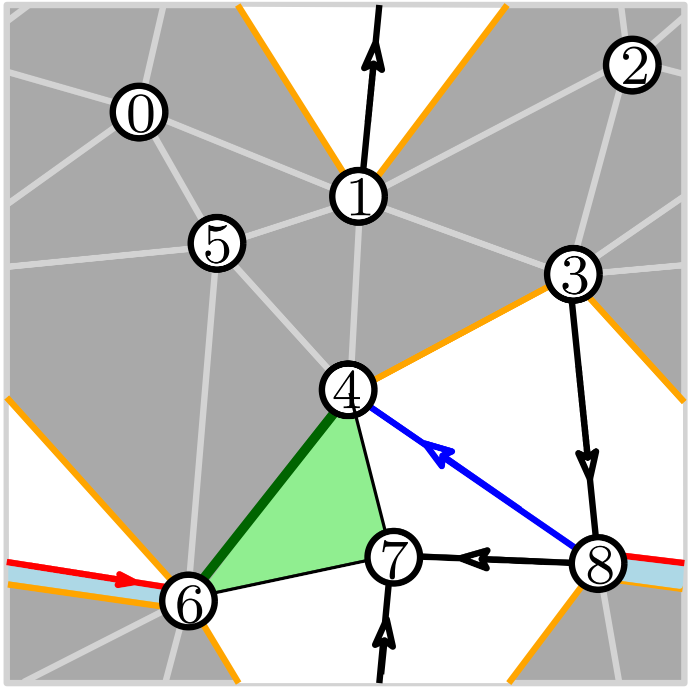


conquer(b_w) + colorient   conquer(b_u) + colorient   conquer(b_u) + colorient

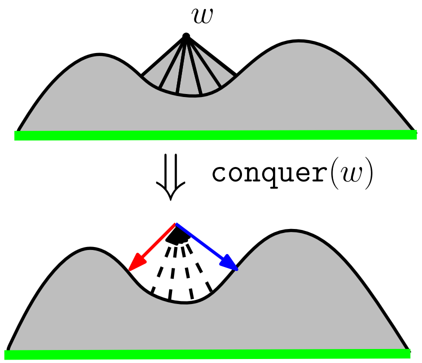
There is one free vertex



split(6, 8)

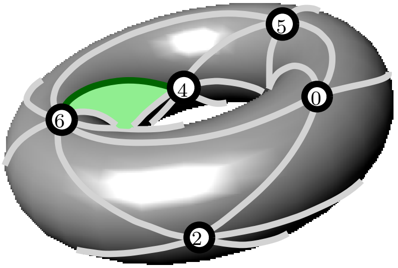


conquer(b_8)



Genus g Schnyder woods: existence

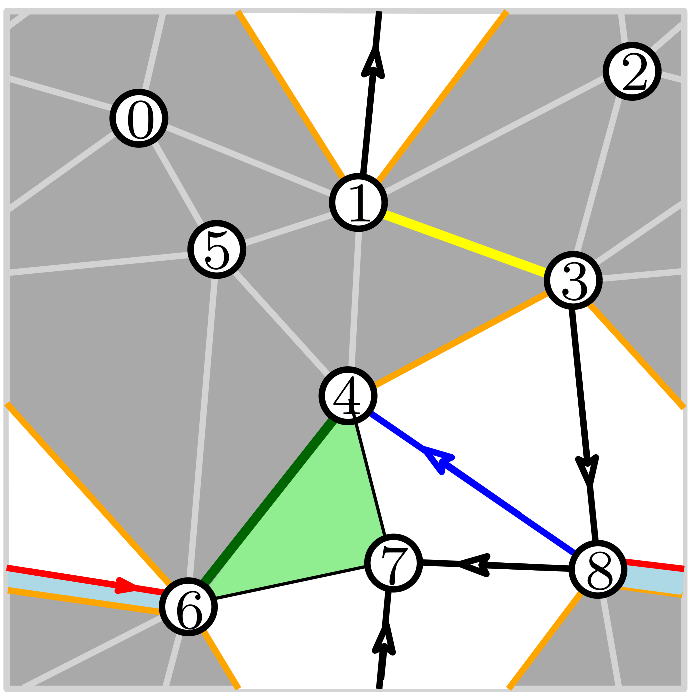
[Castelli-Aleardi Fusy Lewiner, SoCG'08]



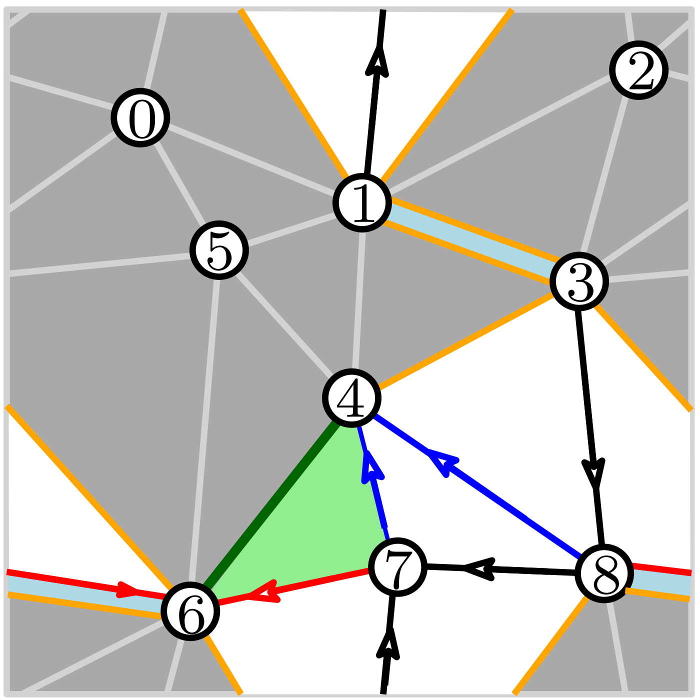
Incremental vertex shelling algorithm

Perform a vertex conquest (as far as you can) until you get stuck

Choose a merge chordal edge (if any)

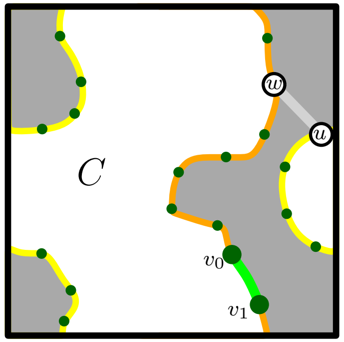


The complement of C is a topological disk

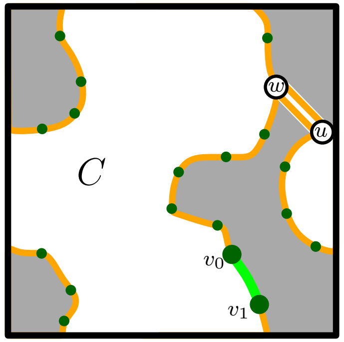


merge$(1, 3)$

two boundaries



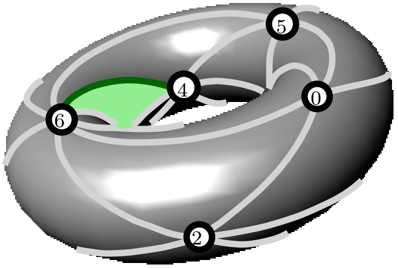
$\Downarrow$ merge(u, w)



one boundary

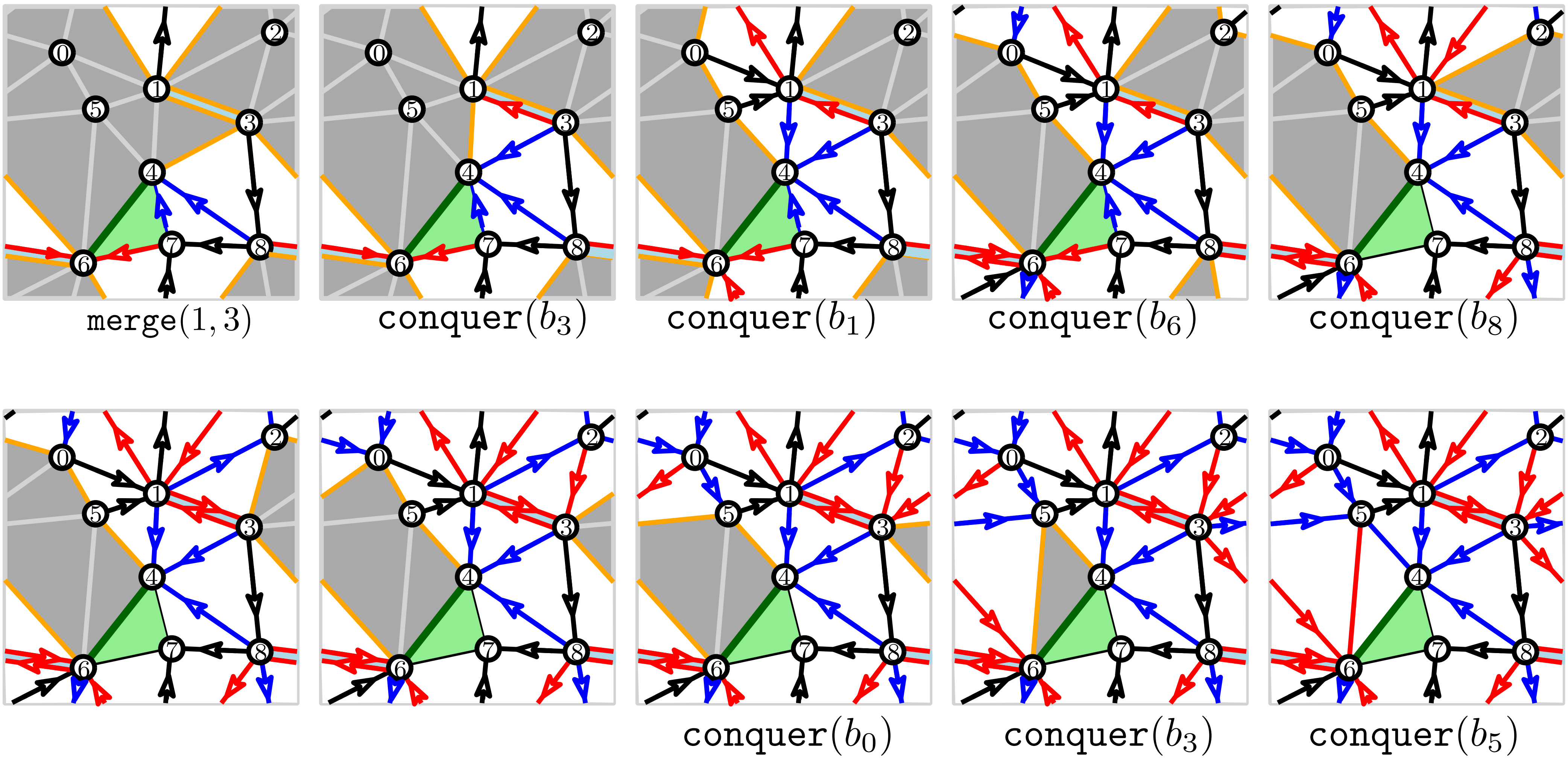
Genus g Schnyder woods: existence

[Castelli-Aleardi Fusy Lewiner, SoCG'08]



Incremental vertex shelling algorithm

The complement of C is a topological disk: just perform vertex conquests (only one boundary)



Schnyder woods for toroidal graphs

Toroidal Schnyder woods: definition

$g = 1 \qquad e = 3n$

$n - e + f = 2 - 2g$

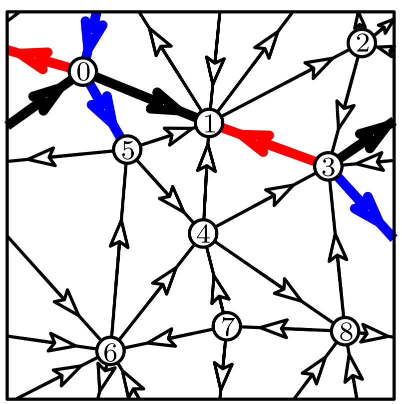
Toroidal Schnyder woods [Goncalves Lévêque, DCG'14]

- 3-orientation + Schnyder local rule valid at each vertex

Toroidal Schnyder woods are **crossing** if

- every monochromatic cycle intersects at least one monochromatic cycle of each color

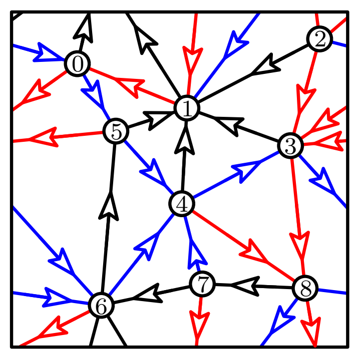
not valid Schnyder wood



3-orientation

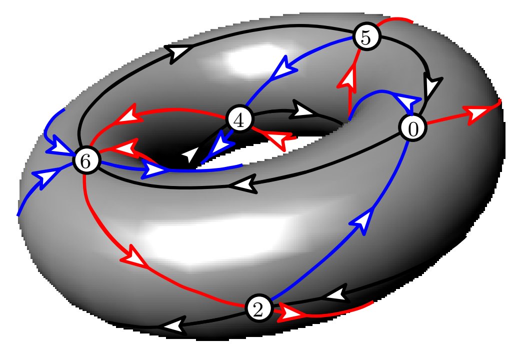
(Local Schnyder rule cannot be propagated everywhere)

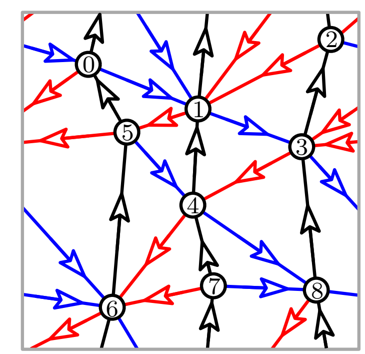
valid Schnyder woods



the Schnyder wood is **not crossing**

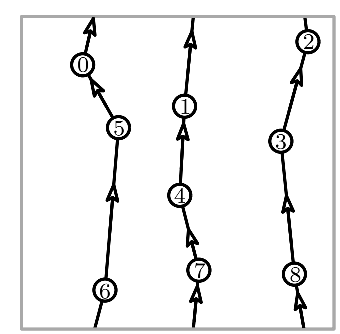
(one mono-chromatic cycle for each color)





crossing Schnyder wood

(there are 3 disjoint mono-chromatic cycles of color 2)

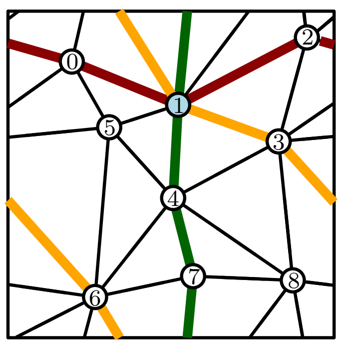


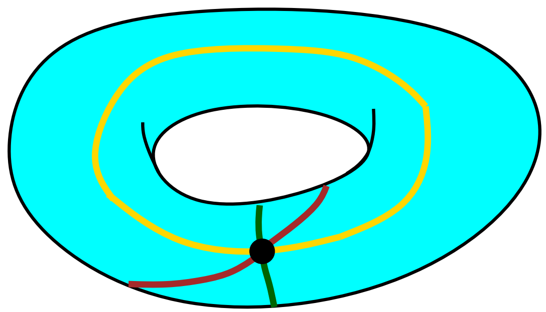
Toroidal Schnyder woods: existence

Thm[Fijavz, unpublished]

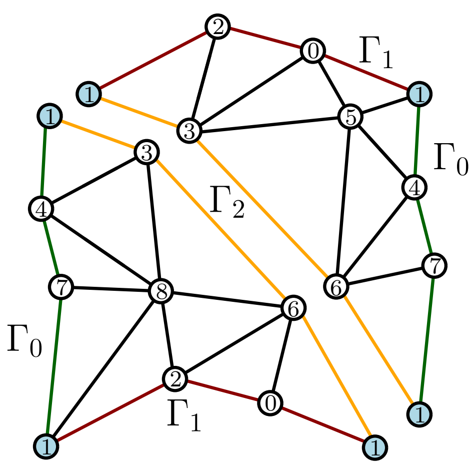
(planar simple triangulations)

A simple toroidal triangulation contains three non-contractible and non-homotopic cycles that all intersect on one vertex and that are pairwise disjoint otherwise.

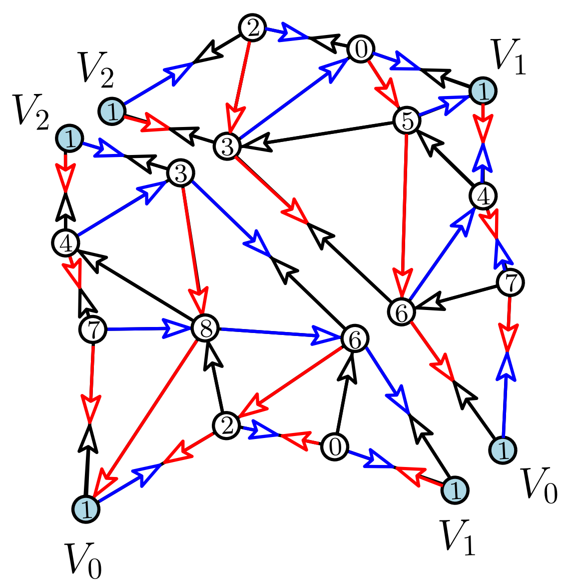




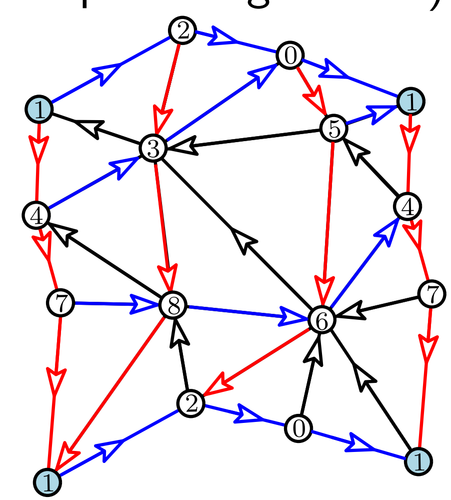
split along Γ_0, Γ_1, Γ_2



(two planar quasi-triangulations)



crossing toroidal Schnyder wood (for simple triangulations)

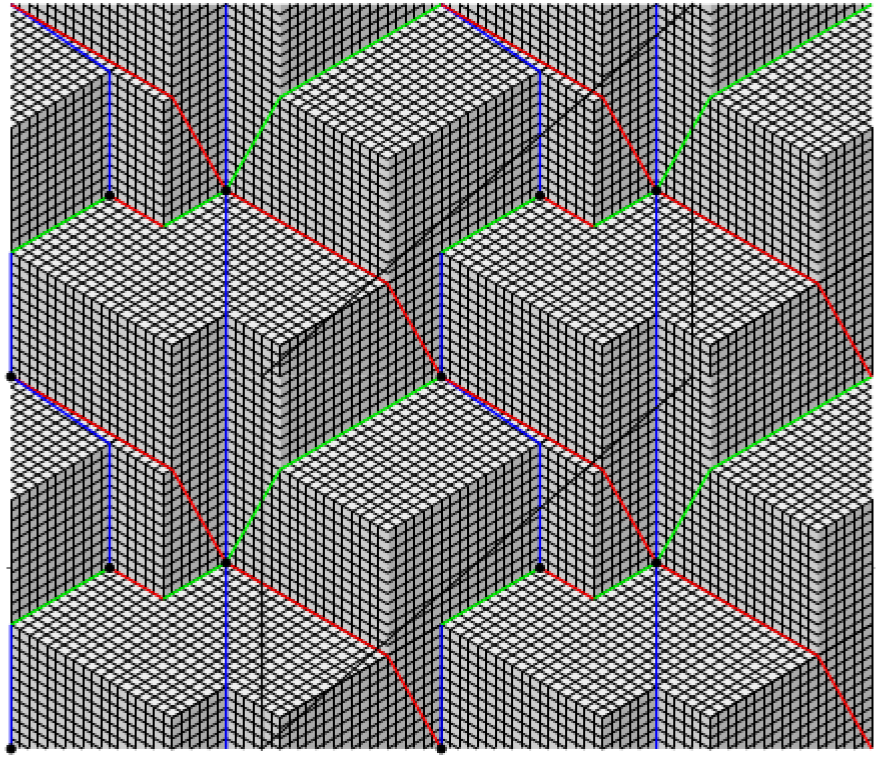


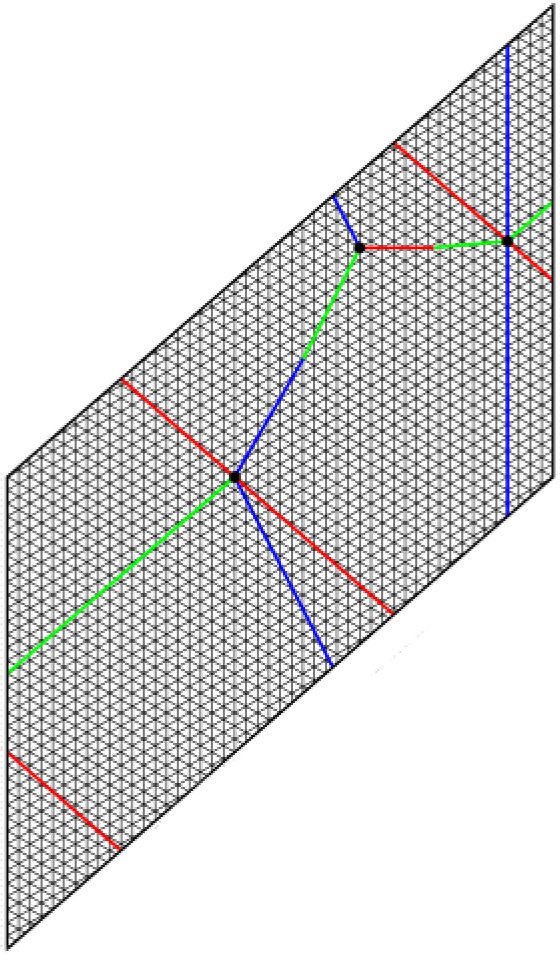
Toroidal Schnyder woods: drawing

(planar simple triangulations)

Thm[Goncalves Lévêque]

A simple toroidal triangulation admits a straight-line periodic drawing on a grid of size $O(n^2 \times n^2)$

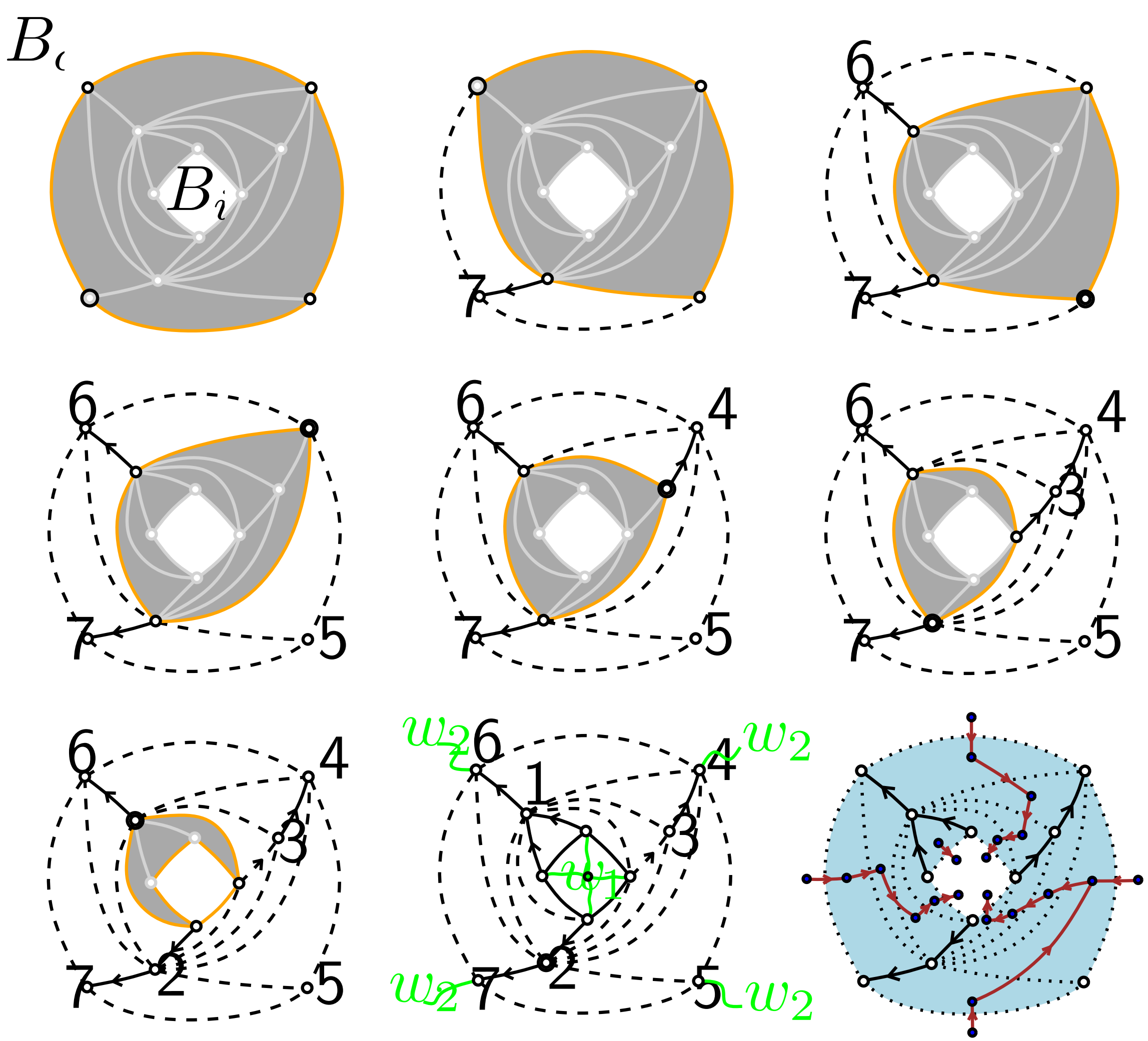




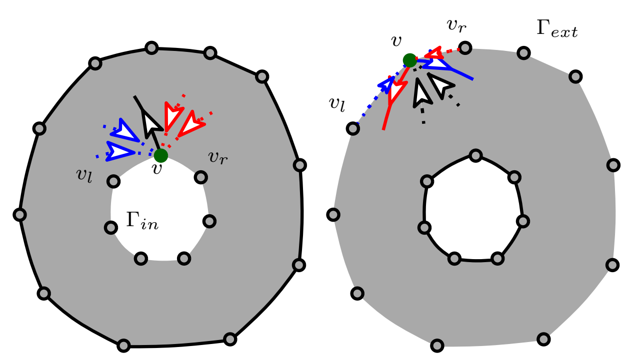
Cylindric Canonical orderings [Castelli Aleardi, Fusy, Devillers]

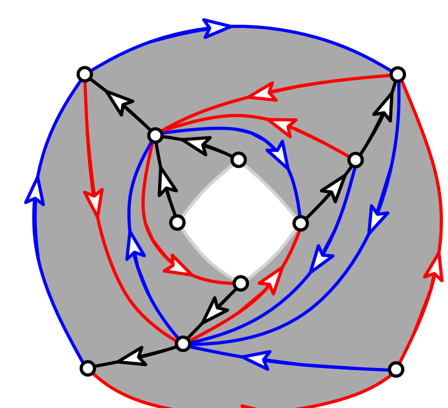
Warning: the interior boundary (defined by Γ_{in}) must be chord-free

- perform vertex shelling starting from exterior boundary B_{out}(orange)



boundary local Schnyder rule

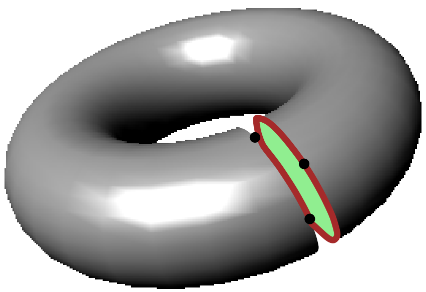




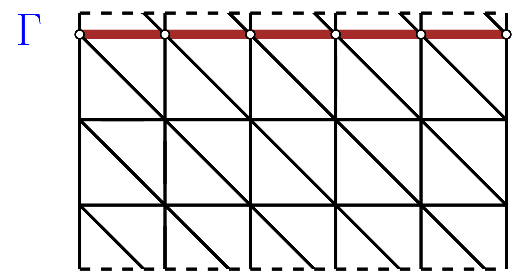
Cylindrical Schnyder wood

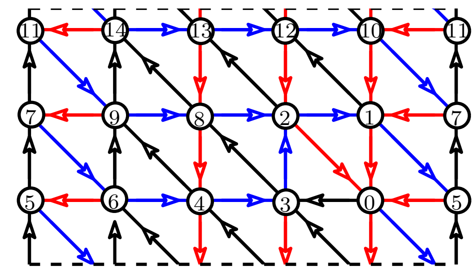
Toroidal (non-crossing) Schnyder woods

Idea: cut the torus along a non-contractible cycle Γ (with no chords on one side)



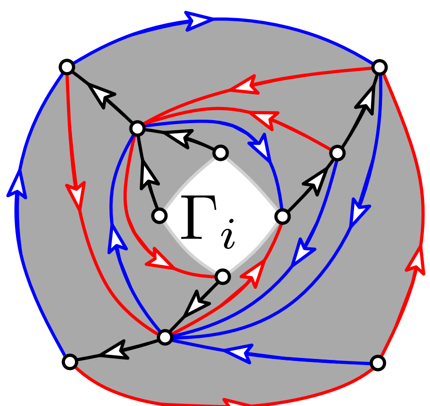
cylindrical triangulation

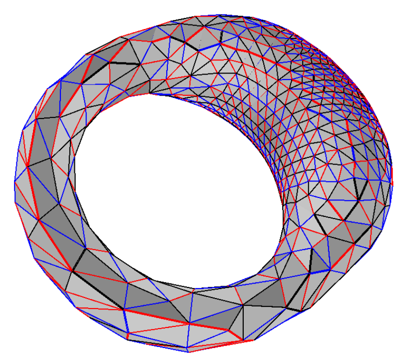


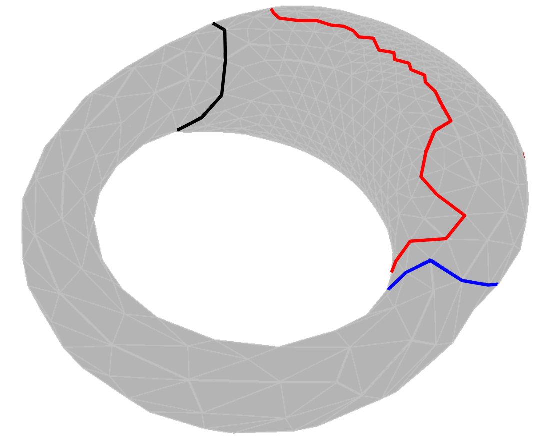


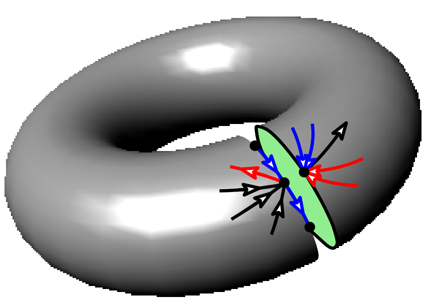
Compute a cylindrical Schnyder wood

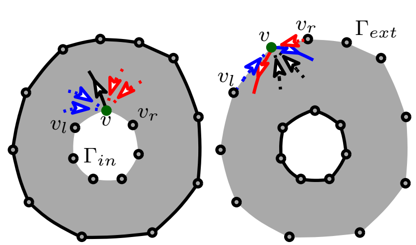
cylindrical canonical ordering











Drawback
The toroidal Schnyder wood is not crossing

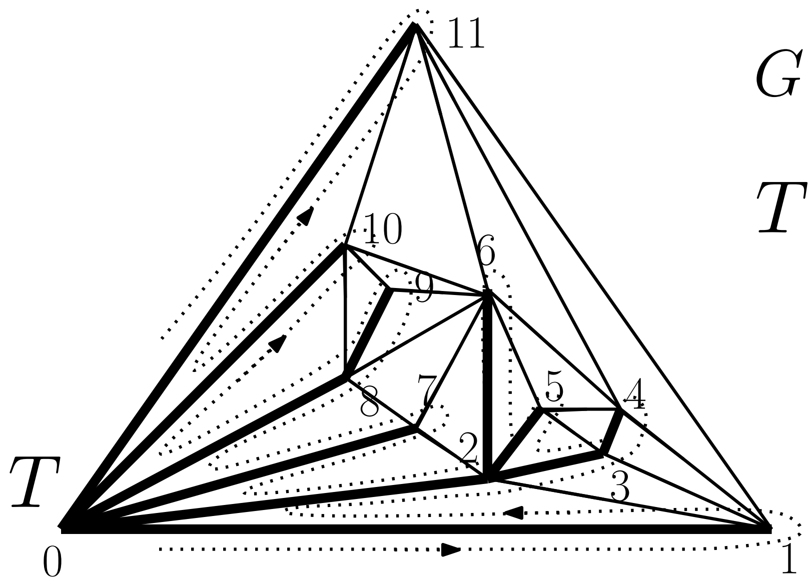
Glue together the two boundaries (local Schnyder rule remains satisfied)

Graph encoding application

A simple encoding scheme

Turan encoding of planar map (1984)

$12n$ bits encoding scheme



$G = (V, E) \qquad |V| = n \qquad |E| = e$

$T :=$ (any) vertex spanning tree of G

T ( ) ((( )) ( ) ( )) ( ) (( )) ( ) ( )

parenthesis word of size $2n$

$G \setminus T$ [[[[ ]]][[[[ ]][ ]][[[[[[ ]][ ]]][[ ]]][ ]]]]

parenthesis word of size $2n$

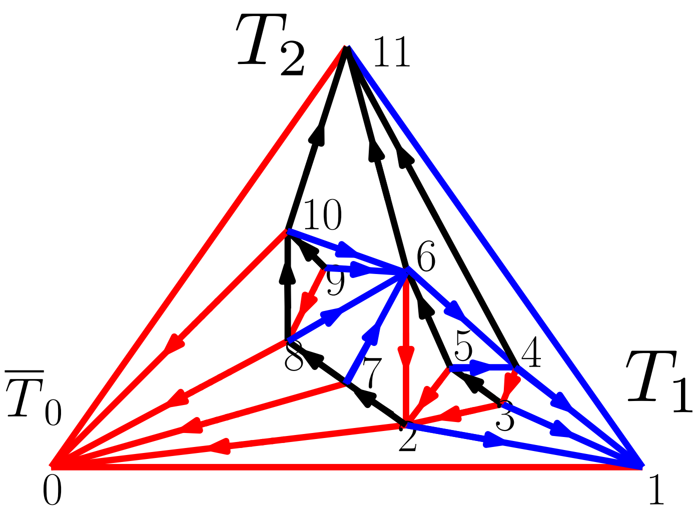
$length(S) = 2e$ symbols

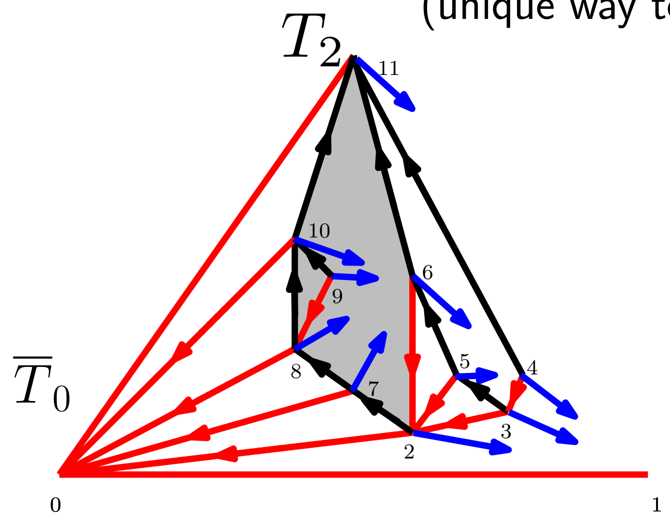
$S(G)$ ([[[)(](][[[)])(]][) . . .

$(2 \log_2 4)e = 4e = 12n$ bits

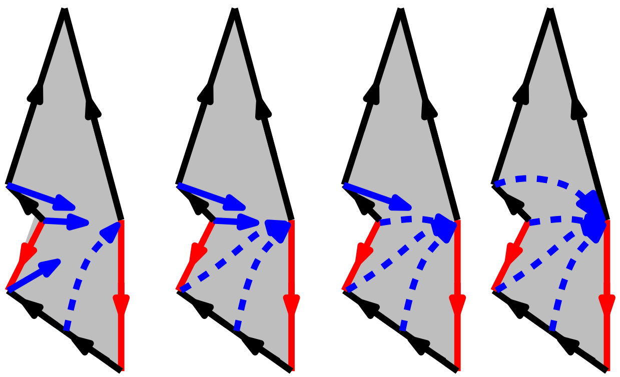
A more efficient encoding

Canonical orderings - Schnyder woods (He, Kao, Lu '99)





(unique way to triangulate each face of $\overline{T}_0 \cup T_2$)

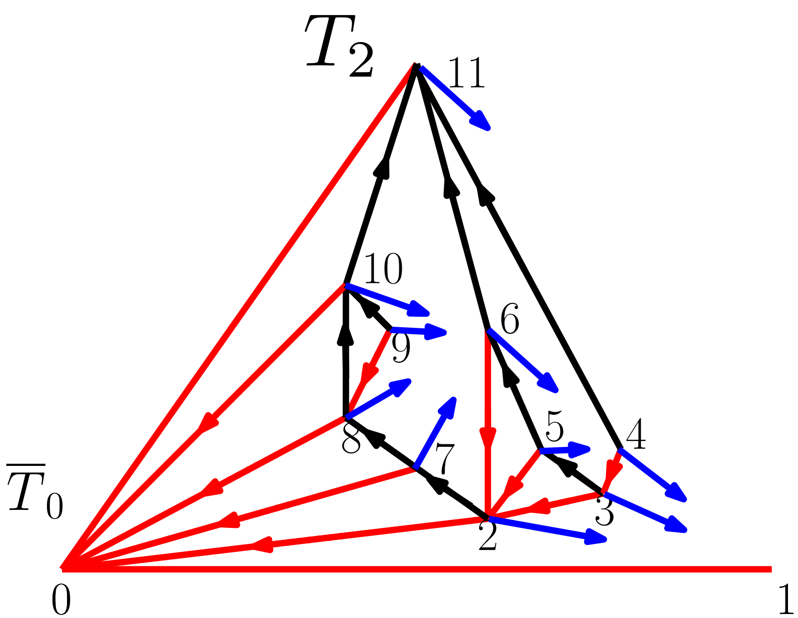


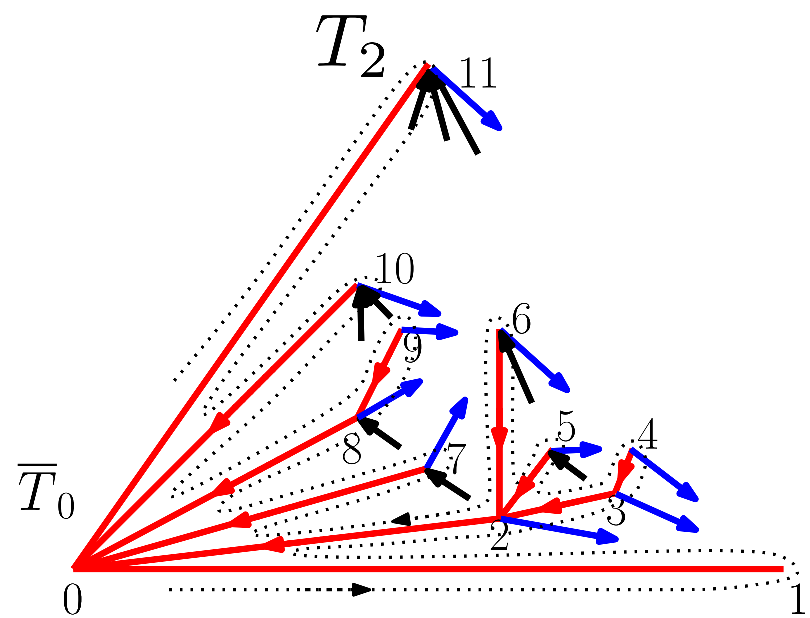
T_1 is redundant: reconstruct from T_0, T_2

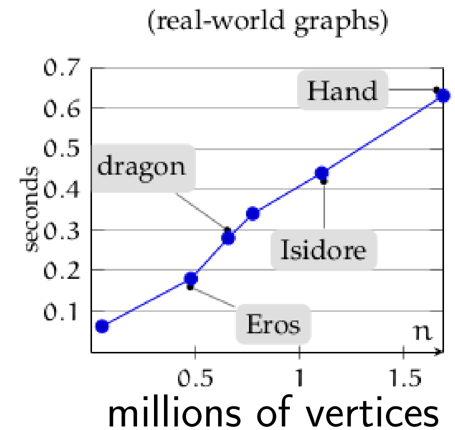
A more efficient encoding

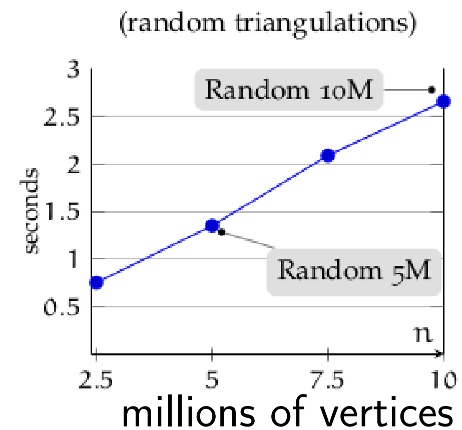
Canonical orderings - Schnyder woods (He, Kao, Lu '99)

$4n$ bits (for triangulations)









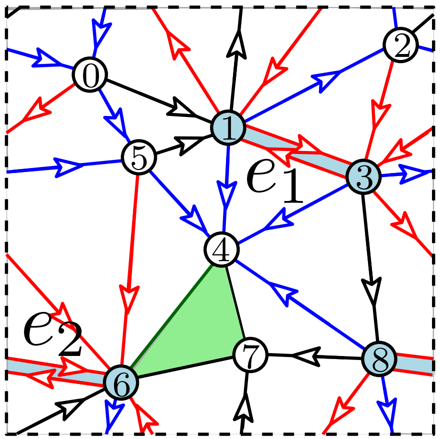
T_2 can be reconstructed from T_0 and the number of ingoing edges (for each node)

$\overline{T}_0$ ( ) ((( )) ( ) ( )) ( ) (( )) ( ) ( ) $2(n-1)$ symbols= $2(n-1)$ bits

$\approx 4n$ bits

$\overline{T}_2$ 00000101010100110111 $(n-1)+(n-3)=2n-4$ bits

Genus g Schnyder woods: application

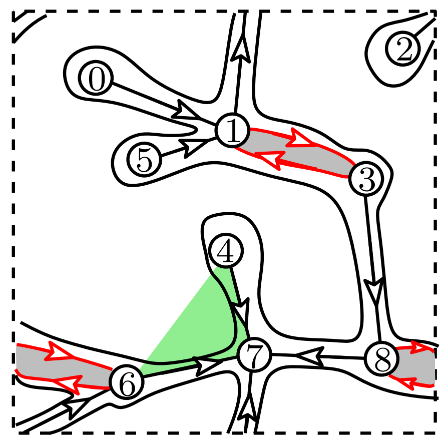


$E^s = \{(1,3),(6,8)\}$

Thm [Castelli-Aleardi Fusy Lewiner, SoCG'08]

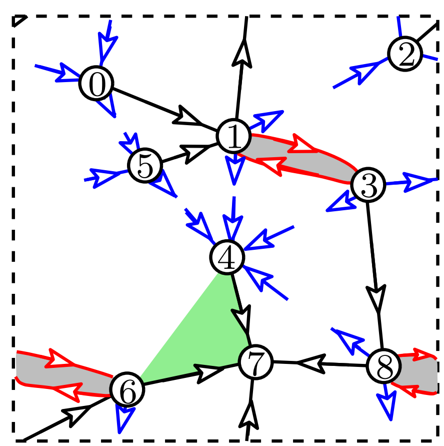
A triangulation of genus g having n vertices can be encoded with at most $4n + O(g \log n)$ bits

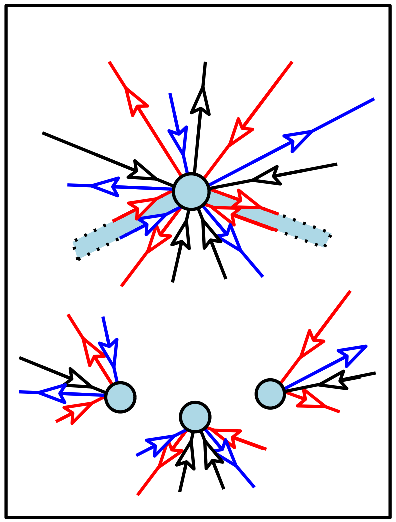
$G_2 = \overline{T}_2 \cup \{e_1, e_2\}$



Encode map $G_2 = \overline{T}_2 \cup E^s$: a tree plus $2g$ edges: $2n + O(g \log n)$ bits

Mark special vertices: $O(g \log n)$ bits





Store outgoing blue edges incident to special edges: $O(g \log n)$ bits

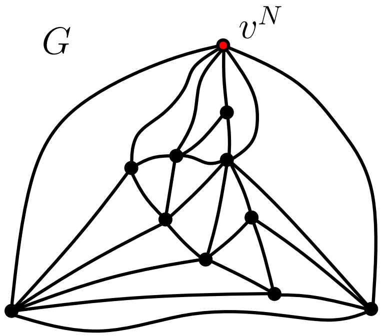
For each node in $T_2 \cup E^s$ store the number of ingoing blue edges (color 1): $2n + O(g \log n)$ bits

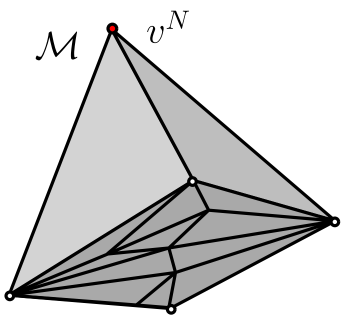
Drawing graphs on surfaces

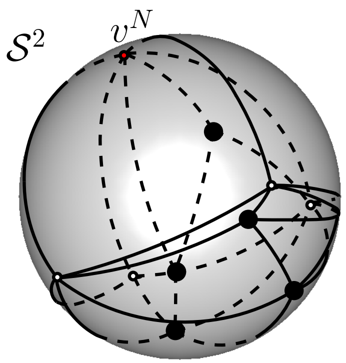
(periodic straight line drawings)

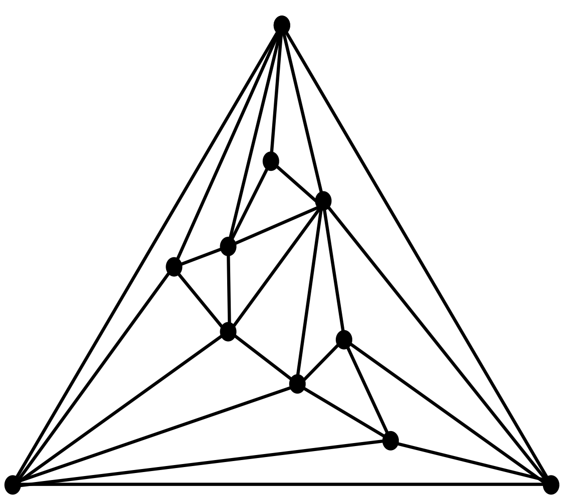
Drawing higher genus graphs

$g = 0$





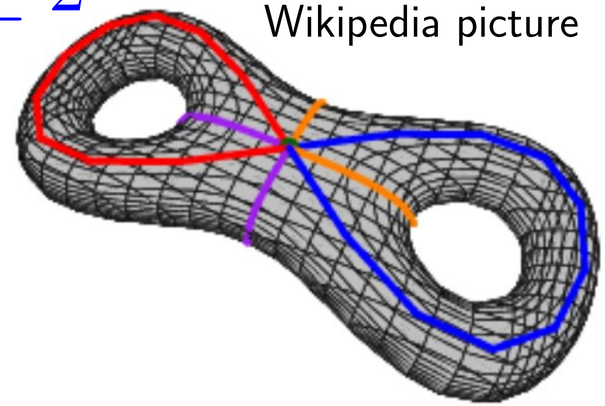




Drawing higher genus graphs

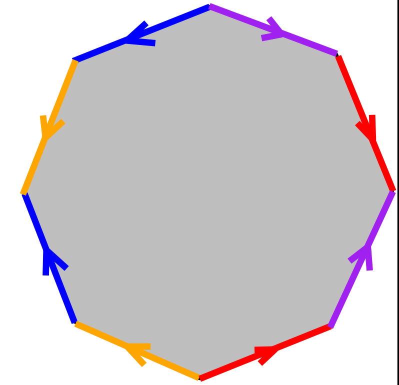
(Palais de la Découverte,Fête de la Science, October 2013)

$g \geq 2$



Wikipedia picture

Polygonal scheme

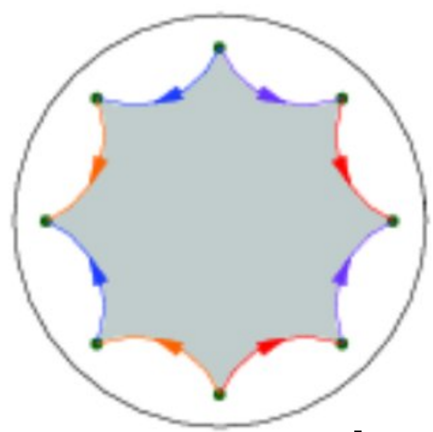


drawing in polynomial area

[Duncan, Goodrich, Kobourov, GD'09]

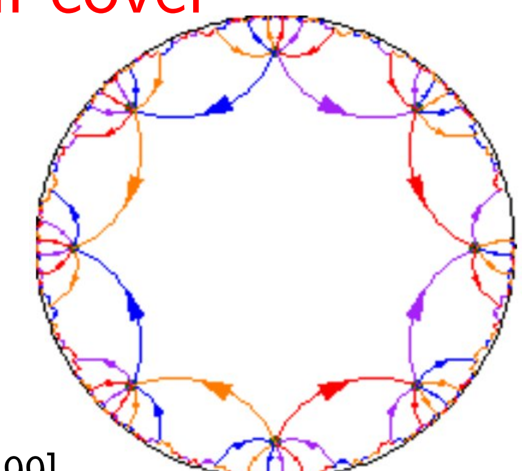
[Chambers, Eppstein, Goodrich, Löffler, GD'10]

Universal cover



[Mohar'99]

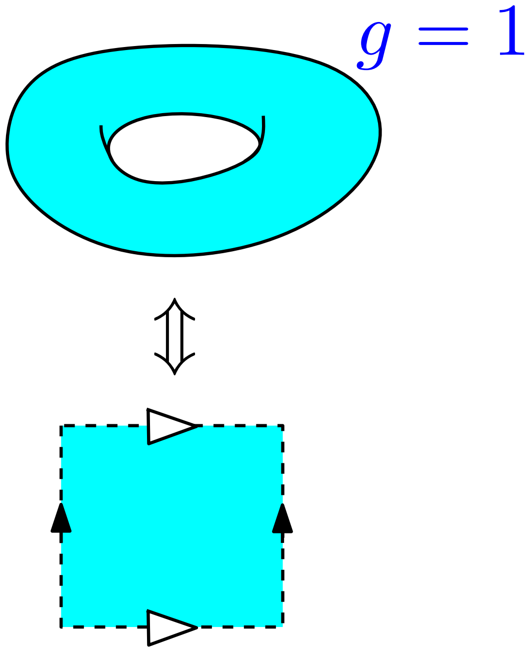
periodic drawing

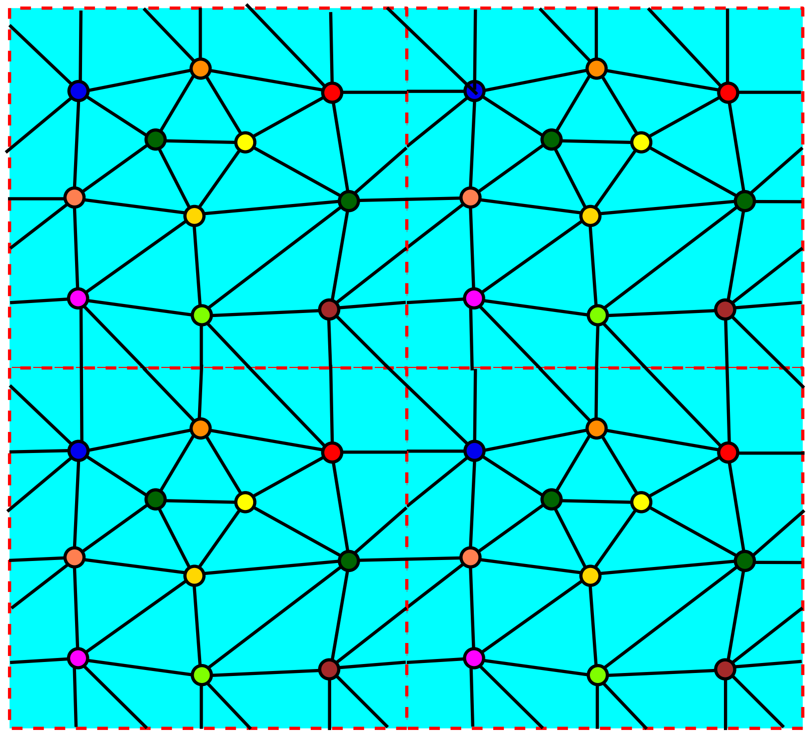


out of circle packing

Drawing toroidal graphs

On the torus

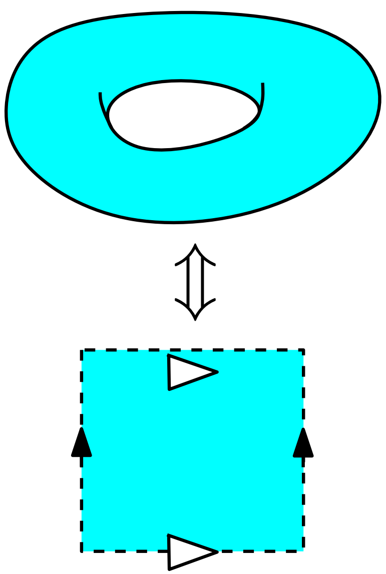


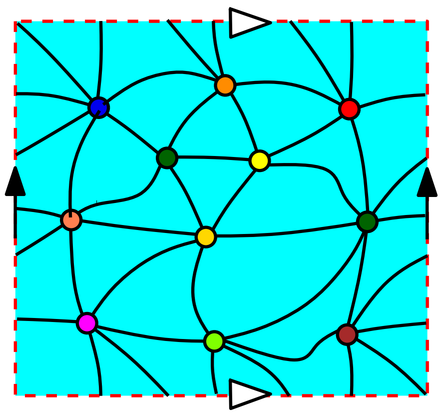


(Palais de la Découverte, Fête de la Science, October 2013)

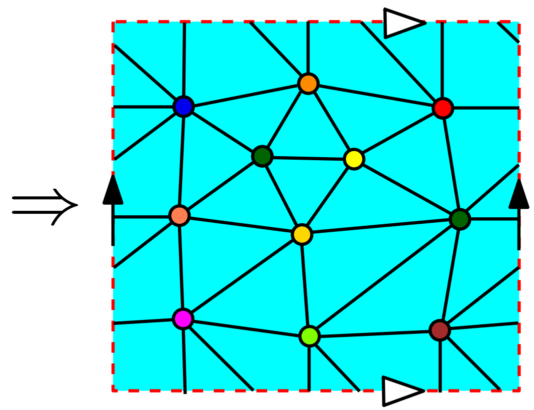
Periodic straight-line drawings

On the torus





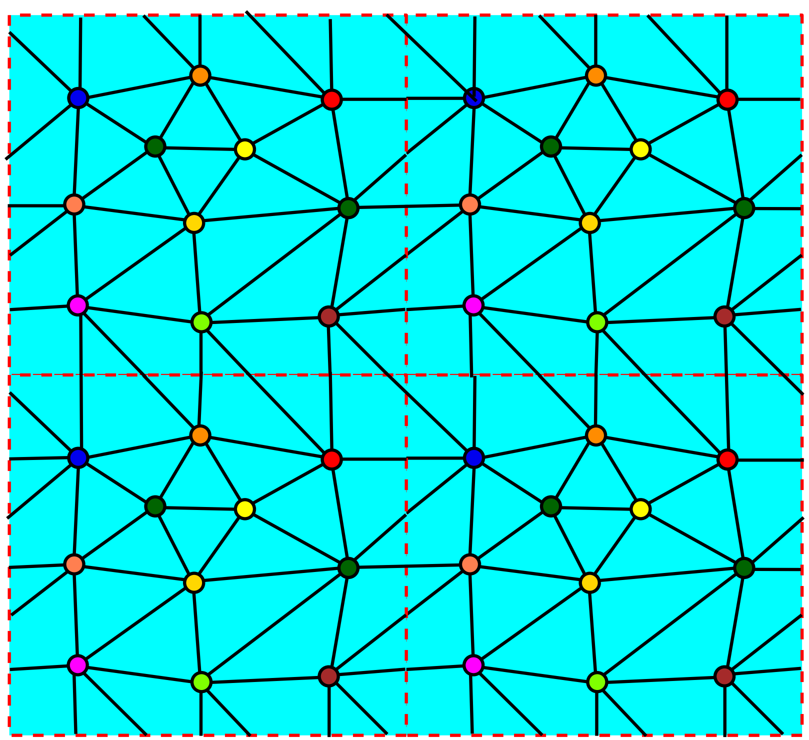
drawing on the flat torus

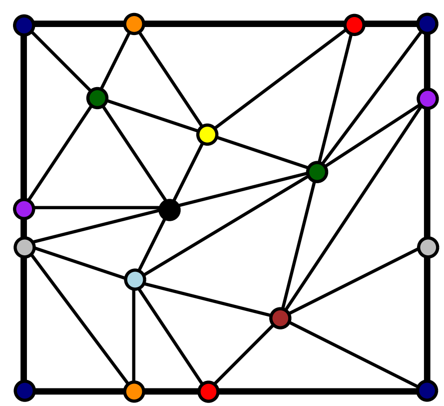


straight-line drawing
x-periodic and
y-periodic drawing

[Castelli Devillers Fusy, GD'12]
$O(n \times n^{\frac{3}{2}})$ **grid**

[Goncalves Lévêque, DCG]
$O(n^2 \times n^2)$ **grid**



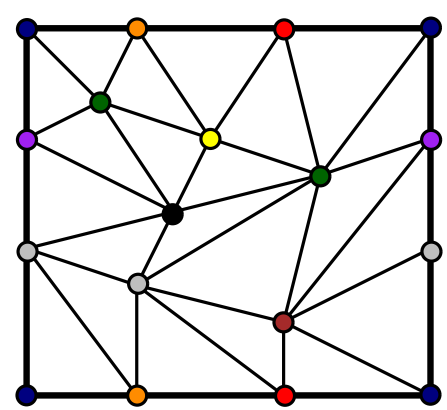


straight-line frame
not x-periodic
not y-periodic

[Chambers et al., GD'10]

[Duncan et al., GD'09]

$O(n \times n^2)$ **grid**



straight-line frame
x-periodic and
y-periodic drawing

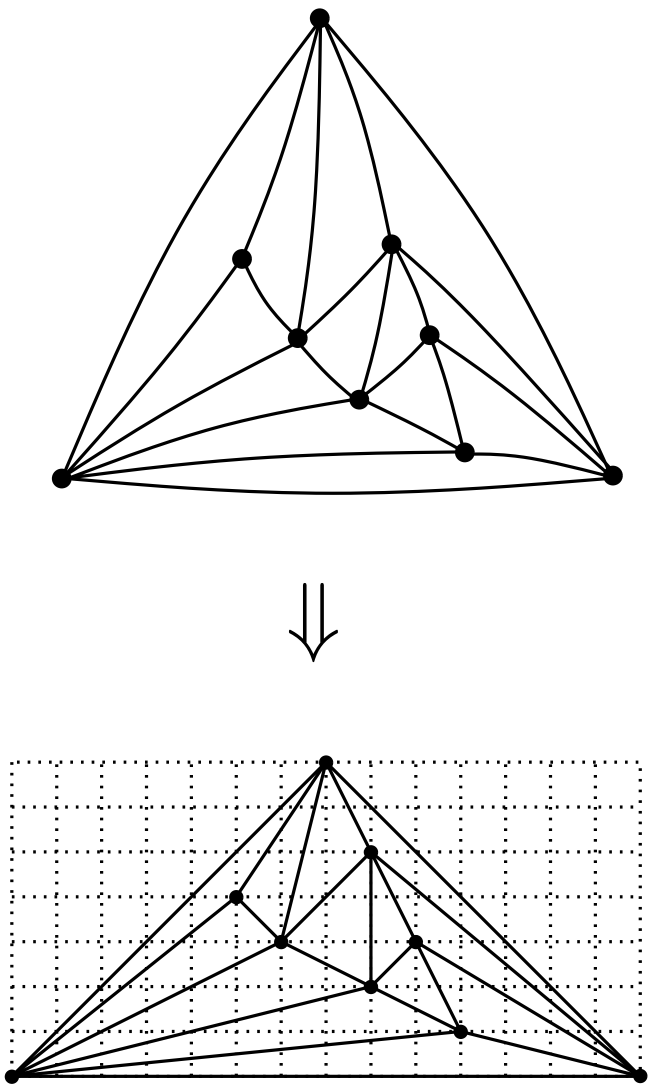
[Castelli Fusy Kostrygin, Latin'14]

A shift-algorithm for the torus

1. Recall algorithm of

[De Fraysseix et al'89]

Plane

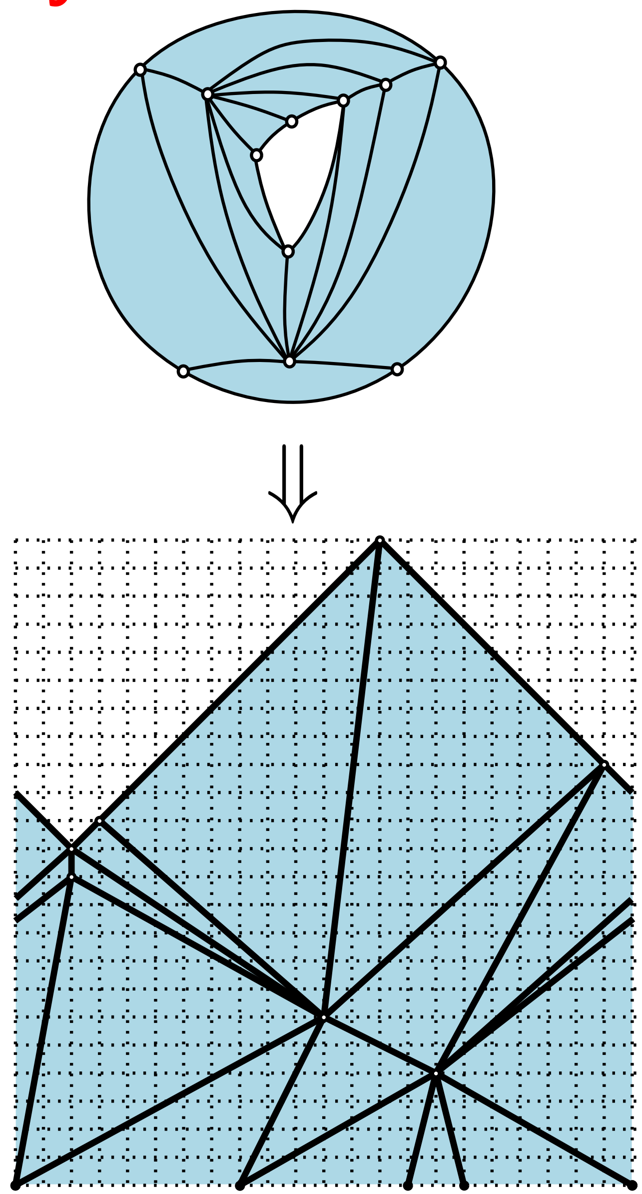


Grid $2n-4 \times n-2$

2. Extend to the cylinder

[Castelli Aleardi Fusy Devillers GD2012]

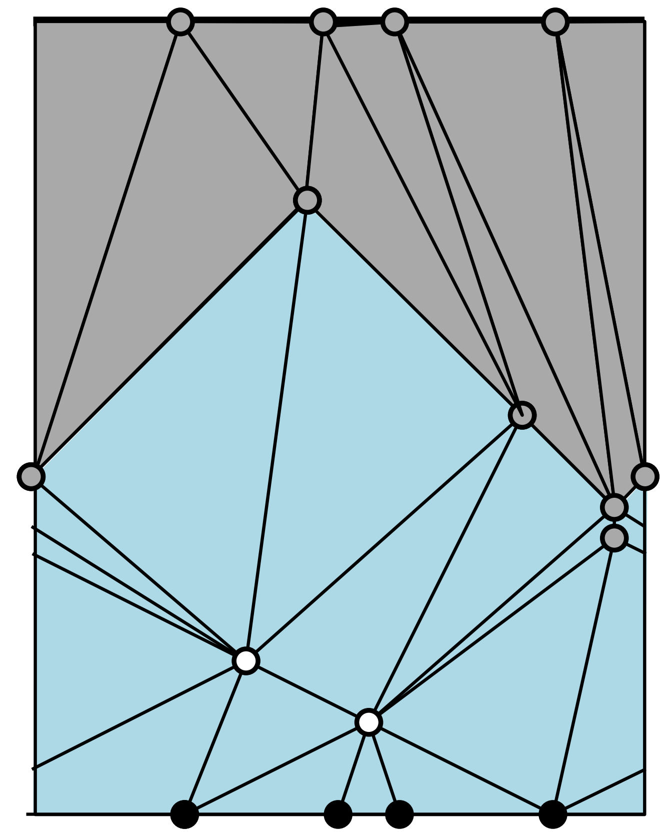
Cylinder



Grid $\leq 2n \times n(2d+1)$

3. Get toroidal drawings

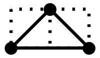
Torus

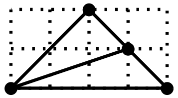


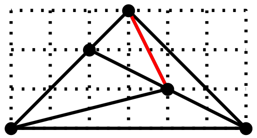
Grid $\leq 2n \times (1+n(2c+1))$

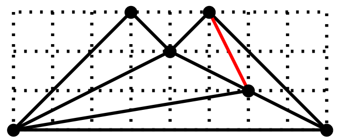
Incremental drawing algorithm

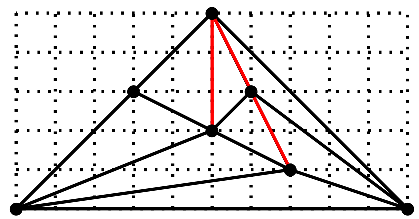
[de Fraysseix, Pollack, Pach'89]

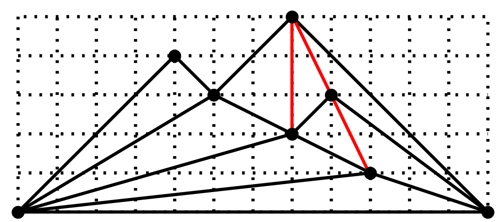
1. 

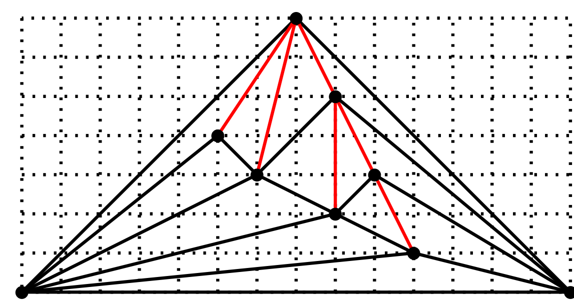
2. 

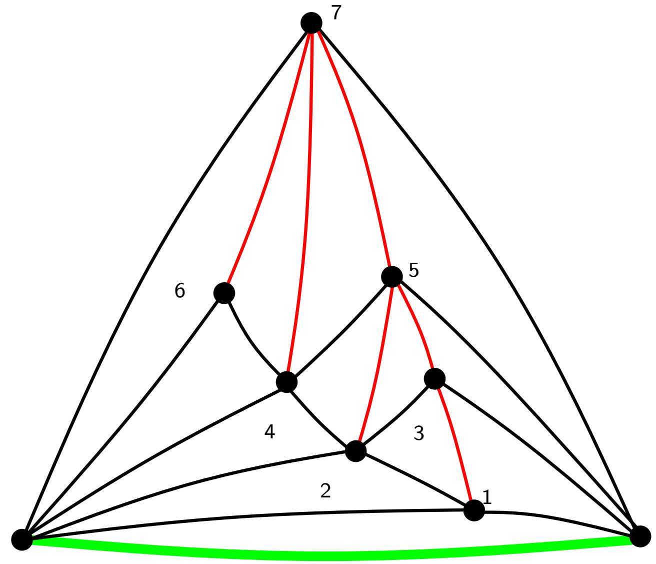
3. 

4. 

5. 

6. 

7. 

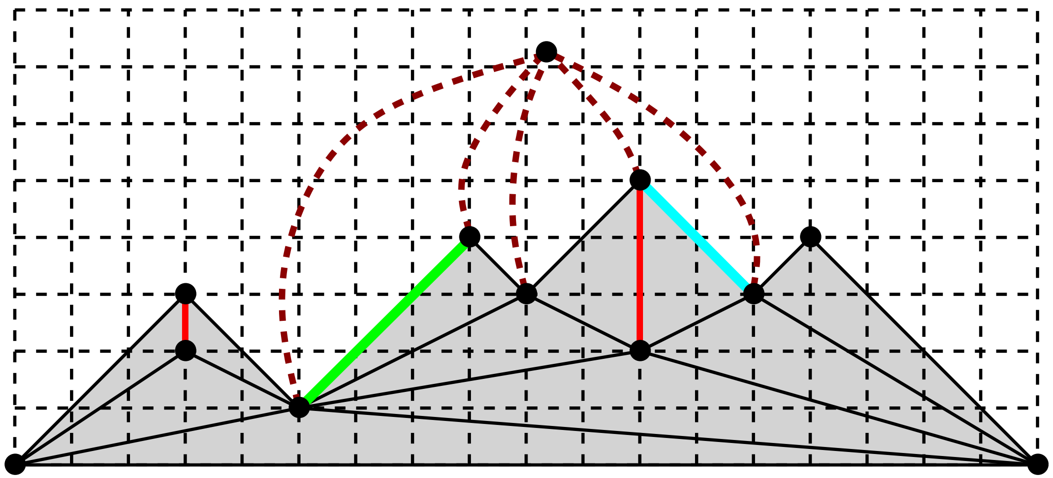


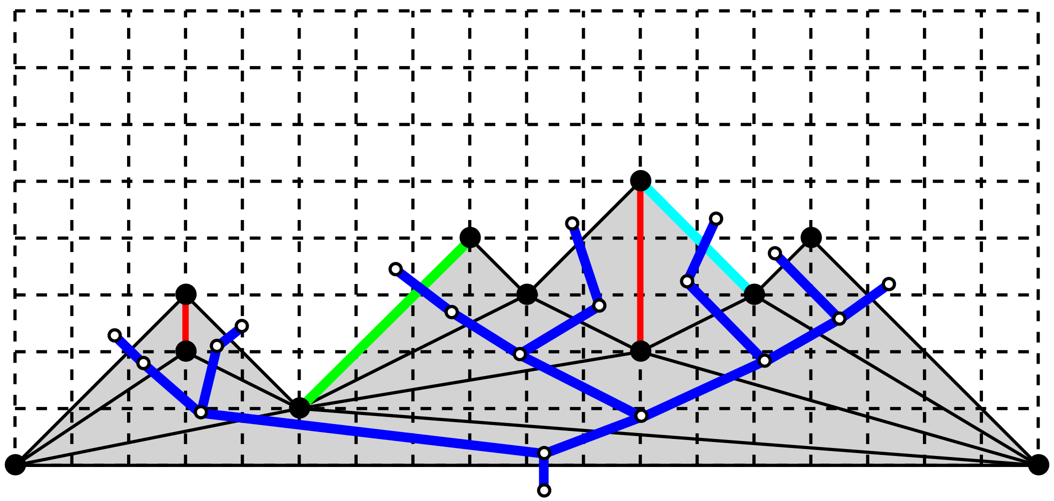
Grid size of G_k: $2k \times k$

Reformulation of the shift-step

At each step: insert two vertical strips of width 1 using the dual tree

G_{k-1}

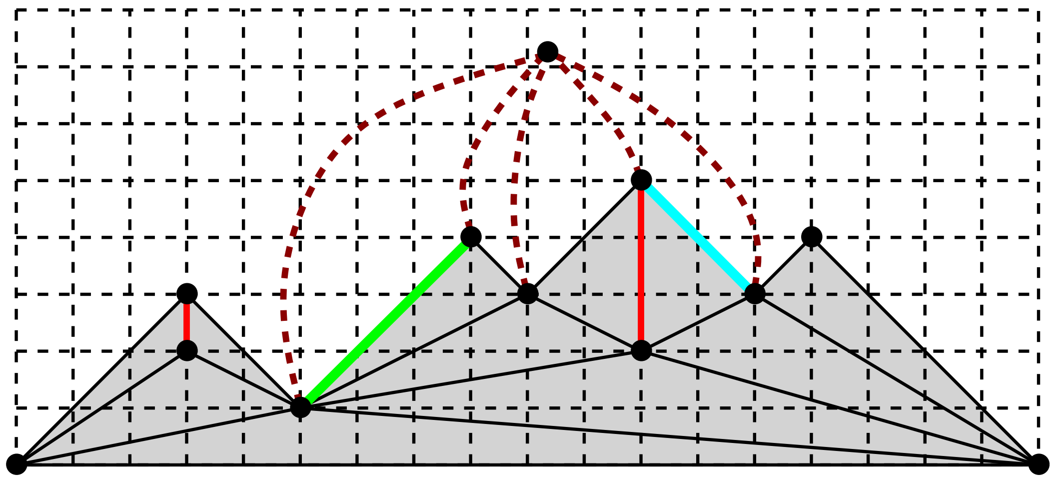


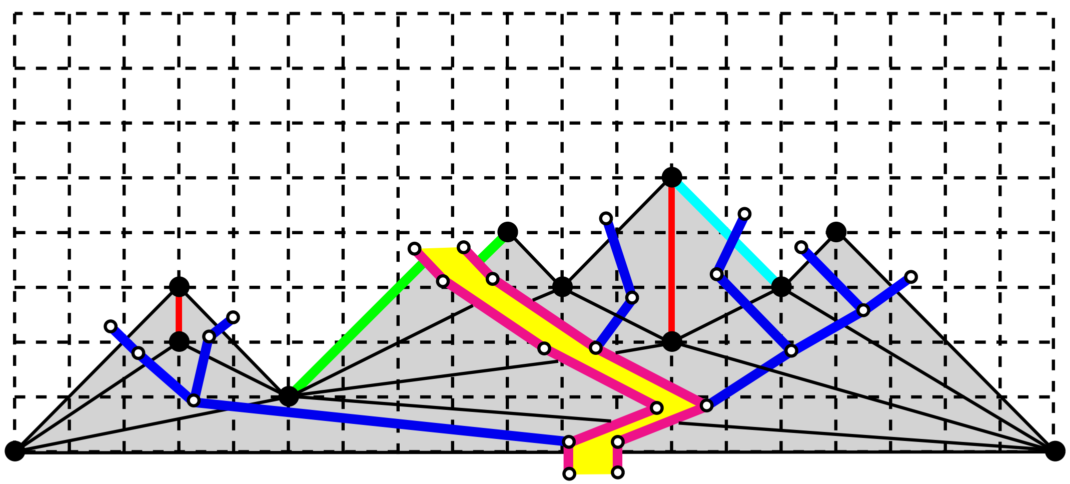


Reformulation of the shift-step

At each step: insert two vertical strips of width 1 using the dual tree

G_{k-1}

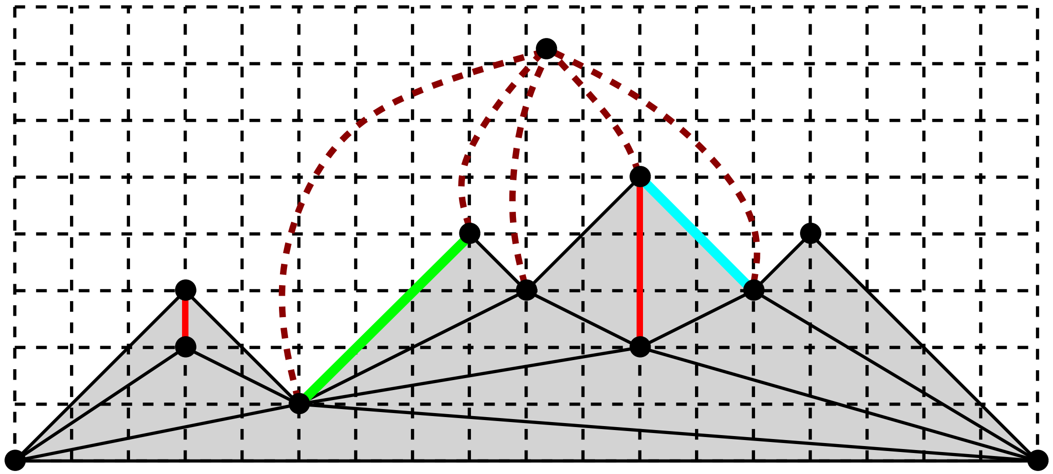


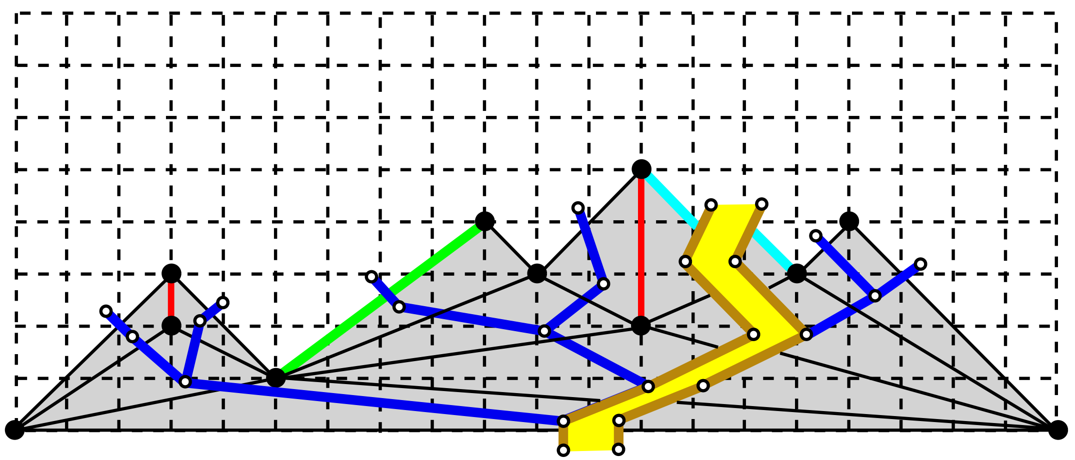


Reformulation of the shift-step

At each step: insert two vertical strips of width 1 using the dual tree

G_{k-1}

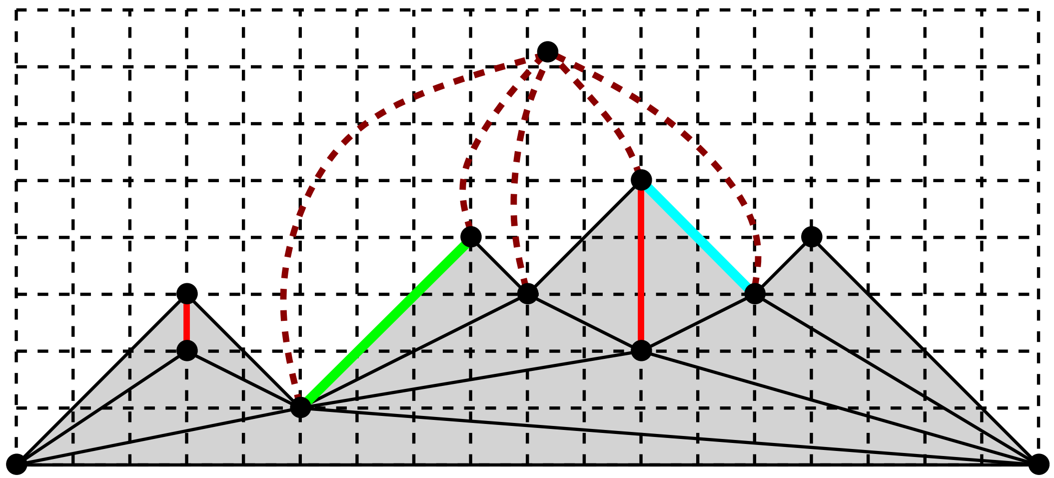




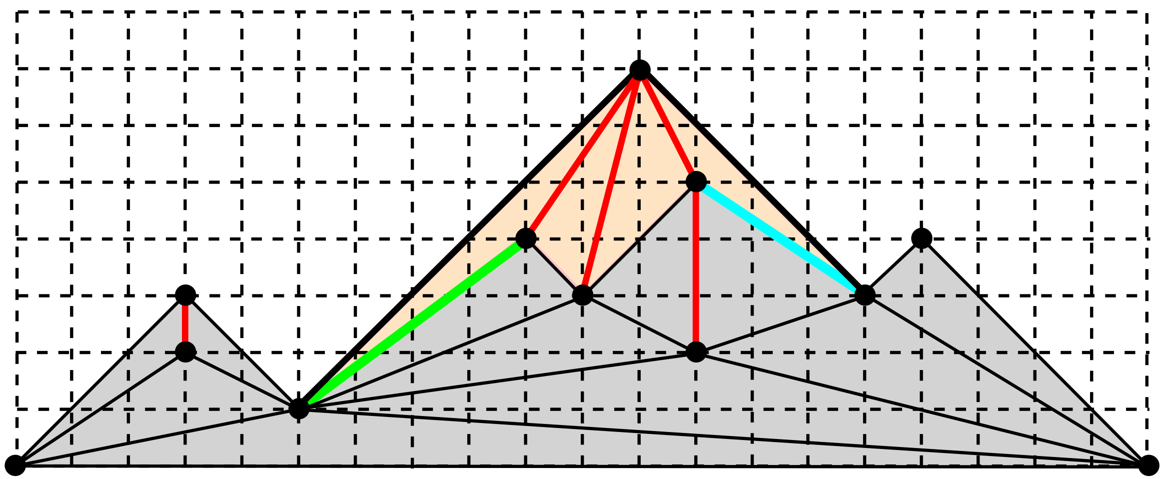
Reformulation of the shift-step

At each step: insert two vertical strips of width 1 using the dual tree

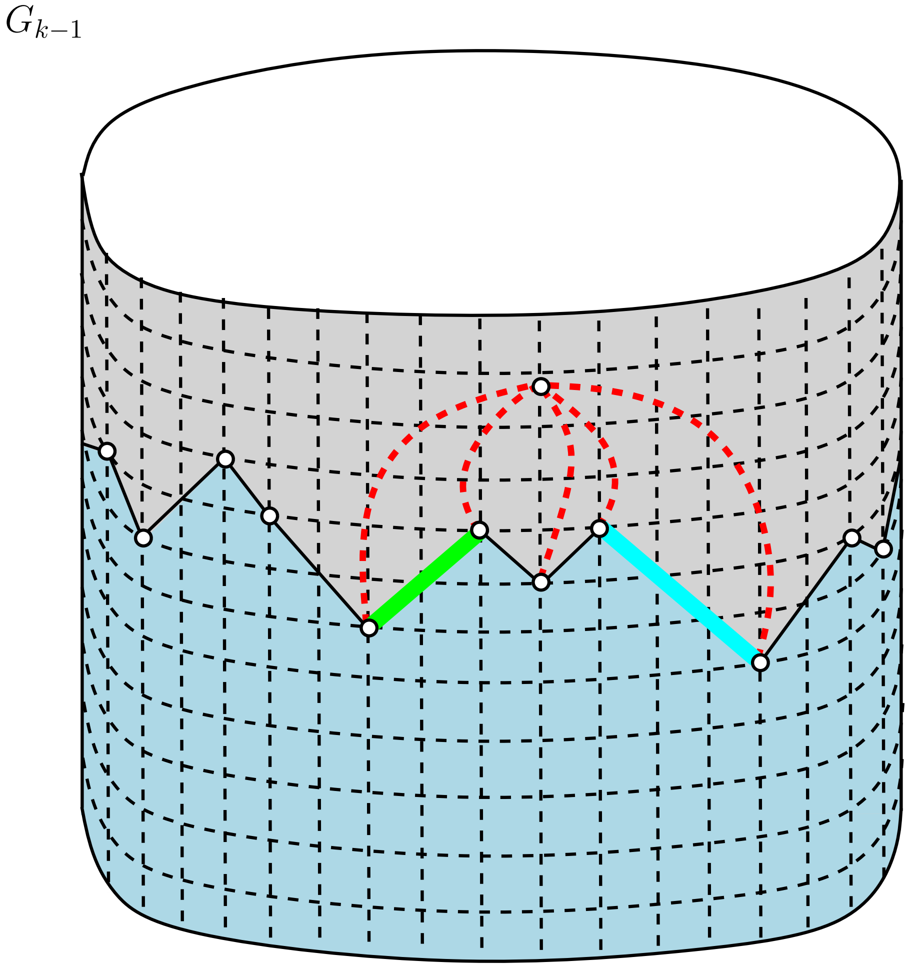
G_{k-1}



G_k

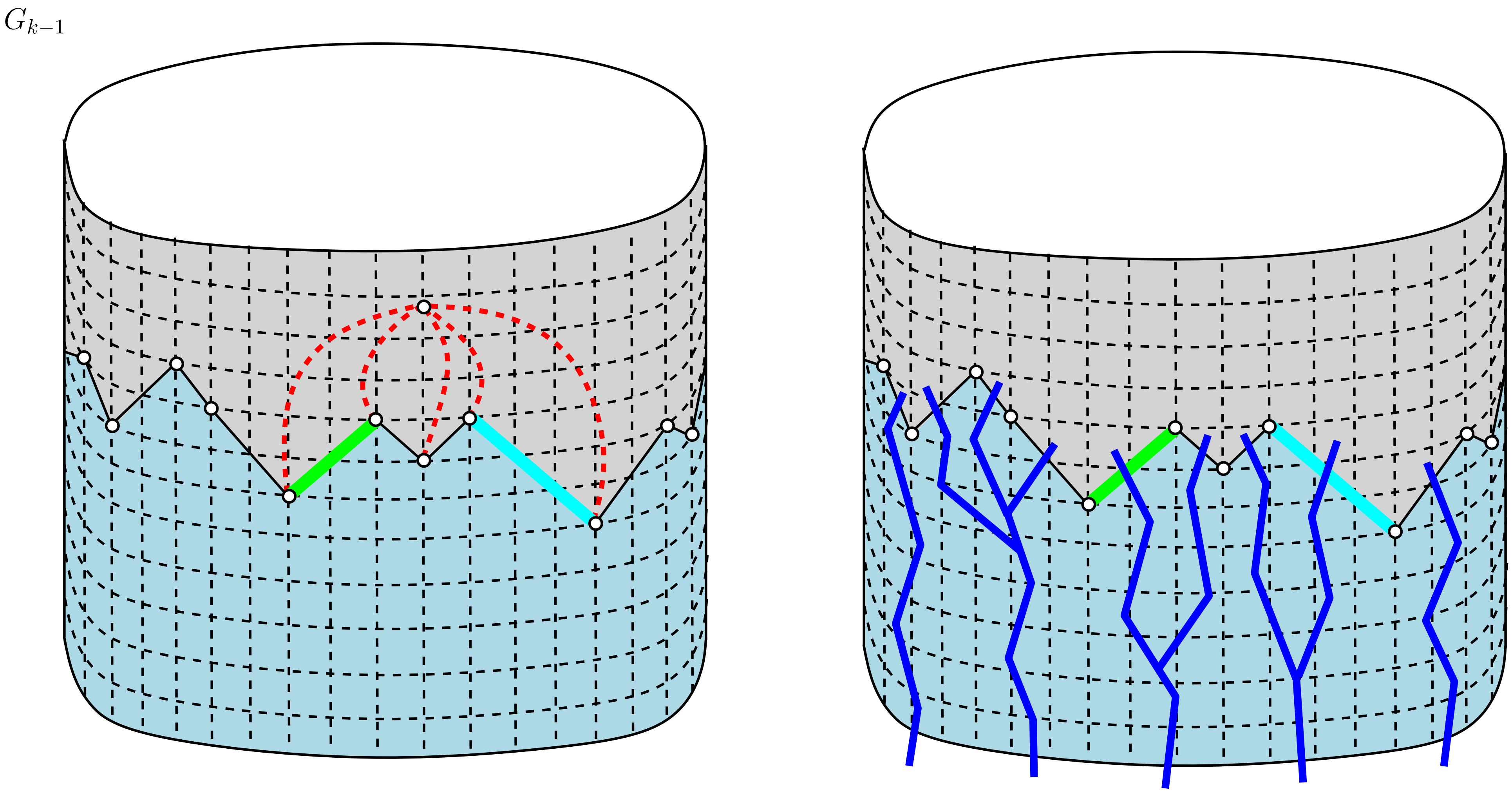


Extension to the cylinder: drawing algorithm



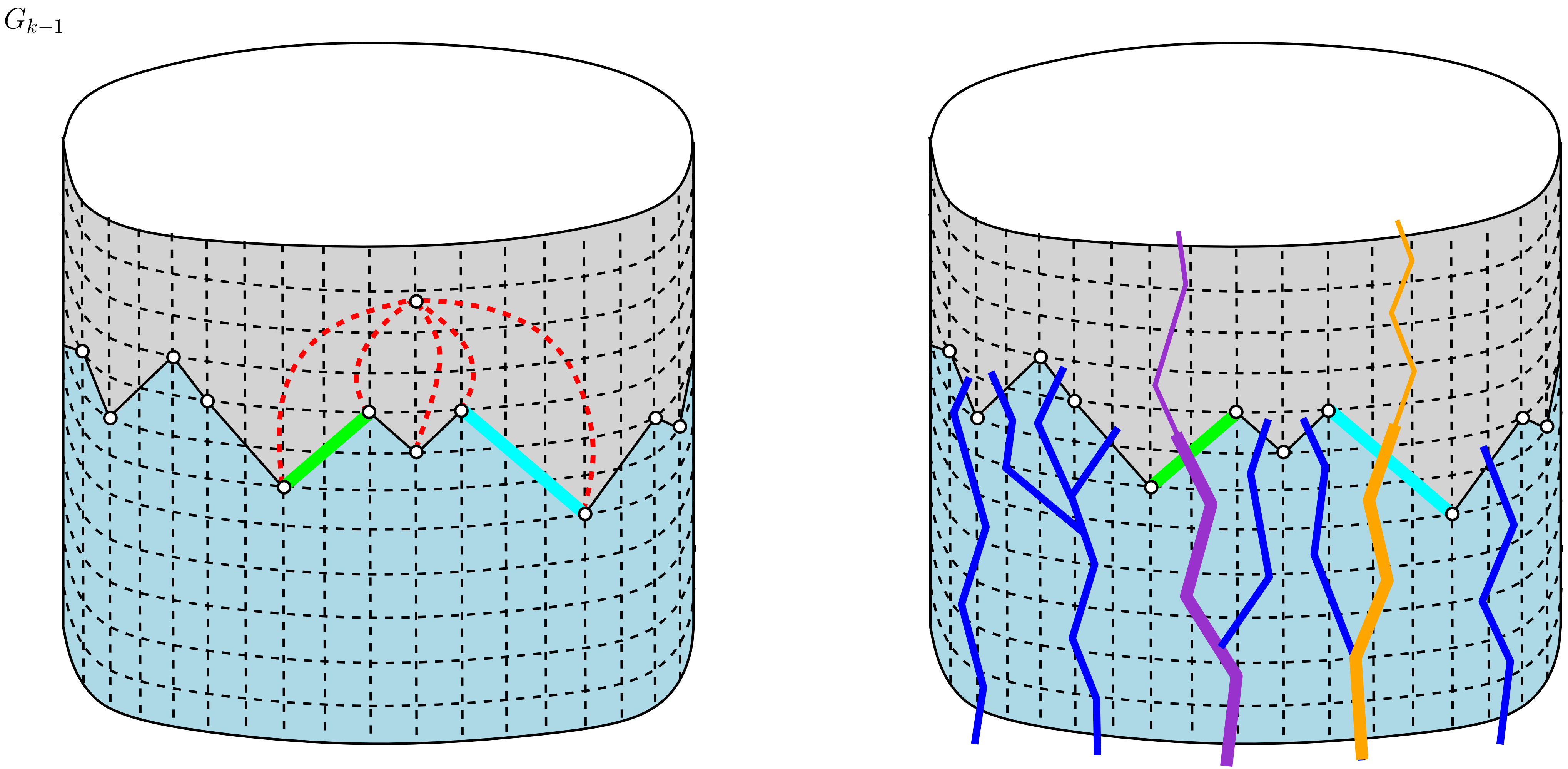
At each step: - insert two vertical strips of width 1
- insert the next vertex as in the planar case

Extension to the cylinder: drawing algorithm



At each step: - insert two vertical strips of width 1
- insert the next vertex as in the planar case

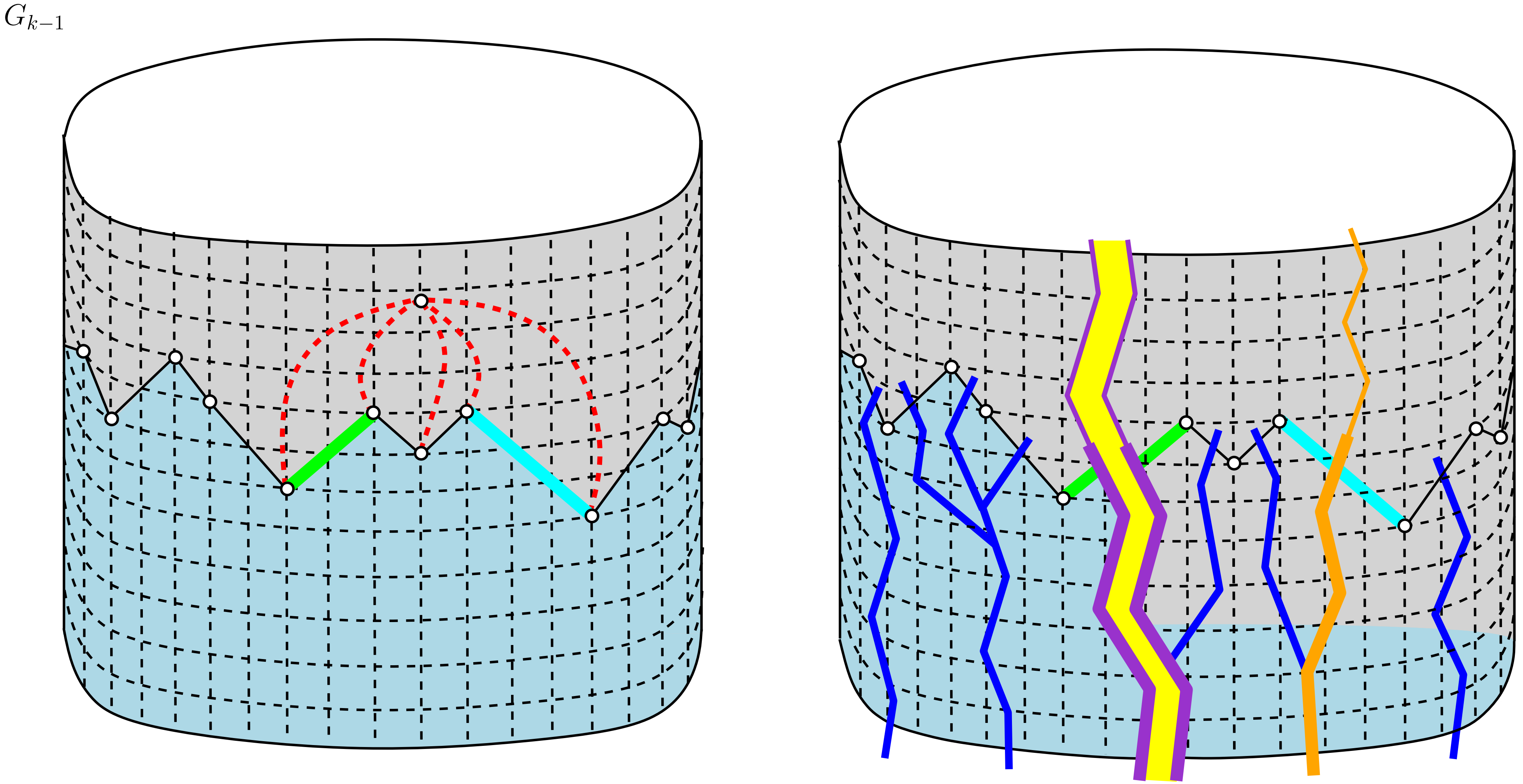
Extension to the cylinder: drawing algorithm



At each step: - insert two vertical strips of width 1

- insert the next vertex as in the planar case

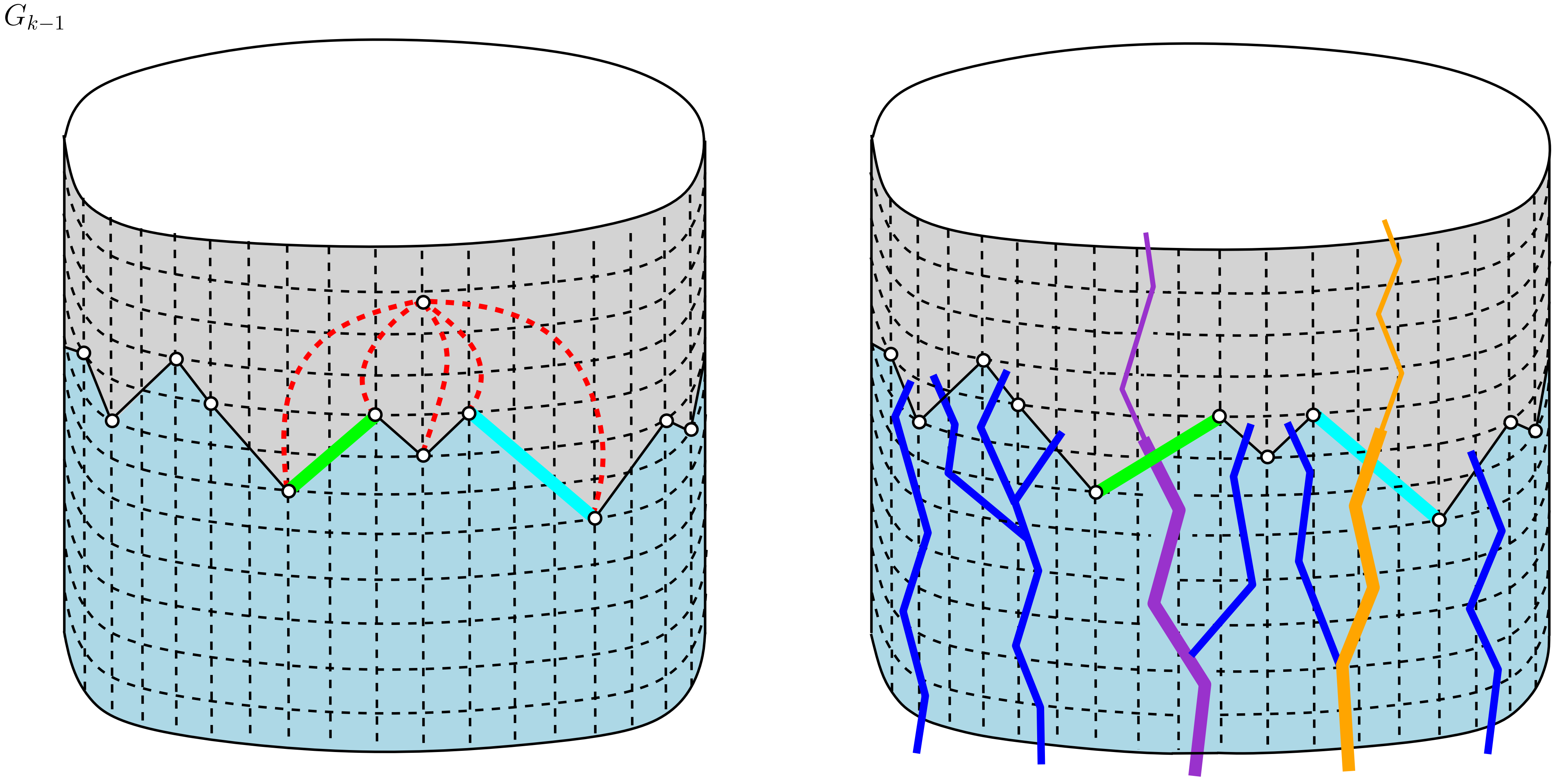
Extension to the cylinder: drawing algorithm



At each step: - insert two vertical strips of width 1

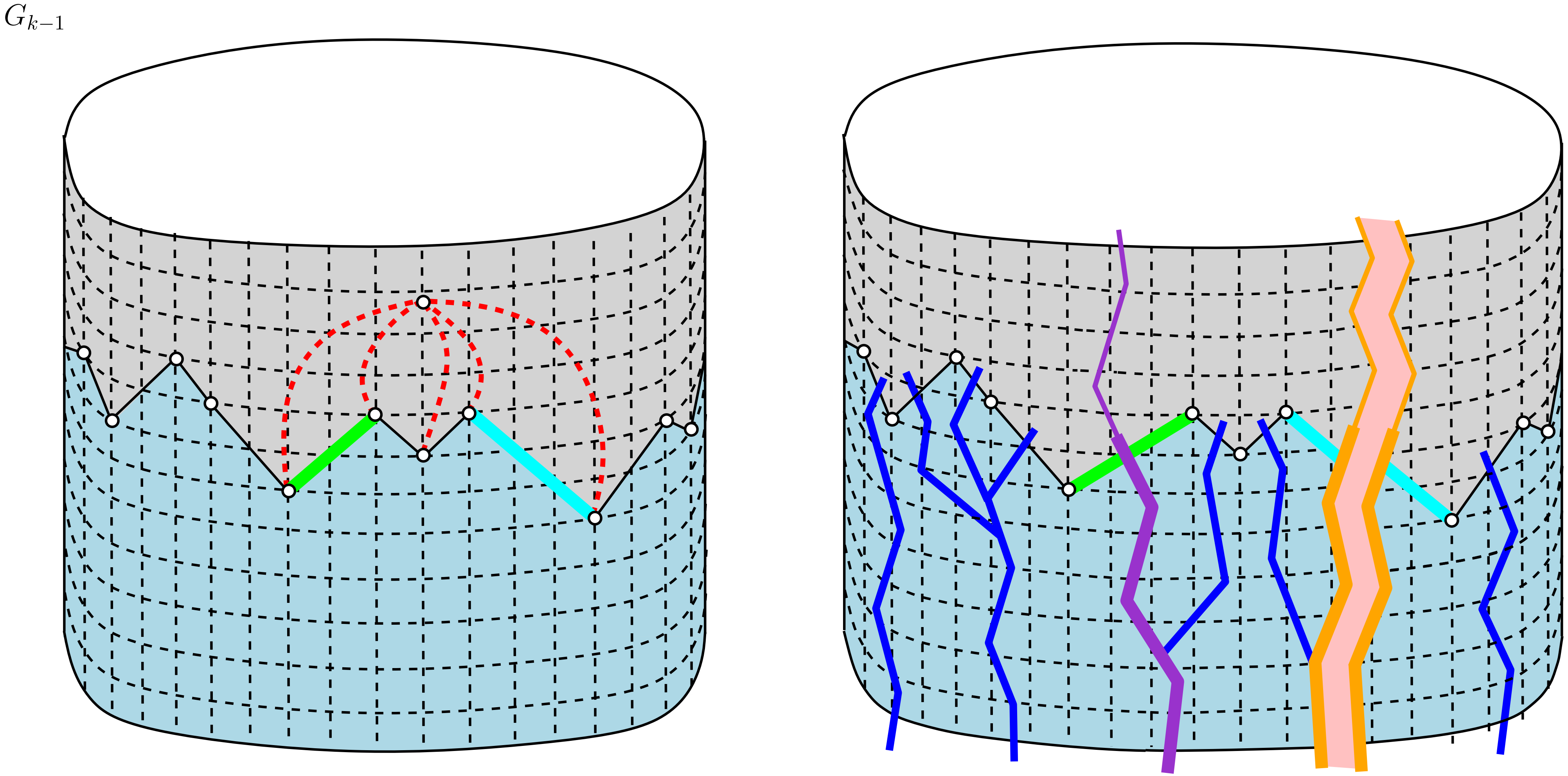
- insert the next vertex as in the planar case

Extension to the cylinder: drawing algorithm



At each step: - insert two vertical strips of width 1
- insert the next vertex as in the planar case

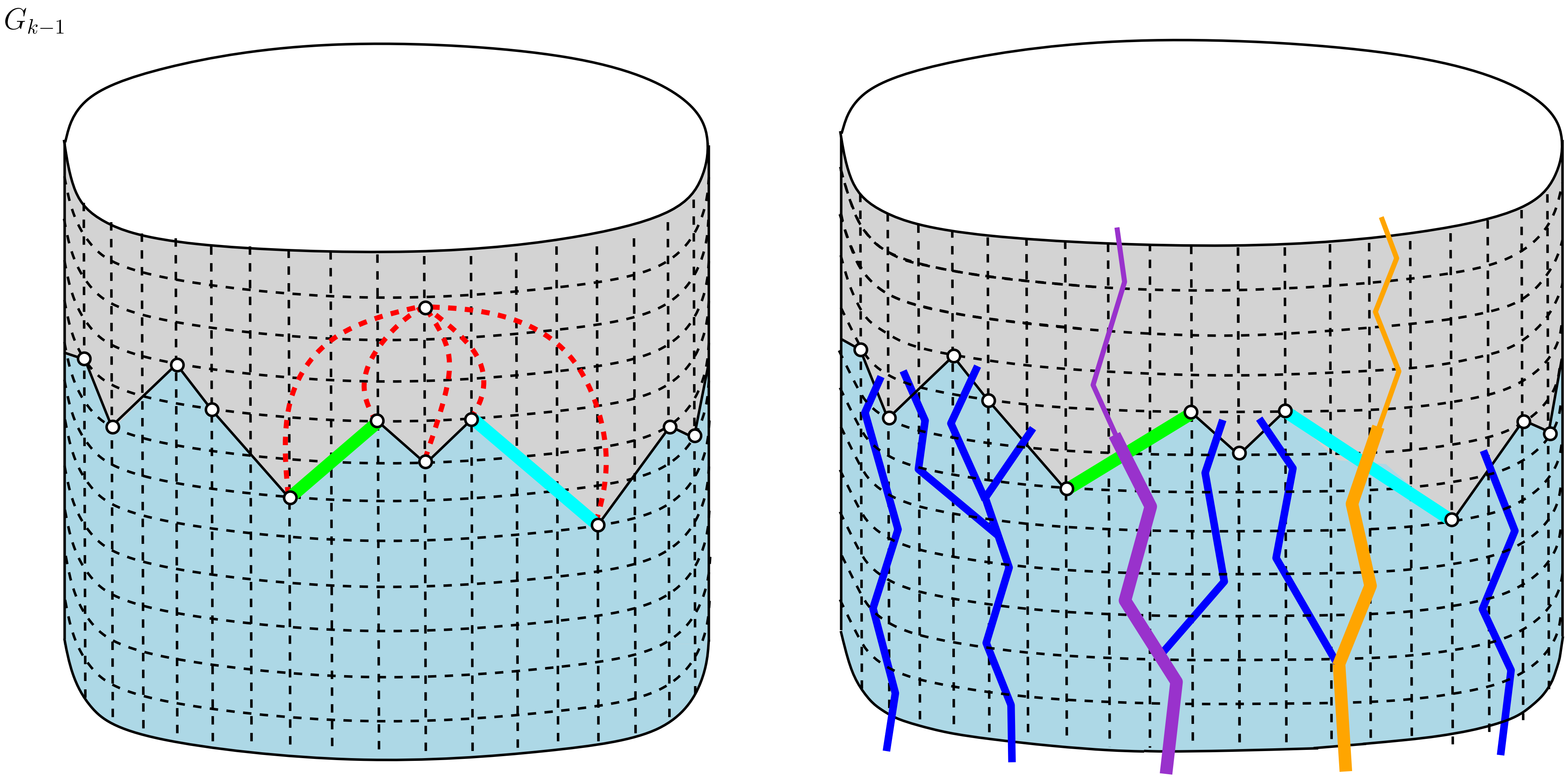
Extension to the cylinder: drawing algorithm



At each step: - insert two vertical strips of width 1

- insert the next vertex as in the planar case

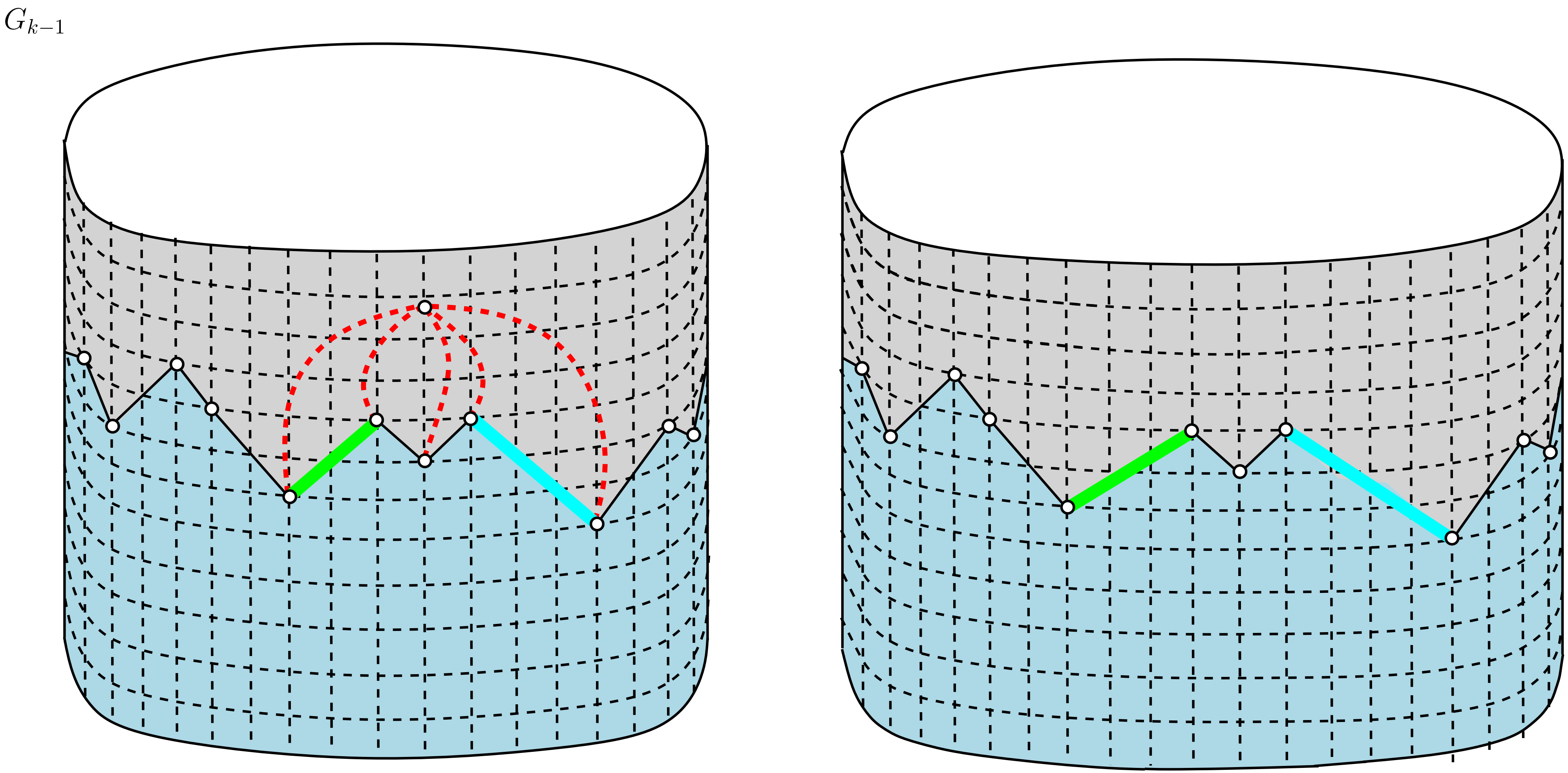
Extension to the cylinder: drawing algorithm



At each step: - insert two vertical strips of width 1

- insert the next vertex as in the planar case

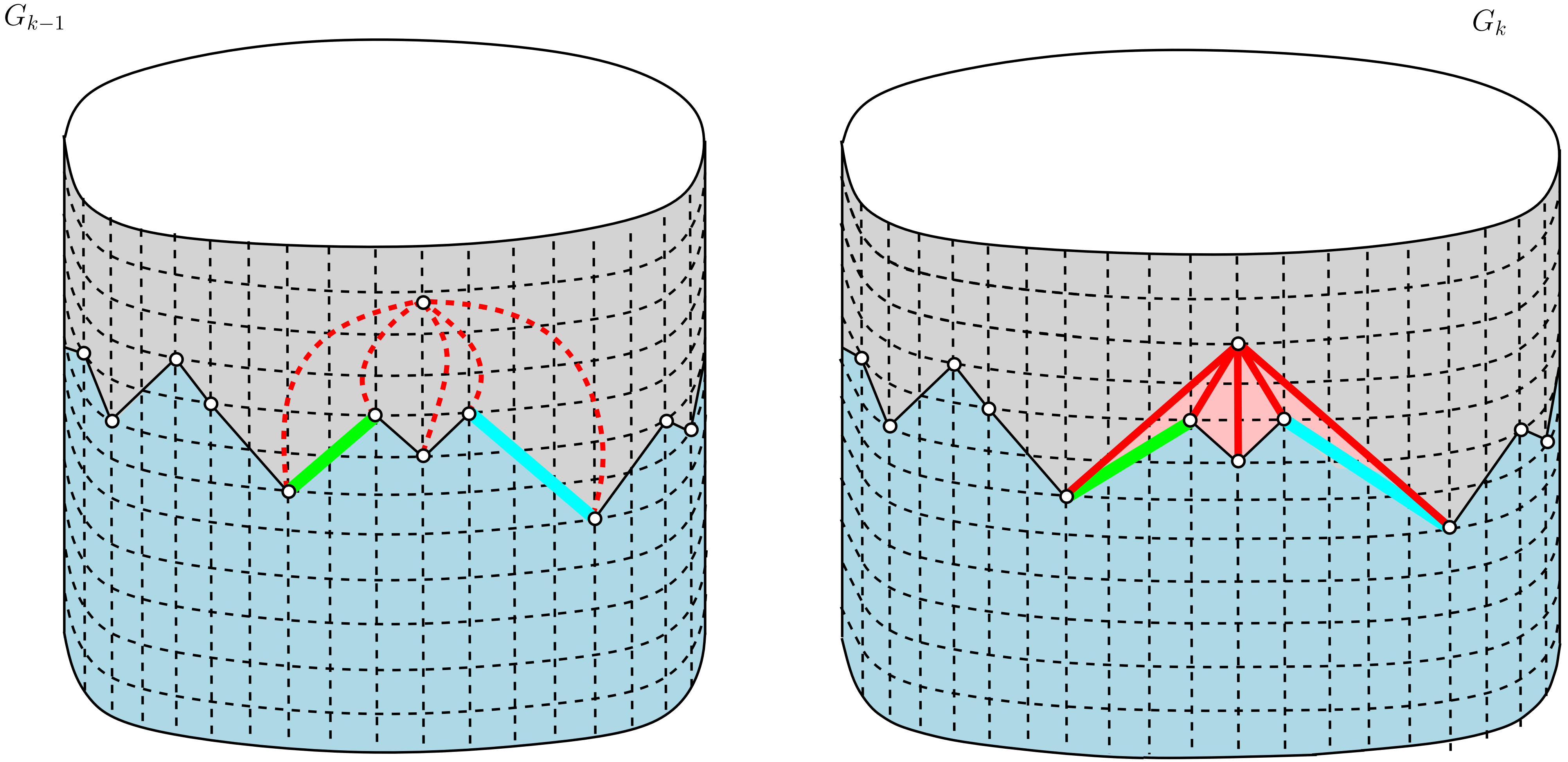
Extension to the cylinder: drawing algorithm



At each step: - insert two vertical strips of width 1

- insert the next vertex as in the planar case

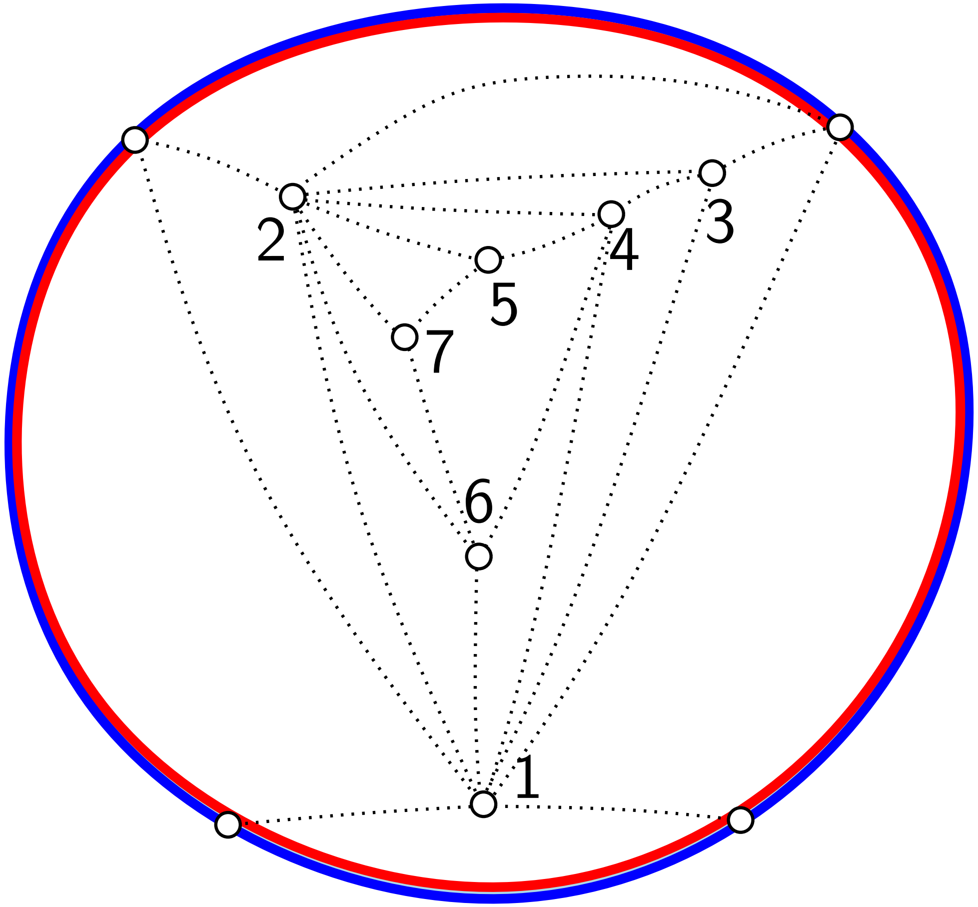
Extension to the cylinder: drawing algorithm

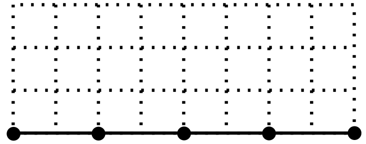


At each step: - insert two vertical strips of width 1

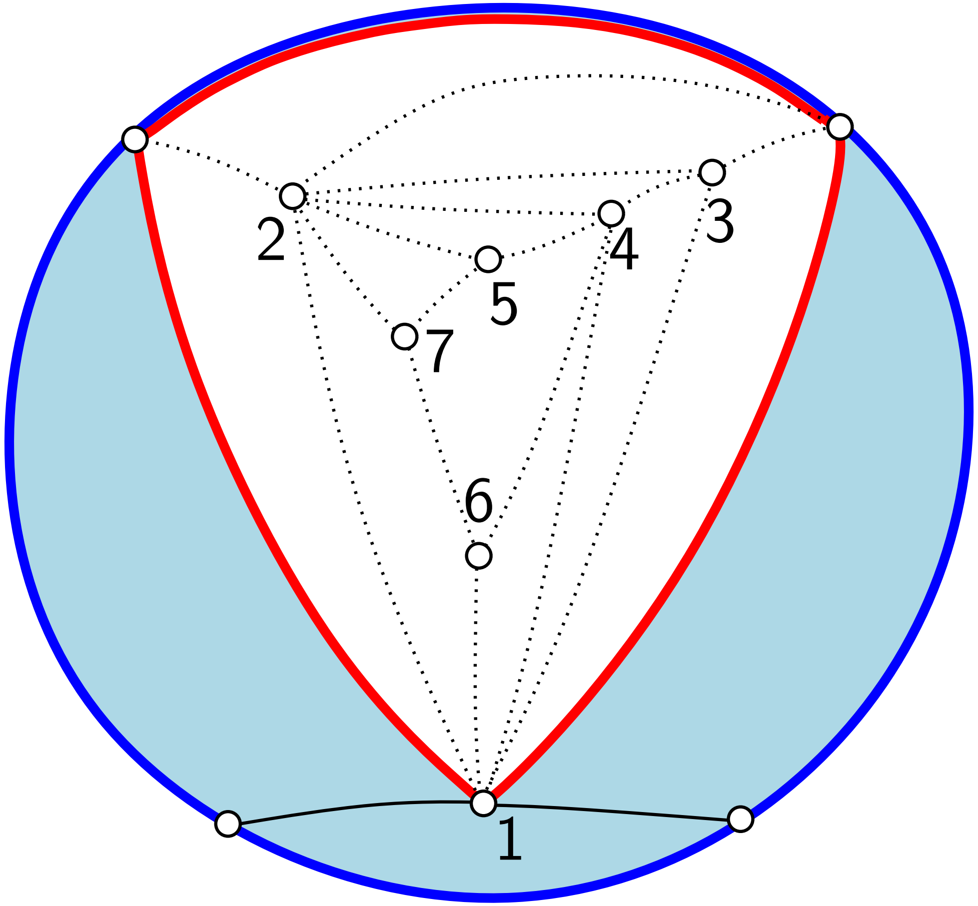
- insert the next vertex as in the planar case

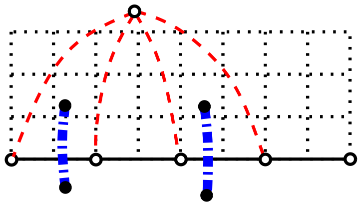
Extension to the cylinder: drawing algorithm

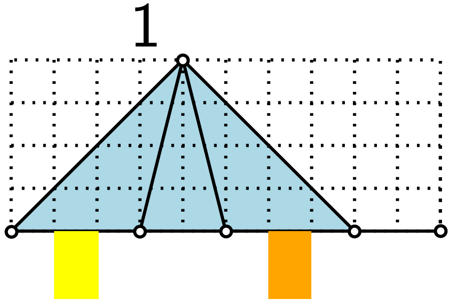




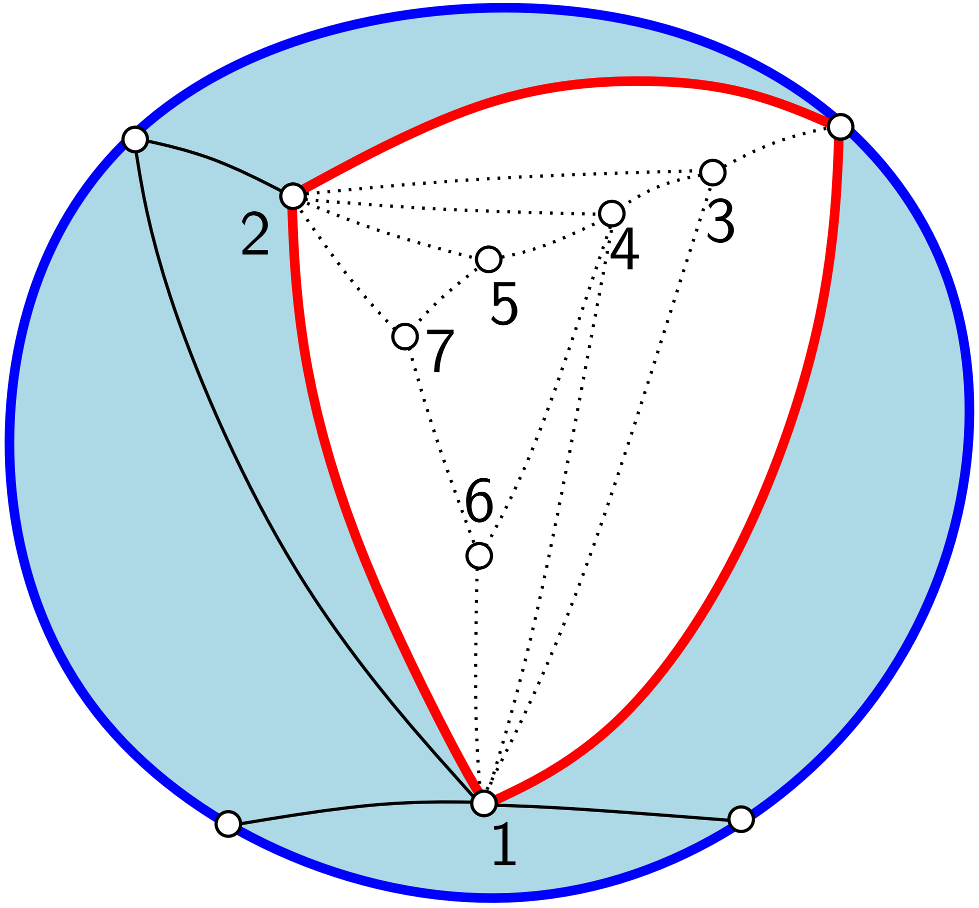
Extension to the cylinder: drawing algorithm

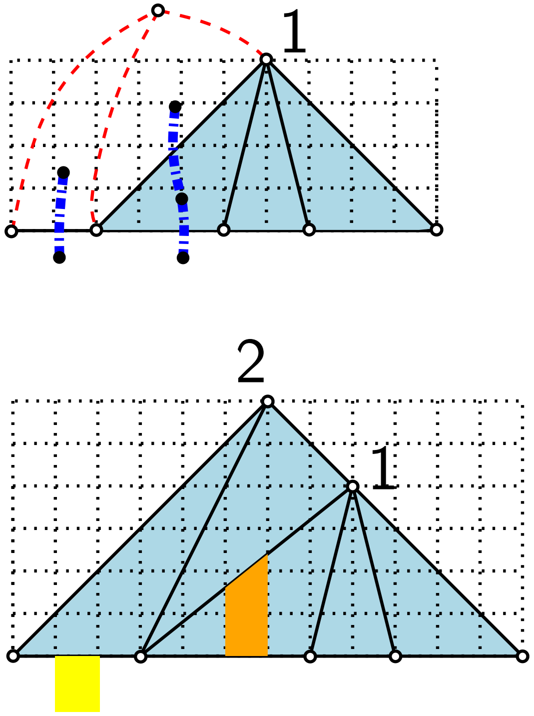




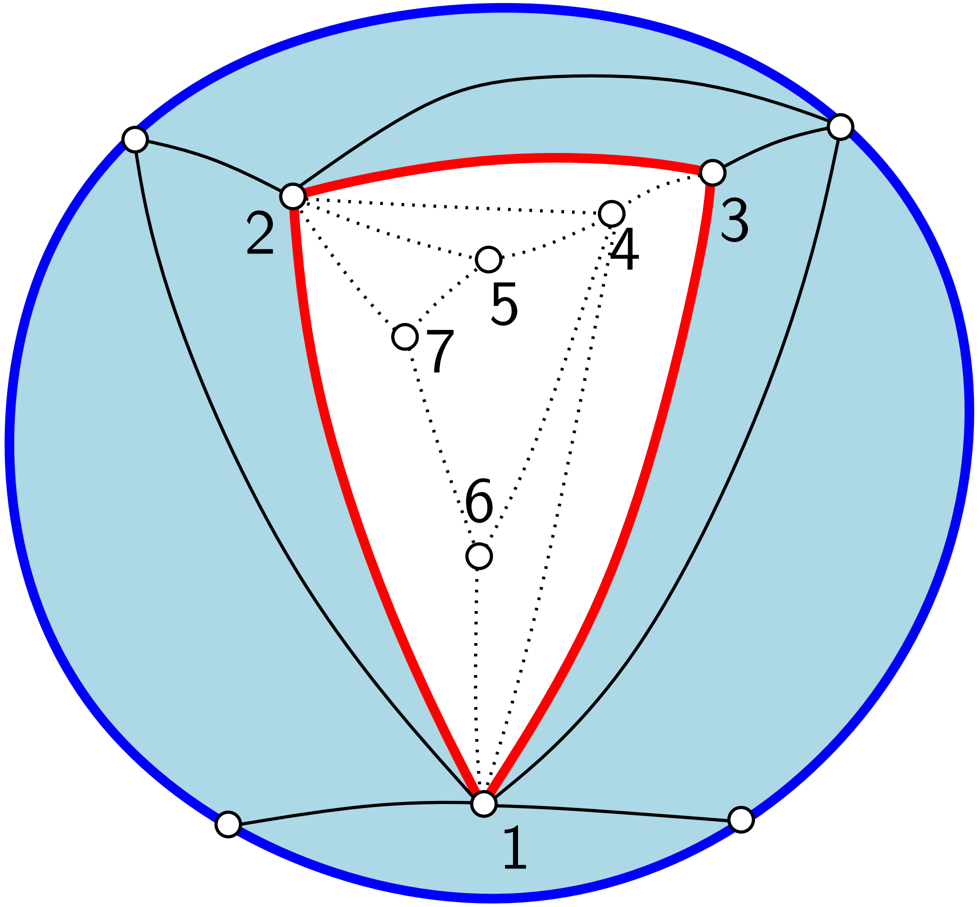


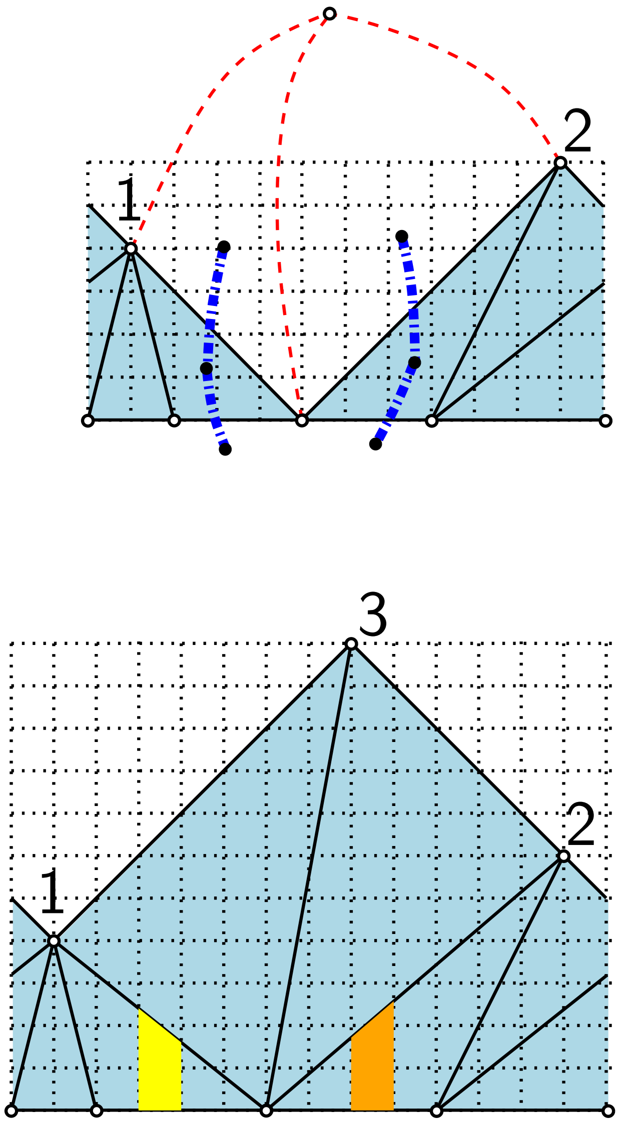
Extension to the cylinder: drawing algorithm



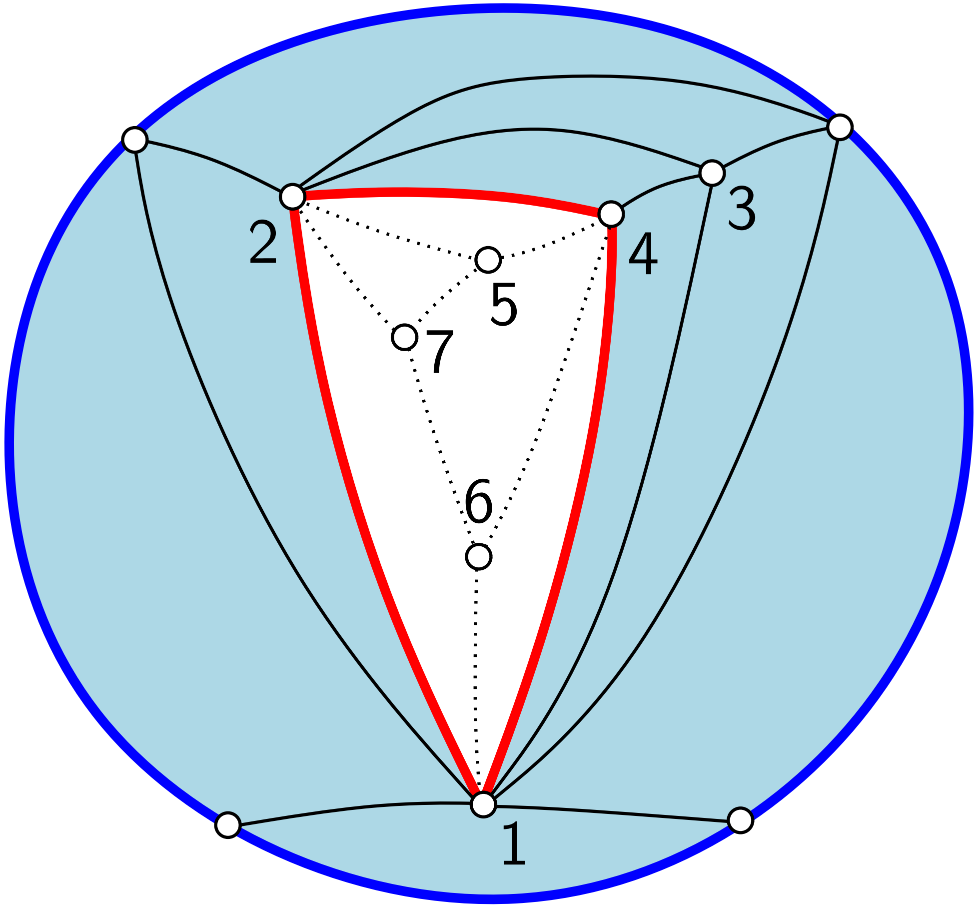


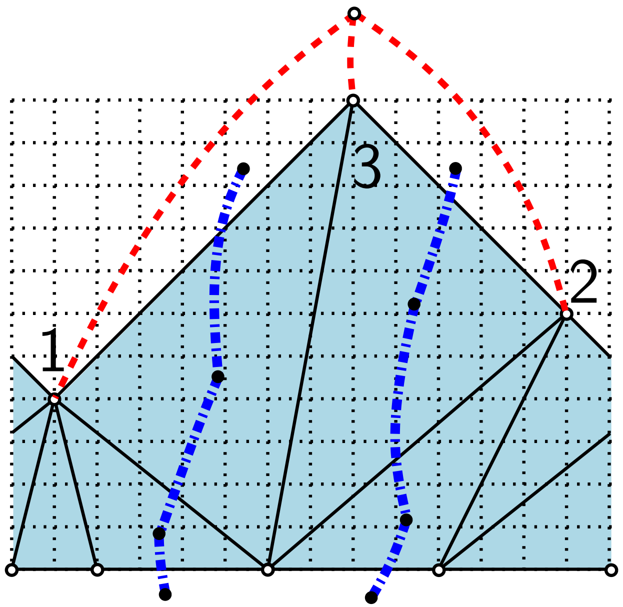
Extension to the cylinder: drawing algorithm

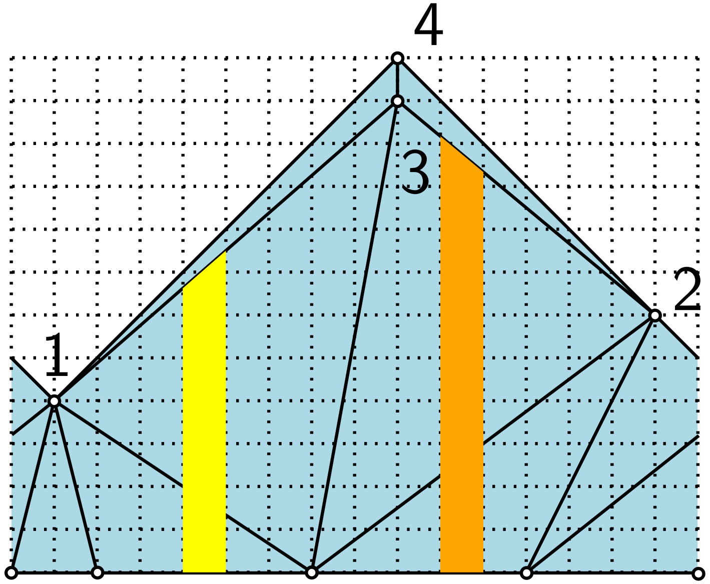




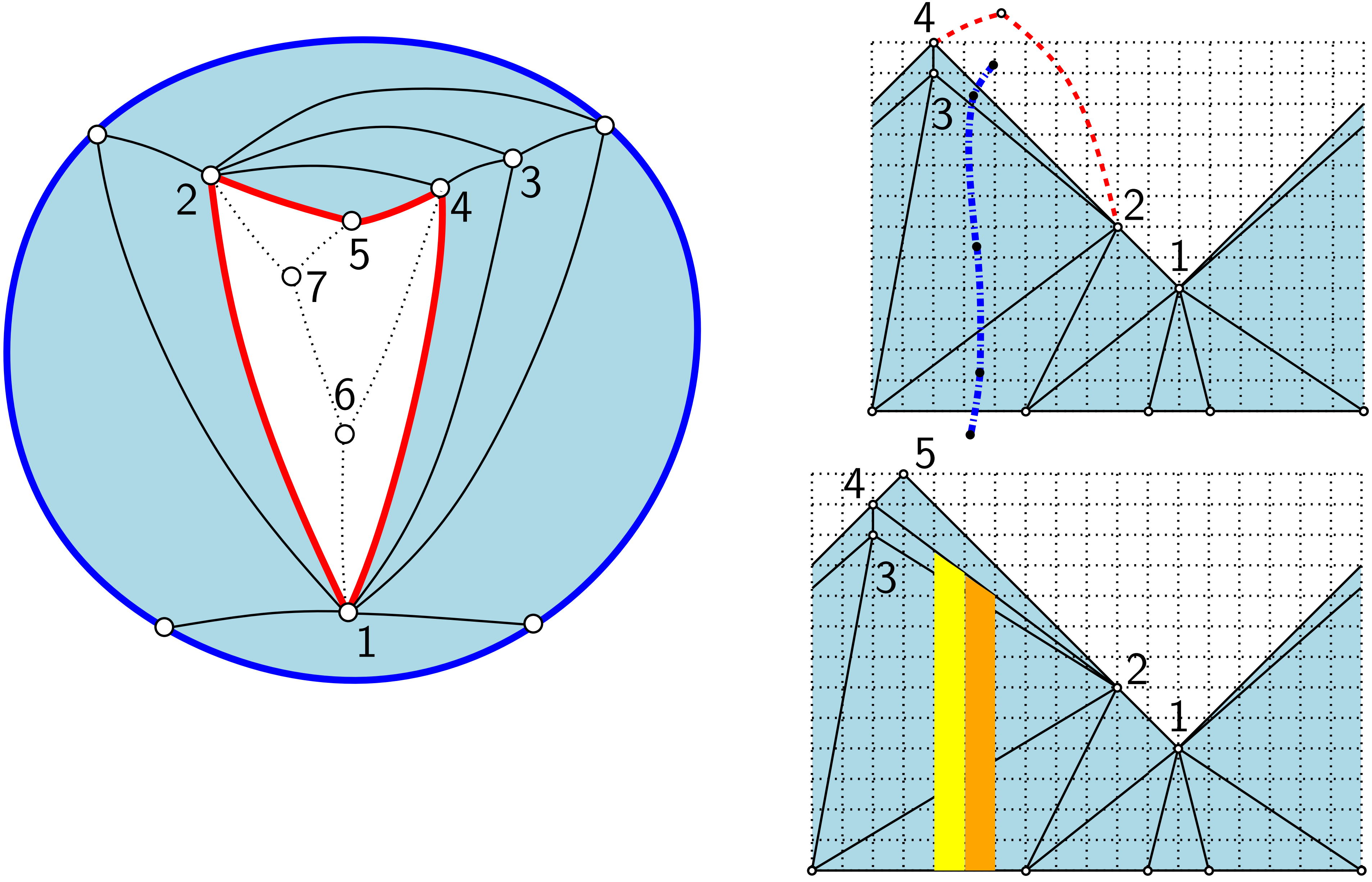
Extension to the cylinder: drawing algorithm



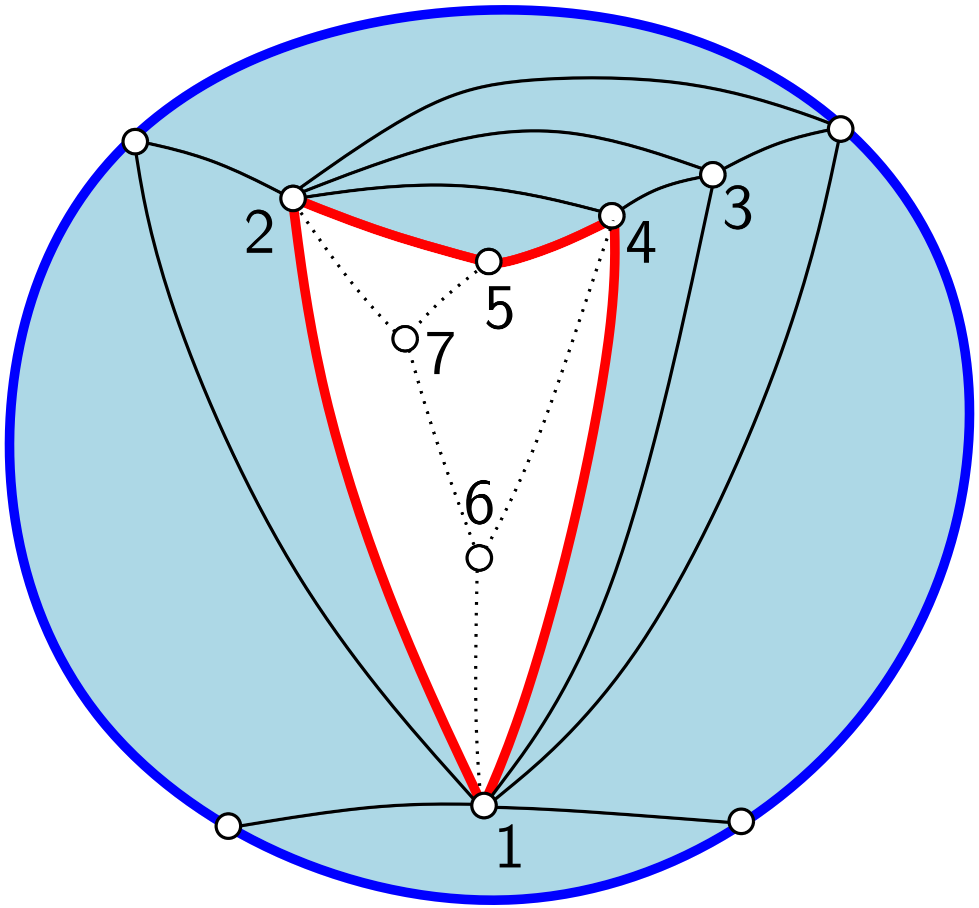


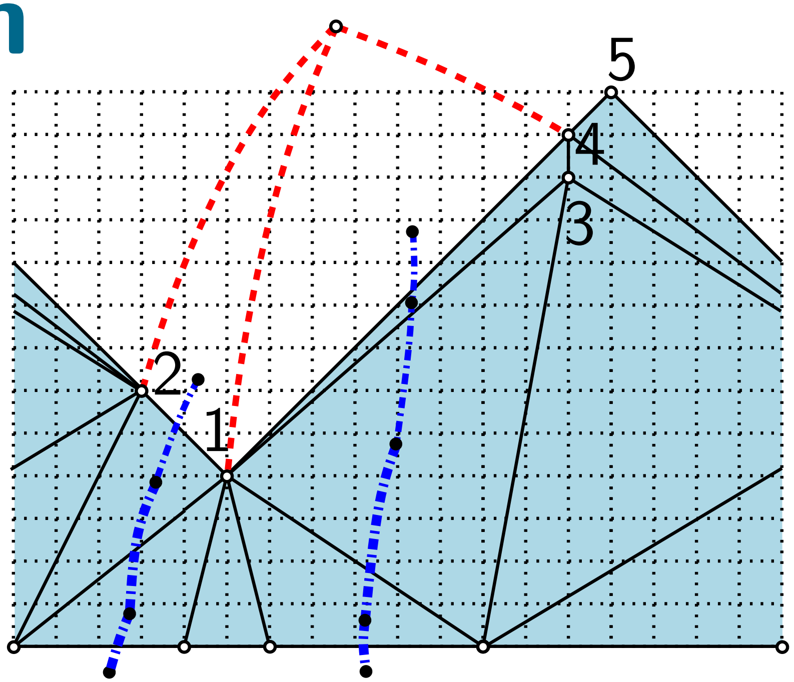


Extension to the cylinder: drawing algorithm

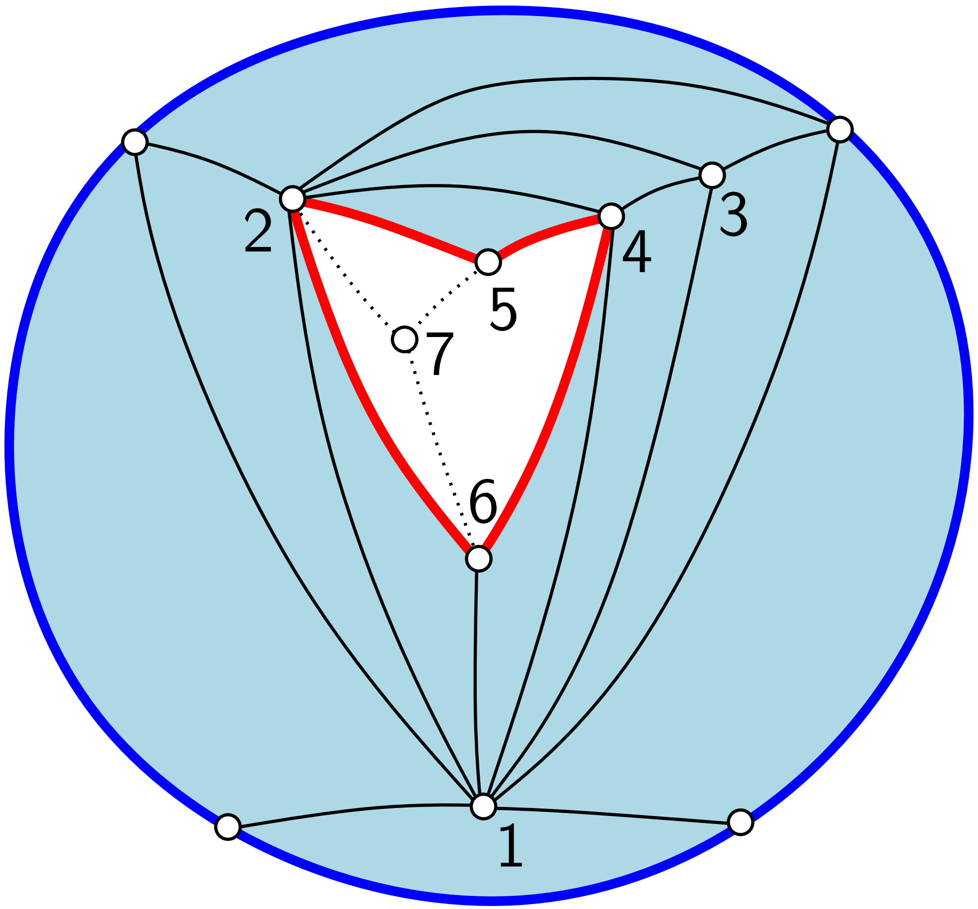


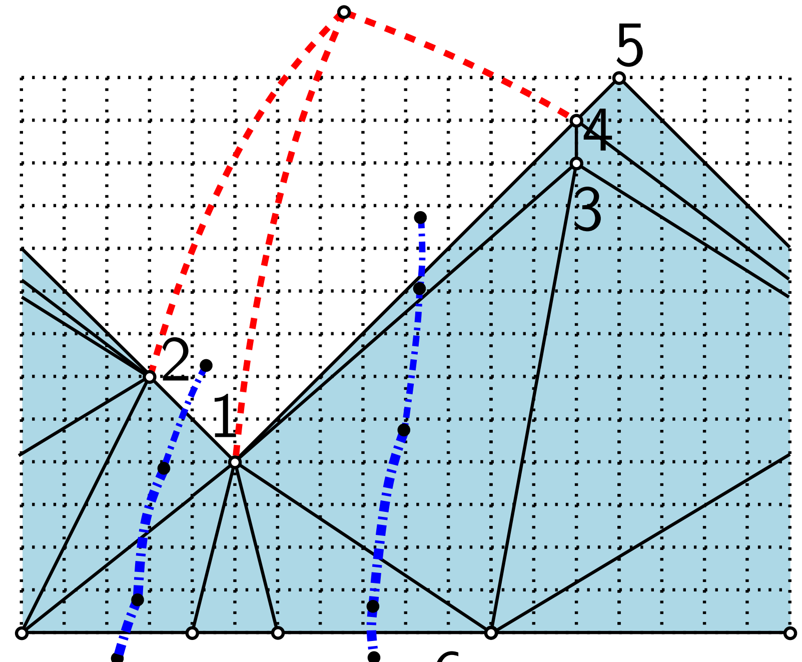
Extension to the cylinder: drawing algorithm

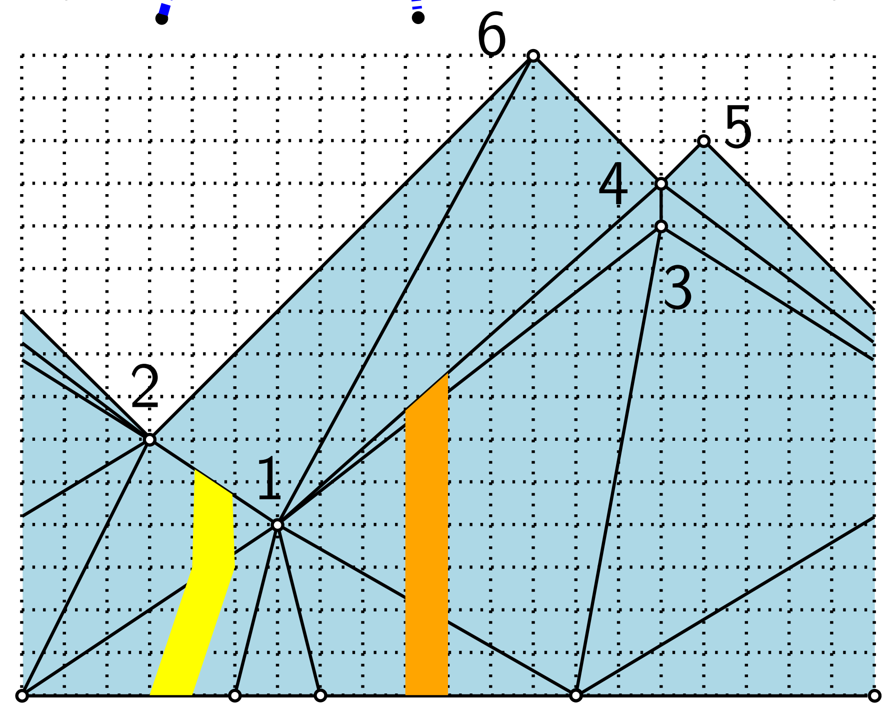




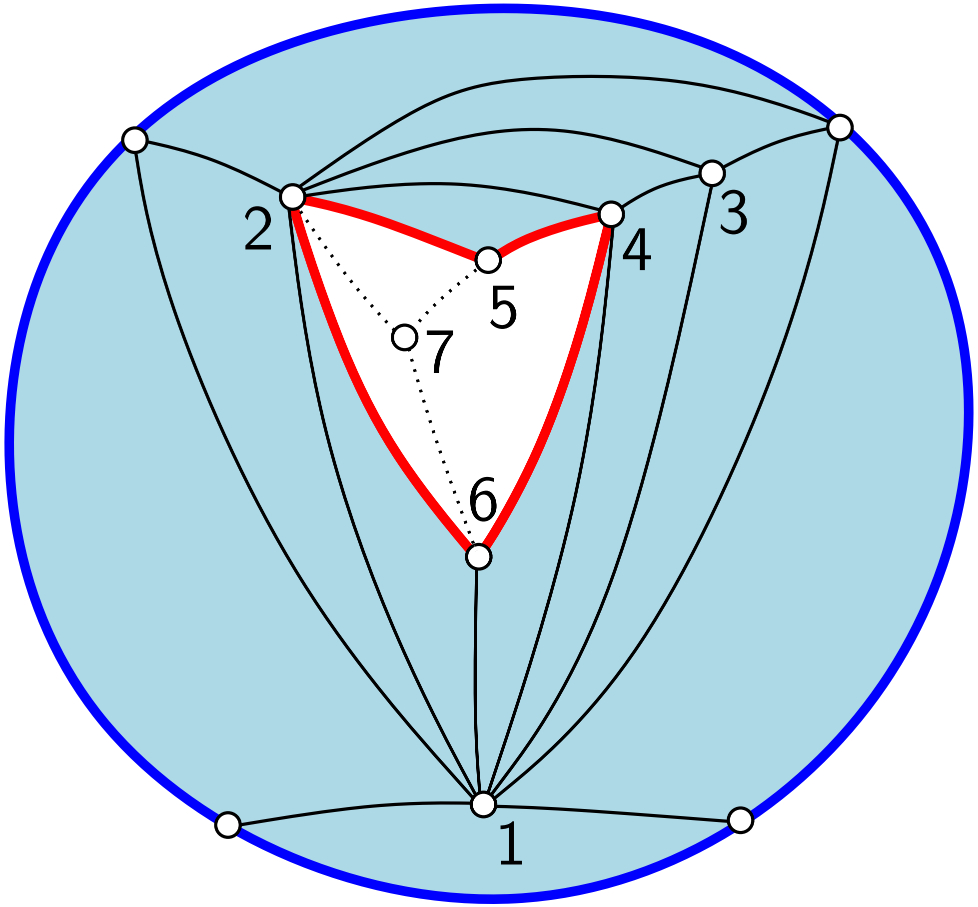
Extension to the cylinder: drawing algorithm

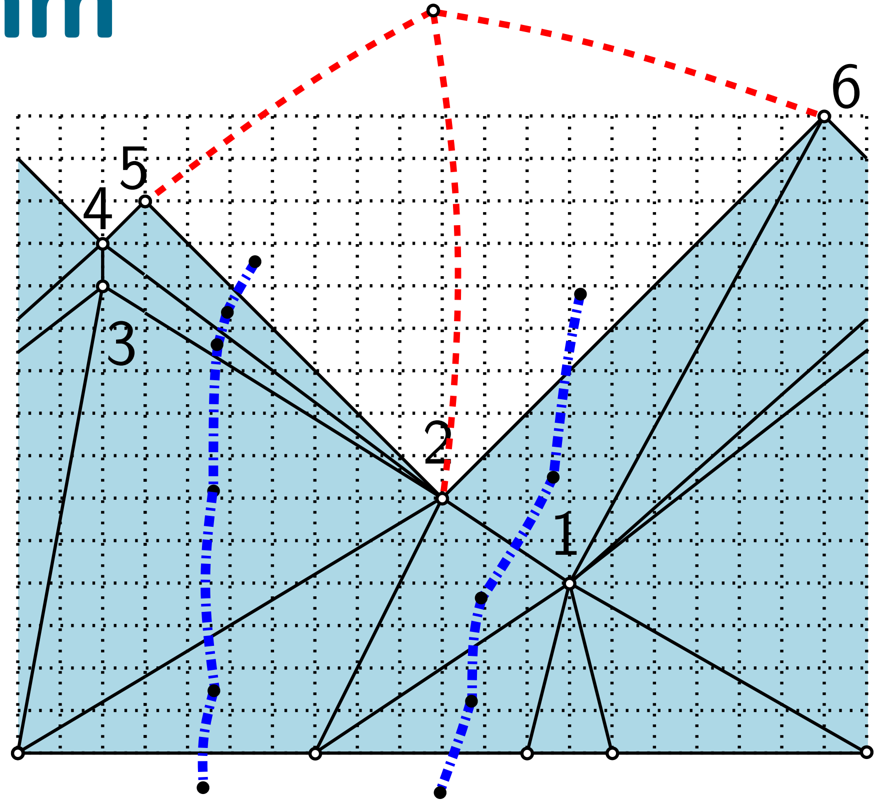




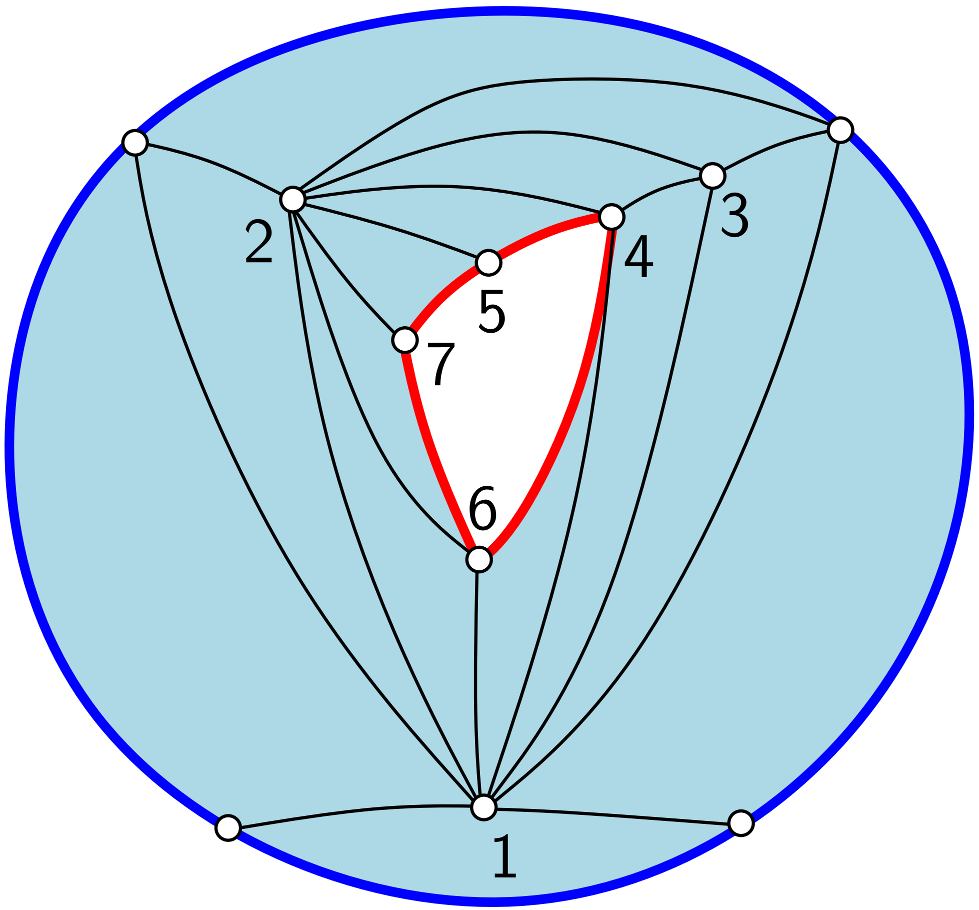


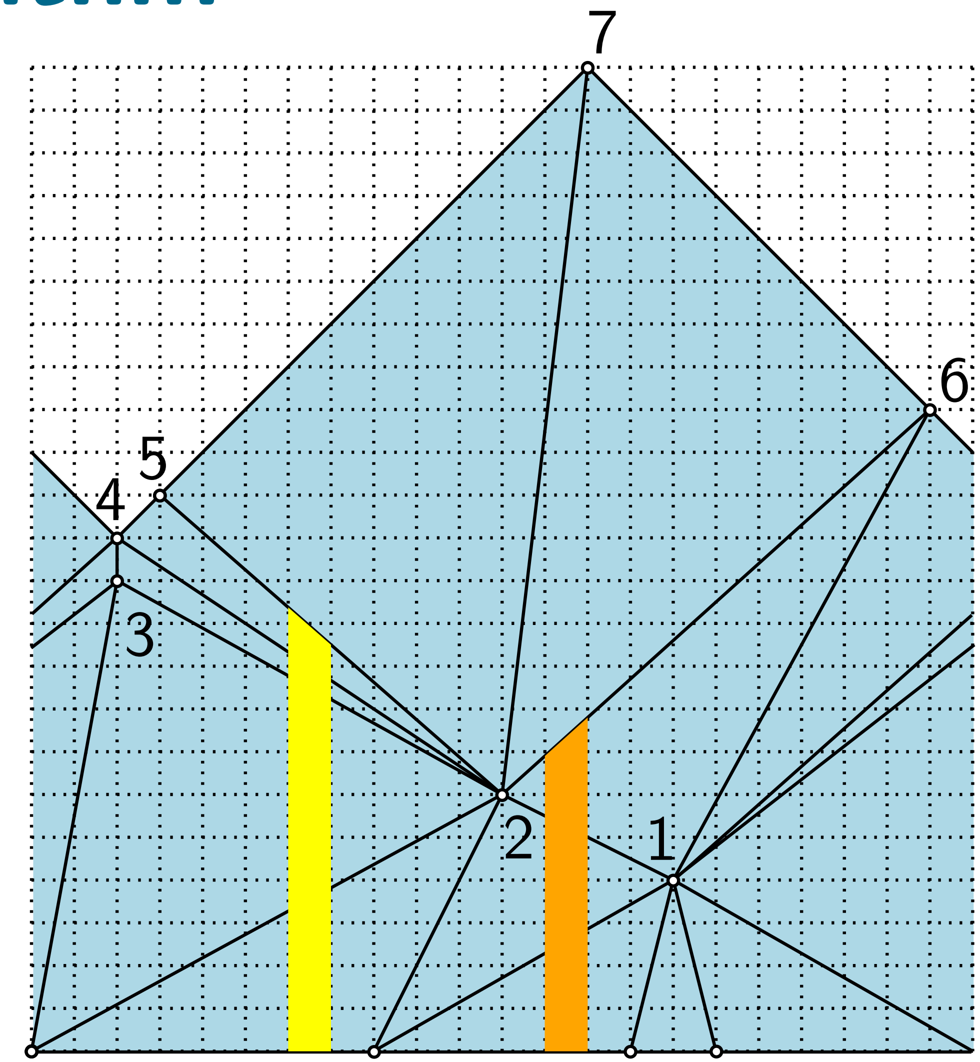
Extension to the cylinder: drawing algorithm





Extension to the cylinder: drawing algorithm

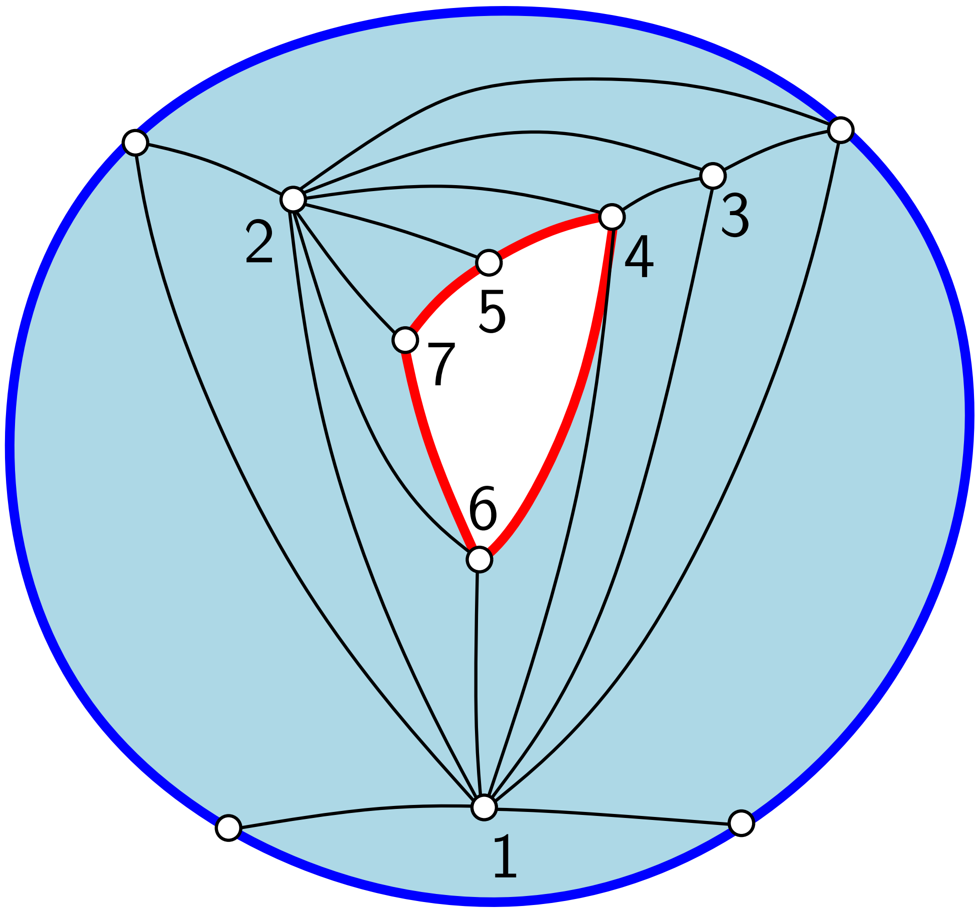


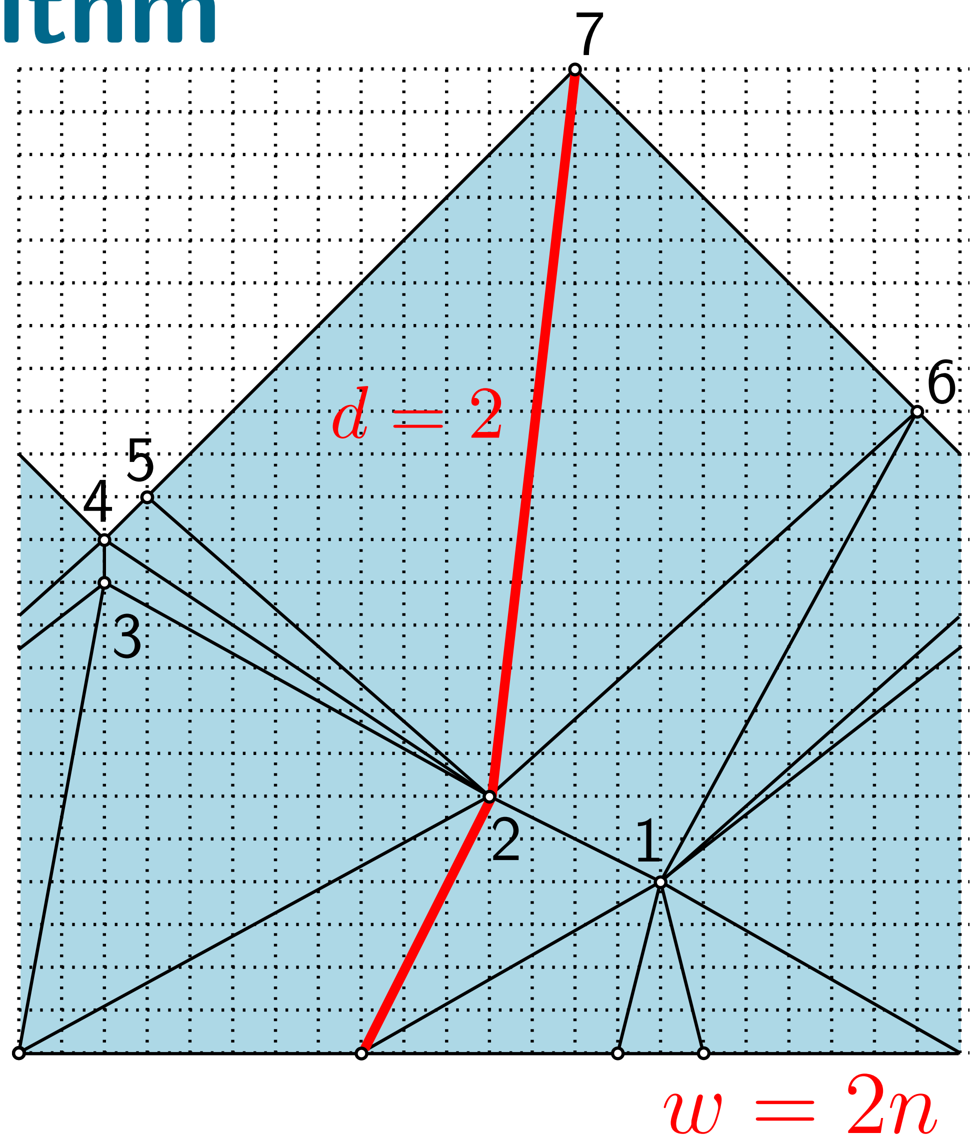


Width $= 2n$ Height $\leq n(n-3)/2$

Can also deal with chordal edges incident to outermost cycle

Extension to the cylinder: drawing algorithm





Each edge has vertical extension at most w

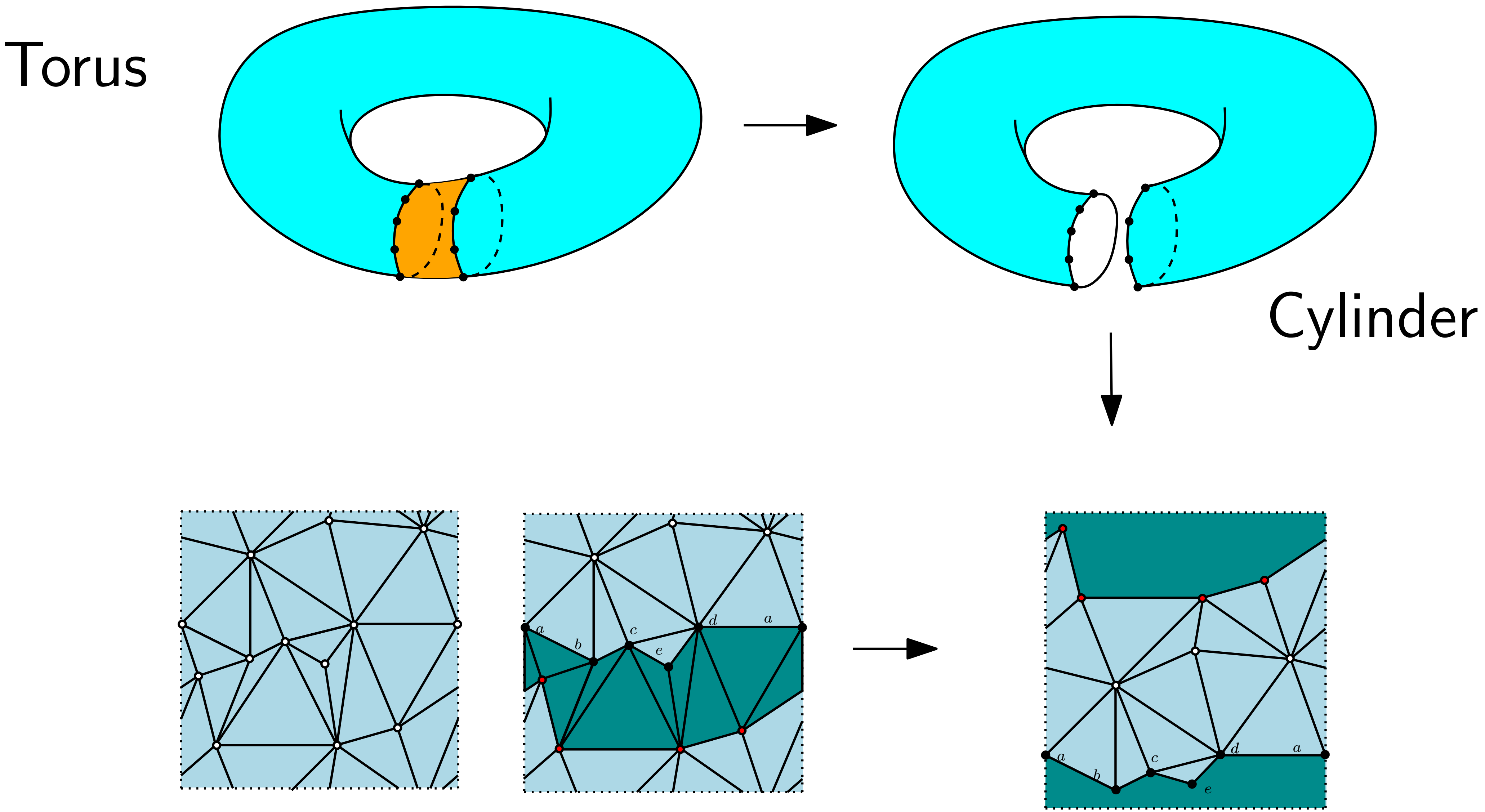
$\Rightarrow h \leq n(2d+1)$

with d the graph-distance between the two boundaries

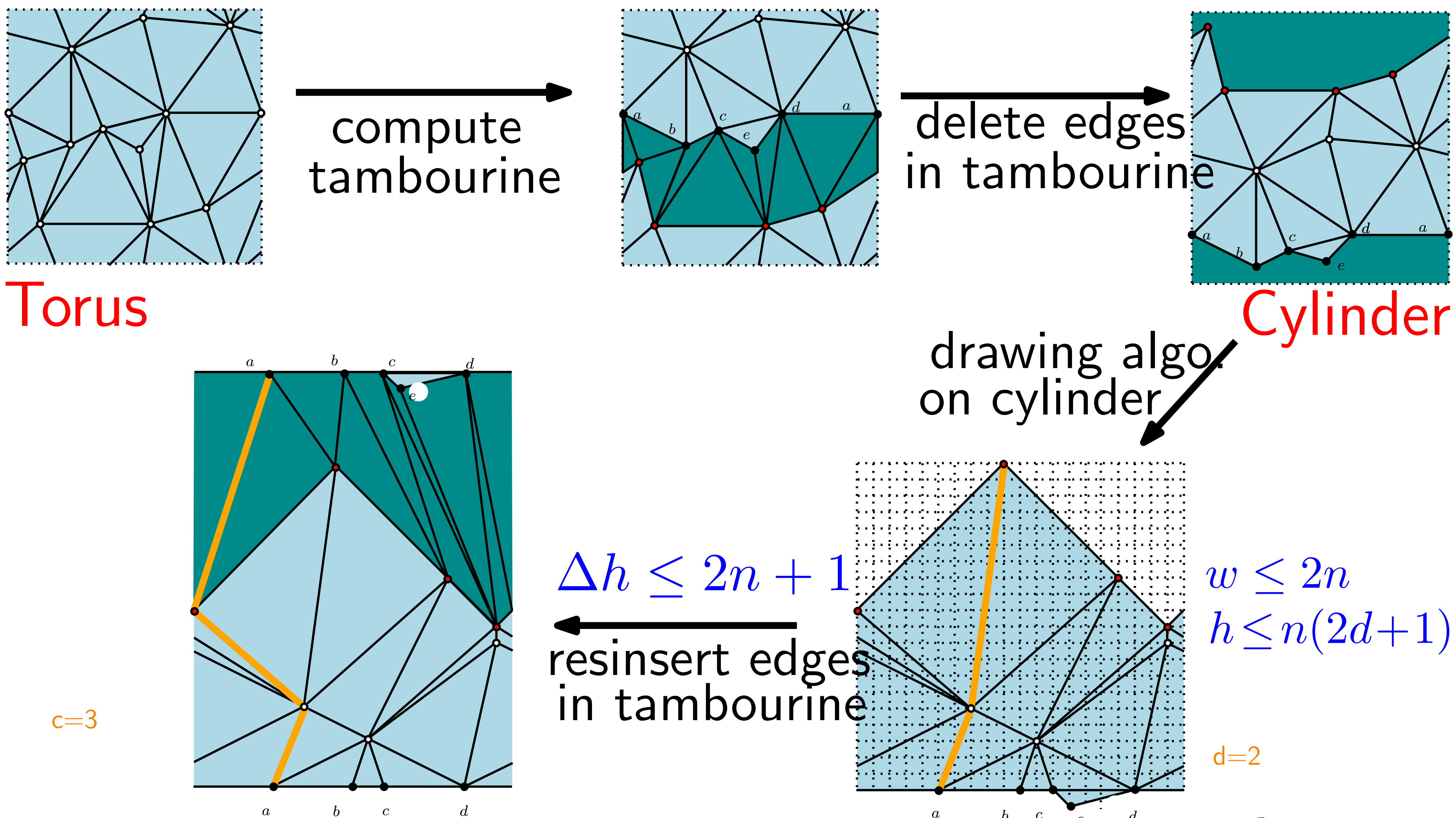
Getting toroidal drawings

Every toroidal triangulation admits a “tambourine”

[Bonichon, Gavoille, Labourel'06]



Getting toroidal drawings

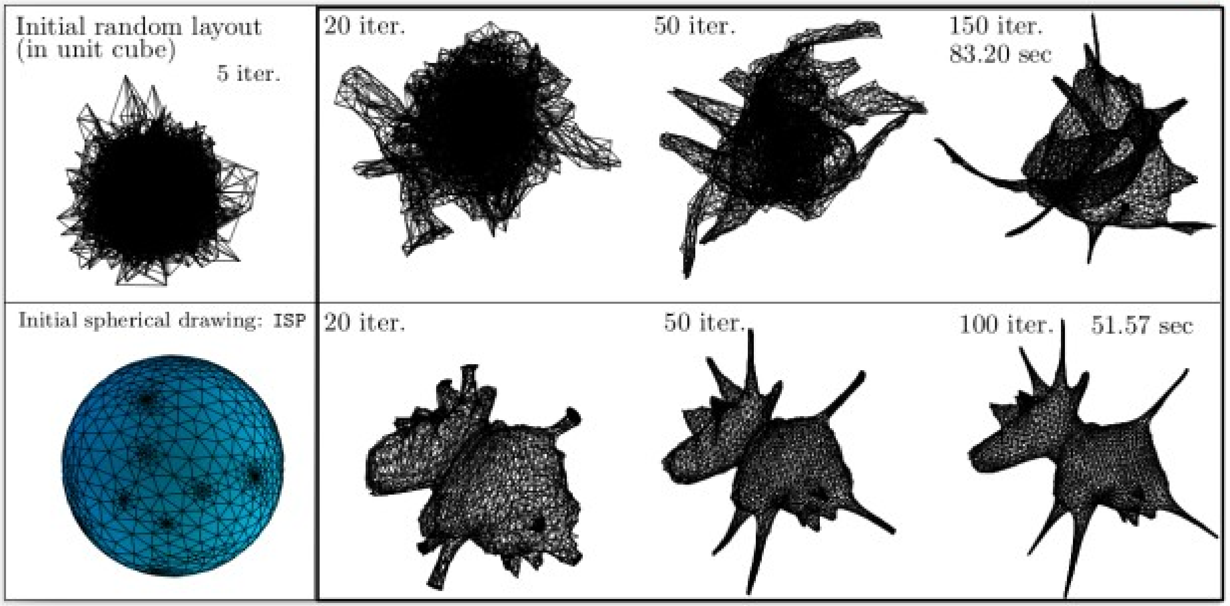


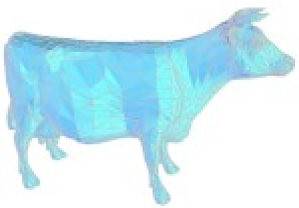
Let c =length shortest non-contractible cycle, $c \leq \sqrt{2n}$ **[Hutchinson, Albert'78]**

Can choose tambourine so that $d < c \Rightarrow h = O(n^{3/2})$

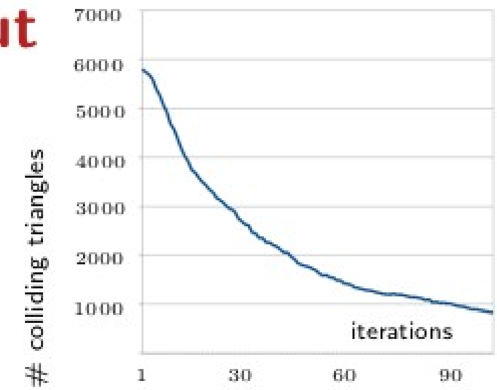
Spherical preprocessing for eulidean spring embedders

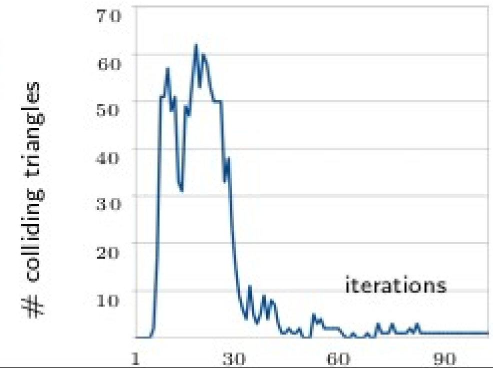
Use spherical drawings as initial layouts for 3D spring embedders: this allows us to better untangle the layout





count triangle collisions





Our Java implementation of the FR91 spring embedder (exact computation of repulsive forces)

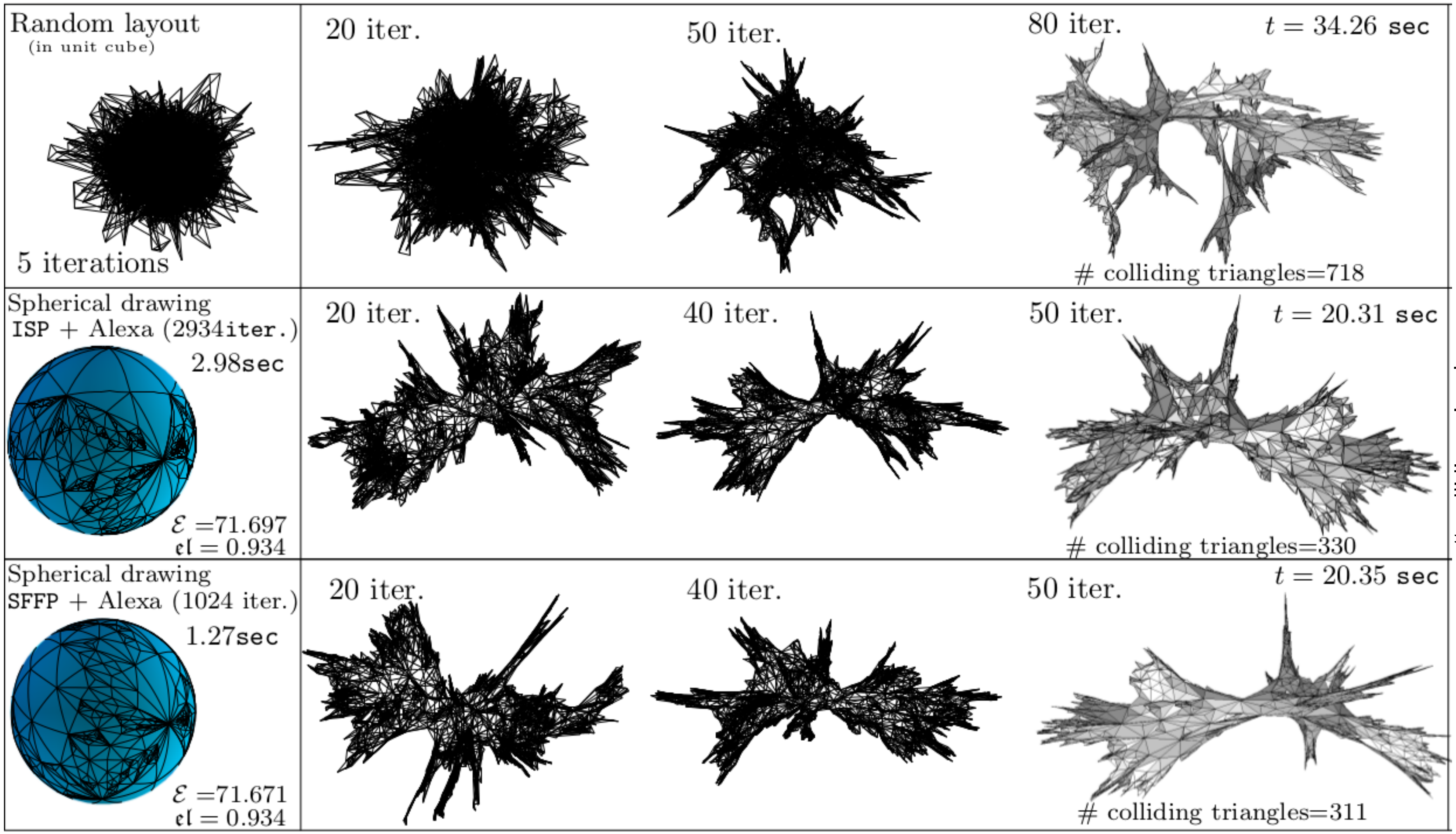
Random initial layout

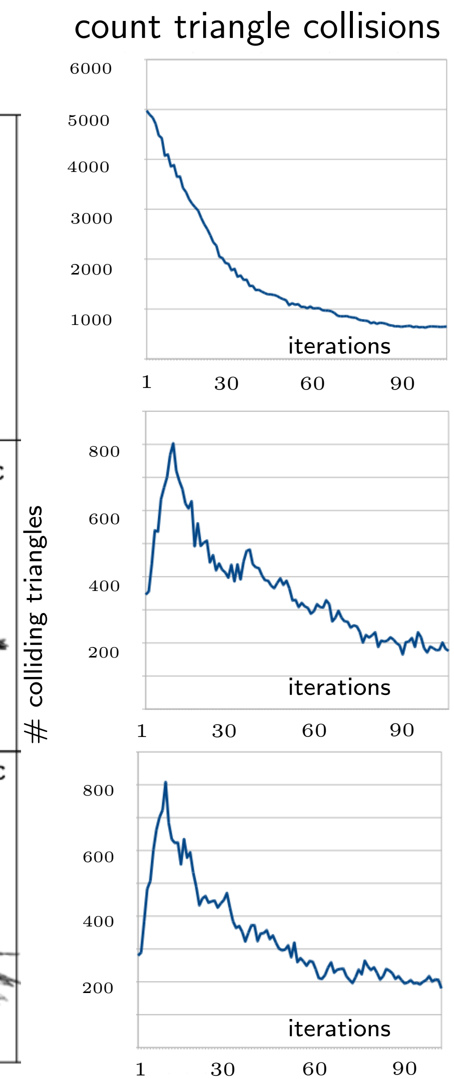
Initial spherical drawing SFPP

Spherical preprocessing for eulidean spring embedders

Use spherical drawings as initial layouts for 3D spring embedders: this allows us to better untangle the layout

`random planar triangulation`
with $5K$ triangles (generated with an uniform random sampler)





Layouts obtained with our Java implementation of the `FR91` spring embedder
(exact computation of repulsive forces)

Experimental results on balanced Schnyder woods

Looking for "nice" Schnyder woods

Counting Schnyder woods: (there are an exponential number)

[Bonichon '05]

\# Schnyder woods of triangulations of size n: $\approx 16^n$

\# planar triangulations of size n: $|\mathcal{T}_n| \approx 2^{3.2451}$

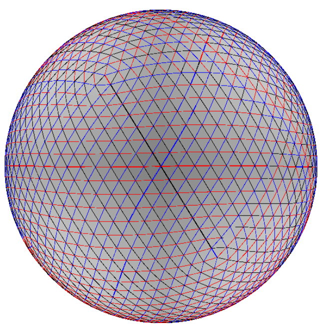
[Felsner Zickfeld '08]

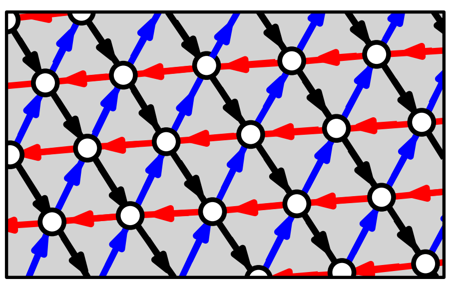
(count of Schnyder woods of a fixed triangulation)

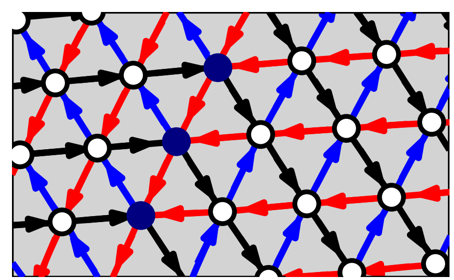
$$2.37^n \leq \max_{T \in \mathcal{T}_n} |SW(T)| \leq 3.56^n$$

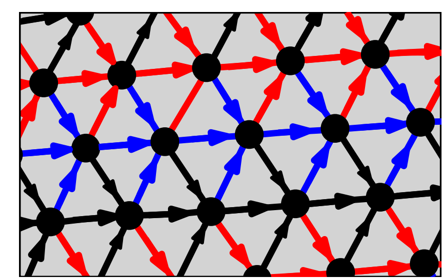
$\mathcal{T}_n :=$ class of planar triangulations of size n

$SW(T) :=$ set of all Schnyder woods of the triangulation T

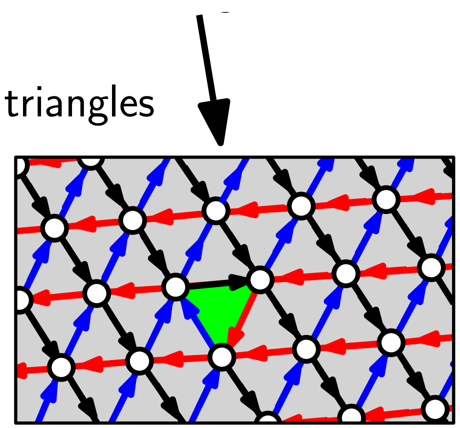




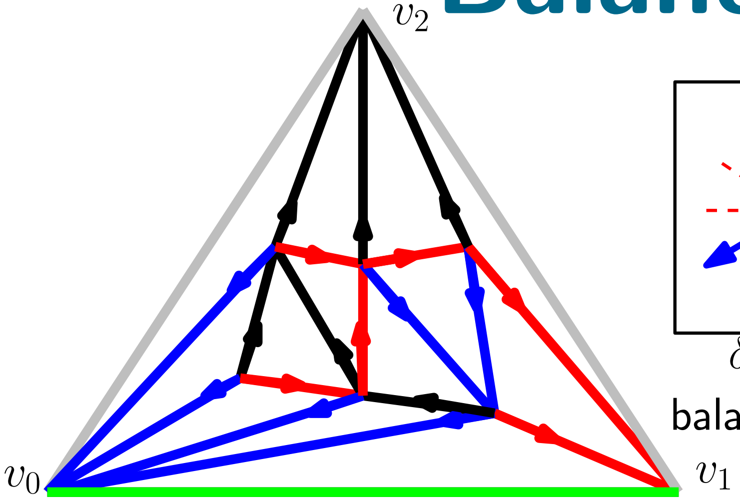


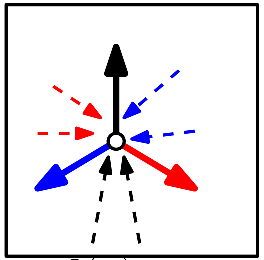


reversal of oriented triangles



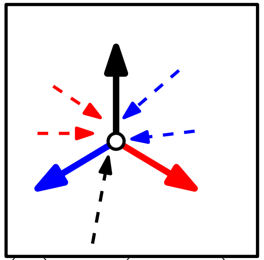
Balanced Schnyder woods





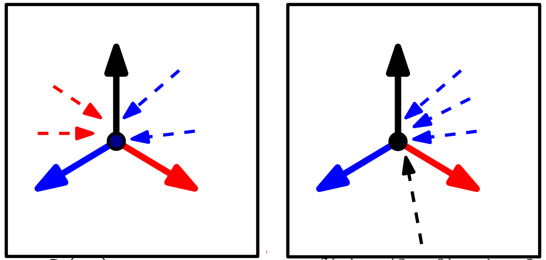
$\delta(v) = 0$

balanced vertex



$\delta(v) = (2-1)-1 = 0$

balanced vertex



$\delta(v) = 1$ $\delta(v) = (3-0)-1 = 2$

unbalanced vertices

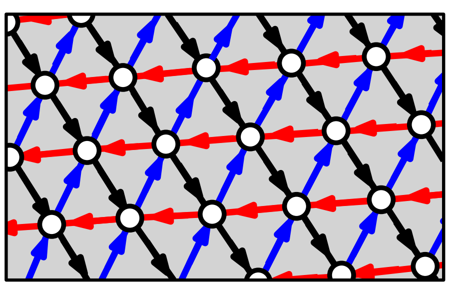
Def

$$\delta(v) := \begin{cases} \max_{i \in \{0,1,2\}} indeg_i(v) - \min_{i \in \{0,1,2\}} indeg_i(v) & \text{if } degree(v) = 3k \\ \max_{i \in \{0,1,2\}} indeg_i(v) - \min_{i \in \{0,1,2\}} indeg_i(v) - 1 & \text{otherwwise} \end{cases}$$

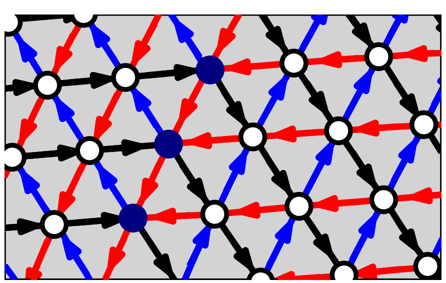
vertex **defect**

$indeg_i(v) :=$ #incoming edges of color i

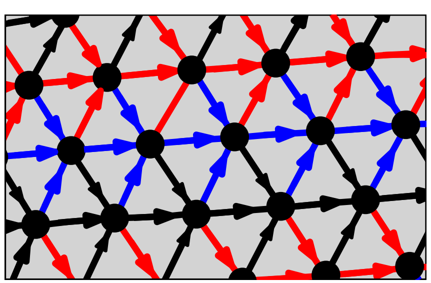
A Schnyder wood is **balanced** if most vertices have a small **defect**



perfectly balanced



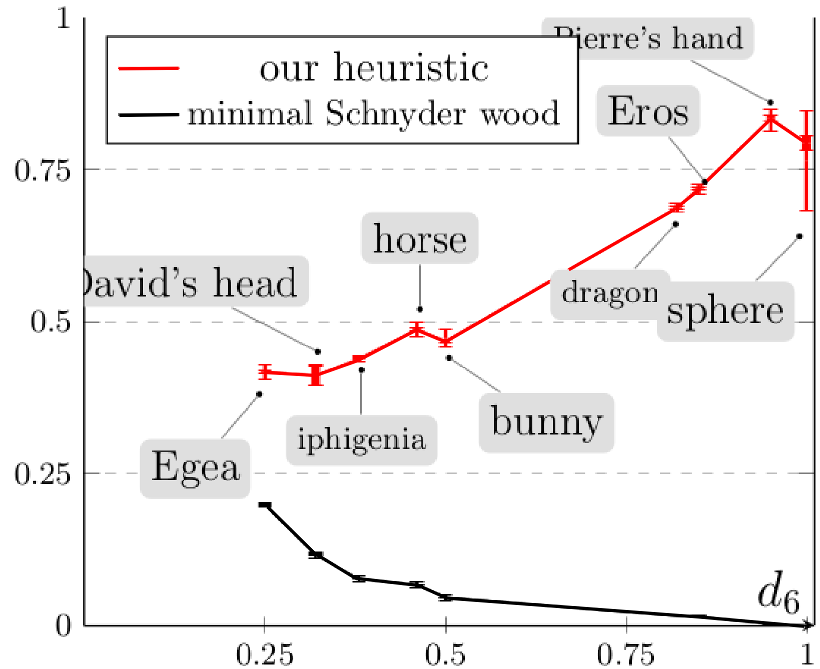
well balanced



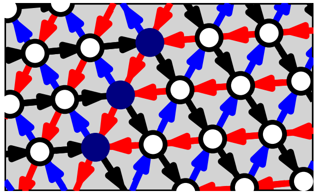
strongly unbalanced

Computing balanced Schnyder woods

Proportion of balanced vertices

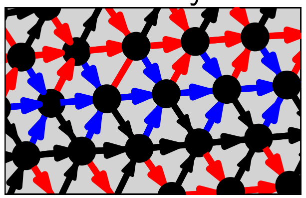


with our heuristic



well balanced

minimal Schnyder wood



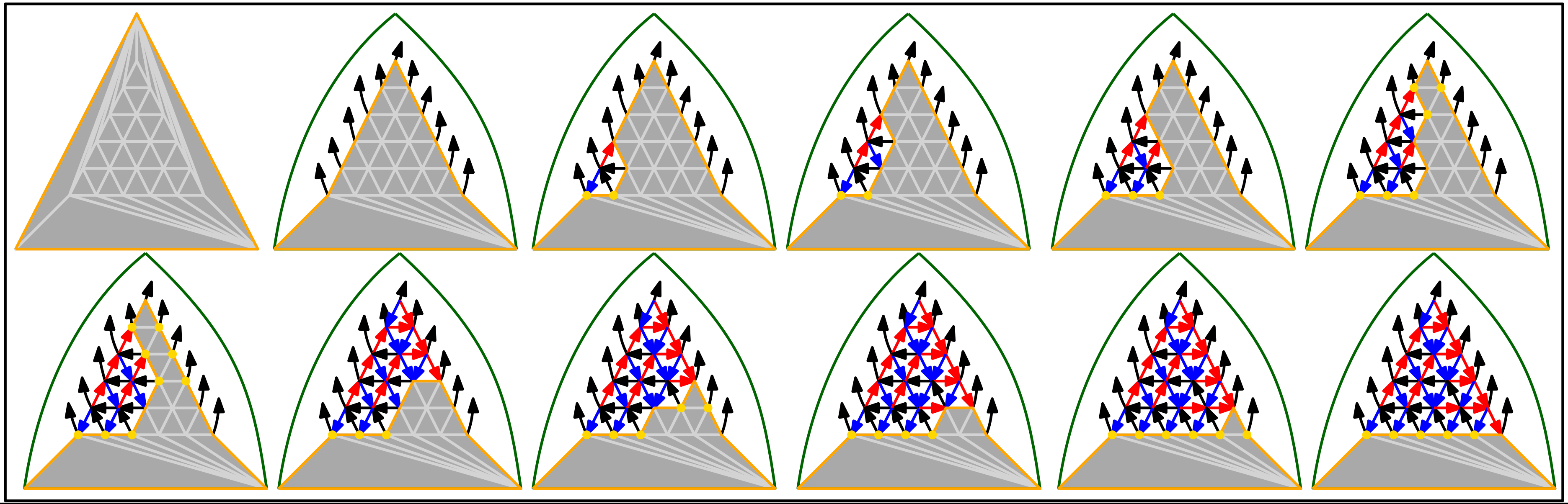
strongly unbalanced

(d_6 := proportion of degree 6 vertices)

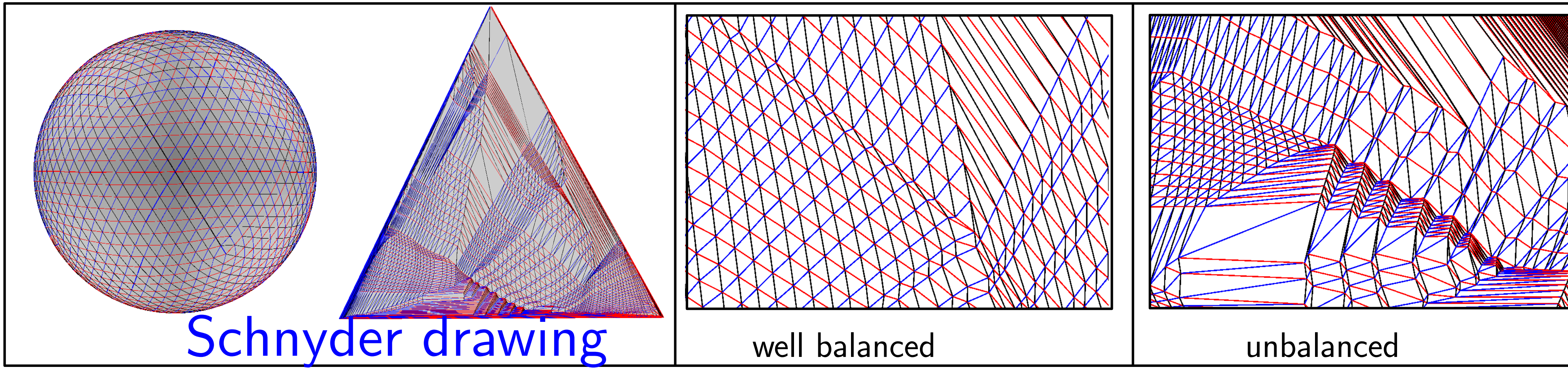
```
balancedSchnyderWood(T, (v0, v1, v2), k)
B = {v0, v1, v2} // initialization
T = new int[n] // priority array
Q0 = ∅, Q1 = ∅, ... Q_{k-1} = ∅ // queue initialization
Q0.addLast(v2)
while(|B| ≠ {v0, v1}) {
  let M be the largest index s.t. Q_M ≠ ∅
  let v = Q_M.poll()
  if(v ∈ B and v is free) {
      let {v_l, v_{j1}, ... v_{jt}, v_r} be the neighbors of v on B
      colorOrient(v)
      conquer(v) // remove v from B
      T[v_l]++, T[v_r]++ // increase priority
      Q_{max(k-1,T[v_l])}.addLast(v_l)
      Q_{max(k-1,T[v_r])}.addLast(v_r)
      Q0.addLast(v_{j1}), ..., Q0.addLast(v_{jt})
  }
}
```

Incremental vertex shelling (Brehm's diploma thesis)

priority driven vertex conquest: remove first boundary vertices with higher number of ingoing edges

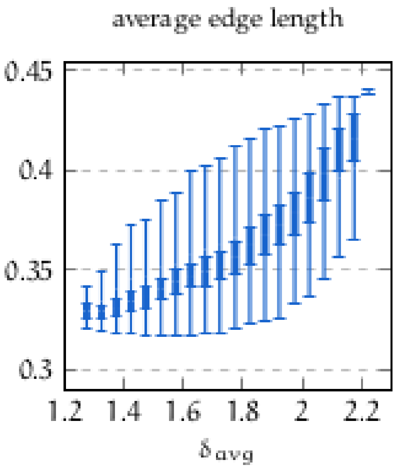


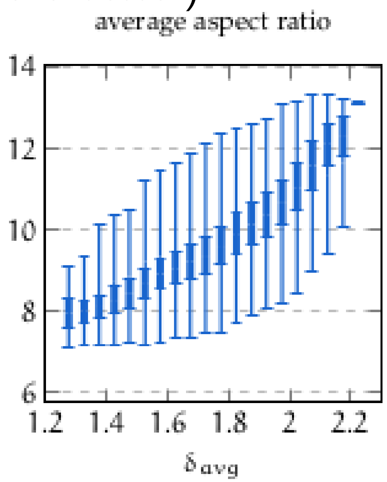
Layout quality for Schnyder drawings



Layout quality for Schnyder drawings

(lower values are better)

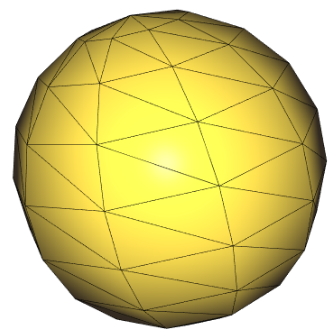




Evaluate layout statistics for all distinct Schnyder woods of a given graph

plot layout statistics as a function of average defect

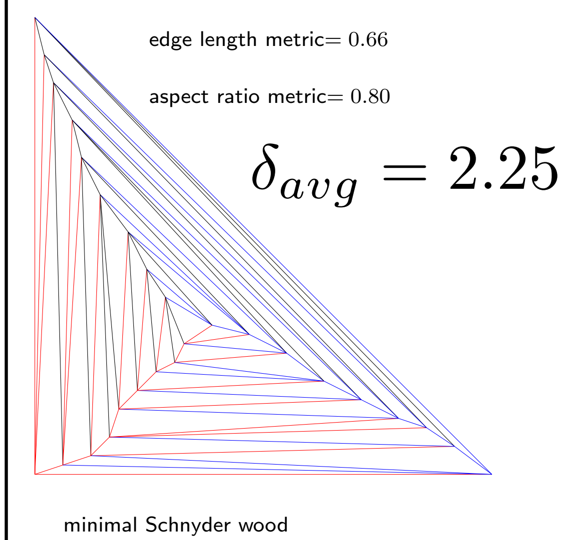
$\delta_{avg} := \frac{1}{n} \sum_v \delta(v)$ (average vertex defect)

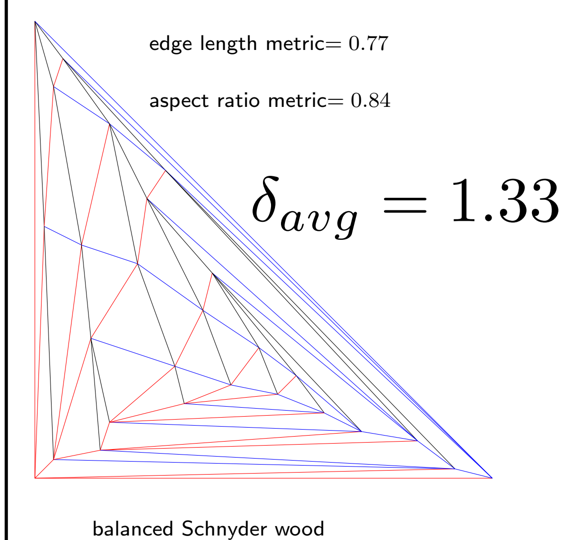


`globe` (regular graph)

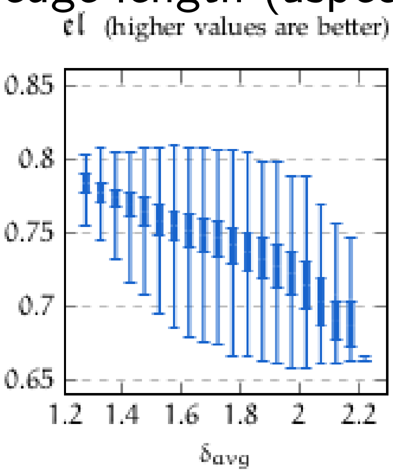
$n = 27$
$d_6 = 0.55$
$d_{max} = 6$
$|\mathcal{S}| = 5\,084\,208$

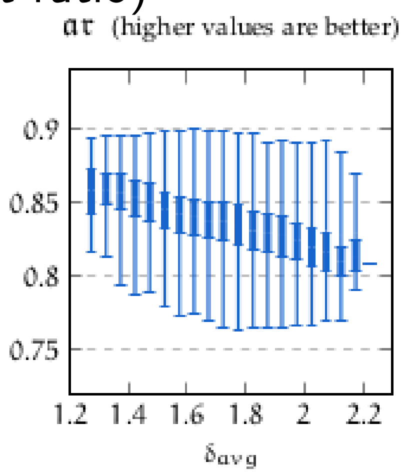
distinct Schnyder woods





high values indicates more uniform edge length (aspect ratio)





average percent deviation of edge length

$$\mathfrak{el} := 1 - \left(\frac{1}{|E|} \sum_{e \in E} \frac{|l(e) - l_{avg}|}{\max(l_{avg}, l_{max} - l_{avg})} \right)$$

(Fowler and Kobourov, 2012)

$l(e) :=$ edge length of e

From Schnyder woods to cycle separators

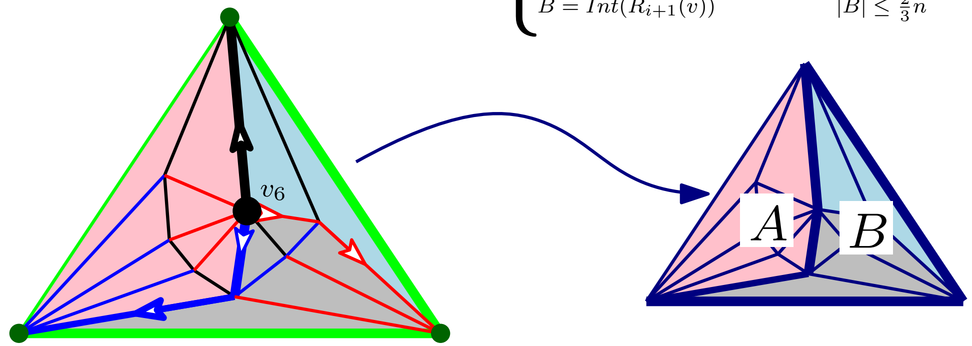
(Fox-Epstein et al. 2016, Holzer et al. 2009)

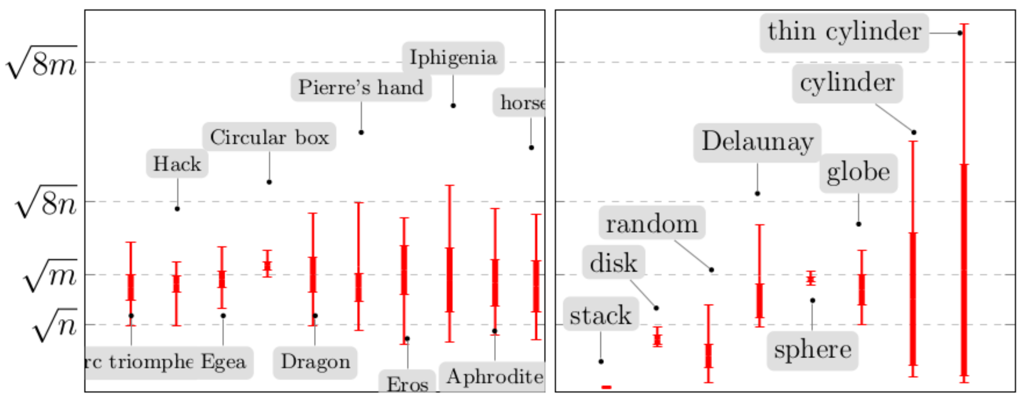
Def (small balanced cycle separators)

A partition (A, B, S) of $V(G)$ such that:

- S defines a simple cycle
- A and B are balanced: $|A| \leq \frac{2}{3}n$, $|B| \leq \frac{2}{3}n$
- the separator is small: $|S| \leq \sqrt{8m}$

choose the best index i and vertex v s.t. $\begin{cases} S = P_i(v) \cup P_{i+2}(v) \cup \{v\} \text{ is minimized} \\ A = Int(R_i(v) \cup R_{i+2}(v)) \quad |A| \leq \frac{2}{3}n \\ B = Int(R_{i+1}(v)) \quad |B| \leq \frac{2}{3}n \end{cases}$

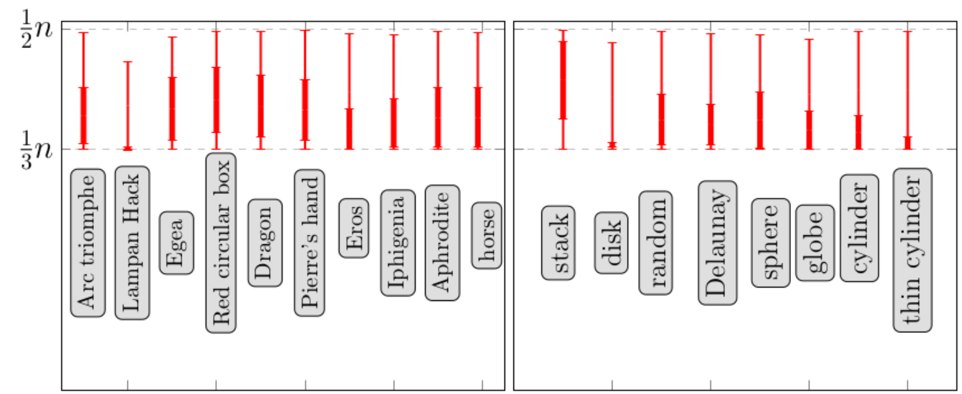




n = number of vertices

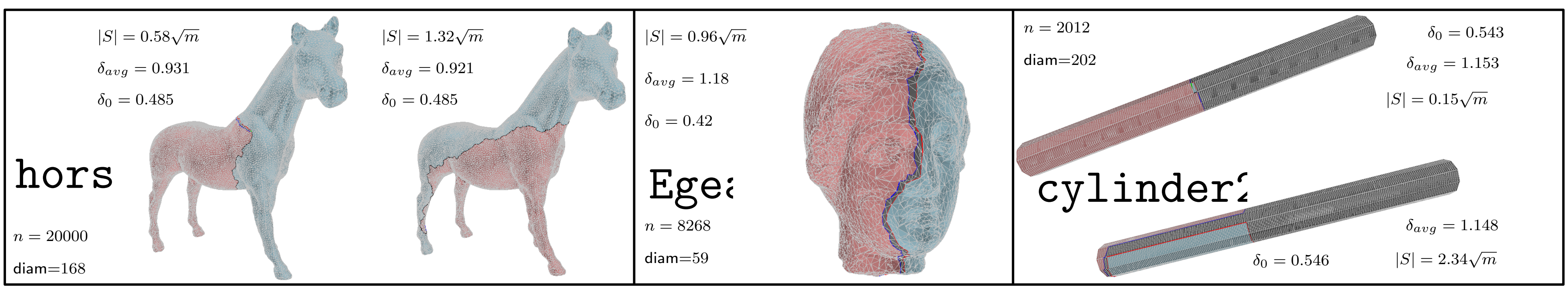
m = number of edges

Boundary size



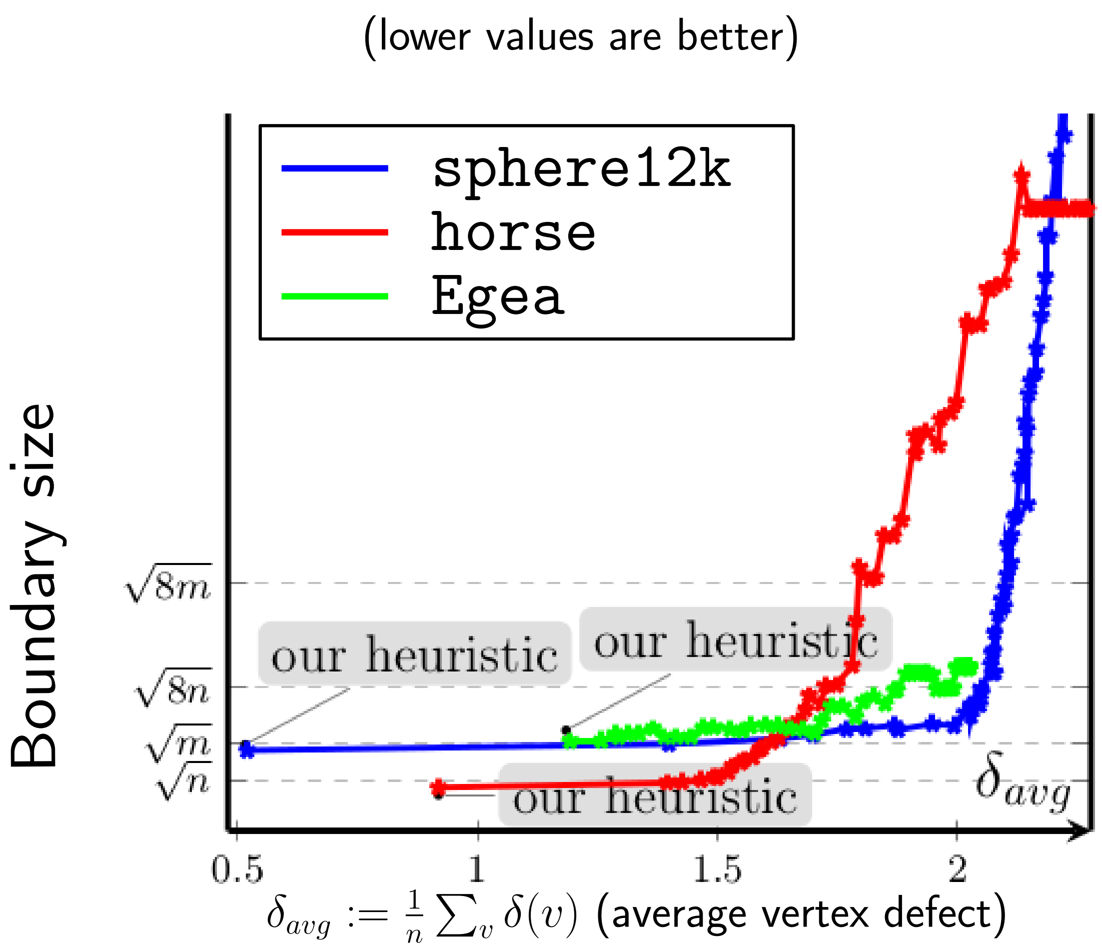
Separator balance

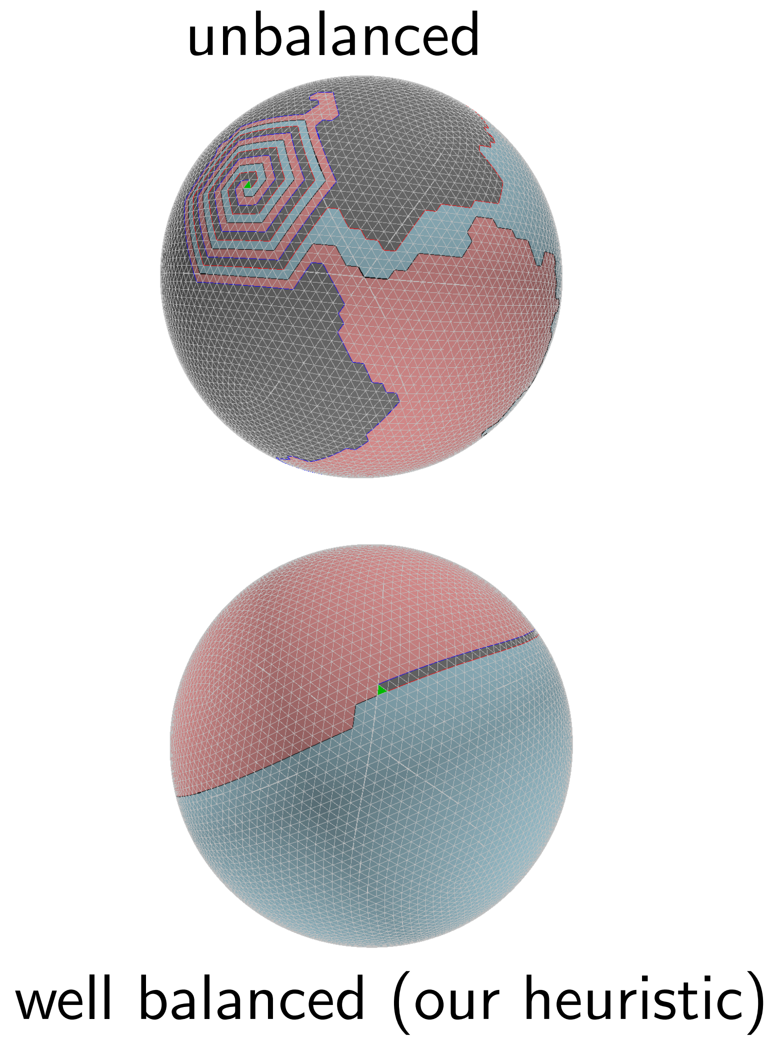
(tests are repated with 200 random choices of the initial **seed**, the root face)



From Schnyder woods to cycle separators

How the separator quality depends on the balance

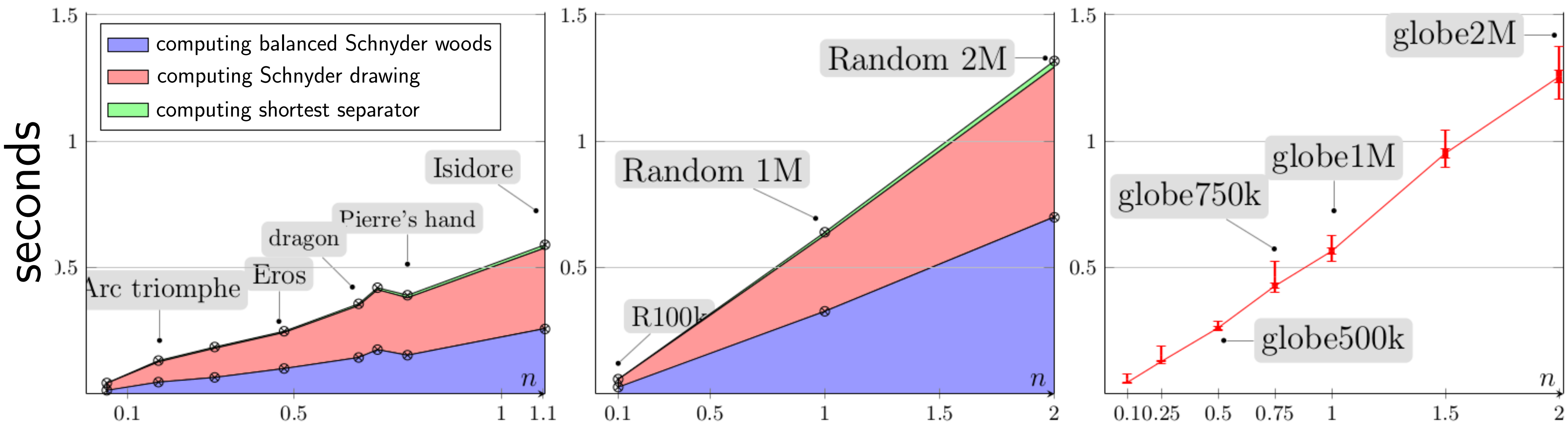




Evaluation of timing costs

average timings (over 100 executions)

total timing costs
(100 choice of random seeds)

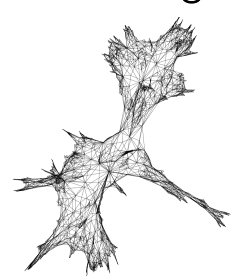


- **Our performances** (pure **Java**, on a core i7-5600 U, 2.60GHz, 1GB Ram):
  We can process $\approx 1.43M - 1.92M$ vertices/seconds
- Metis can process $\approx 0.7M$ vertices/seconds (**C**, on a Intel core i7-5600 2.60GHz)
- Previous works can process $\approx 0.54M - 0.62M$ vertices/seconds
  (Fox-Epstein et al. 2016, Holzer et al. 2009) (**C/C++**, on a Xeon X5650 2.67GHz)

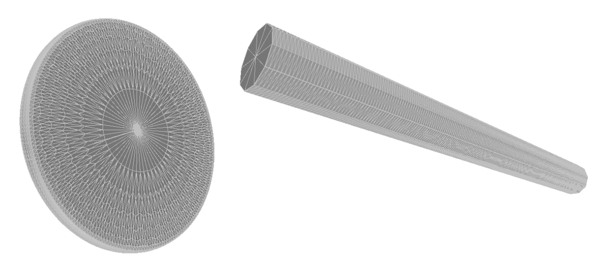
Our datasets (several tens of real-world, random and synthetic graphs)

3d meshes from **aim@shape** and **Thingi 10k**

Random triangulations

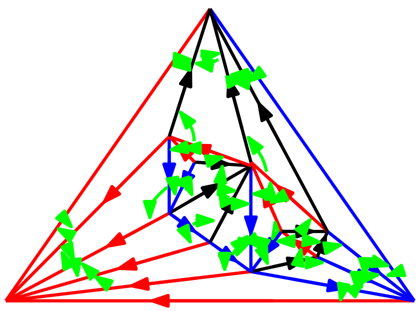


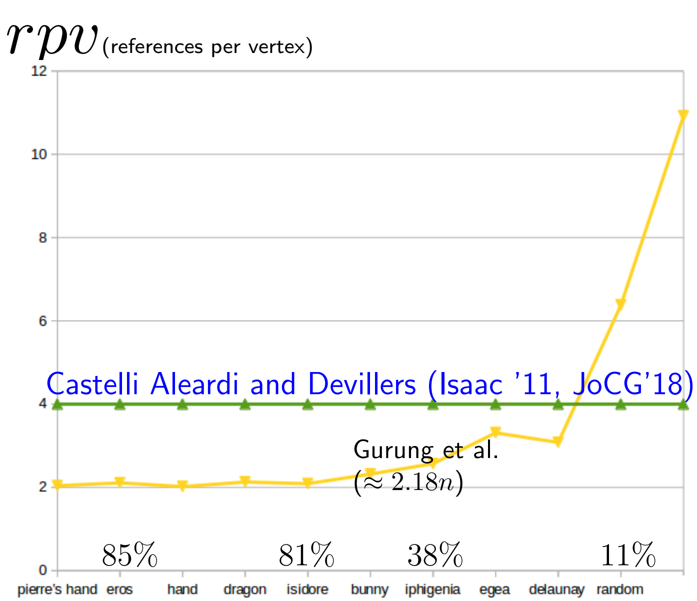
Synthetic graphs

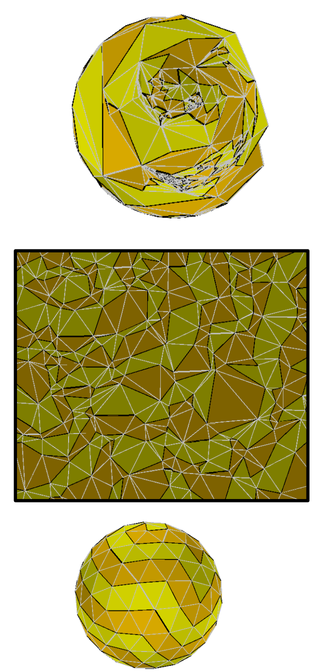


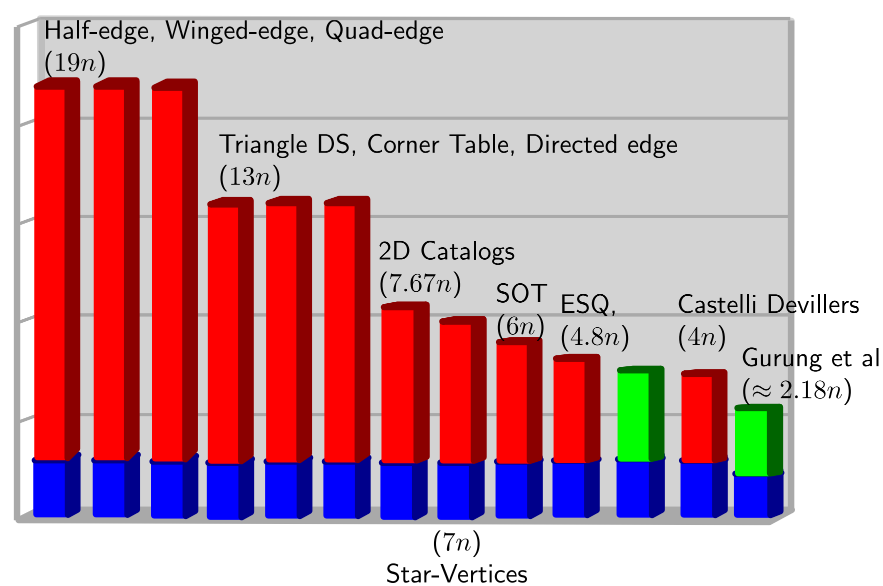
Practical mesh data structure

	Data Structure	size	navigation time	vertex access	dynamic
(non compact) data structures	Half-edge/Winged-edge/Quad-edge	$18n + n$	$O(1)$	$O(1)$	*yes*
	Triangle based DS / Corner Table	$12n + n$	$O(1)$	$O(1)$	*yes*
compact data structures	Directed edge (Campagna et al. '99)	$12n + n$	$O(1)$	$O(1)$	*yes*
	2D Catalogs (Castelli Aleardi et al., '06)	$7.67n$	$O(1)$	$O(1)$	*yes*
	Star vertices (Kallmann et al. '02)	$7n$	$O(d)$	$O(1)$	*no*
	TRIPOD (Snoeyink, Speckmann, '99)	$6n$	$O(1)$	$O(d)$	*no*
	SOT (Gurung et al. 2010)	$6n$	$O(1)$	$O(d)$	*no*
	SQUAD (Gurung et al. 2011) (ε between 0.09 and 0.3)	$(4+\varepsilon)n$	$O(1)$	$O(d)$	*no*
	ESQ (Castelli Aleardi, Devillers, Rossignac'12)	$4.8n$	$O(1)$	$O(d)$	*yes*
	Castelli Aleardi and Devillers (Isaac '11, JoCG'18)	$4n$ (or $5n$)	$O(1)$	$O(d)$ (or $O(1)$)	*no*
	LR (Gurung et al. 2011) (δbetween 0.2 and 0.3)	$(2+\delta)n$	$O(1)$	$O(1)$	*no*





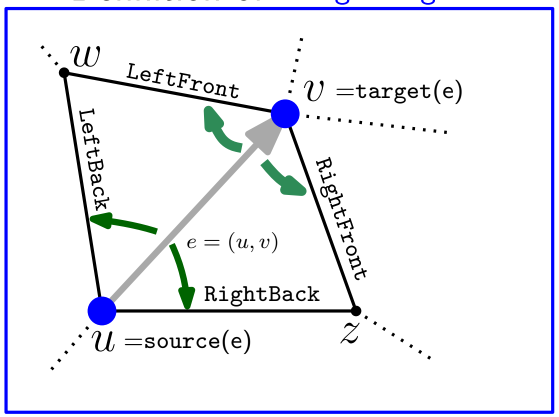




Winged Edge DS (size $19n$)

(Baumgart, 1972)

Definition of Winged-edge



$0 \leq e \leq (3n-6)-1$

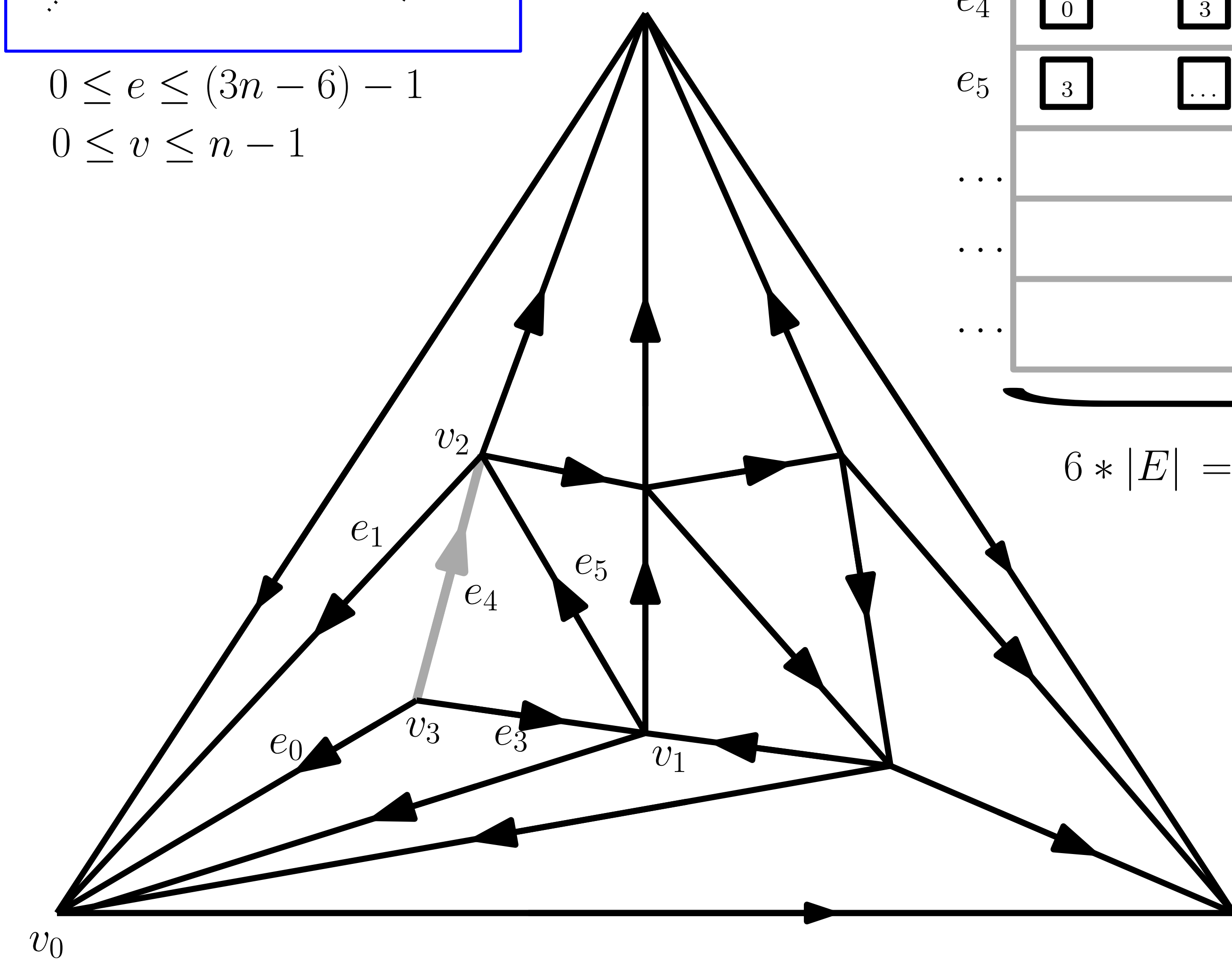
$0 \leq v \leq n-1$

	LeftBack	RightBack	LeftFront	RightFront	source	target
e_0						
e_1						
e_2						
e_3						
e_4	0	3	1	5	3	2
e_5	3	...	4	...	1	2
...						
...						
...						

$6 * |E| = 6 * (3n-6) \approx 18n$ entries

	incident edge
v_0	e_0
v_1	e_3
v_2	
v_3	e_4
v_4	
v_5	
	...
	...
	...

n entries



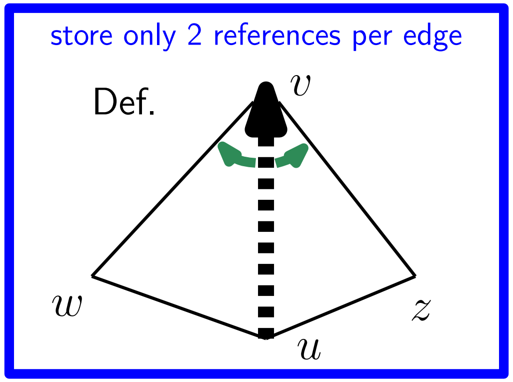

```
Degree(u)
{
        e = Edge(u);
        f = e; d = 0;
        do{
                if u =Source(f) f =LeftBack(f) ;
                else            f =RightFront(f) ;
                d = d + 1;
        } while f ≠ e;
        return d;
}
```

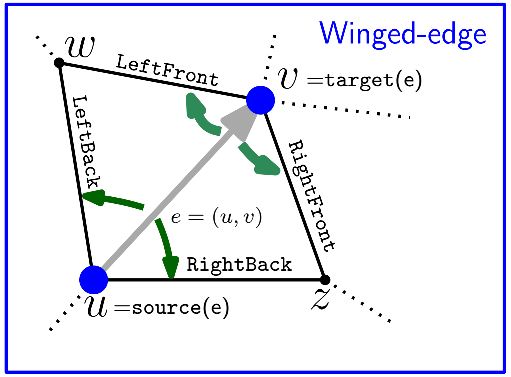
Our first simple Compact DS (size $6n$)

(Castelli Aleardi, Devillers, 2011)

$e := (u, v) \qquad 0 \leq v \leq n-1$

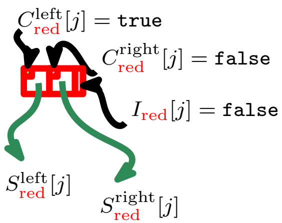
$0 \leq e \leq 3n$





retrieve

$$(w, u) := \begin{cases} (e+1)\%3 & \text{case 1} \\ (T[e]+2)\%3 & \text{case 2} \\ T[T[e]] & \text{case 3} \end{cases}$$



	red		blue		black	
0	6	6	2	6	1	2
1	6	4	0	0	3	0
2	3	7	7	0		
3	8	2	1	8	2	1
4	7	9	6	7	9	9
5						
6	4	1	0	4	1	1
7	2	4				
8	9	3	3	9	3	3
9	4	8	8	1	8	8

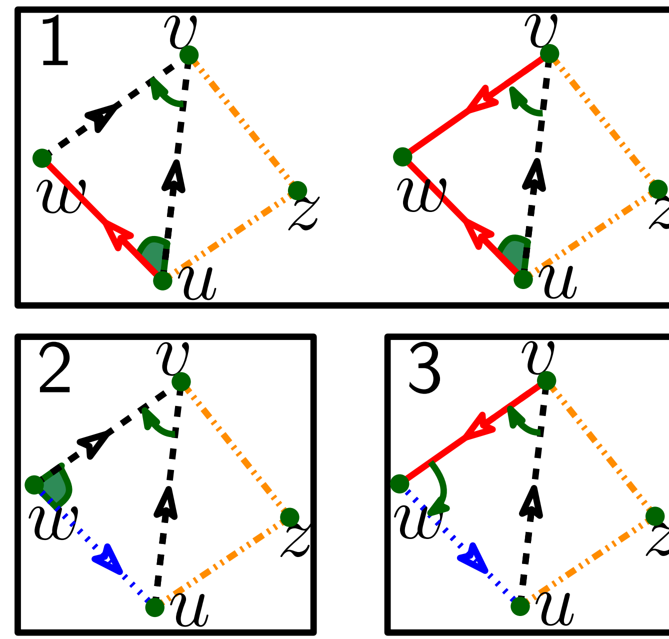
$6 * n$ entries

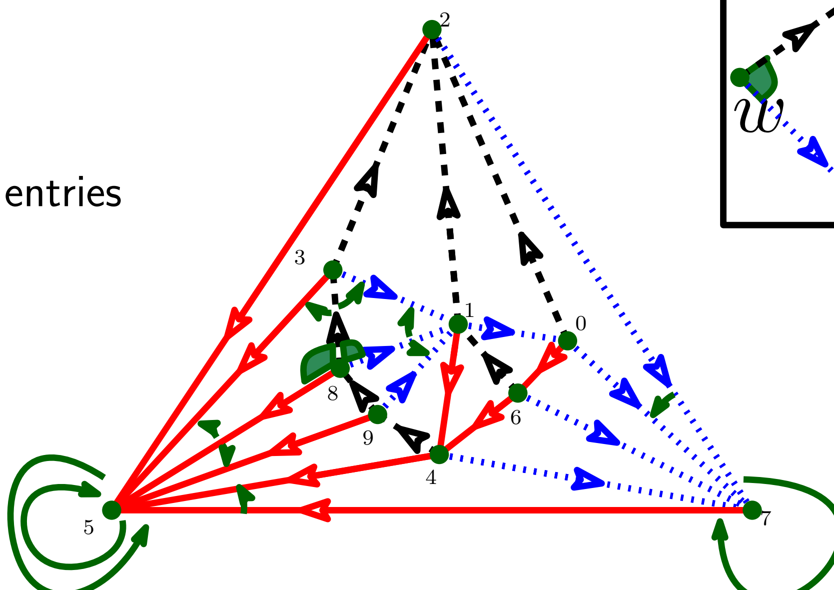
$u :=$ `source`$(e) = e/3$

(w,v):=T[2e]

$(z, v) := T[2e+1]$

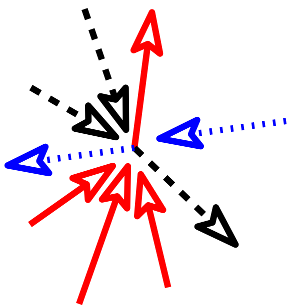
`color`$(e) = e\%3$



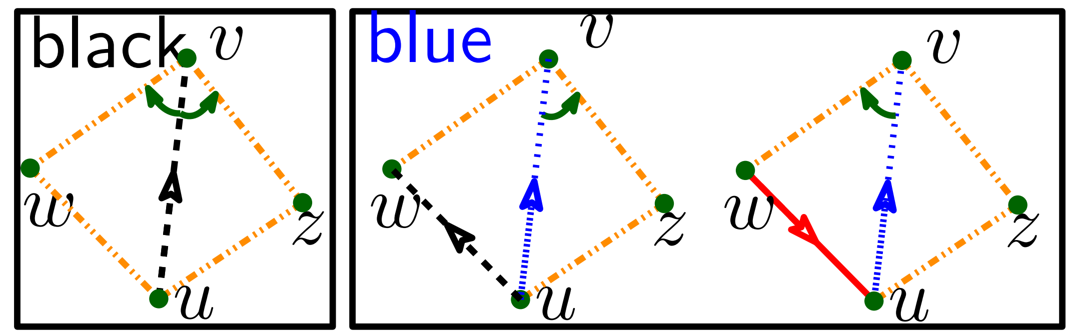


More compact DS (size $5n$): use maximal Schnyder woods

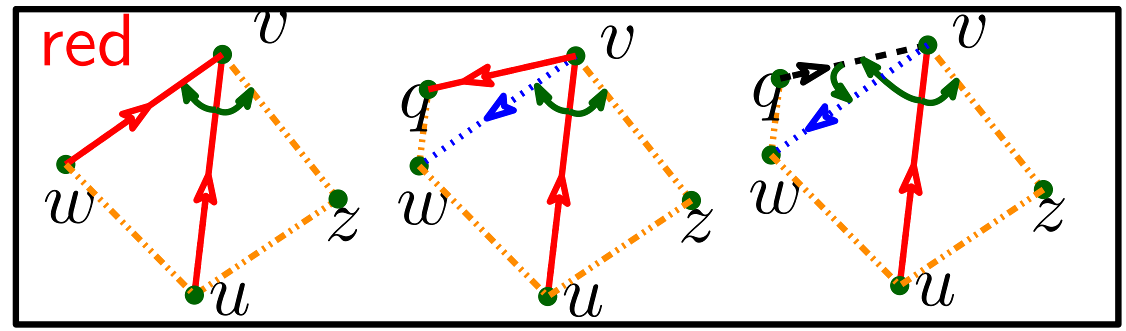
(less redundant and "more difficult to implement")

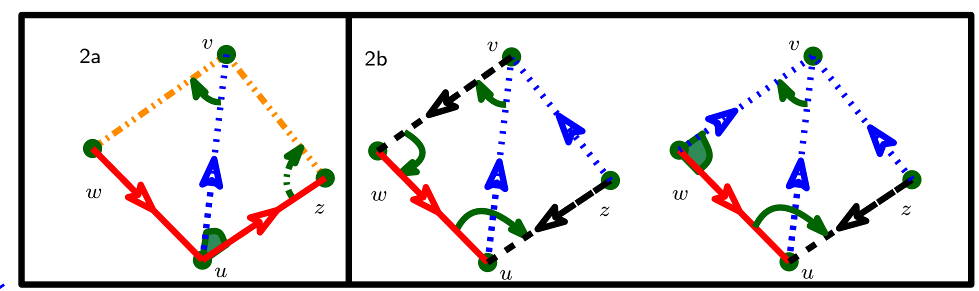


remove one blue column

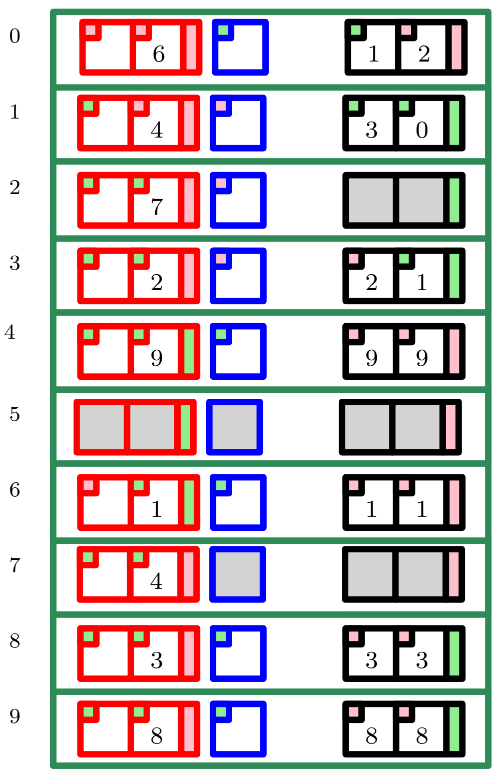


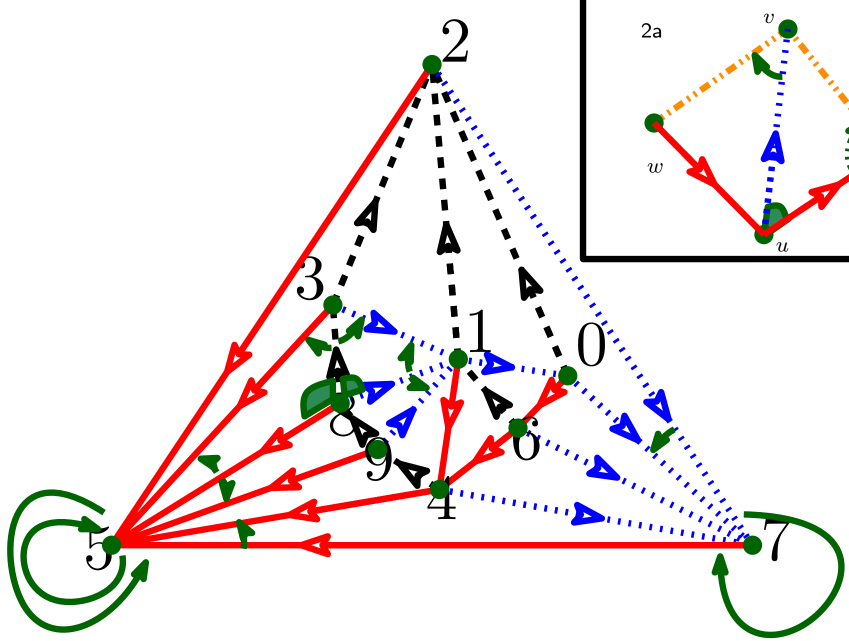
(sometimes) change one red left reference (for red edges)

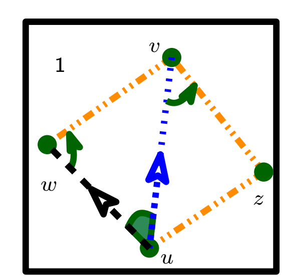




only difficult case



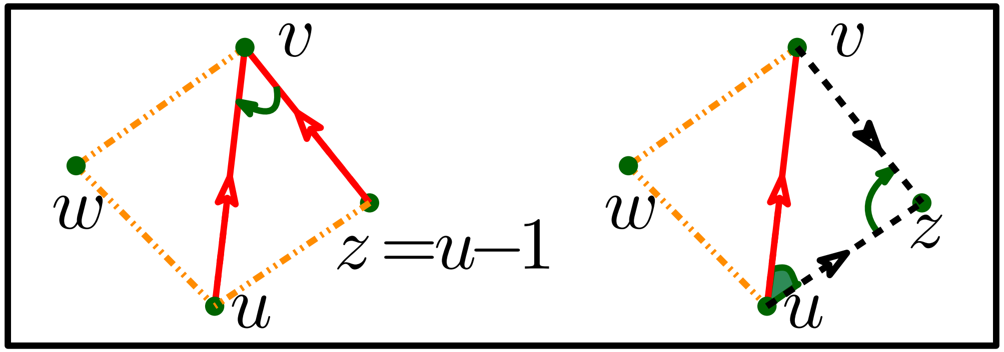


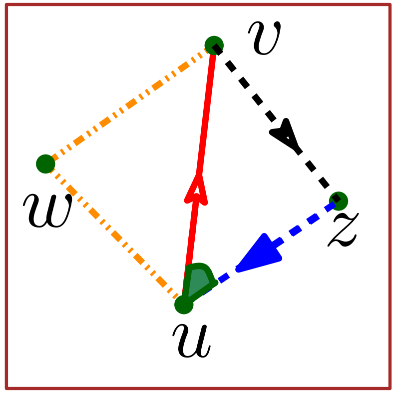


More compact DS (size $4n$): use maximal Schnyder woods

(reorder vertices according to a BFS traversal of T_0)

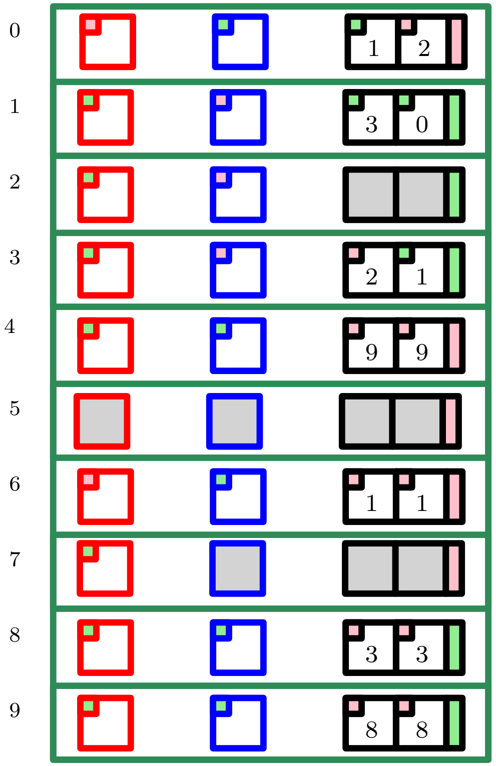
("slightly more difficult to implement")

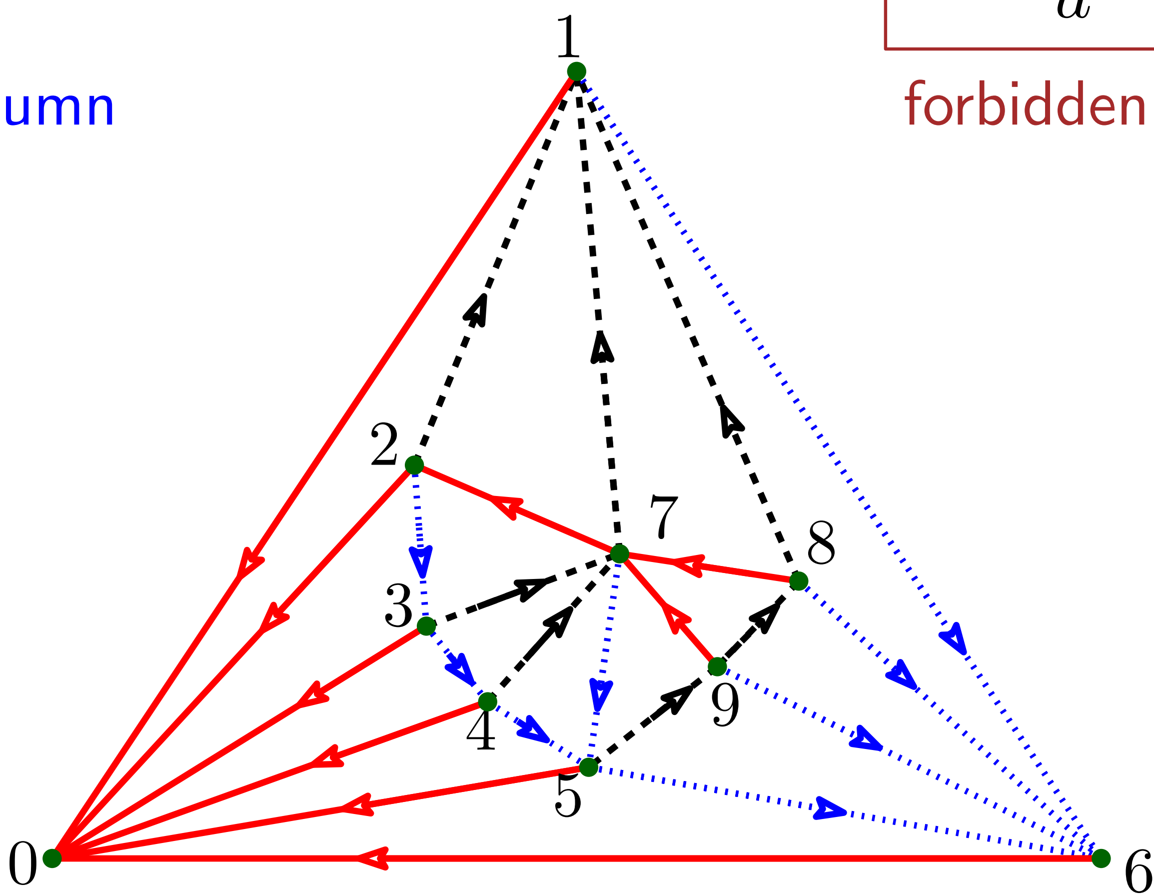




forbidden case

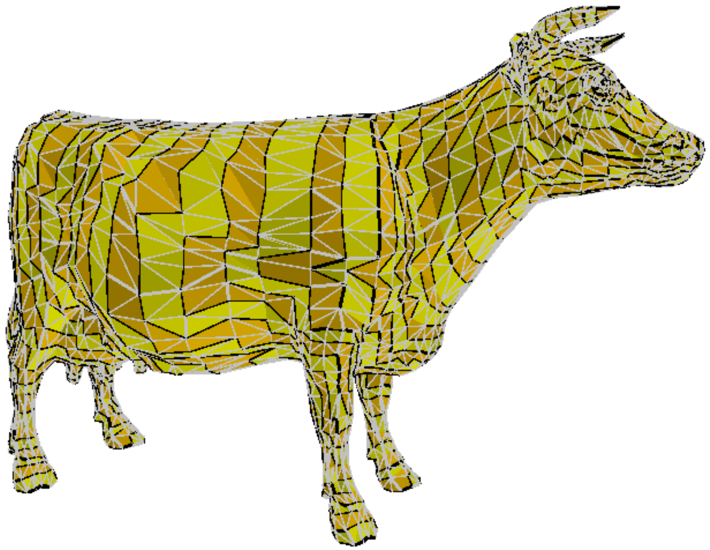
remove one red column

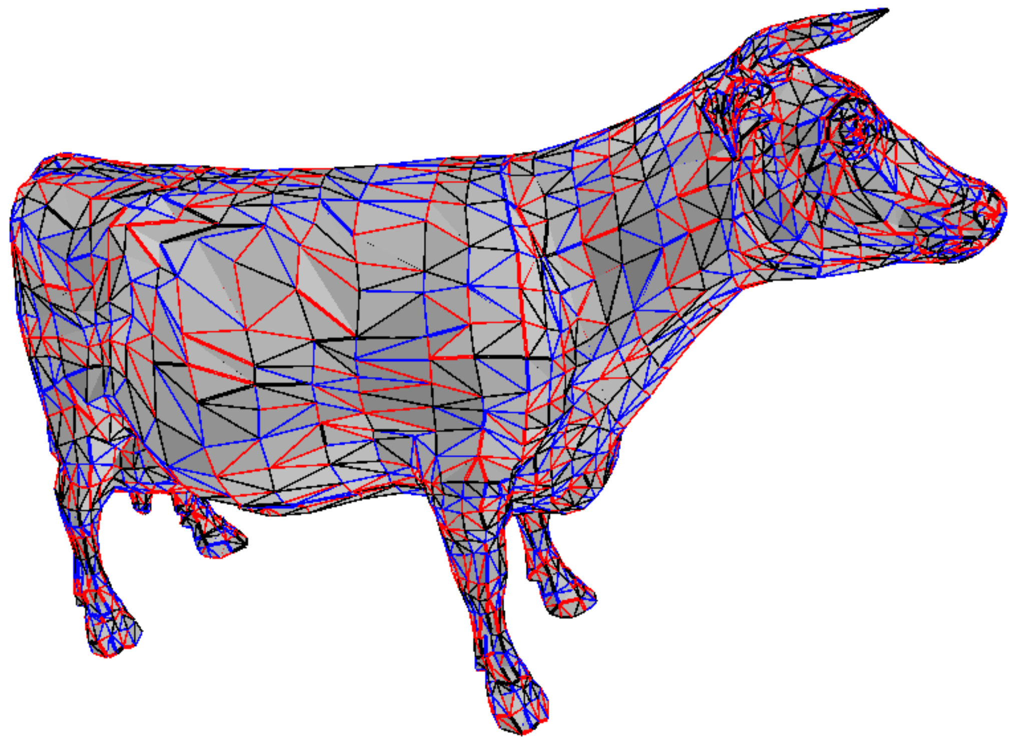




More compact DS: size $< 4n$?

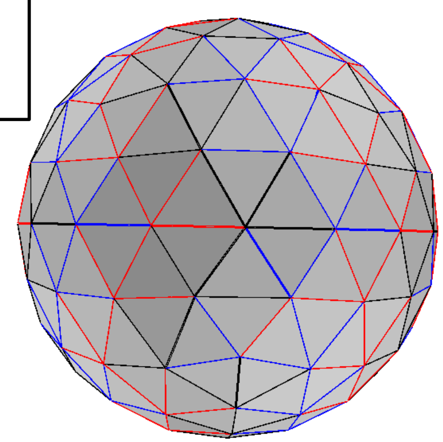
(can we exploit the regularity of the triangulation?)

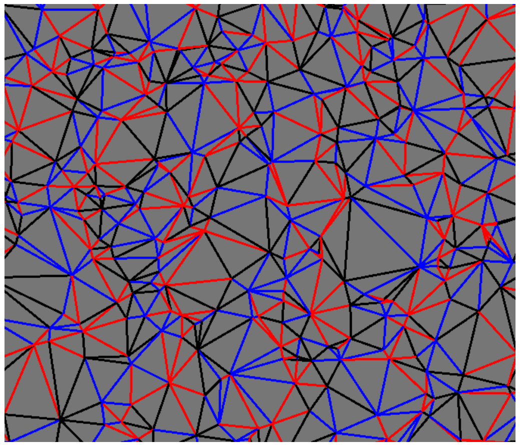




upper bound depending of vertex degree distribution

$$size(n) = 3n + 2(\sum_{i=k+3}^{n-1} p_i)$$





for $k = 4$

$$size(n) = 3n + 2(\sum_{i=7}^{n-1} p_i)$$

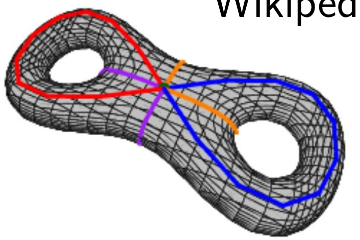
Concluding remarks and perspectives

(Schnyder woods and related combinatorial structures
have still many wonderful surprises in store for us)

Schyder woods for higher genus surfaces

Thm (3-orientations for graphs on surfaces, of arbitrary genus)
[Albar Goncalves Knauer, 2014]

Any triangulation of a surface (the sphere and the projective plane) admits a '3-orientation': orientation without sinks s.t. every vertex has outdegree divisible by three

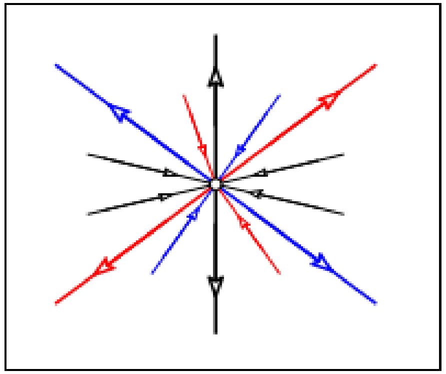


Wikipedia picture

Conjecture (Existence of Schnyder woods for higher genus triangulations)
[Goncalves Knauer Lévêque, 2016]

Multiple local Schnyder condition: the outdegree of every vertex is a positive multiple of 3.

(there are no **sinks**)



Thm [Suagee, 2021]

Simple triangulations of genus $g \geq 1$ having large **edgewidth** do admit Schnyder woods

edgewidth $\geq 40(2^g - 1)$

Experimental confirmation

exaustive generation of all 3-orientations for all triangulations with $g = 2$, $n \leq 11$

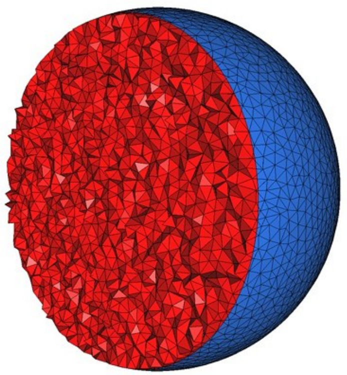
All simple triangulations of genus $g = 2$ and size ≤ 11 admit Schnyder woods

n	# irreducible triangulations	#triangulations ($g = 2$)
7	–	–
8	–	–
9	–	–
10	865	865
11	26276	113506

surftri software [Sulanke, 2006]

Schyder woods for higher dimension complexes

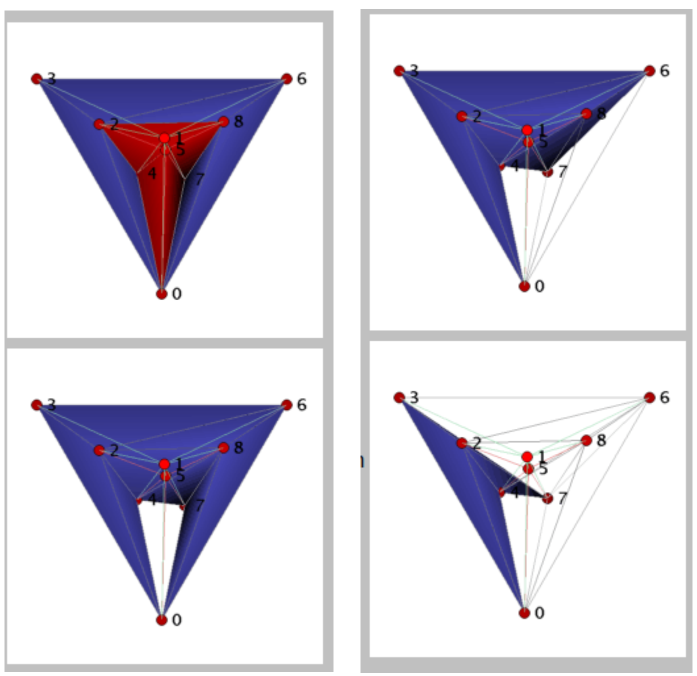
What about higher dimensional complexes?



(CGAL mesher)

Very challenging problems... things are far more complicated

No hope to generalize canonical orderings easily

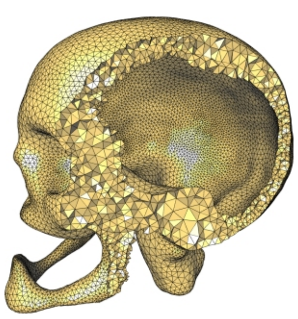


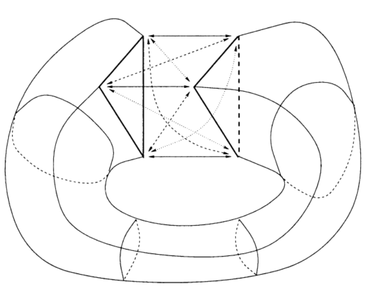
Non shellable simplicial 3-ball, $n = 9$ (Lutz)

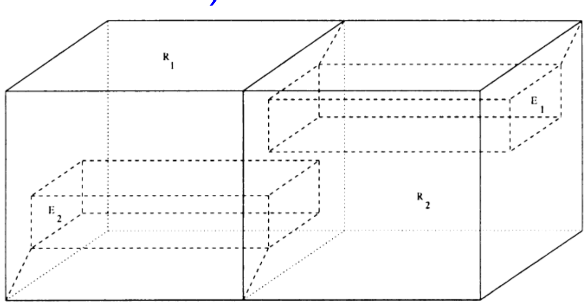
Tetrahedral mesh compression

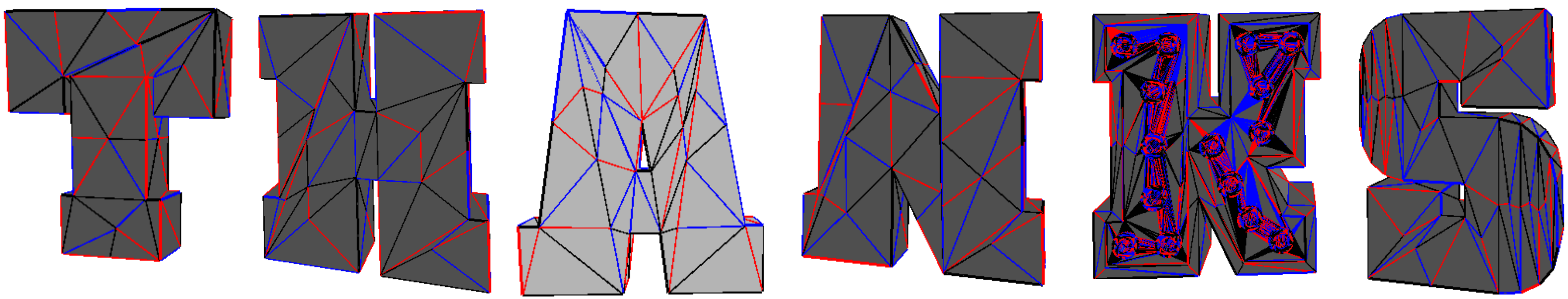
Grow&Fold (Szymczak Rossignac '00): $\approx 7t$ bits

Cut-border machine (Gumhold et al. '99)








THANKS