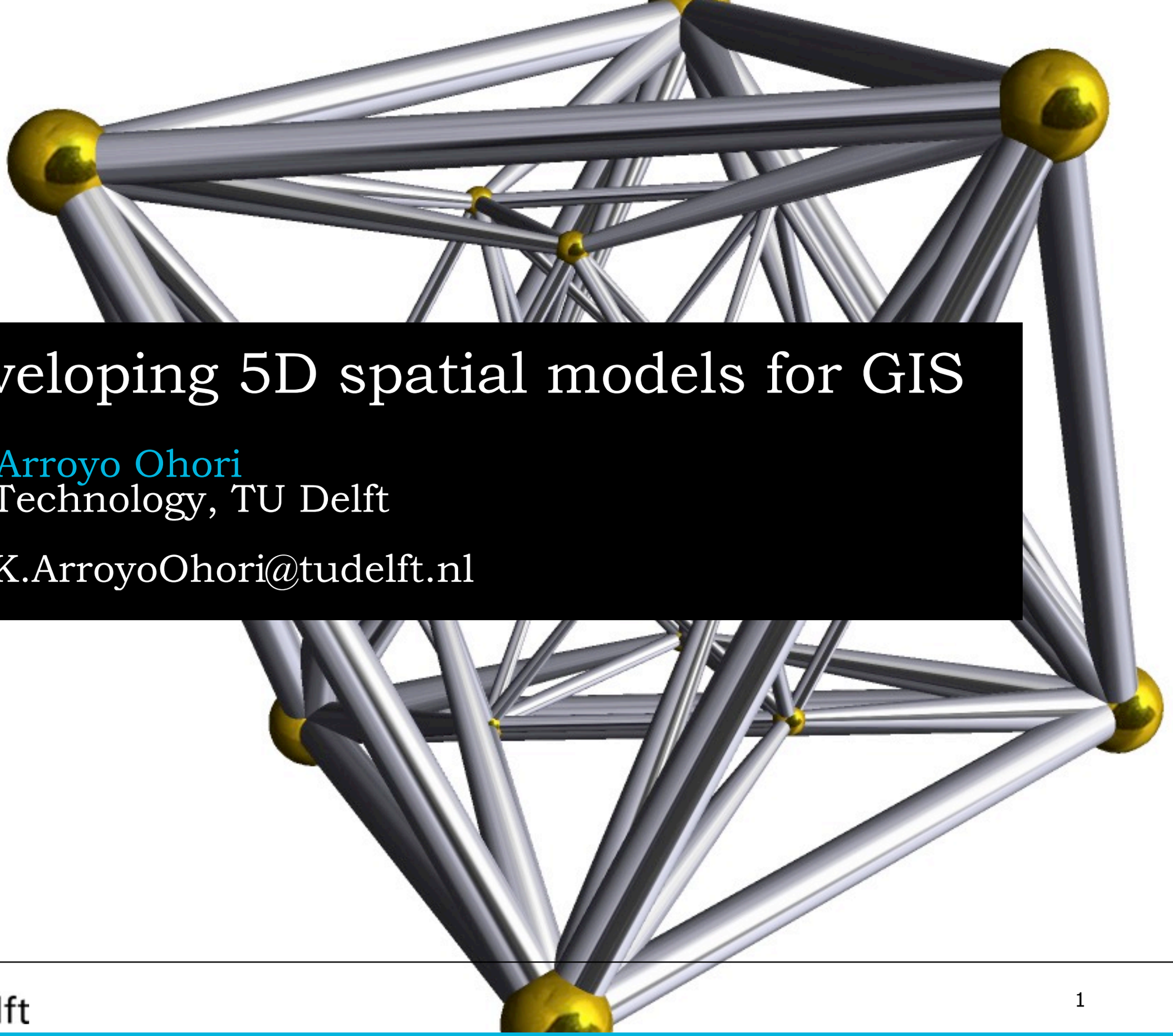


Developing 5D spatial models for GIS

Ken Arroyo Ohori
GIS Technology, TU Delft

G.A.K.ArroyoOhori@tudelft.nl

The 5D project

- Develop a 5D model that integrates 3D space + time + scale
- Vidi grant from Jantien Stoter
- GIS Technology group at the Delft University of Technology
- 5 years, 2 PhDs (4 years), 1 Postdoc (1 year), 2 part-time researchers

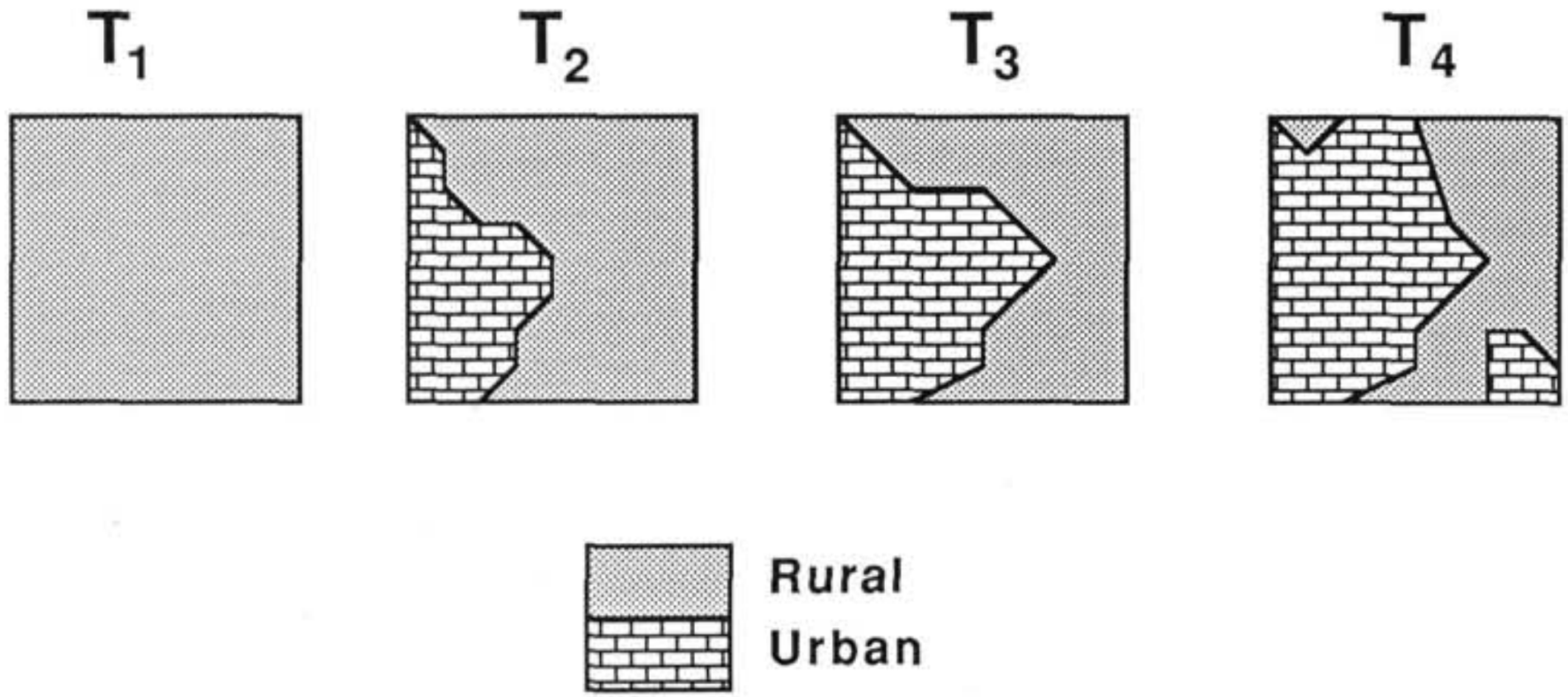
Spatial models

- Several developed independently (CAD, geology, CS/CG, GIS)
- All related, but with different characteristics (topology, allowed geometries, attributes)
- Some trends:
  - Increasing topological information
  - 2D, 3D, nD

Modelling space+time

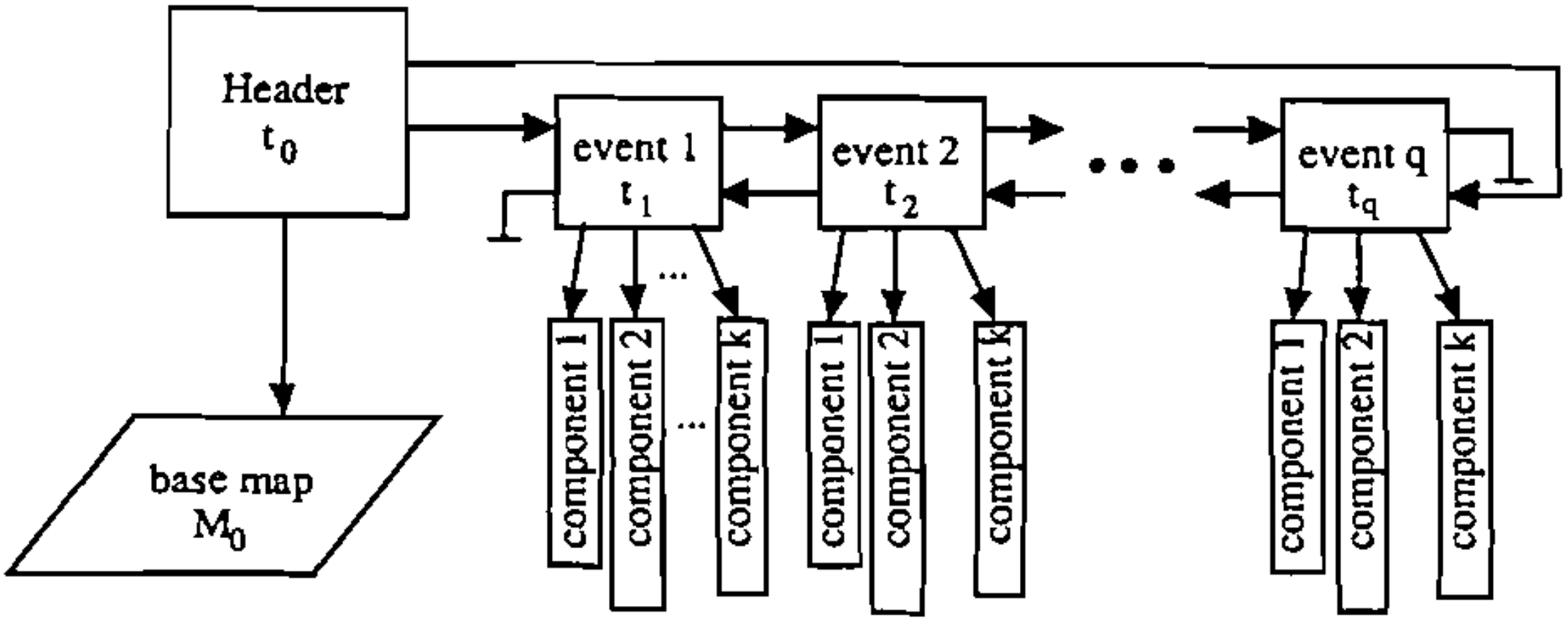
- Naïve approach (snapshot)
- Time specific models, linking objects together
- Treating it as spatial (full topology)

Modelling space+time



Langran & Chrisman, 1988

Modelling space+time



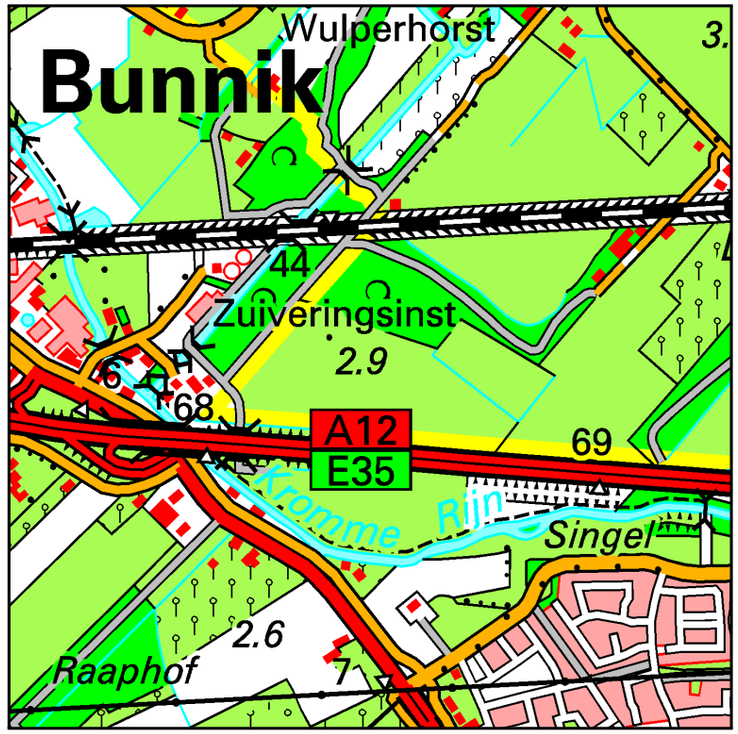
Peuquet & Duan, 1995

Modelling space+scale

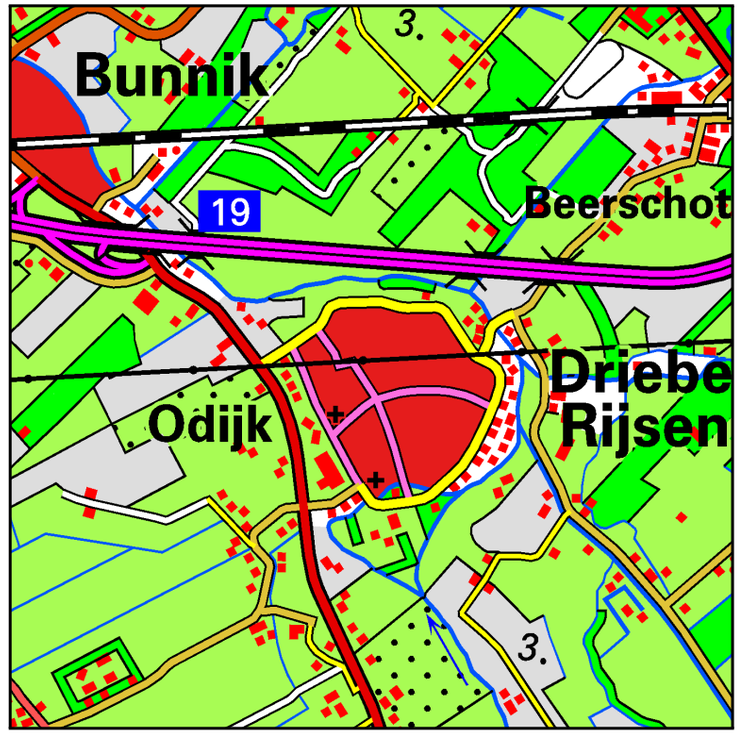
- Naïve approach (multiple representations)
- Linking objects together
- Treating it as spatial

Modelling space+scale

(a) 1:25 000



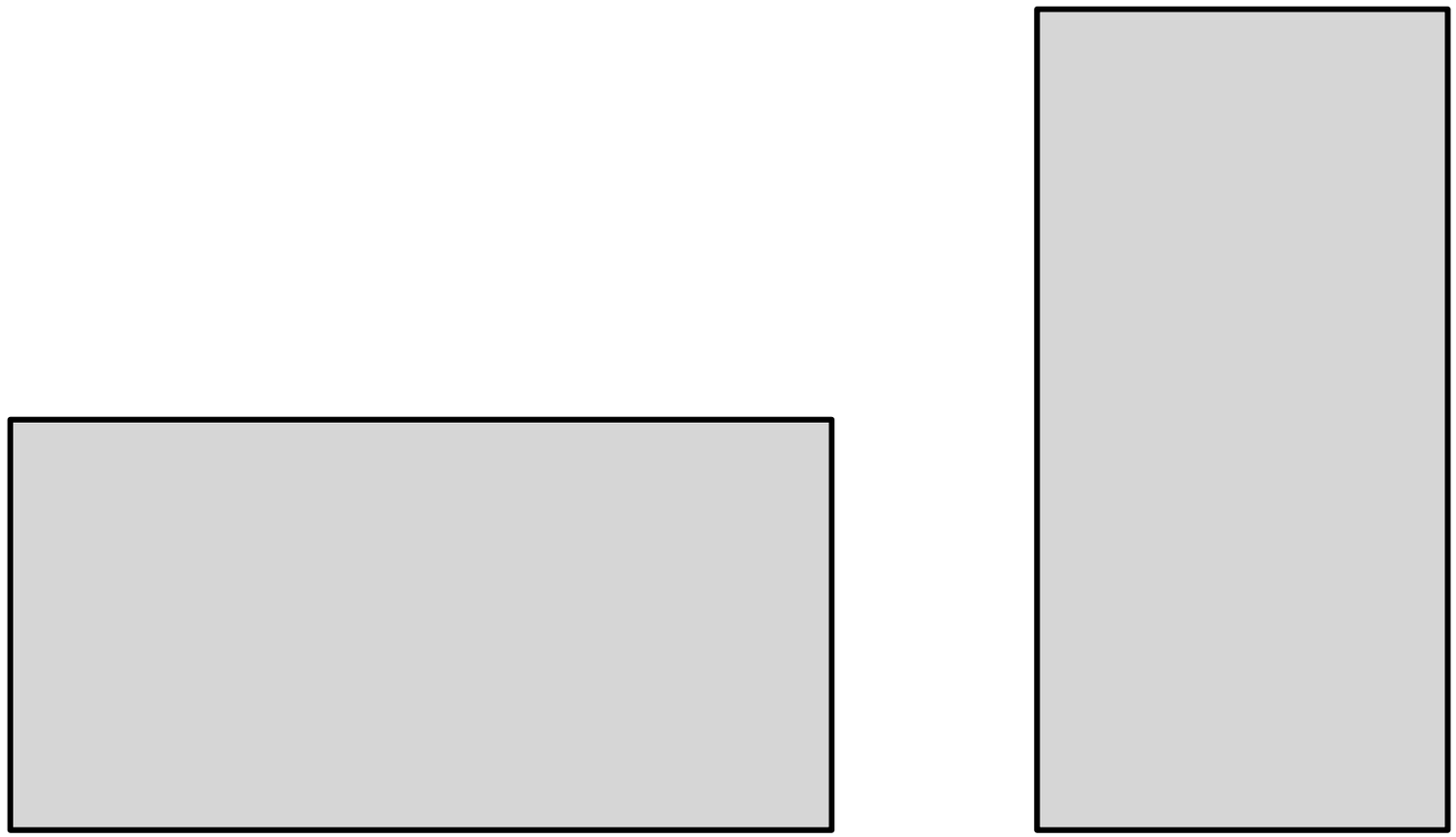
(b) 1:50 000

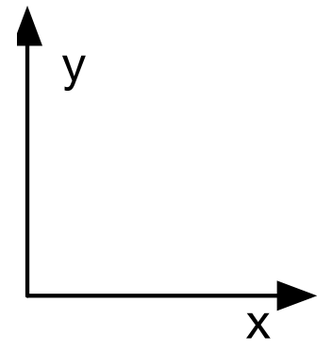


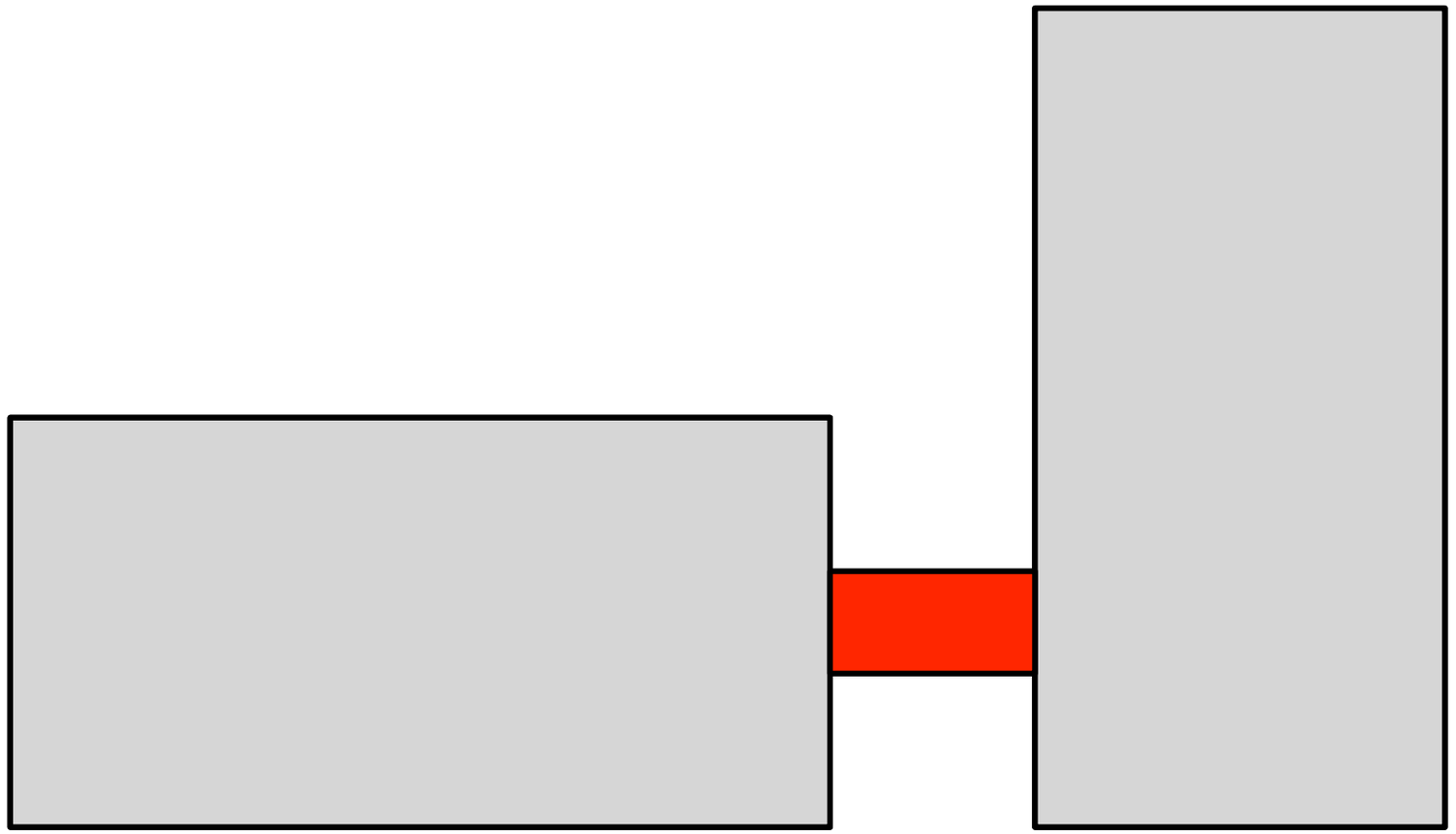
(c) 1:100 000

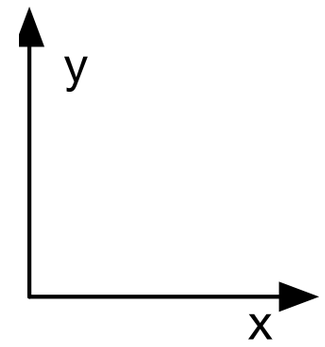
Meijers, 2011

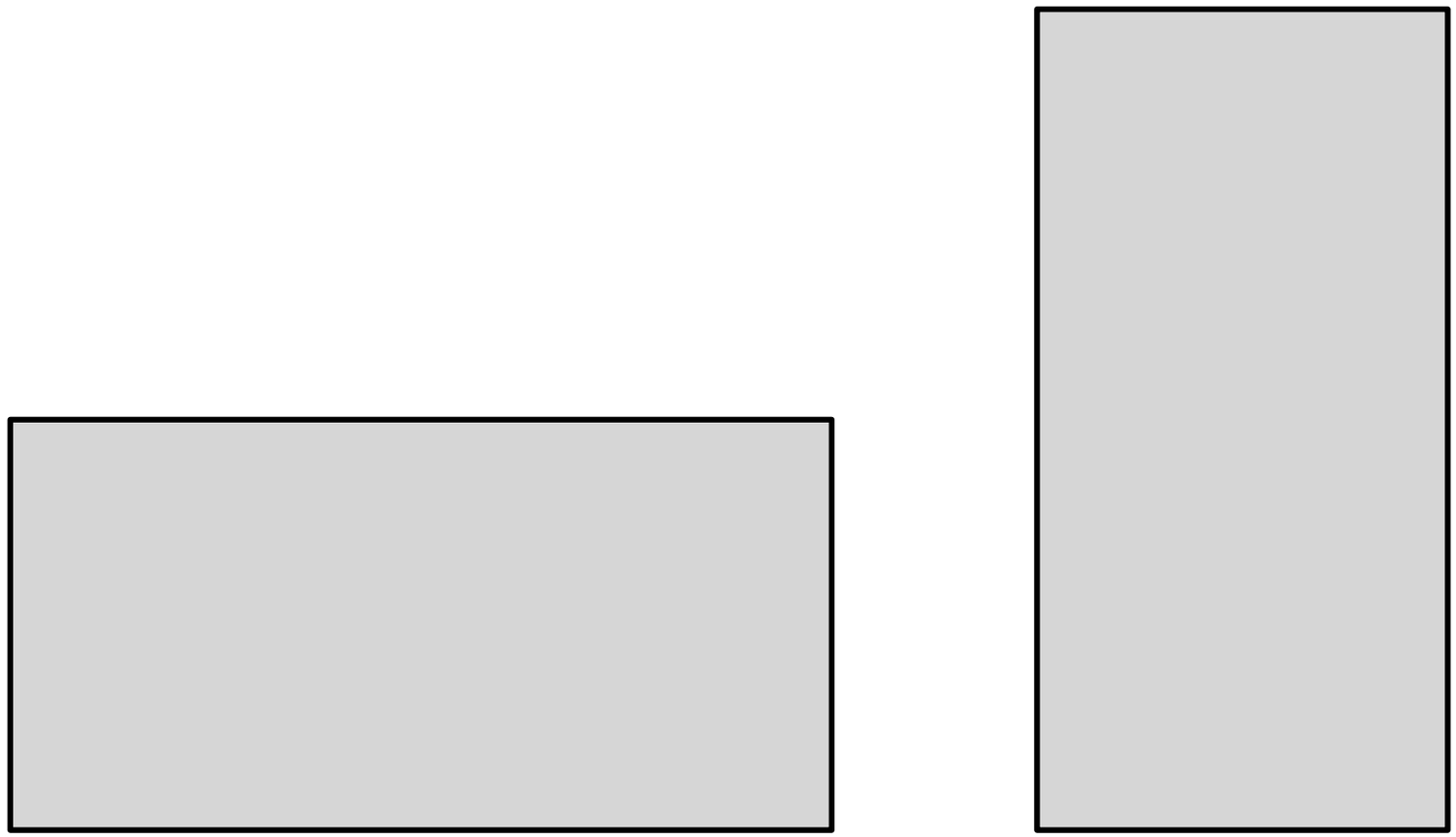
Wikimedia Commons

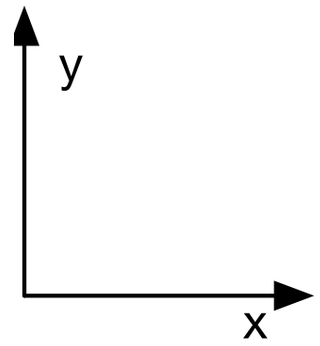


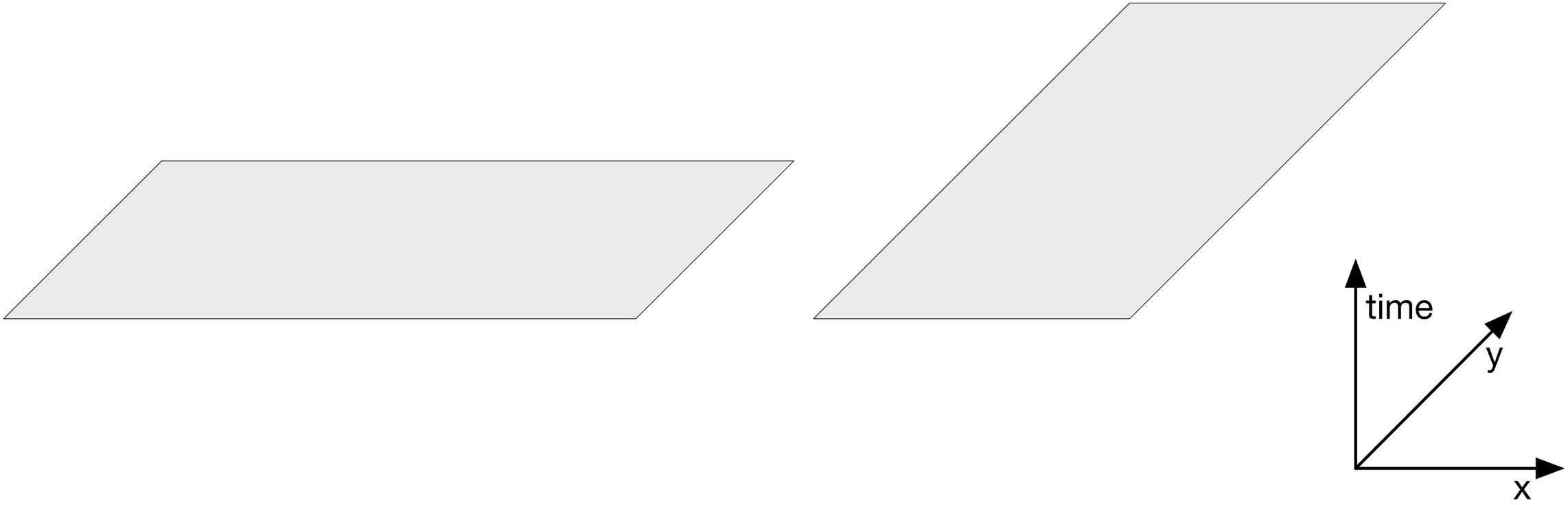

y
x

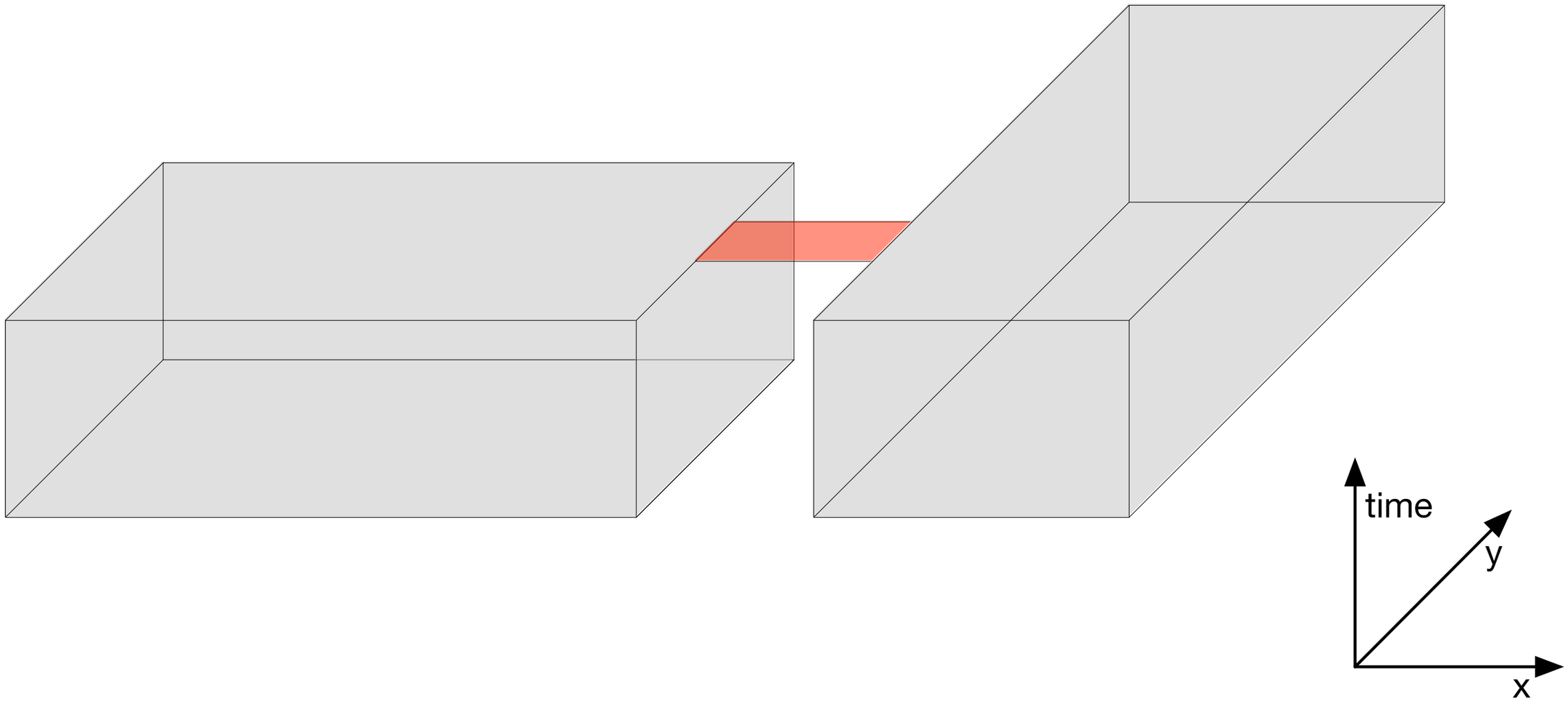


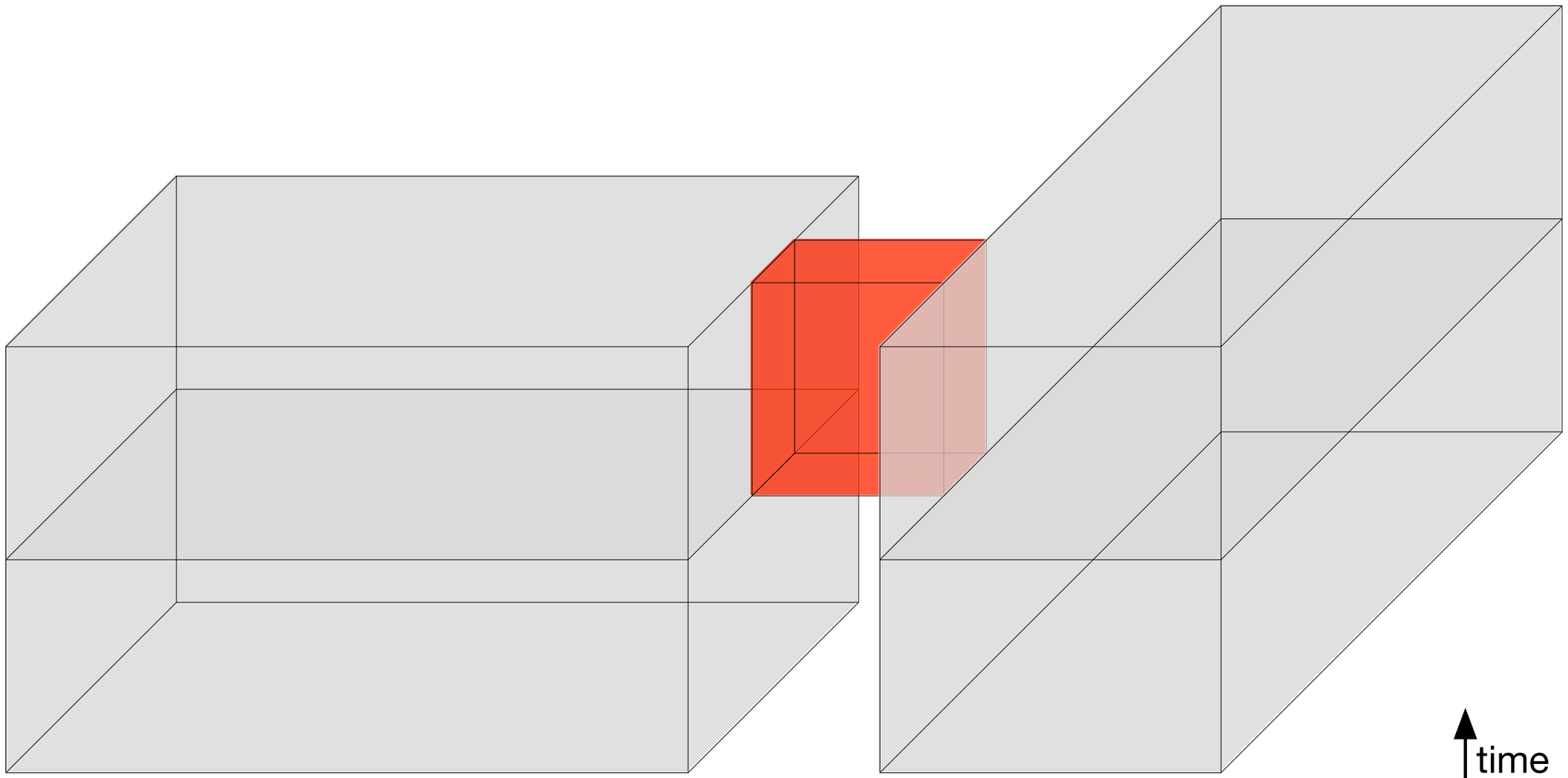

y
x

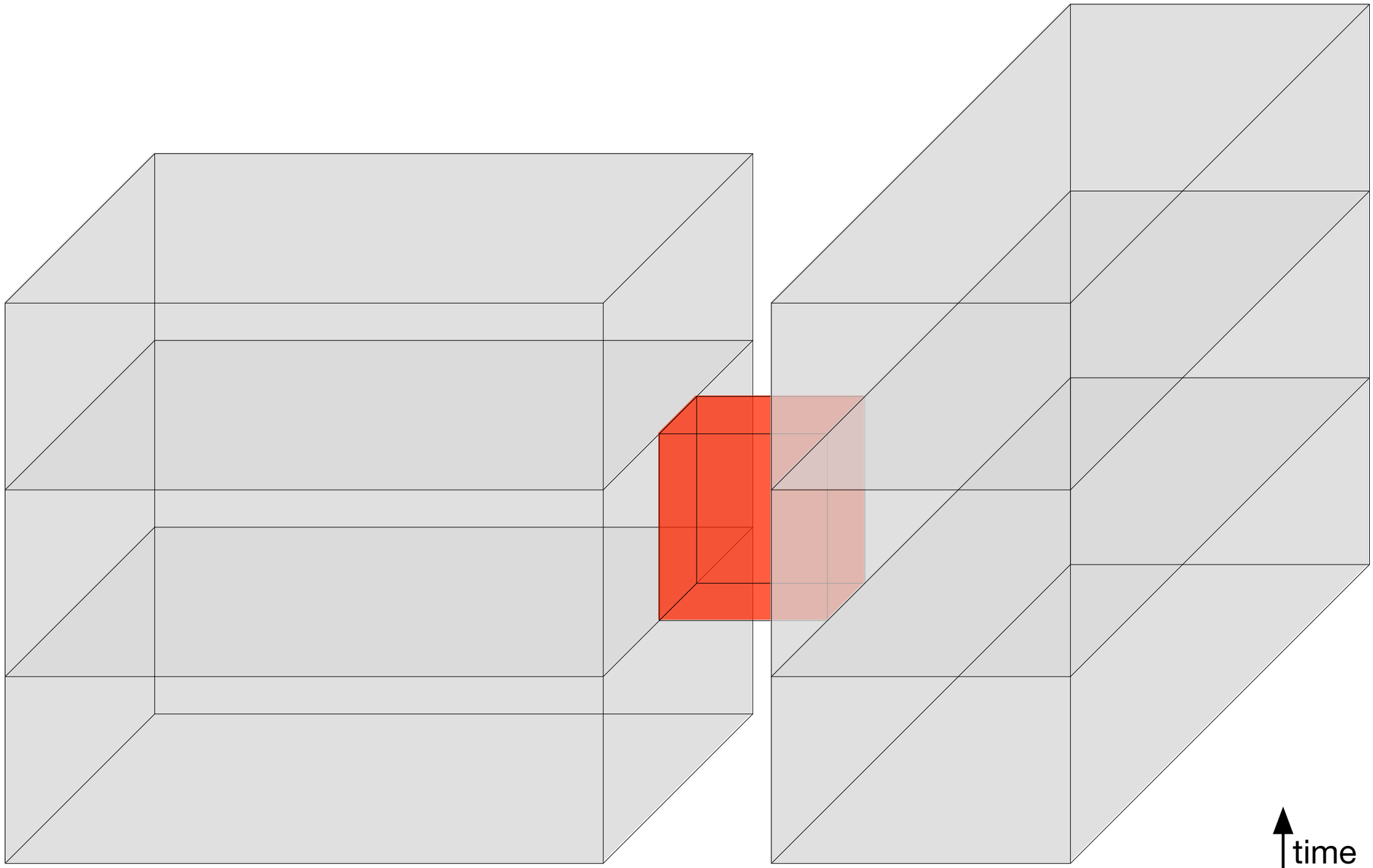


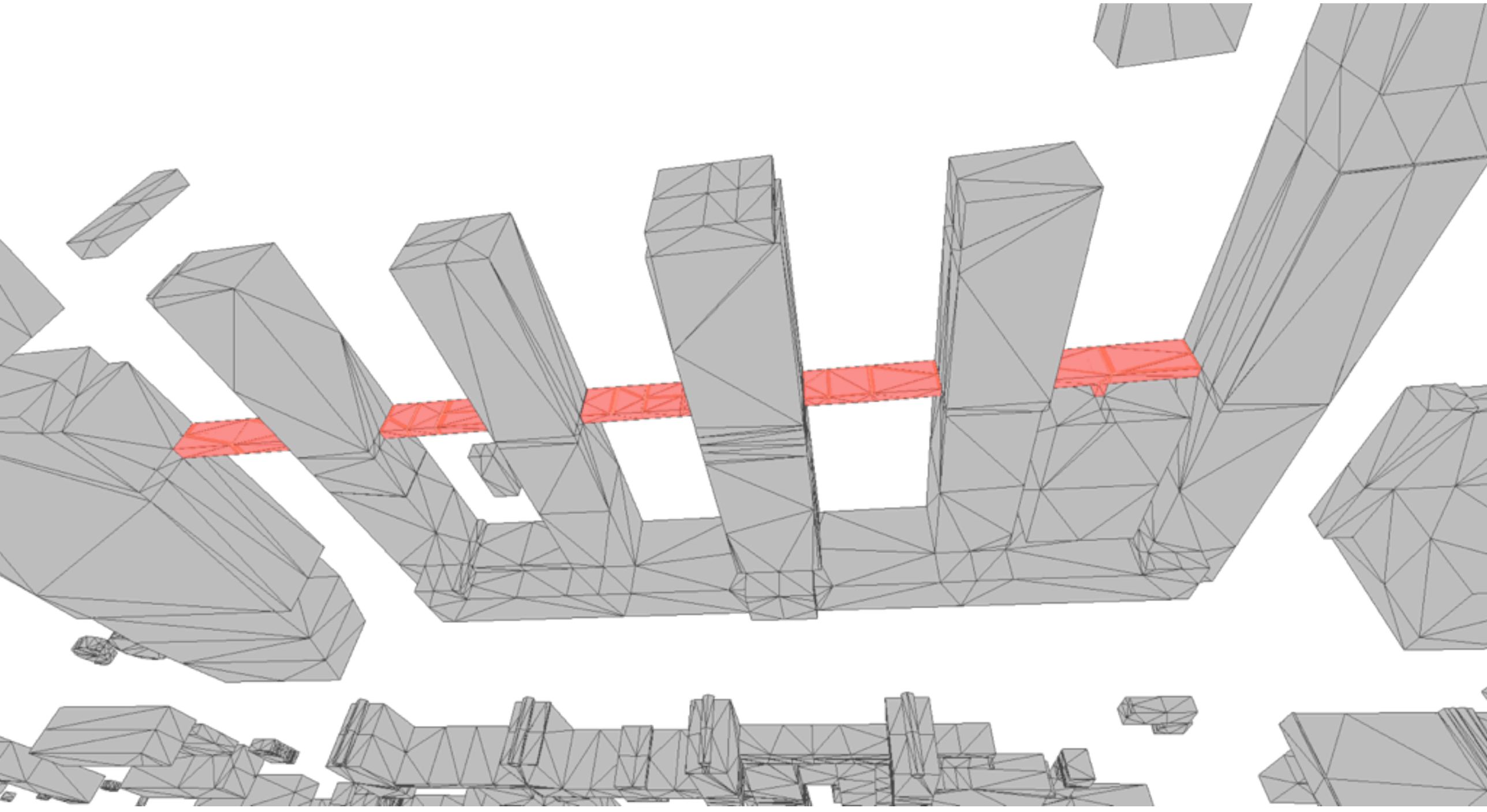

y
x

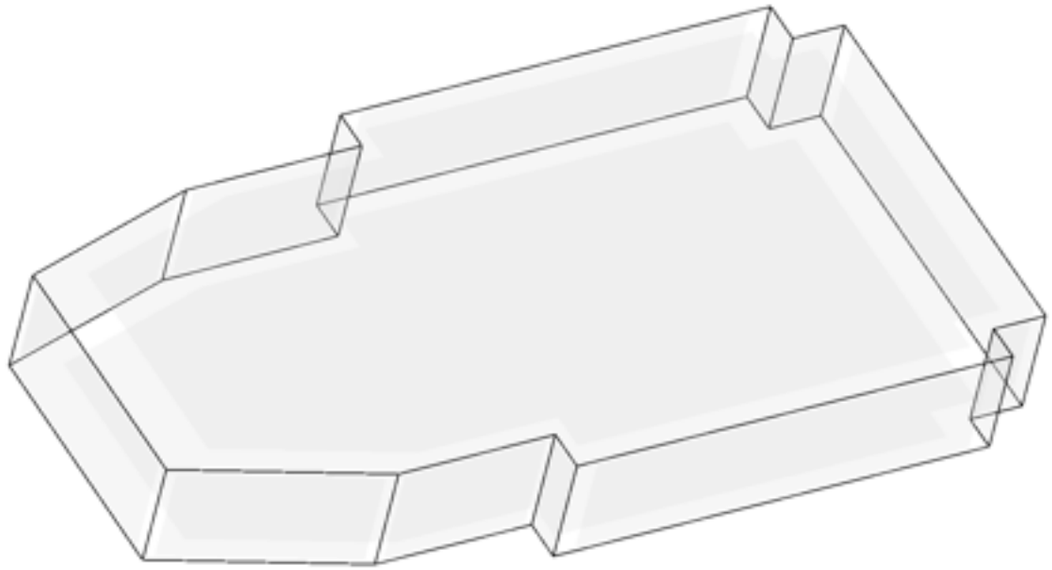

time
y
x

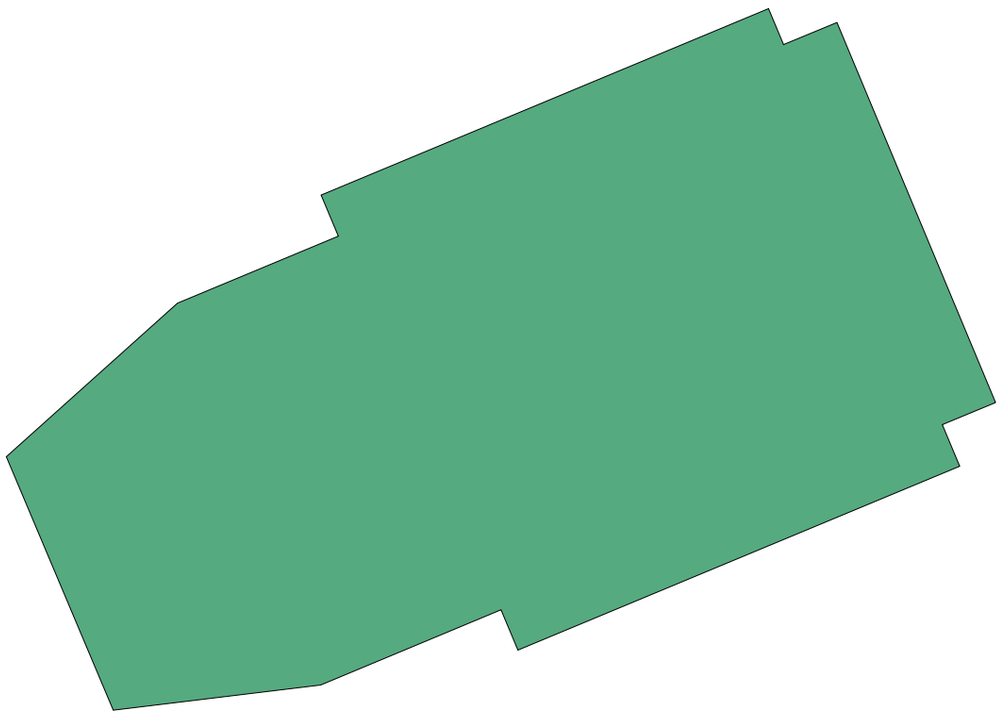

time
y
x

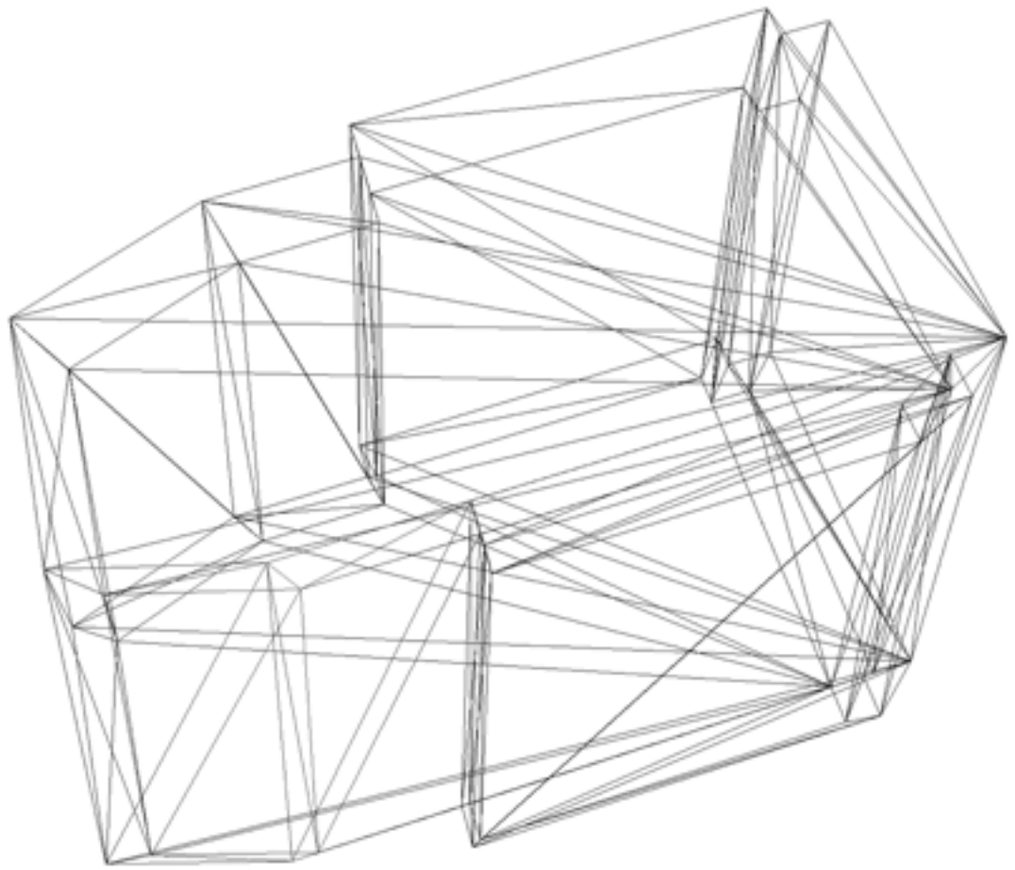

time
y
x


time
y
x









Possible base models

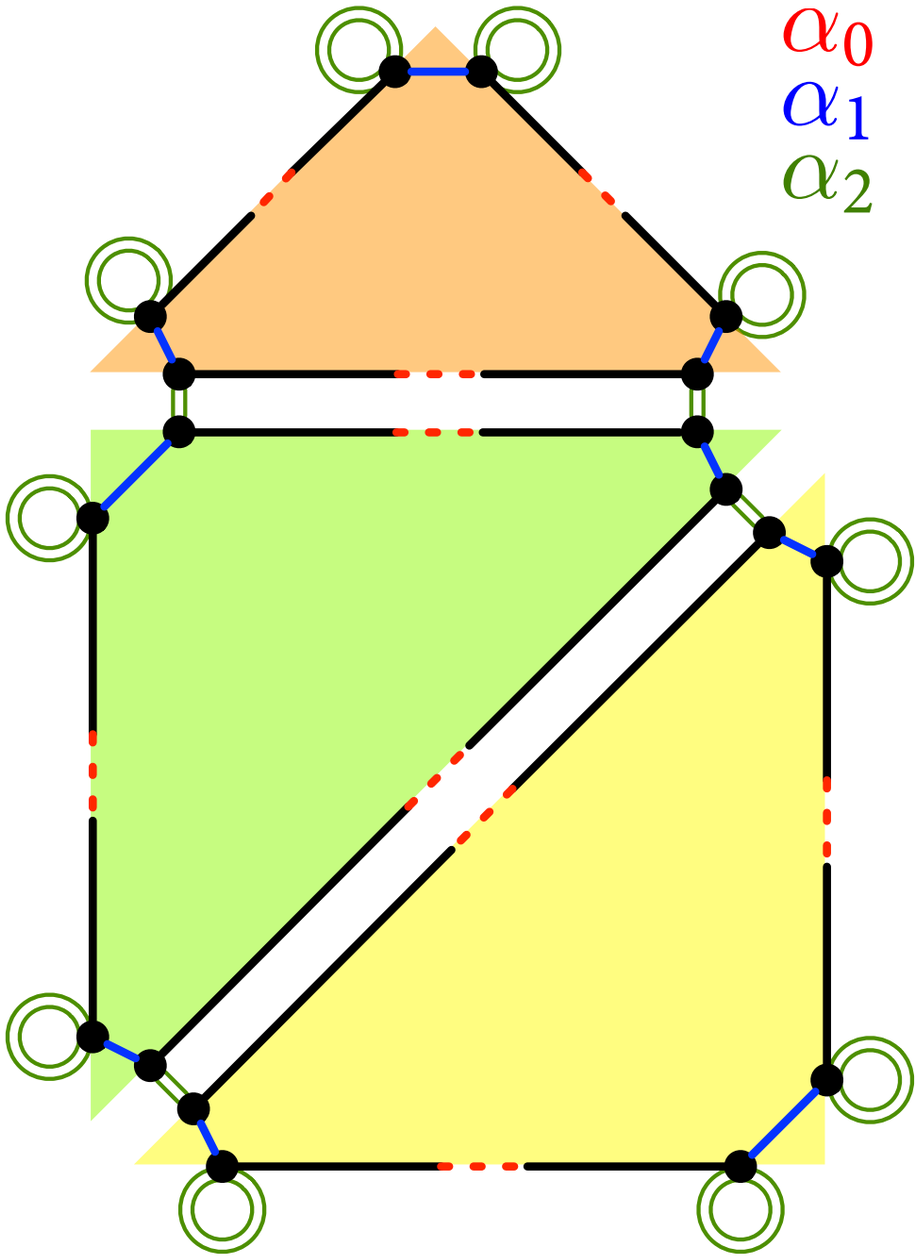
- Simplicial complexes (triangulations)
- Boundary representation
- Constructive solid geometry
- Nef polyhedra

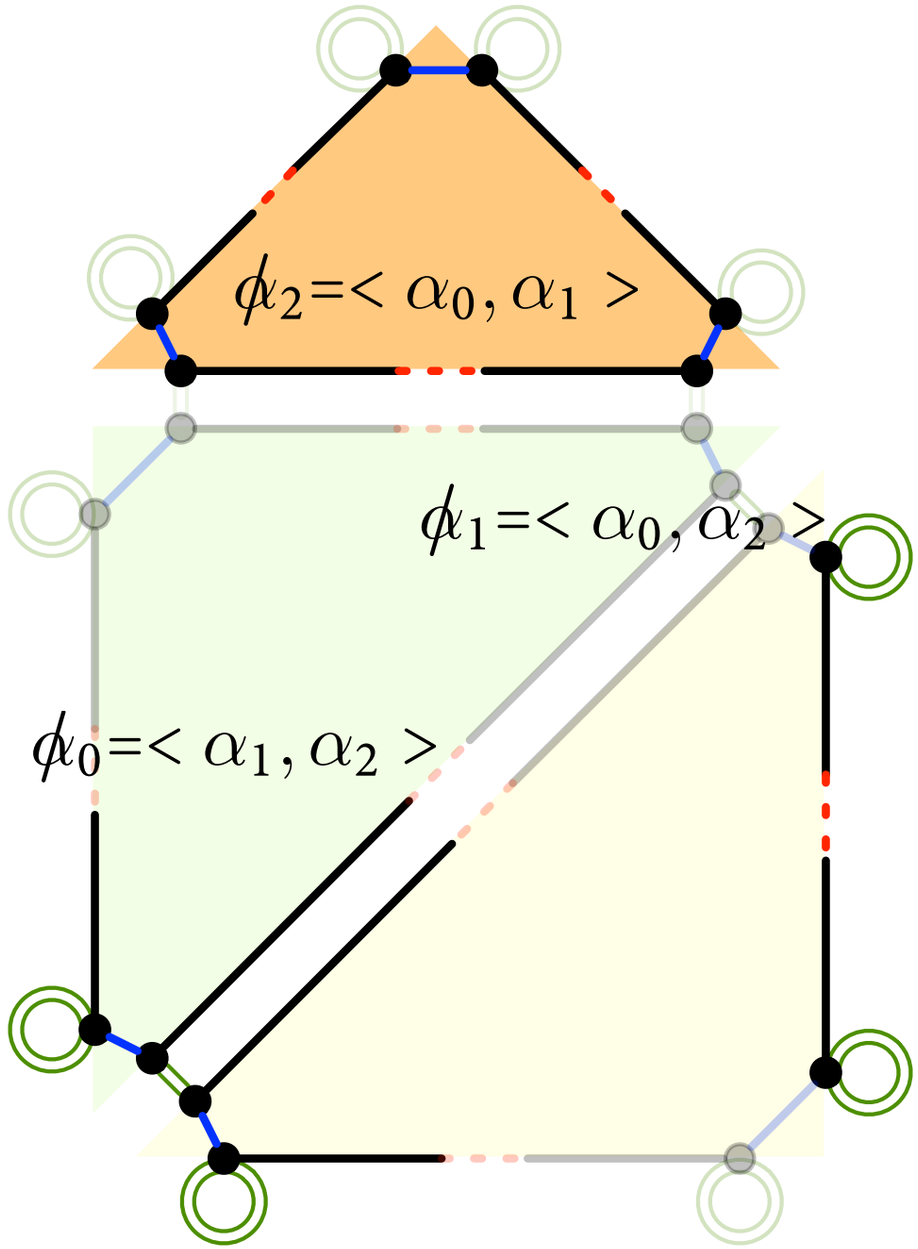
Possible base models

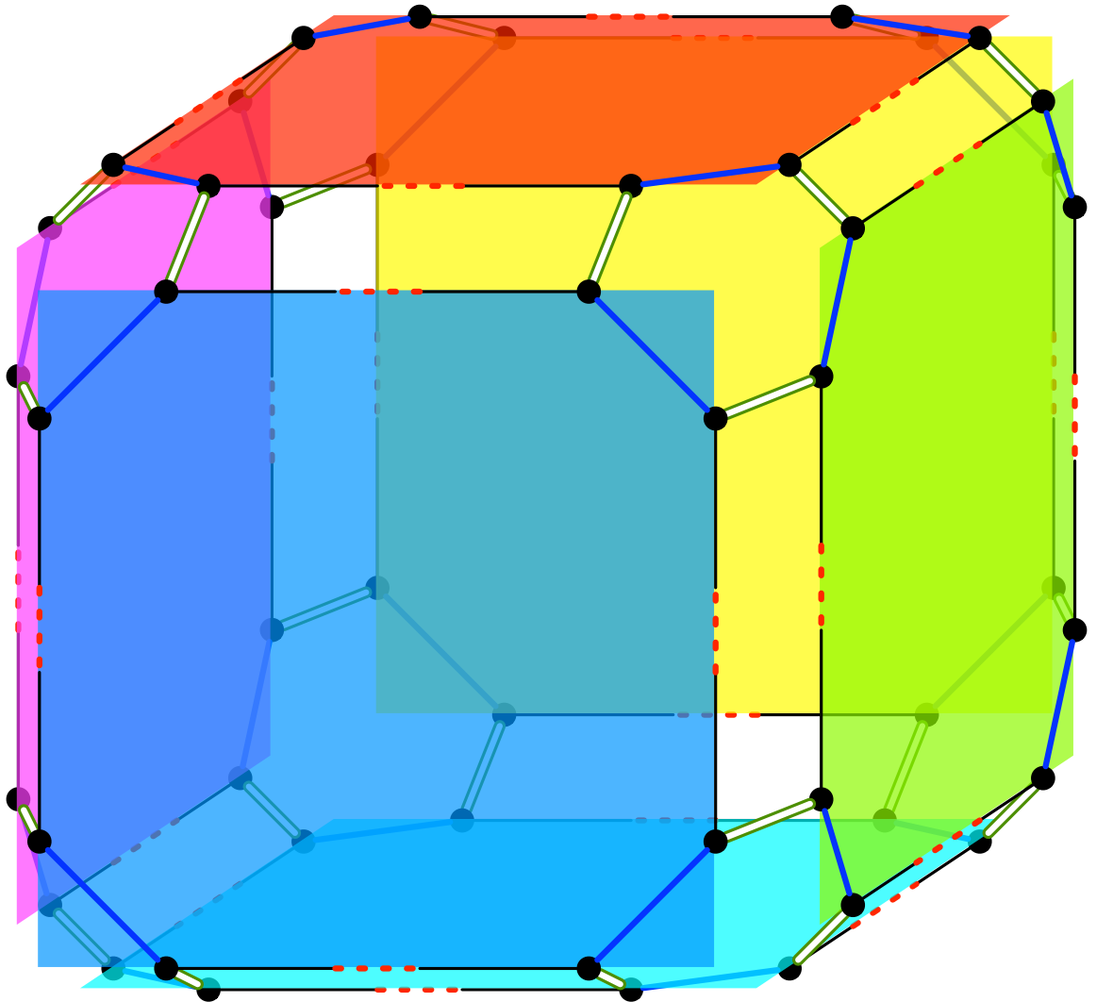
- Simplicial complexes (triangulations)
- **Boundary representation**
- Constructive solid geometry
- Nef polyhedra

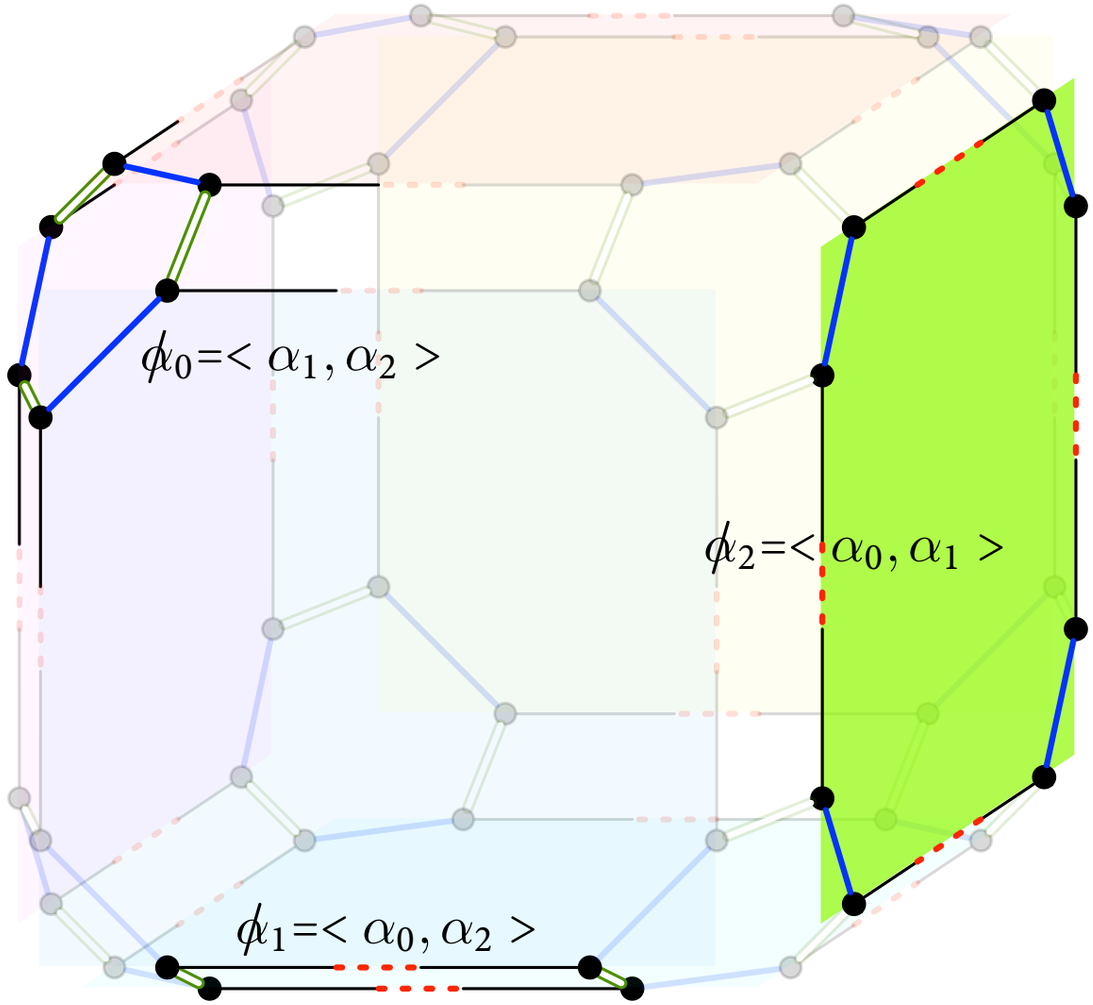
Generalised maps

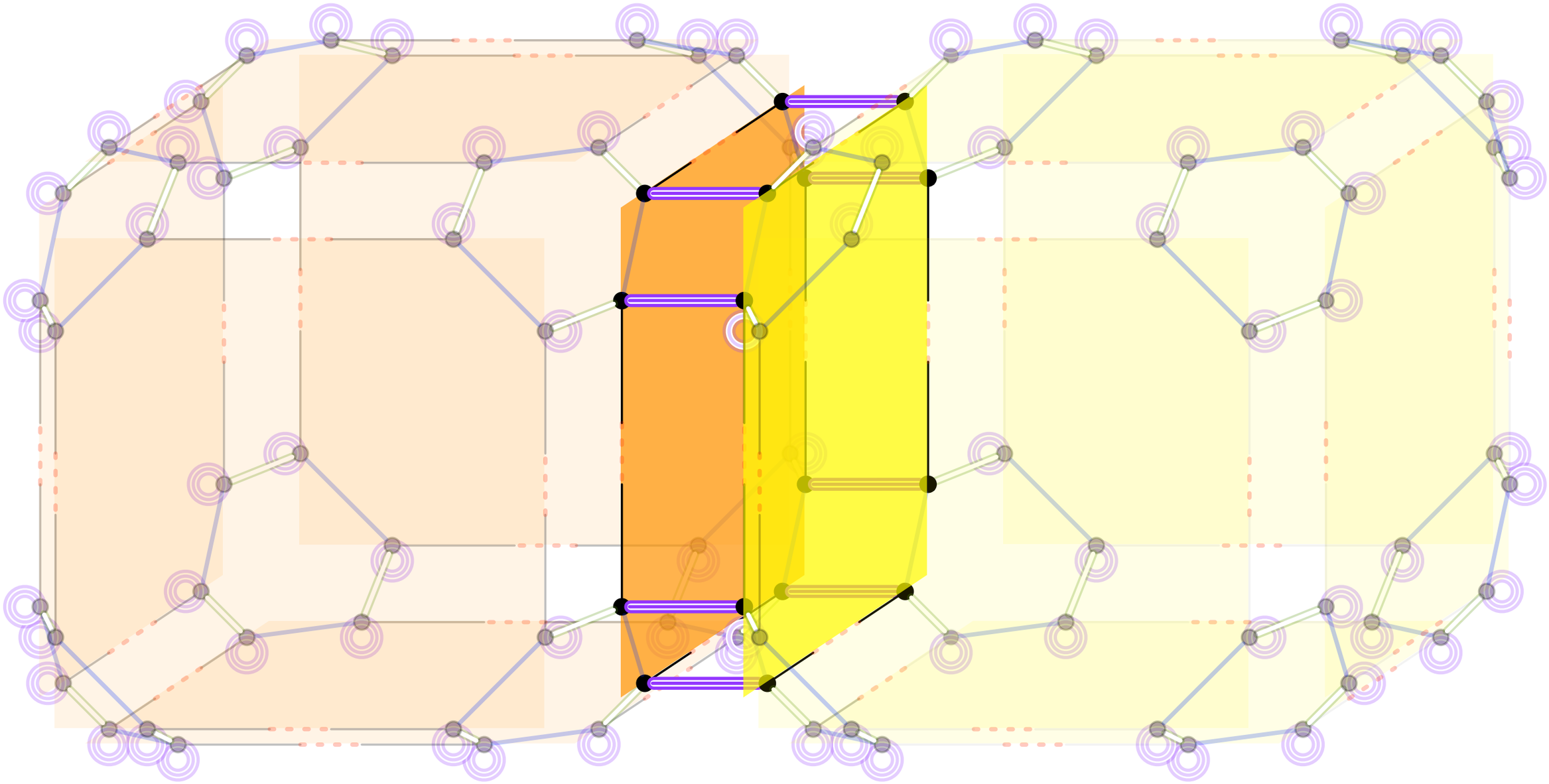
- Strong mathematical foundation (with ongoing work)
- Used in practice: GOCAD (geology) & Moka (geometric modeller)


α_0
α_1
α_2


$\phi_2 = < \alpha_0, \alpha_1 >$
$\phi_1 = < \alpha_0, \alpha_2 >$
$\phi_0 = < \alpha_1, \alpha_2 >$




$\phi_0 = < \alpha_1, \alpha_2 >$
$\phi_2 = < \alpha_0, \alpha_1 >$
$\phi_1 = < \alpha_0, \alpha_2 >$



What can we represent with usual GIS data structures?

- 2D (3D objects are usually represented by their 2D boundary)
- Manifold
  - Not all topological relationships can be kept
- Orientable
  - Data sets are filled with badly oriented objects
- Holes?

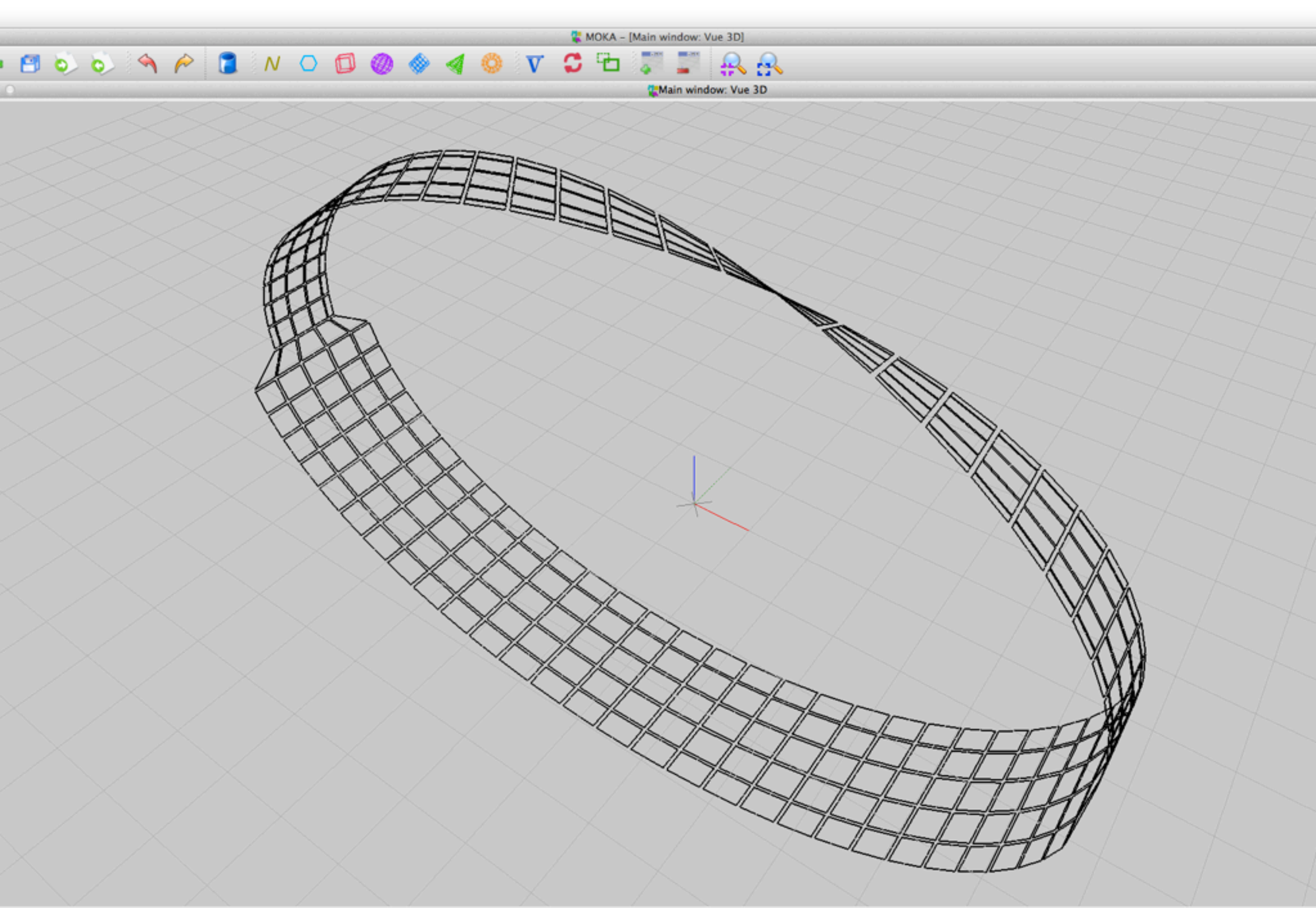
In higher dimensions, objects become more complex

- 2D topology is no longer sufficient
- There might be holes in any dimension
- Visualisation is not trivial

Wikimedia Commons

MOKA – [Main window: Vue 3D]

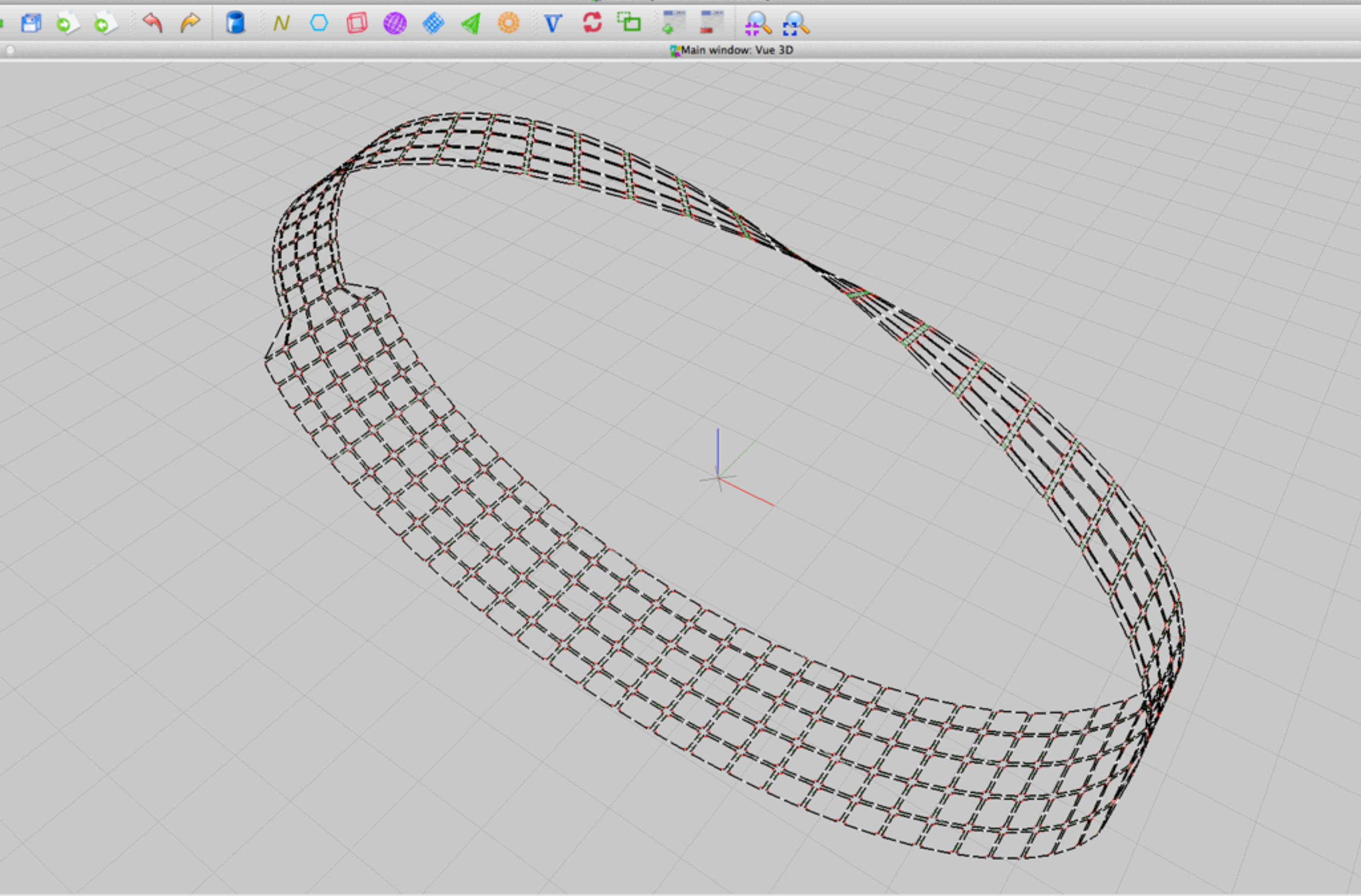
Main window: Vue 3D



ed: 0; Vertices: 360; Darts unmarked

MOKA – [Main window: Vue 3D]

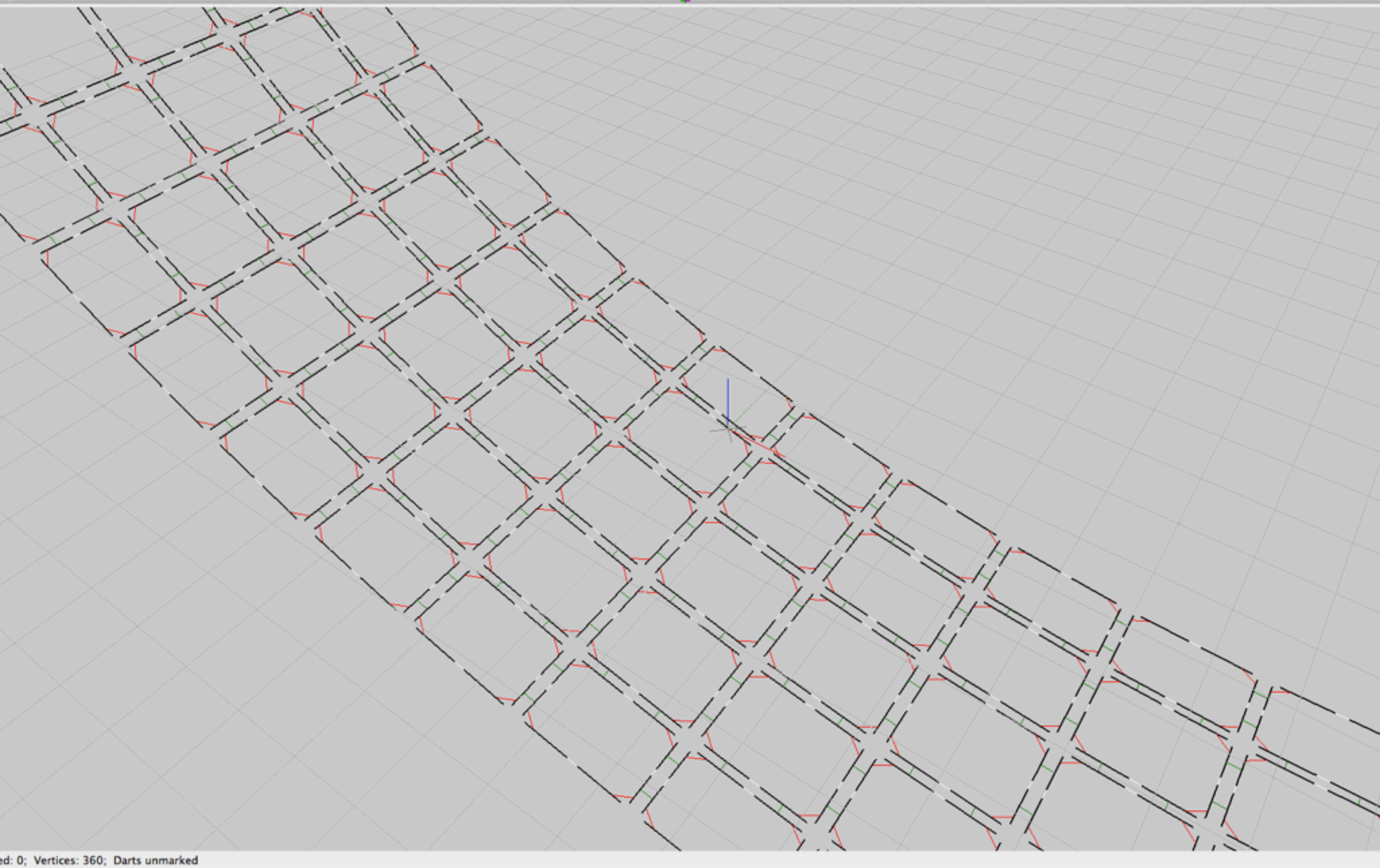
Main window: Vue 3D



ed: 0; Vertices: 360; Darts unmarked

MOKA – [Main window: Vue 3D]

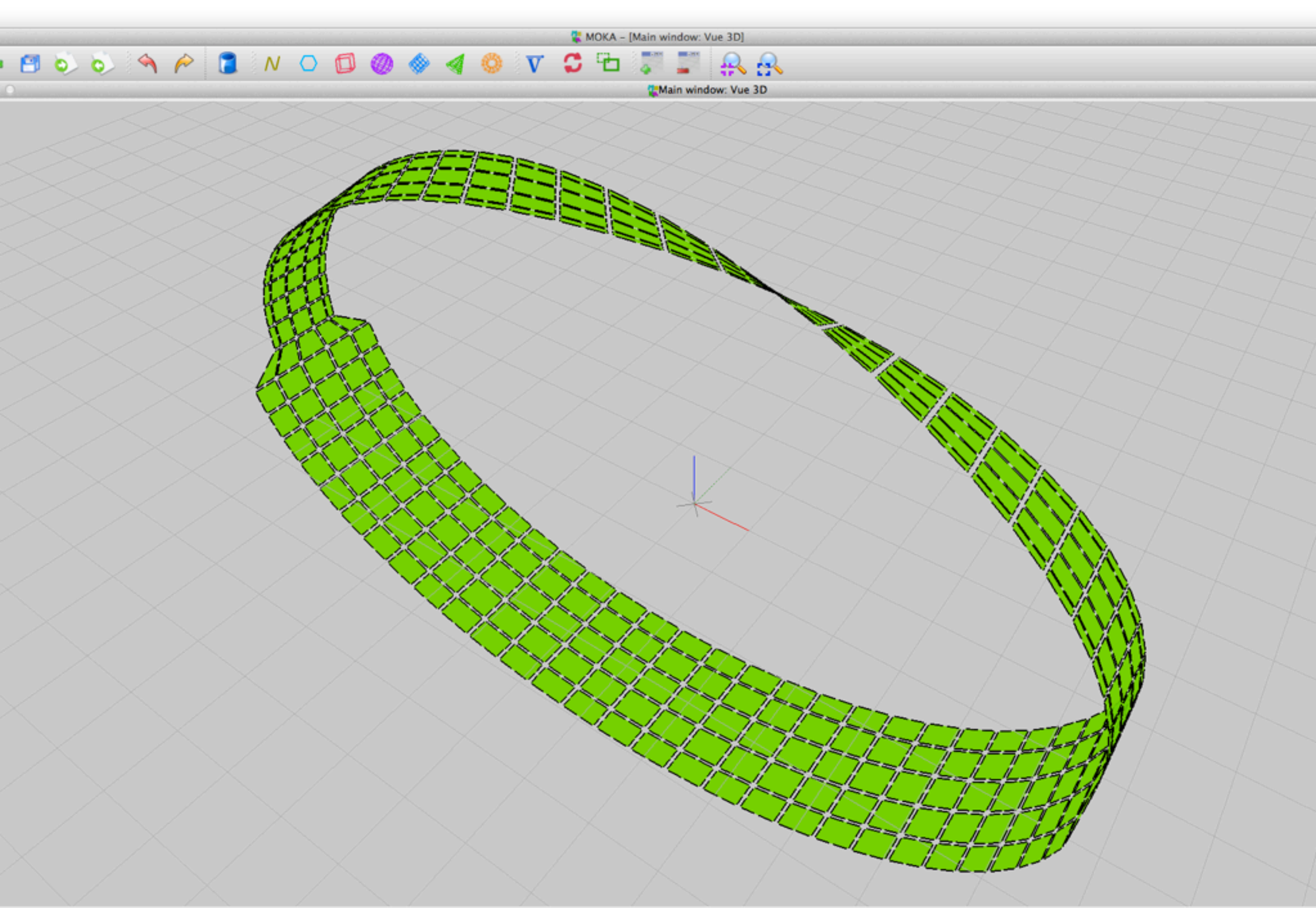
Main window: Vue 3D



ed: 0; Vertices: 360; Darts unmarked

MOKA – [Main window: Vue 3D]

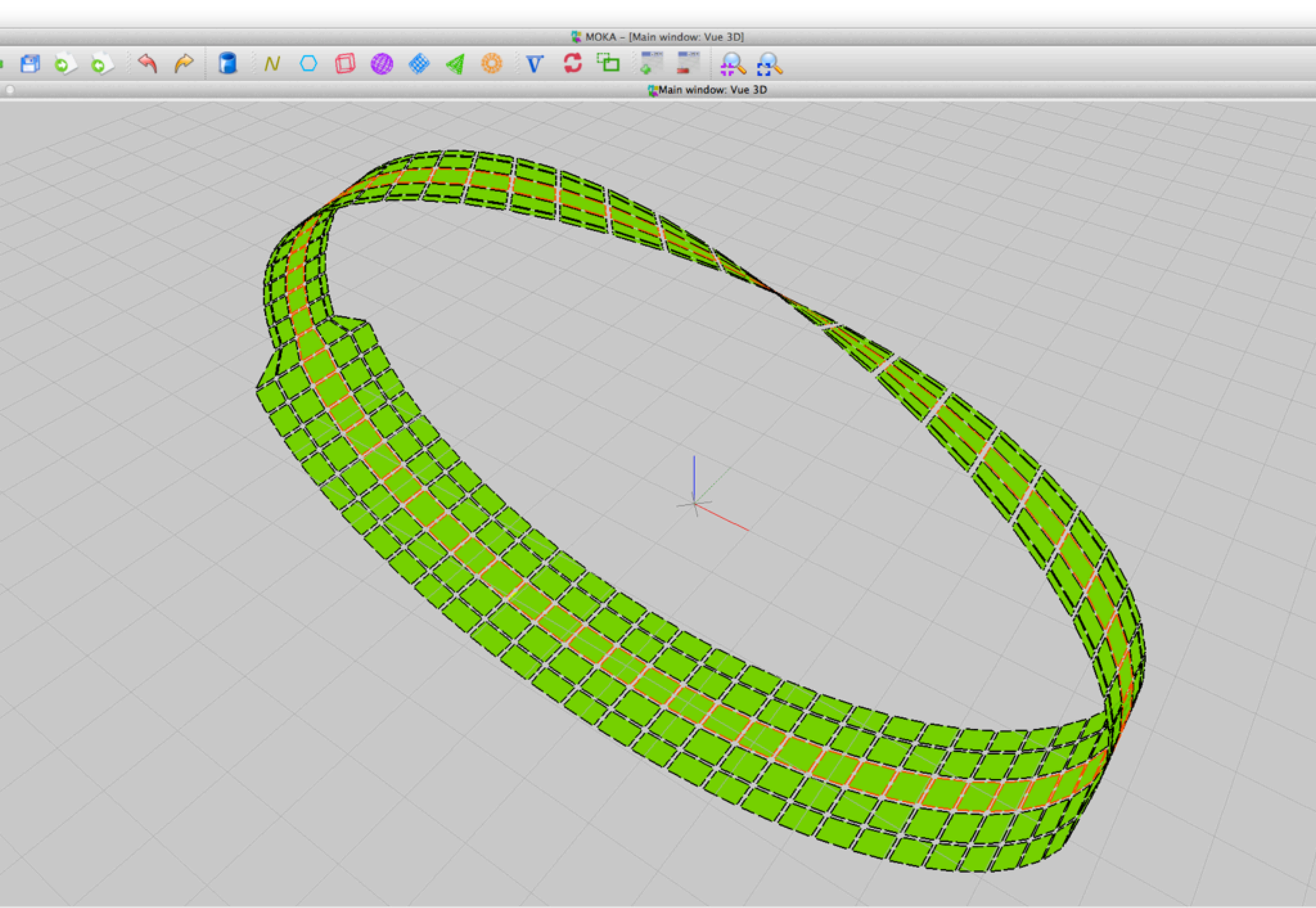
Main window: Vue 3D



ed: 0; Vertices: 360; Darts unmarked

MOKA – [Main window: Vue 3D]

Main window: Vue 3D



ed: 480; Vertices: 360; Forward selection done

Topological characteristics

Globals

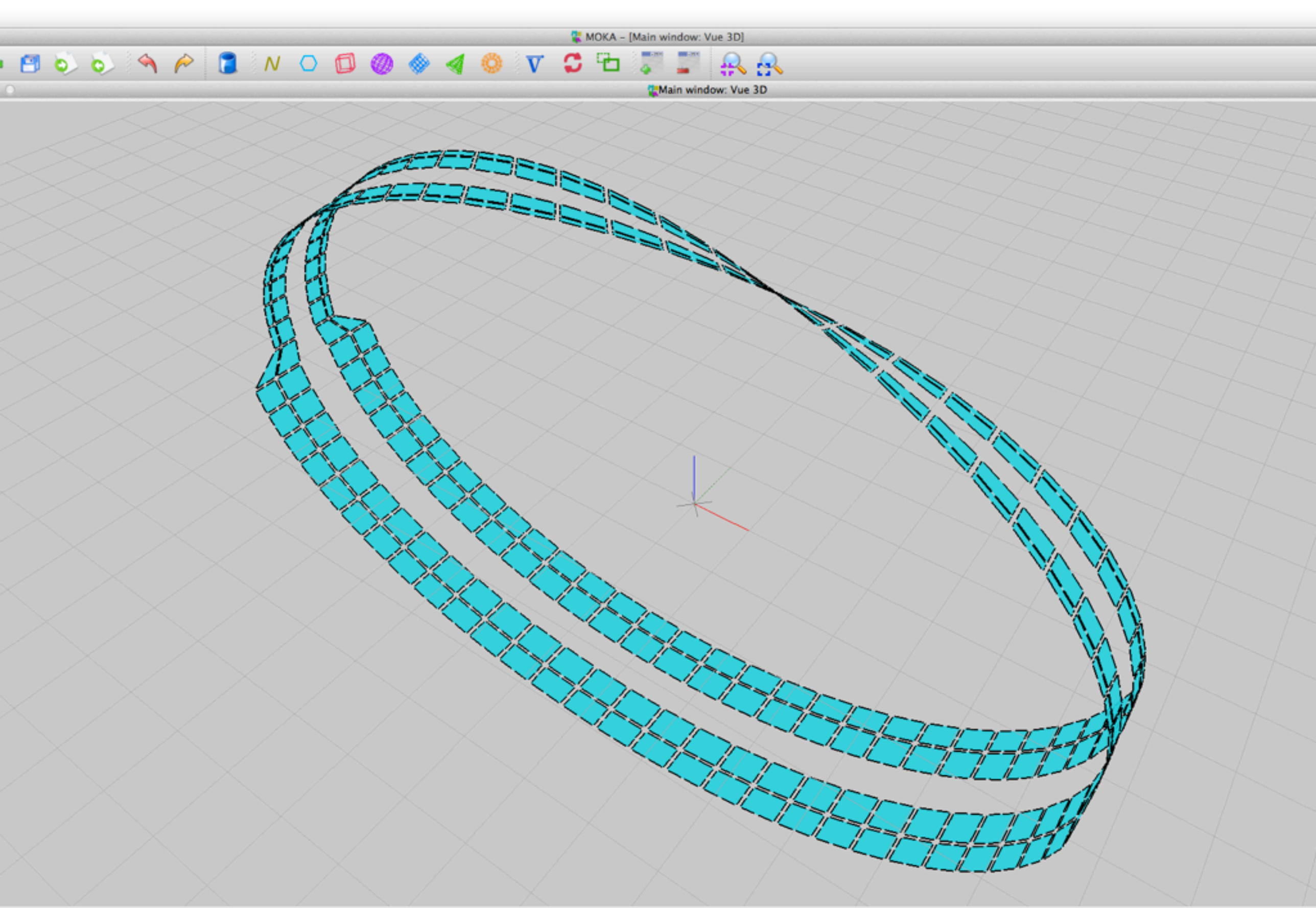
darts:	2400		
vertices:	360	0-borders:	0
edges:	660	1-borders:	0
faces:	300	2-borders:	1
volumes:	1	3-borders:	1
composants:	1		

Volume incident to 'last'

darts:	2400	0-borders:	0
vertices:	360	1-borders:	0
edges:	660	2-borders:	1
faces:	300		
Euler characteristic:	0		
Orientability coefficient:	1	S(1,1,0): Möbius strip	
Genus:	0		

MOKA – [Main window: Vue 3D]

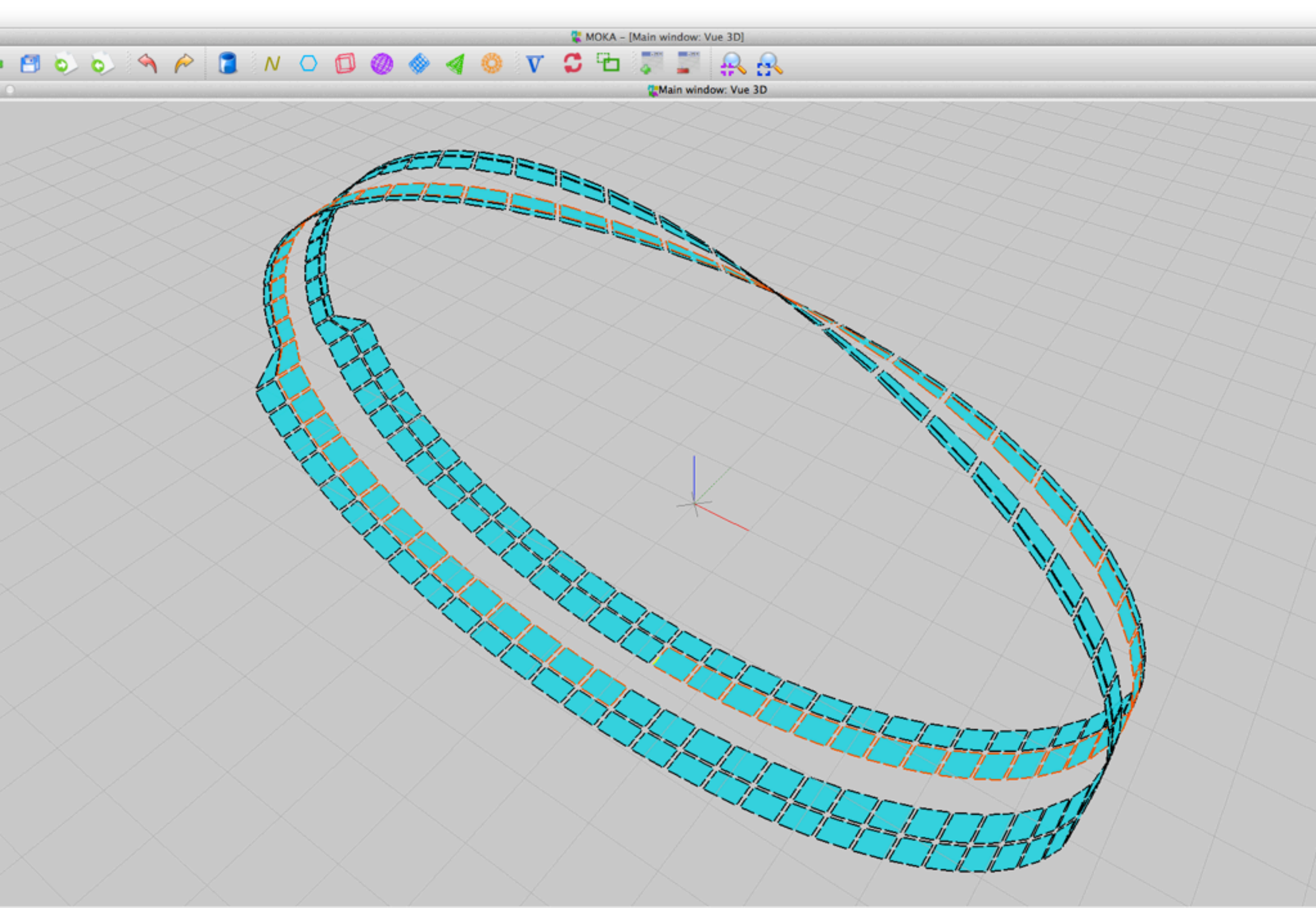
Main window: Vue 3D



ed: 0; Vertices: 360; Redo [65 * 0]

MOKA – [Main window: Vue 3D]

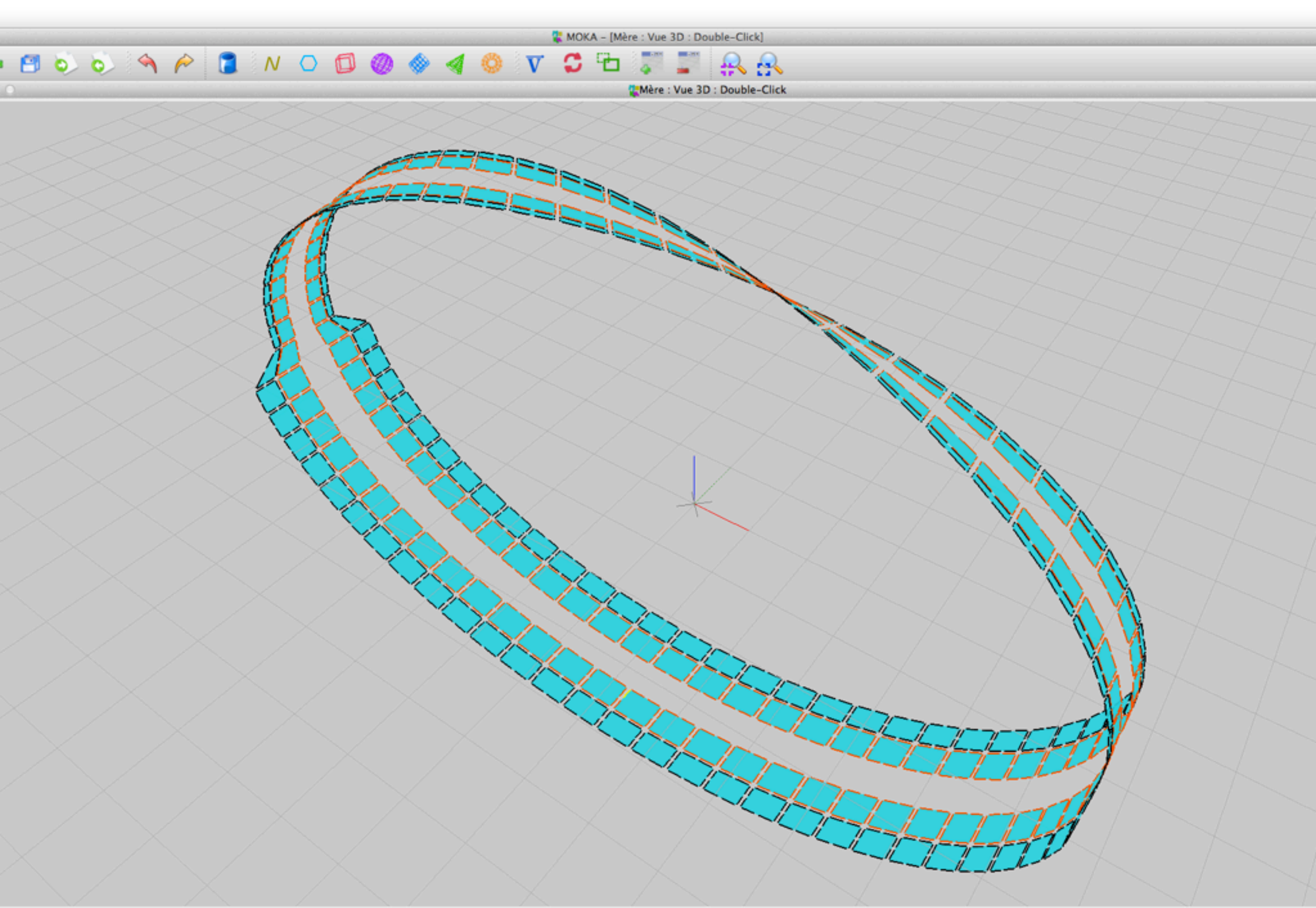
Main window: Vue 3D



ed: 480; Vertices: 360; Forward selection done

MOKA – [Mère : Vue 3D : Double-Click]

Mère : Vue 3D : Double-Click



ed: 960; Vertices: 360; Forward selection done

Topological characteristics

Globals

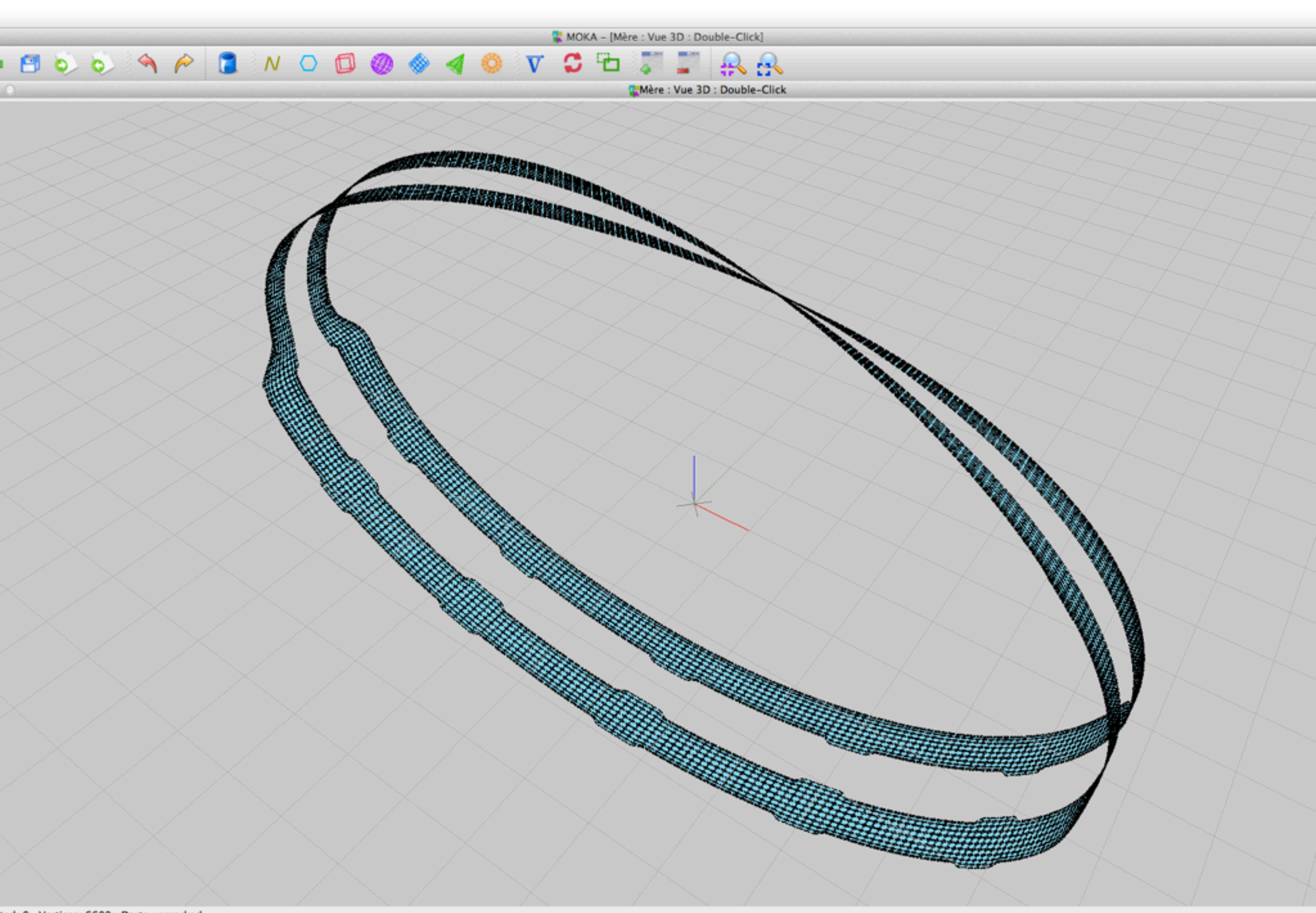
darts:	1920		
vertices:	360	0-borders:	0
edges:	600	1-borders:	0
faces:	240	2-borders:	2
volumes:	1	3-borders:	1
composants:	1		

Volume incident to 'last'

darts:	1920	0-borders:	0
vertices:	360	1-borders:	0
edges:	600	2-borders:	2
faces:	240		
Euler characteristic:	0		
Orientability coefficient:	0	S(2,0,0): Strip	
Genus:	0		

MOKA – [Mère : Vue 3D : Double-Click]

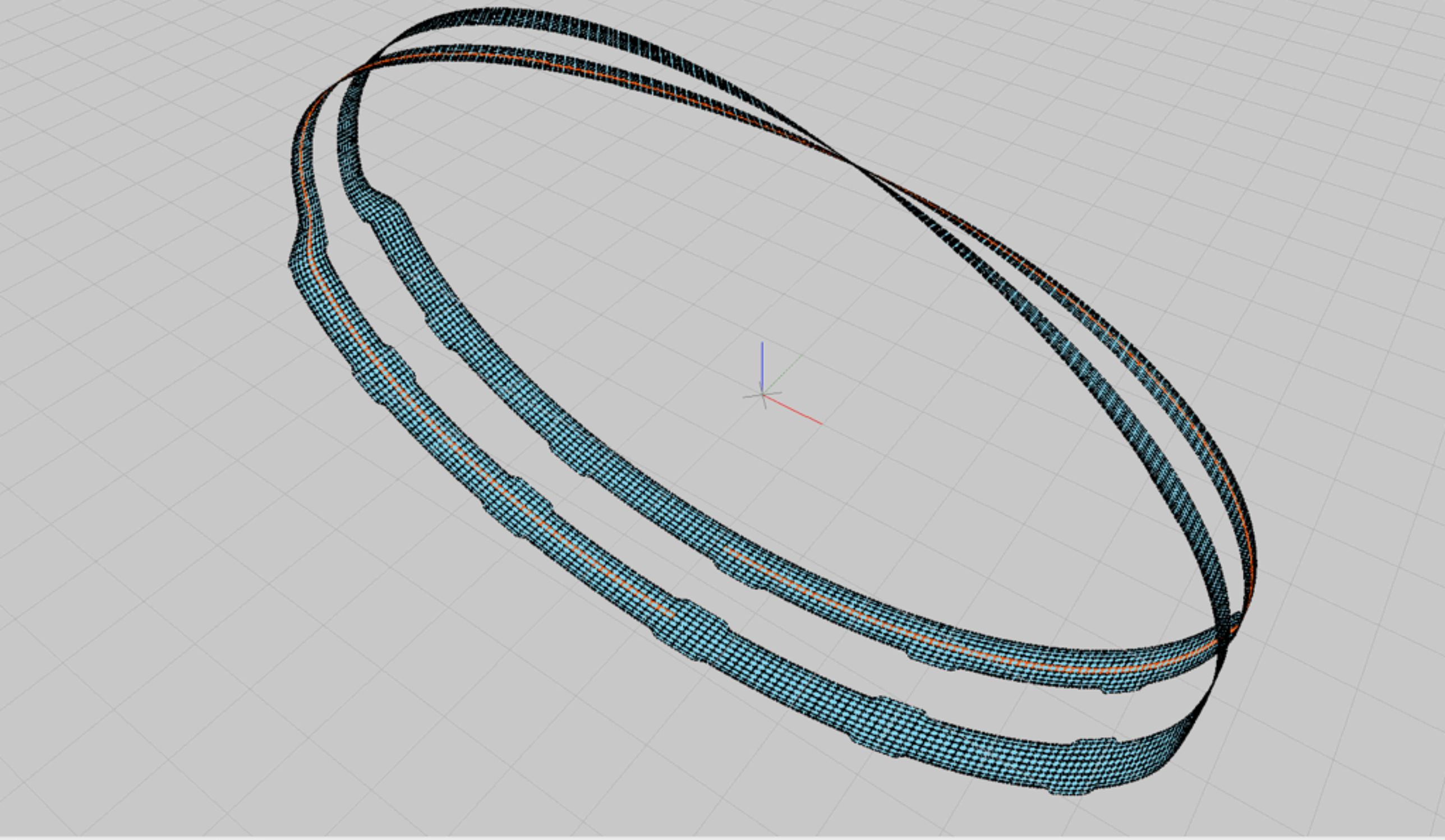
Mère : Vue 3D : Double-Click



ted: 0; Vertices: 6600; Darts unmarked

MOKA – [Mère : Vue 3D : Double-Click]

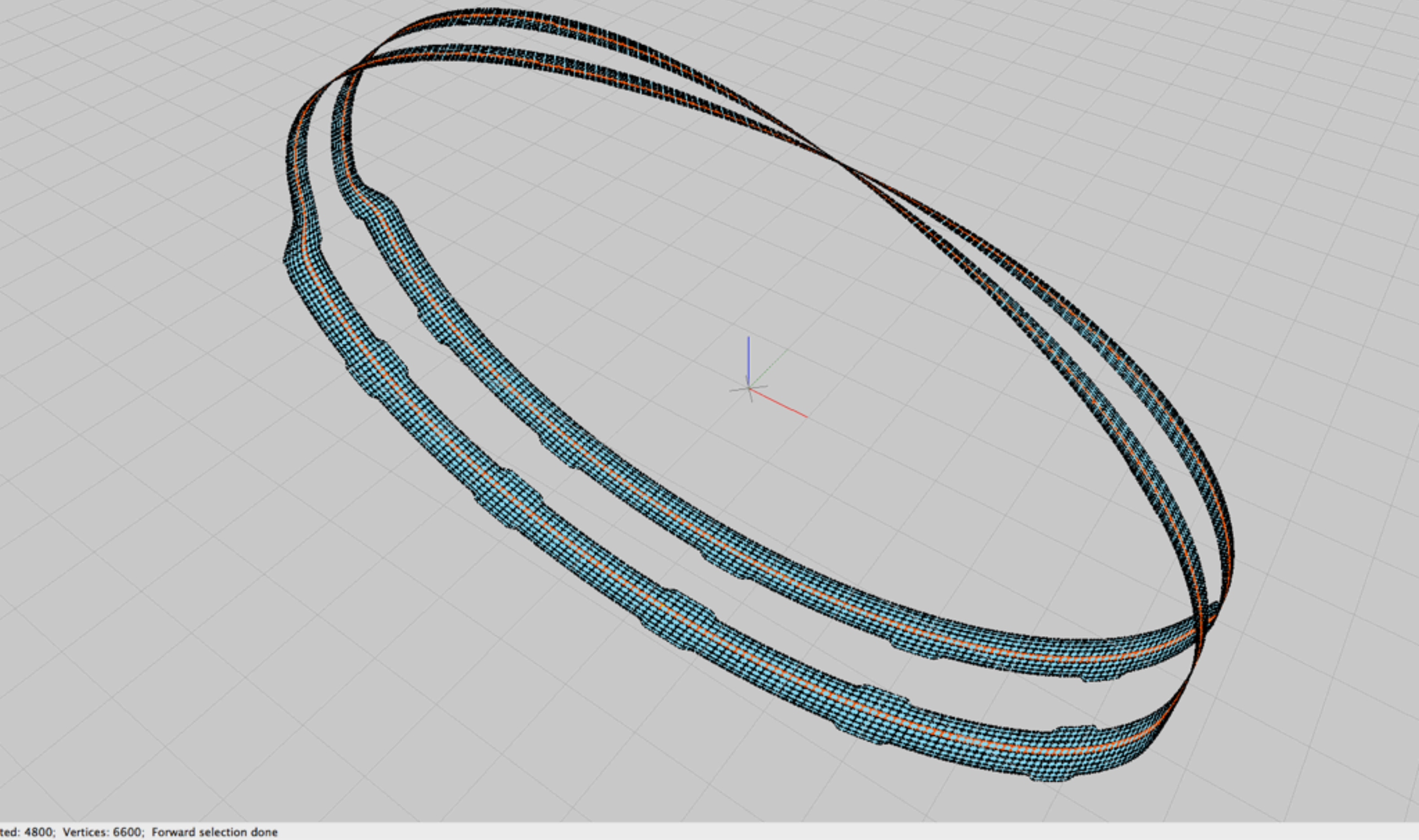
Mère : Vue 3D : Double-Click



ted: 2408; Vertices: 6600; Forward selection done

MOKA – [Mère : Vue 3D : Double-Click]

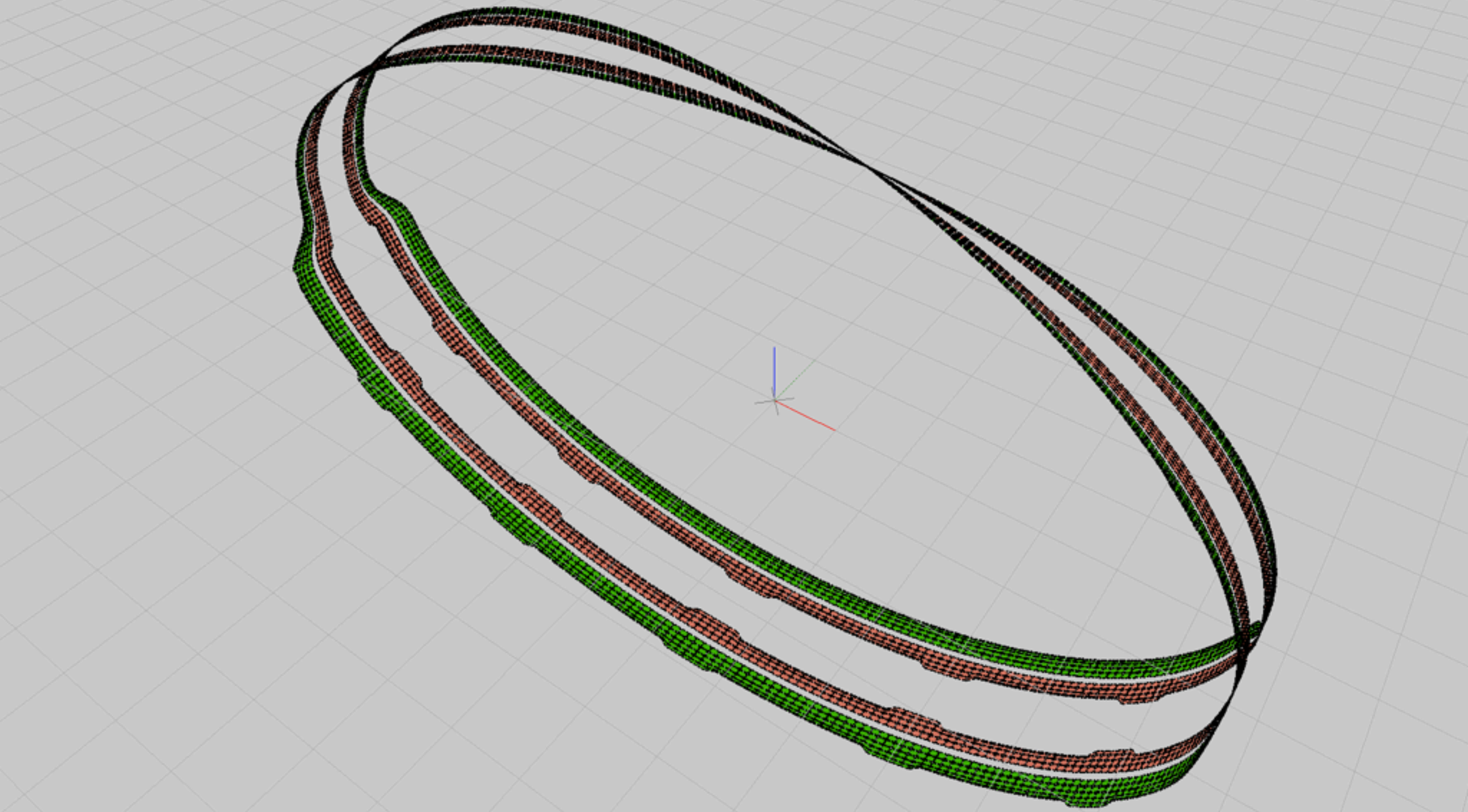
Mère : Vue 3D : Double-Click



ted: 4800; Vertices: 6600; Forward selection done

MOKA – [Mère : Vue 3D : Double-Click]

Mère : Vue 3D : Double-Click



ted: 0; Vertices: 6600; Redo [68 * 0]

Topological characteristics

Globals

darts:	43200		
vertices:	6600	0-borders:	0
edges:	12000	1-borders:	0
faces:	5400	2-borders:	4
volumes:	2	3-borders:	2
composants:	2		

Volume incident to 'last'

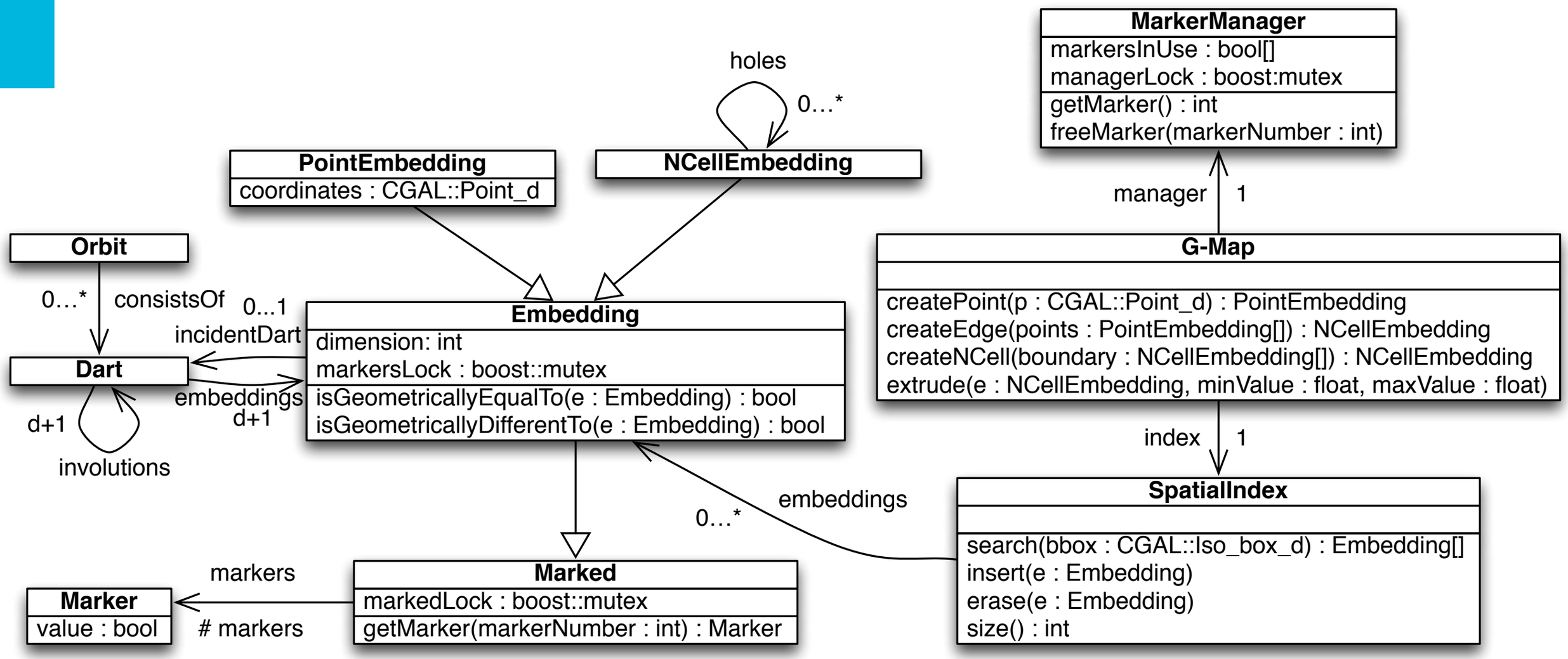
darts:	24000	0-borders:	0
vertices:	3600	1-borders:	0
edges:	6600	2-borders:	2
faces:	3000		
Euler characteristic:	0		
Orientability coefficient:	0	S(2,0,0):	Strip
Genus:	0		

What is needed?

- Storing geometry, topology and computed values
- Complex handling of attributes: numeric, text, classes
- Construction from invalid or non topological data
- Queries: geometric, topological, attribute based, or a combination
- Holes, in possibly every dimension > 0
- Disconnected objects

Operations

- Traversing the 5D structure
- Construction of 5D objects
- Efficient access to disconnected objects (0D-5D), which might be disconnected
- Visualisation in 2D & 3D (slicing & projections)


MarkerManager
markersInUse : bool[]
managerLock : boost:mutex
getMarker() : int
freeMarker(markerNumber : int)
holes
0…*
PointEmbedding
coordinates : CGAL::Point_d
NCellEmbedding
manager
1
Orbit
0…*
consistsOf
0...1
G-Map
createPoint(p : CGAL::Point_d) : PointEmbedding
createEdge(points : PointEmbedding[]) : NCellEmbedding
createNCell(boundary : NCellEmbedding[]) : NCellEmbedding
extrude(e : NCellEmbedding, minValue : float, maxValue : float)
Embedding
dimension: int
markersLock : boost::mutex
isGeometricallyEqualTo(e : Embedding) : bool
isGeometricallyDifferentTo(e : Embedding) : bool
incidentDart
Dart
embeddings
d+1
d+1
involutions
index
1
SpatialIndex
search(bbox : CGAL::Iso_box_d) : Embedding[]
insert(e : Embedding)
erase(e : Embedding)
size() : int
embeddings
0…*
markers
Marker
value : bool
markers
Marked
markedLock : boost::mutex
getMarker(markerNumber : int) : Marker

Thank you! Questions?