Simplicial surfaces in GAP

Markus Baumeister
(j/w Alice Niemeyer)

Lehrstuhl B für Mathematik
RWTH Aachen University

30.08.2017

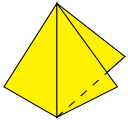
Basic data

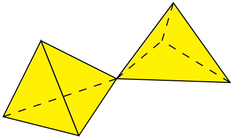
- Package name: `SimplicialSurfaces`
  - Not yet generally available
- Authors: Alice Niemeyer, Markus Baumeister
- Based on current research at Lehrstuhl B: Plesken, Strzelczyk and others
- Internally used packages:
  - `AttributeScheduler` by Gutsche
  - `Digraphs` by De Beule, Mitchell, Pfeiffer, Wilson et al.
  - `GAPDoc` by Lübeck
  - `AutoDoc` by Gutsche, Horn

Motivation

Goal: Investigate rigid paper folding in $\mathbb{R}^3$

- consider surfaces built from triangles (simplicial surfaces)
  - not closed under folding
  - allow more general structures:





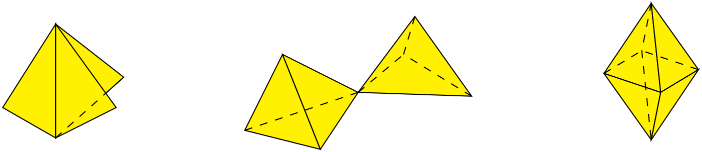
- embeddings are difficult to compute
  - some embeddings of asymmetric icosahedrons are infeasible to compute

$\rightsquigarrow$ focus on intrinsic properties

$\rightsquigarrow$ incidence geometry

Reasons for implementation in GAP

- can describe incidence geometry
  - allows flexible access to the incidence geometry (`AttributeScheduler`)
- can manage hierarchy of structures:



- works well with group–theoretic descriptions
- difference to `FinInG`–package by De Beule, Neunhöffer et al.
  - we only have two dimensions but can work with colourings and foldings

Table of contents

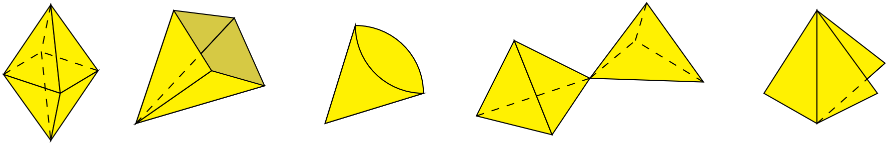
1. General simplicial surfaces
2. Edge colouring and group properties
3. Abstract folding

Table of contents

1. General simplicial surfaces
2. Edge colouring and group properties
3. Abstract folding

Triangular complexes

We want to describe different structures:

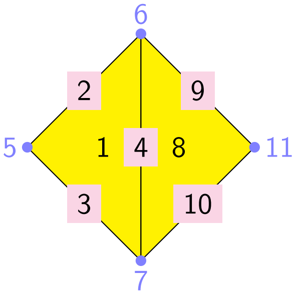


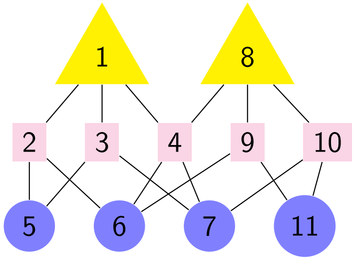
$\rightsquigarrow$ **triangular complexes**

- sets of vertices, edges and faces
- incidence relation between them
- every face is a triangle
- every vertex lies in an edge and every edge lies in a face

Isomorphism testing

Incidence structures can be interpreted as coloured graphs:





$\rightsquigarrow$ reduce to coloured graph isomorphism problem (with fixed colours)
$\rightsquigarrow$ solved efficiently in practice by `nauty/Traces` (McKay, Piperno)
In GAP: `NautyTracesInterface` (Gutsche, Niemeyer, Schweitzer)

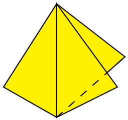
- calls C–functions directly without writing files
- also returns automorphism group

General properties

Some properties can be computed for all triangular complexes:

- Connectivity
- Euler–Characteristic

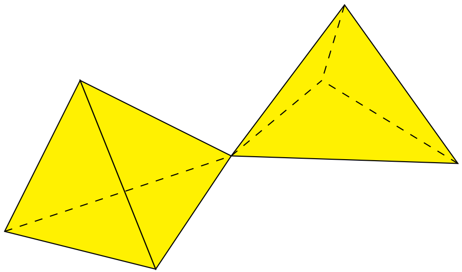
Orientability is **not** one of them. Counterexample:



$\Rightarrow$ every edge lies in at most two faces (for well–definedness)
$\rightsquigarrow$ **ramified simplicial surfaces**

Why ramified?

Typical example of a ramified simplicial surface:

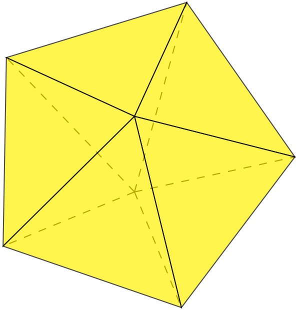


$\Rightarrow$ It is not a surface — there is a *ramification* at the central vertex
A **simplicial surface** does not have these ramifications.

Classification

Plesken/Strzelczyk classified all closed simplicial surfaces up to 20 triangles.

- building blocks are those without a 3–cycle of edges (not around face)
- e. g. exactly 87 non–isomorphic surfaces with 20 triangles
- e. g. only one surface with 10 triangles:



Progress report of triangular complexes

Already implemented:

- surface hierarchy
- elementary properties (e. g. connectivity, orientability [if defined])
- isomorphism testing
- classification of small surfaces (as a data base)

Not yet implemented:

- automorphism group
- advanced properties (any wishes?)

Table of contents

1. General simplicial surfaces
2. Edge colouring and group properties
3. Abstract folding

Embedding questions

Given: A simplicial surface

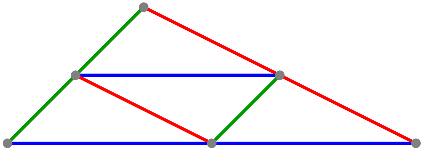
- For which edge lengths embeddable into $\mathbb{R}^3$? In how many ways?

Simplification: All triangles are isometric

Goal: Find an abstract structure that encodes this isometry condition

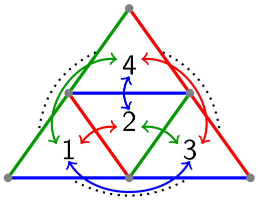
$\rightsquigarrow$ Consider a general triangle (all side lengths different)

$\rightsquigarrow$ Edge–colouring encodes different lengths



Colouring as permutation

Consider a tetrahedron with an edge colouring



simplicial surface $\Rightarrow$ at most two faces at each edge

$\rightsquigarrow$ every edge defines a transposition of incident faces

$\rightsquigarrow$ every colour class defines a permutation of the faces

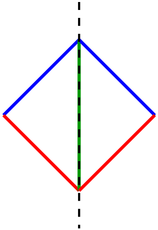
- (1,2)(3,4) , (1,3)(2,4) , (1,4)(2,3)

$\rightsquigarrow$ group theoretic considerations

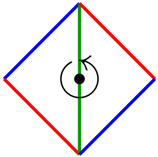
- The connected components of the surface correspond to the orbits of $\langle \sigma_a, \sigma_b, \sigma_c \rangle$ on the faces (fast computation for permutation groups)

How do faces fit together?

Consider a face of the surface and a neighbouring face
The neighbour can be coloured in two ways:



mirror (m)



rotation (r)

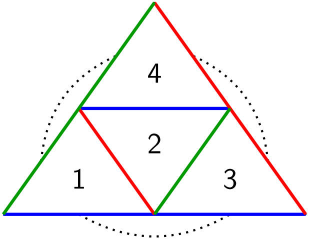
$\rightsquigarrow$ **mr–assignment** for the edges of the surface

Theorem

Permutations and mr–assignment uniquely determine the surface.

Constructing surfaces from groups

A general mr–assignment leads to complicated surfaces.
Simplification: edges of same colour have the same mr–type
Example



has only r–edges.

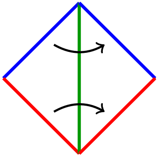
The mirror–case

If all edges are mirrors, the situation is simple

Theorem

A simplicial surface has only mirror–edges iff it covers a single triangle, i. e. there is a surjective incidence–preserving map to the simplicial surface consisting of exactly one face.

Consider



$\Rightarrow$ Unique map that preserves incidence

- Covering pulls back a mirror–colouring of the triangle
- Mirror–colouring defines a map to the triangle

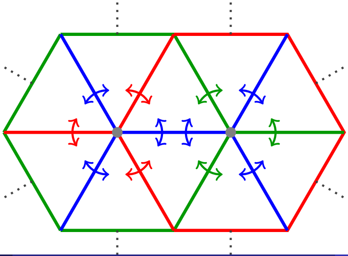
Construction from permutations

Start with three involutions σ_a, σ_b, σ_c in permutation representation (like generators of a finite group)

Theorem

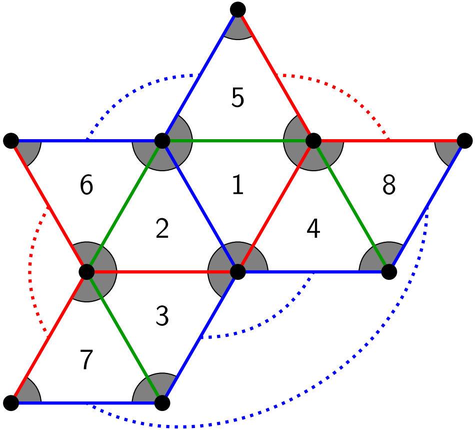
There exists a coloured surface with the given involutions where all edges are mirror edges.

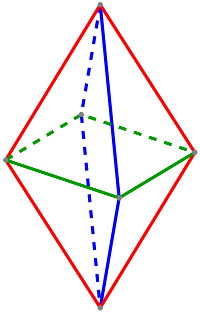
- The faces are the points moved by the involution triple
- The edges are the cycles of the involutions
- The vertices are the orbits of $\langle \sigma_a, \sigma_b \rangle$ on the faces (for all pairs)



Construction example

$\sigma_a = (1,2)(3,4)(5,6)(7,8)$
$\sigma_b = (1,4)(2,3)(5,8)(6,7)$
$\sigma_c = (1,5)(2,6)(3,7)(4,8)$

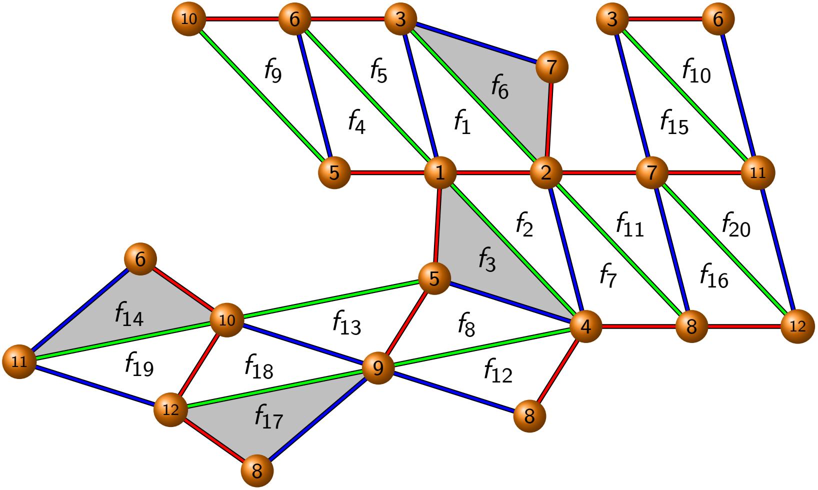




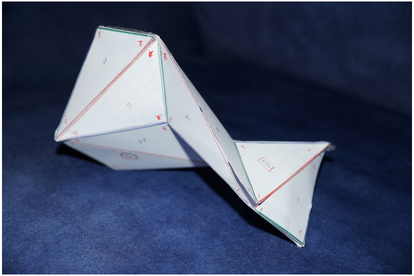
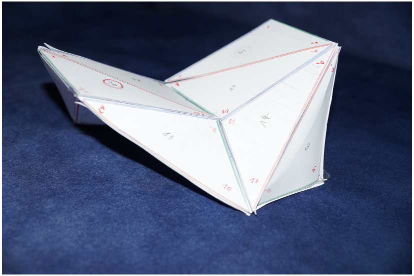
`iko`: coloured icosahedron

```
gap> DrawSurfaceToTikZ(iko,"NetIko.tex");
```

- Has to be manually untangled



Embedded icosahedron

Progress report of edge colouring

Implemented:

- computing all colourings of a given simplicial surface
- constructing all surfaces with given involutions
  1. up to (coloured) isomorphism
  2. with given mr–assignment
- drawing of simplicial surfaces
- constructing various coloured coverings

Still missing:

- for which lengths are the surfaces embeddable?
- does varying the lengths lead to another embedding?

Table of contents

1 General simplicial surfaces

2 Edge colouring and group properties

3 Abstract folding

What kind of folding?

- Folding of surface in $\mathbb{R}^3$
- Fold only at given edges (no introduction of new folding edges)
- Folding should be rigid (no curvature)

Goal: Classify possible folding patterns (given a net)

Embeddings are very hard

- At every point in time the folding process has to be embedded
- We can only show foldability for specific small examples
  - Usually using regularity (like crystallographic symmetry)
  - No general method
- It is very hard to define iterated foldings in an embedding

Folding without embedding

Central idea:

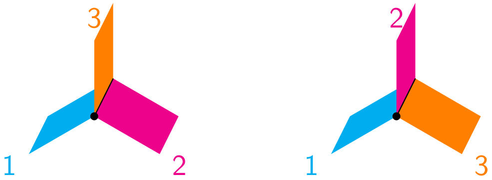
- Don't model the folding process (needs embedding)
- Describe starting and final folding state
  - Only consider changes in the topology (like identification of faces)
  - allows abstraction from embedding

$\rightsquigarrow$ Incidence geometry (triangular complex/simplicial surface)

- Captures some folding restrictions (rigidity of tetrahedron)
- Has to be refined

- Concept should allow reversible folding
- We need an ordering of the faces:

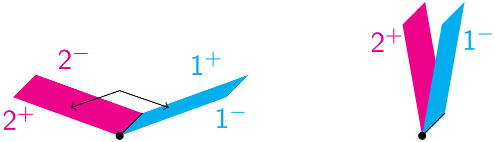
- We also need a cyclic order of the faces around an edge:



$\rightsquigarrow$ **folding complex**

How to describe folding?

Needs specification of two face sides:

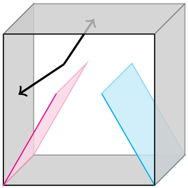


$\Rightarrow$ Characterize folding by specifying the two face sides that touch
$\rightsquigarrow$ **folding plan**

How does folding plan work?

Folding of two faces can induce folding of other faces:

- Can apply to arbitrarily many faces
- The induced folding is not unique



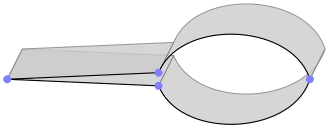
How does folding plan work?

Folding of two faces can induce folding of other faces:

- Can apply to arbitrarily many faces
- The induced folding is not unique

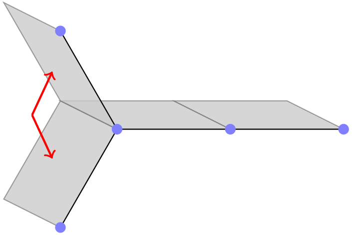
$\Rightarrow$ Identify only two faces at a time (non–uniqueness becomes choice)

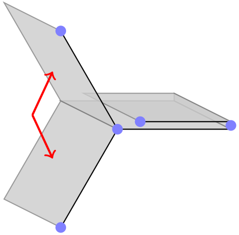
- $\rightsquigarrow$ Relax the rigidity–constraint:
  - Allow non–rigid configurations as transitional states



Structure of multiple foldings

With folding plans we can perform the same folding in different folding complexes

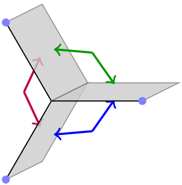


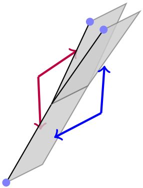


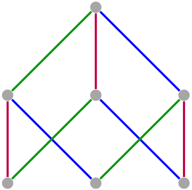
$\rightsquigarrow$ more structure on the set of possible foldings

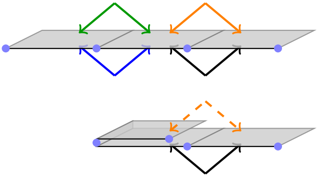
Folding graph

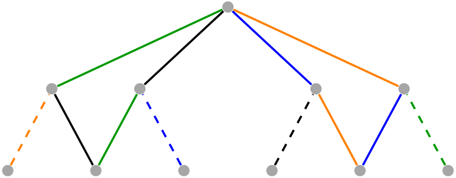
- Vertices are folding complexes (modelling folding states)
- Edges are folding plans connecting two folding complexes



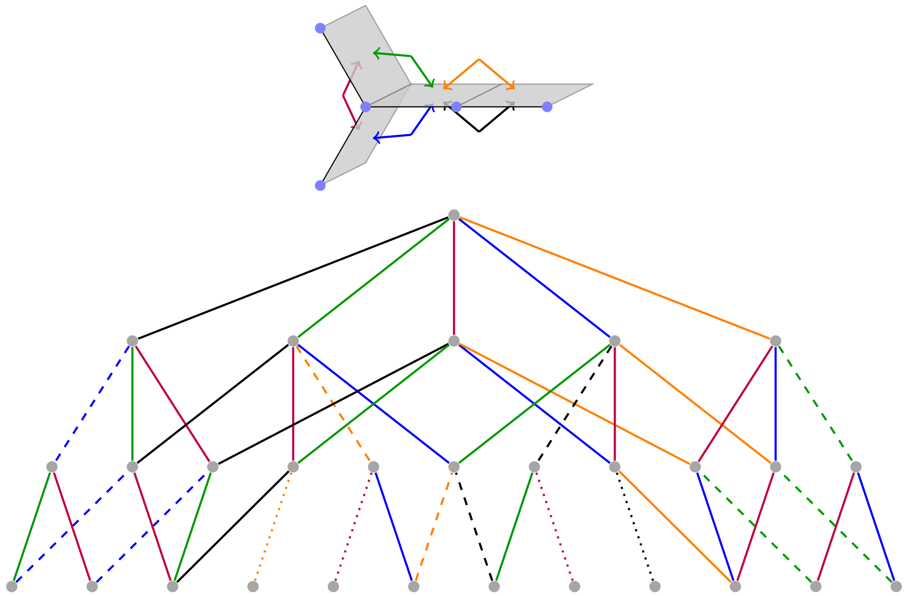






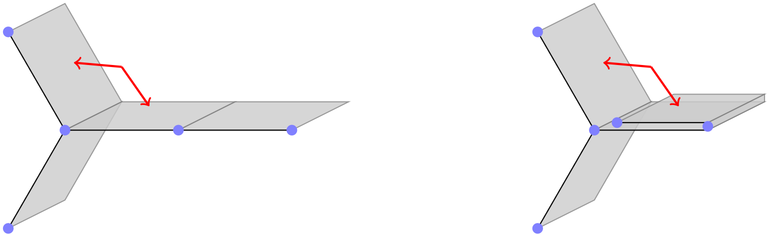


Larger graph



Drawback of folding plans

Some foldings that "should" be the same, aren't:



⇒ If you know the folding structure of a small complex, you can't easily find the folding structure of an extended complex

⇝ Folding plans are not optimal to model folding

Progress report of abstract folding

In development:

- folding complex
- folding plans
- folding graph

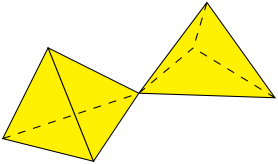
Missing:

- better folding description
- extension of folding graphs
- properties of folding graphs

Summary: SimplicialSurfaces

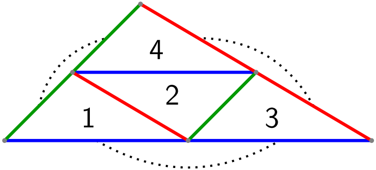
Triangular complexes

- mostly complete



Edge colouring

- current theory implemented
- a lot of theory still to discover



Abstract folding

- framework exists
- needs proper implementation