FlatSurf Demo 2

last edited Jun 10, 2016, 10:38:41 AM by admin

Constructing the square:

```
from flatsurf.geometry.polygon import Polygons
```

The following is the parent for polygons with coordinates in the field K.

Remarks:

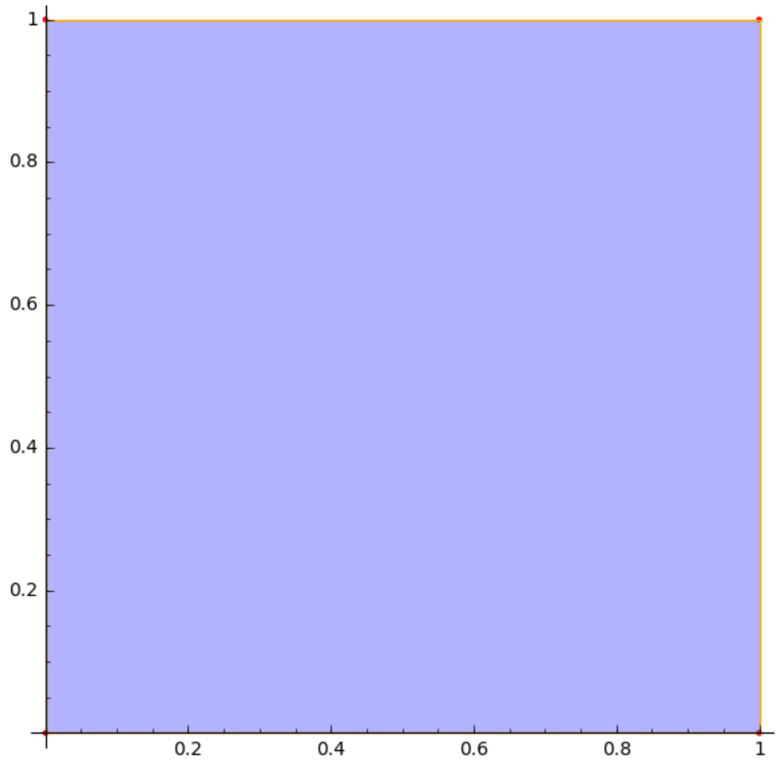
- The first vertex of our polygons is always the origin.
- All our polygons are convex.

To construct a polygon, use the parent to build the parent. Passing a list of edge vectors will produce the polygon. The edge vectors must sum to zero.

```
square = Polygons(QQ)([(1,0),
(0, 1),
(-1,0),
(0, -1)
])
```

```
    Polygon: (0, 0), (1, 0), (1, 1), (0, 1)
```

```
square.plot()
```



Defining the staircase

```
from flatsurf.geometry.surface import Surface
from flatsurf import *
```

evaluate

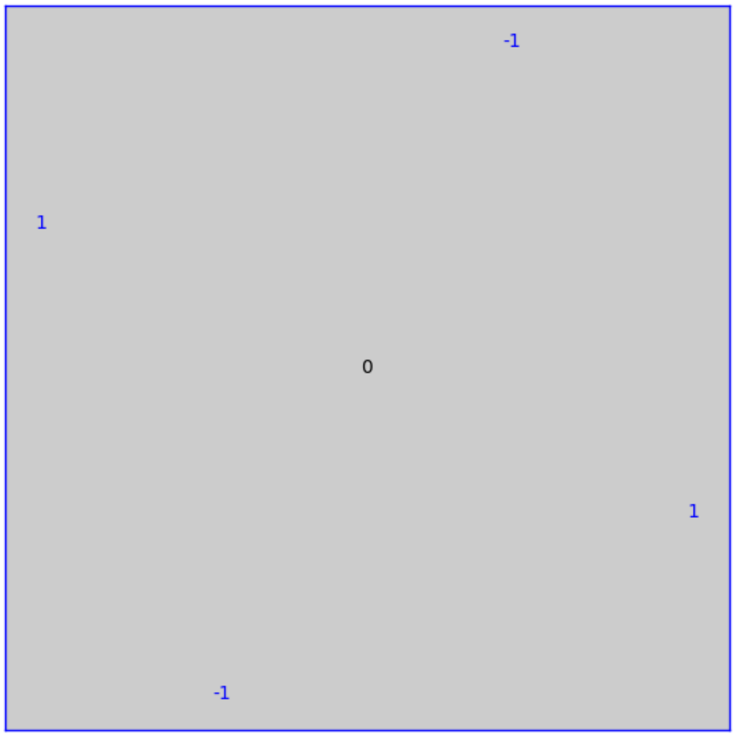
```
class StaircaseSurface(Surface):
    r"""The Staircase surface."""

    def __init__(self):
        # Store the square:
        self._square = Polygons(QQ)([(1,0), (0, 1), (-1,0), (0, -1)])
        # The surface will be defined by polygons with vertices with rational coordinates,
        # will have a base label as zero, and will be infinite
        Surface.__init__(self, QQ, 0, finite=False)

    def polygon(self, lab):
        return self._square

    def opposite_edge(self, p, e):
        if e==0 or e==2:
            if p%2==0:
               return p-1, (e+2)%4
            else:
                return p+1, (e+2)%4
        else:
            if p%2==0:
                return p+1, (e+2)%4
            else:
                return p-1, (e+2)%4
```

We think of this surface as a TranslationSurface.

```
s = TranslationSurface(StaircaseSurface())
```

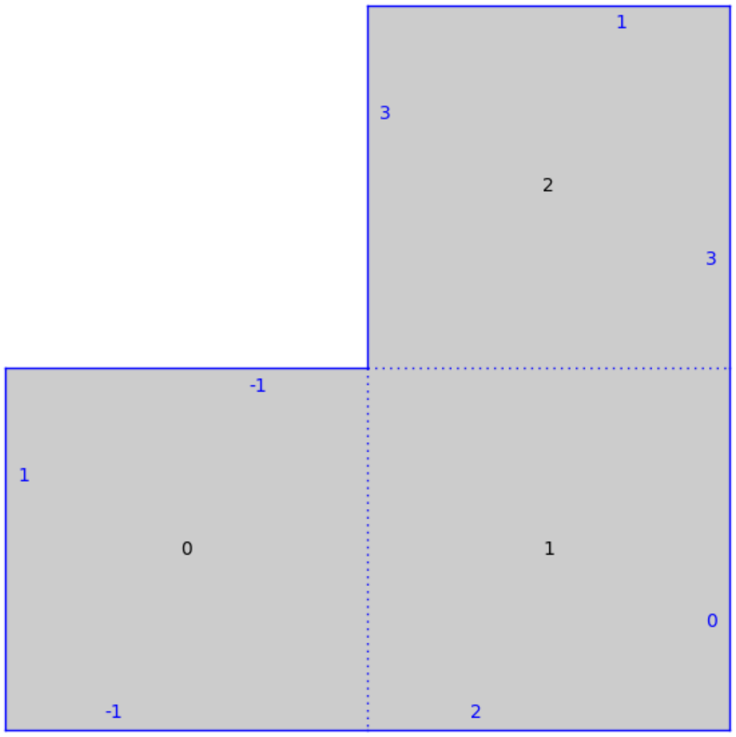
```
gs = s.graphical_surface()
```

```
gs.plot()
```

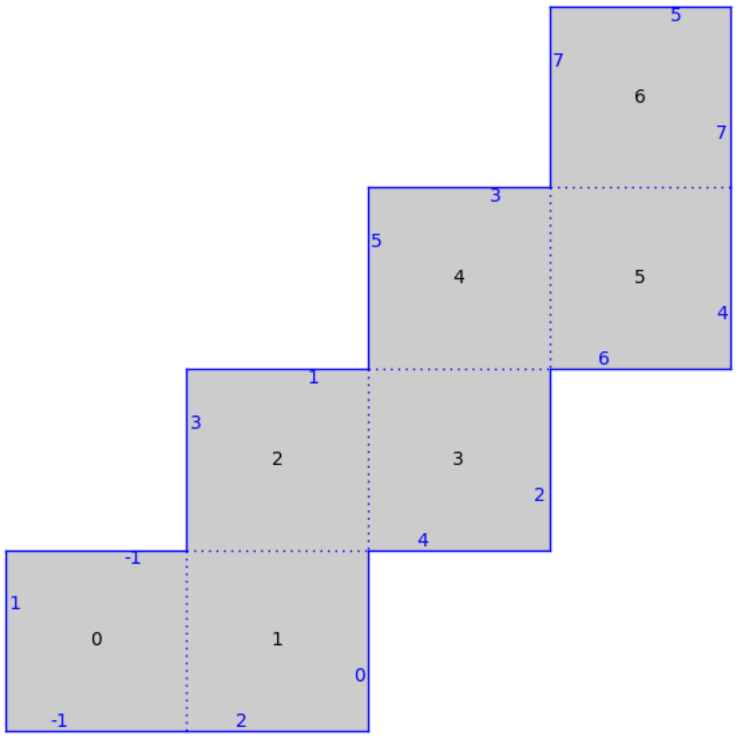


```
gs.make_adjacent_and_visible(0,1)
gs.make_adjacent_and_visible(1,2)
```

```
gs.plot()
```



```
gs = s.graphical_surface()
for i in range(3):
    gs.make_adjacent_and_visible(2*i,1)
    gs.make_adjacent_and_visible(2*i+1,2)
gs.plot()
```



Straight-Line Flow

```
from flatsurf.geometry.tangent_bundle import SimilaritySurfaceTangentBundle
```

We will flow in a direction of slope given by the golden mean, phi. This builds a number field and defines phi.

```
K.<phi> = NumberField(x**2-x-1, embedding=1.6)
```

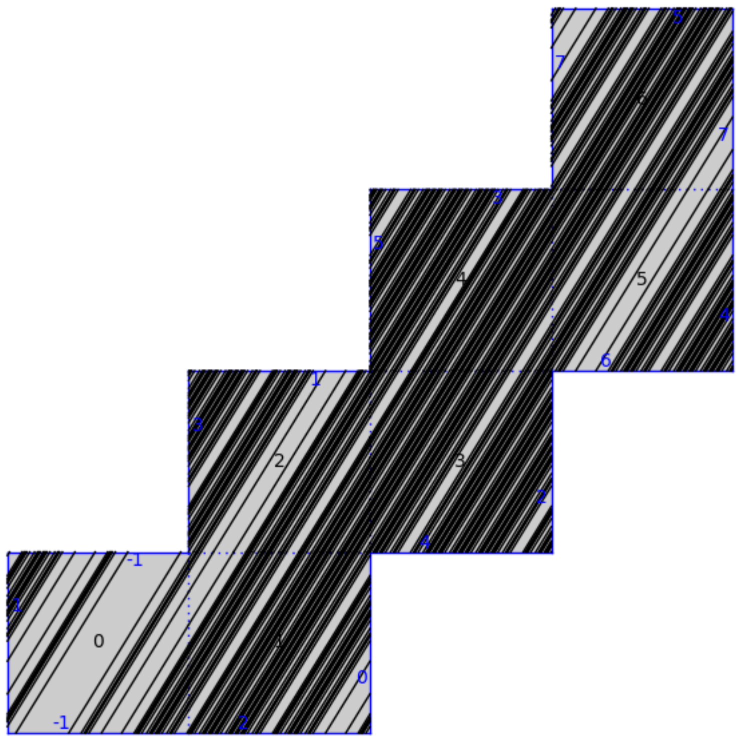
```
v=s.tangent_vector(4,(0,0),(1,phi),ring=K)
```

```
traj=v.straight_line_trajectory()
traj.flow(1000)
```

```
from flatsurf.graphical.straight_line_trajectory import GraphicalStraightLineTrajectory
```

```
gtraj = GraphicalStraightLineTrajectory(gs,traj)
```

```
gs.plot()+gtraj.plot()
```

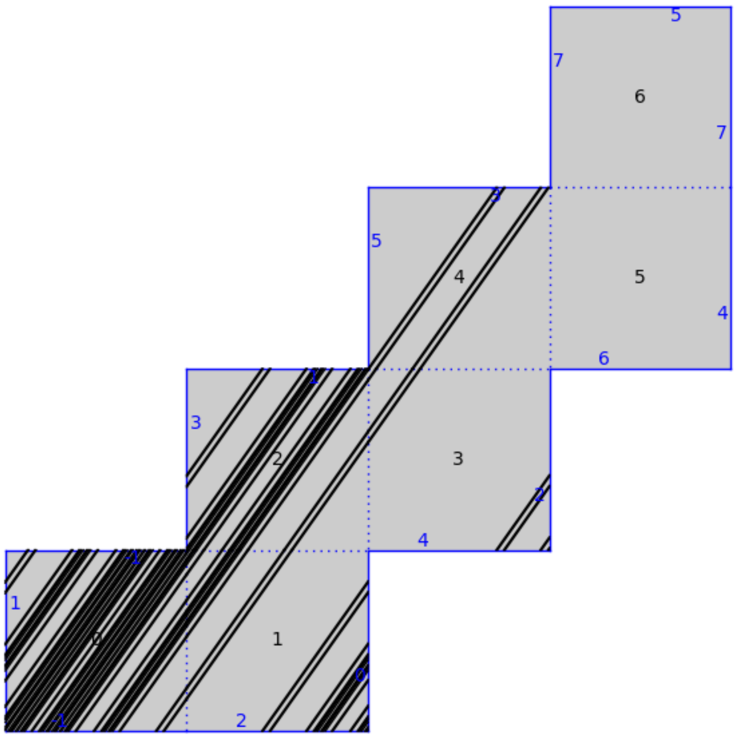


```
K.<rt2> = NumberField(x**2 - 2, embedding=1.4)
v=s.tangent_vector(4,(0,0),(1,rt2),ring=K)
```

```
traj = v.straight_line_trajectory()
traj.flow(1000)
```

```
gtraj = GraphicalStraightLineTrajectory(gs,traj)
```

```
gs.plot()+gtraj.plot()
```


5
7
6
7
5
4
5
4
6
3
2
3
2
4
1
0
1
1
2
0