Superficies cuadriculadas en sage

August 17, 2021

1 Superficies cuadriculadas en sage

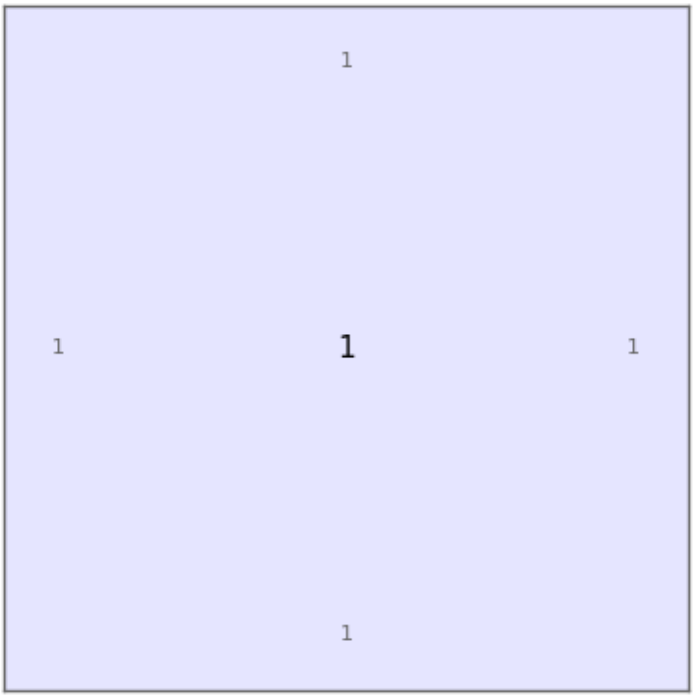
Vamos a demonstrar el uso del paquete surface-dynamics en el programa sage. Ese paquete puede calcular muchas cosas relacionadas a superficies cuadriculadas. Todo esos calculs puden hacerse en línea https://sagecell.sagemath.org/.

[7]:
```
from surface_dynamics import *
```

El toro

[3]:
```
o1 = Origami('(1)', '(1)')
o1.plot()
```

[3]:

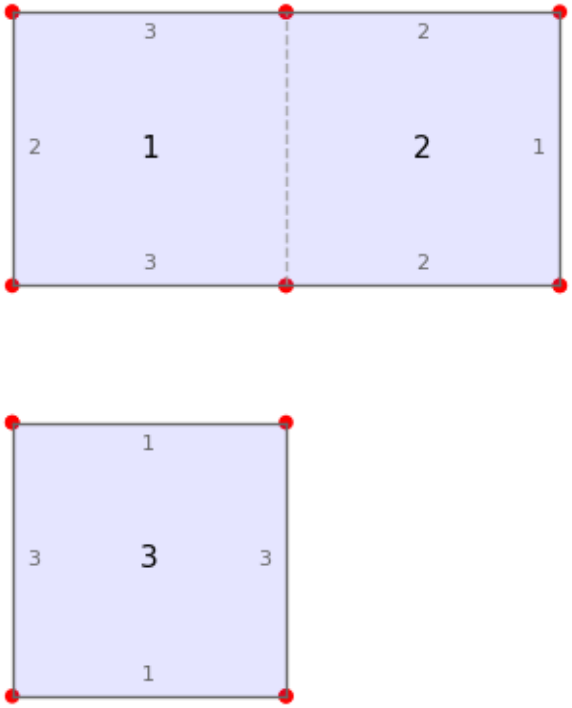


[ ]:

La superficie en forma de L

[4]:
```
o2 = Origami('(1,2)', '(1,3)')
o2.plot()
```

[4]:



[ ]:
```

```

Calculo de angulos (o estrato), $\mathcal{H}_g(k_1, k_2, \ldots, k_\sigma)$

[6]:
```
print(o1.stratum())
print(o2.stratum())
```

```
H_1(0)
H_2(2)
```

[ ]:
```

```

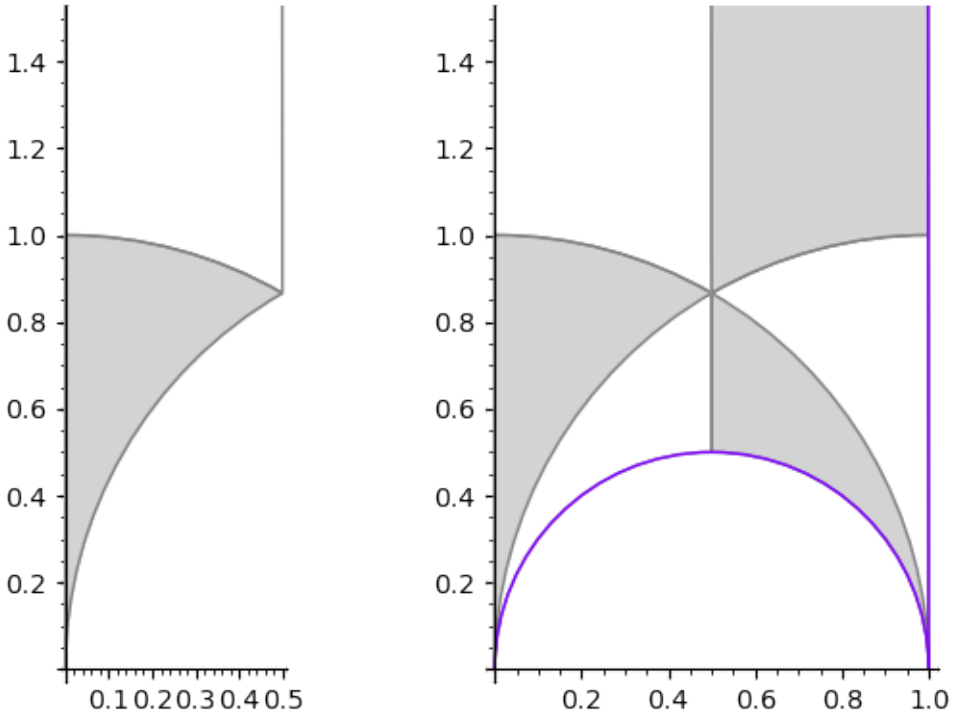
Grupo de Veech (o estabilizador de la $\mathrm{SL}(2,\mathbb{Z})$-action)

[8]:
```
G1 = o1.veech_group()
print(G1)
G2 = o2.veech_group()
print(G2)
```

```
Arithmetic subgroup with permutations of right cosets
 S2=()
```

```
 S3=()
 L=()
 R=()
Arithmetic subgroup with permutations of right cosets
 S2=(2,3)
 S3=(1,2,3)
 L=(1,2)
 R=(1,3)
```

[16]:
```
P1 = G1.farey_symbol().fundamental_domain().plot()
P2 = G2.farey_symbol().fundamental_domain().plot()
P1.axes_range(ymax=1.5)
P2.axes_range(ymax=1.5)
graphics_array([P1, P2]).show()
```



[ ]:
```
```

[18]:
```
print(G1.index())
print(G1.is_congruence())
```

```
1
True
```

[19]:
```
print(G2.index())
print(G2.is_congruence())
```

```
3
True
```

[ ]:
```

```

1.1 Mas ejemplos en $\mathcal{H}_2(2)$

Abajo, calculemos las curvas de Teichmüller aritmeticas de origamis con menos de 10 cuadrados. Por cada una, calculemos el tamaño (ie el cardinal de la $\mathrm{SL}(2,\mathbb{Z})$-órbita))

[21]:
```
H = AbelianStratum(2).unique_component()
```

[33]:
```
for N in range(3,10):
    A = H.arithmetic_teichmueller_curves(N)
    print("N={}: {} arithmetic curves".format(N, len(A)))
    for T in A:
        G = T.veech_group()
        print("  size   : {}".format(G.index()))
        print("  congr. : {}".format(G.is_congruence()))
        print()
```

```
N=3: 1 arithmetic curves
  size   : 3
  congr. : True

N=4: 1 arithmetic curves
  size   : 9
  congr. : False

N=5: 2 arithmetic curves
  size   : 18
  congr. : False

  size   : 9
  congr. : False

N=6: 1 arithmetic curves
  size   : 36
  congr. : False

N=7: 2 arithmetic curves
  size   : 54
  congr. : False

  size   : 36
```

```
  congr. : False

N=8: 1 arithmetic curves
  size   : 108
  congr. : False

N=9: 2 arithmetic curves
  size   : 81
  congr. : False

  size   : 108
  congr. : False
```

[ ]:

[ ]: