In [1]:
```
from pdesign import canvas, shapes, lines
from pdesign import transforms as trans
import numpy as np
from shapely.geometry import MultiLineString, LineString, Point, Polygon, Mul
from shapely.ops import unary_union
import matplotlib.pyplot as plt

from shapely.geometry import box as Box
from shapely import affinity
```

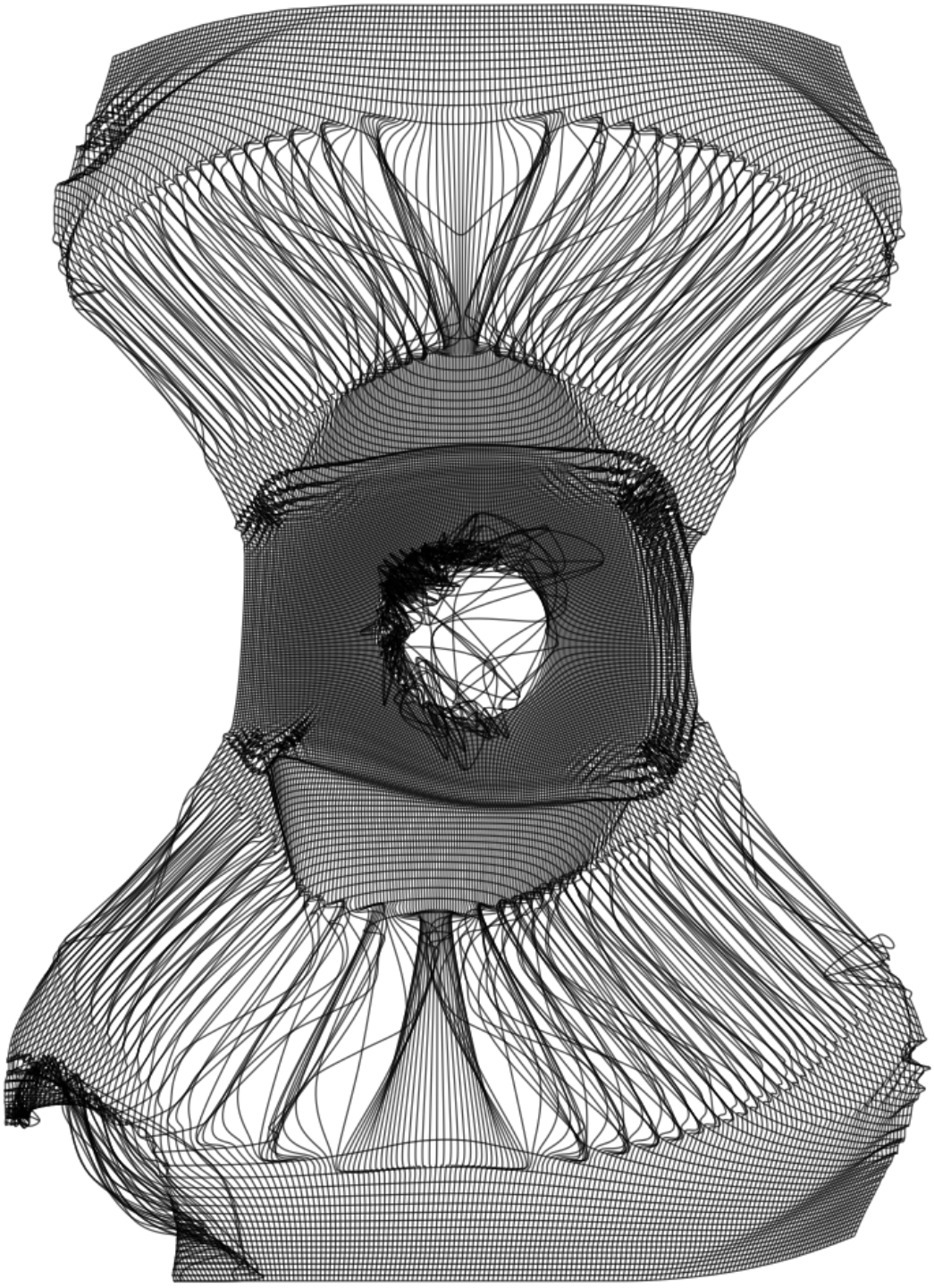
In [2]:
```
linewidth = 0.01771654*72
picture = canvas.Canvas(paper_size=(11,14), margin_percent=0.0, origin='corne

dp = {
    "alpha":0.7,
    "linewidth":0.45*0.0393701*72,
    "clear":False,
}

picture_bbox = Box(picture.bbox[0,0], picture.bbox[0,1], picture.bbox[1,0], p
```

```
<Figure size 792x1008 with 0 Axes>
```

In [8]:

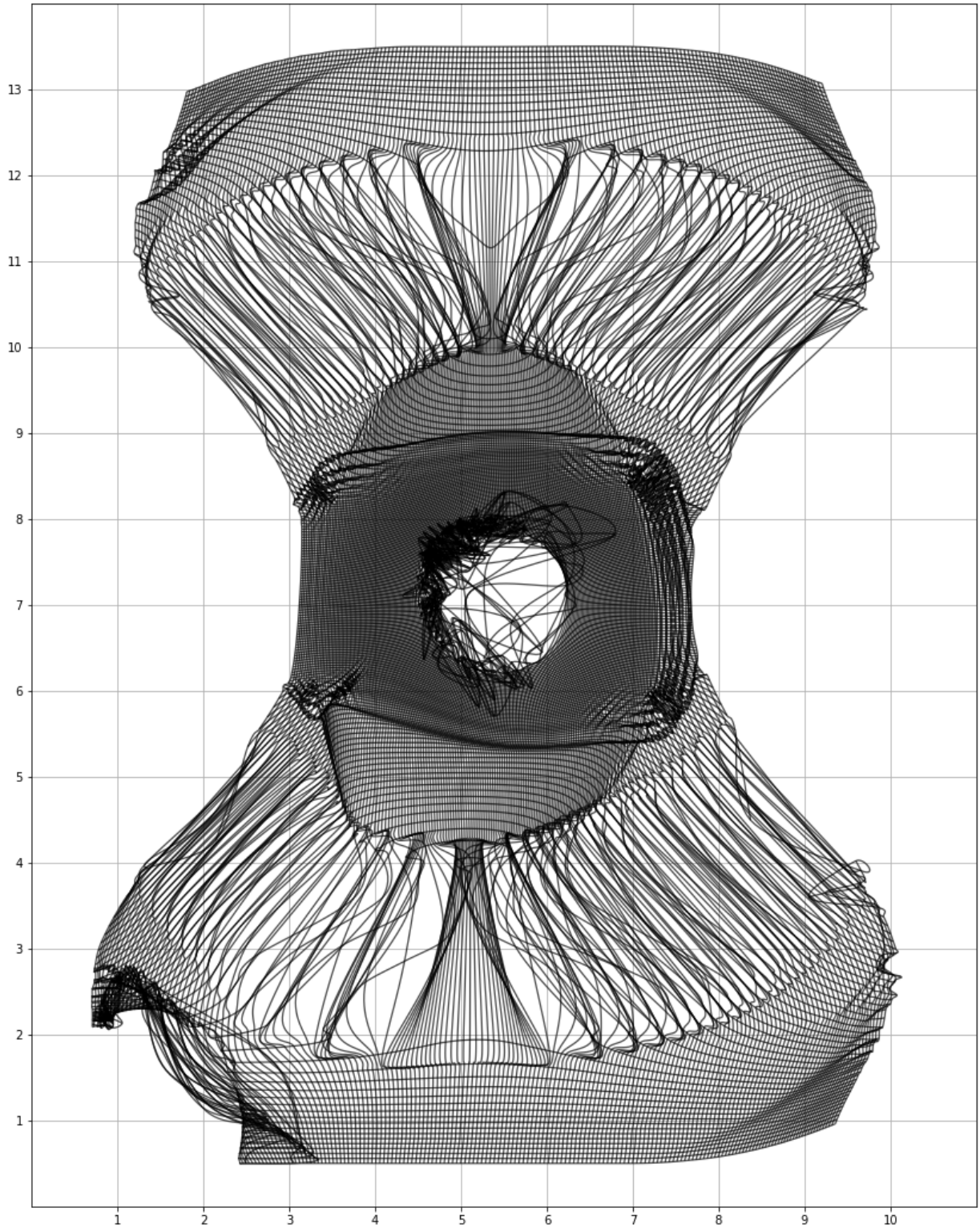
```
size = 175

m = 2

x, y = np.meshgrid(np.linspace(m, 11-m, size), np.linspace(m, 14-m, size), in


delta = 0.2
max_step = 0.1
for _ in range(75):

    r = np.sqrt((x-11/2)**2 + (y-7)**2)
    th = np.arctan2(x, y)


    f = np.sin(5*th) + np.cos(r) + 2*np.sin(r)
    dx, dy = np.gradient(f)


    x += delta*np.clip(dx, -max_step, max_step)
    y += delta*np.clip(dy, -max_step, max_step)


grid = []

for i in range(size):

    l = np.dstack([x[i], y[i]])[0][::(-1)**i]
    grid.append(LineString(lines.cardinal_spline(l)).simplify(1e-4))

    l = np.dstack([x[:, i], y[:, i]])[0][::(-1)**i]
    grid.append(LineString(lines.cardinal_spline(l)).simplify(1e-4))


grid = MultiLineString(grid)
#grid = affinity.scale(grid, 0.9, 0.9, origin=(11/2, 7))


picture.make_canvas()
picture.plot_shapes([grid], **dp)
picture.display_overlays(False)
pict
picture.fig
```

Out[8]:



In [10]:
```
picture.make_canvas()
picture.plot_shapes([grid], **dp)
#picture.display_overlays(False)
picture.add_grid(11,14)
picture.fig
```

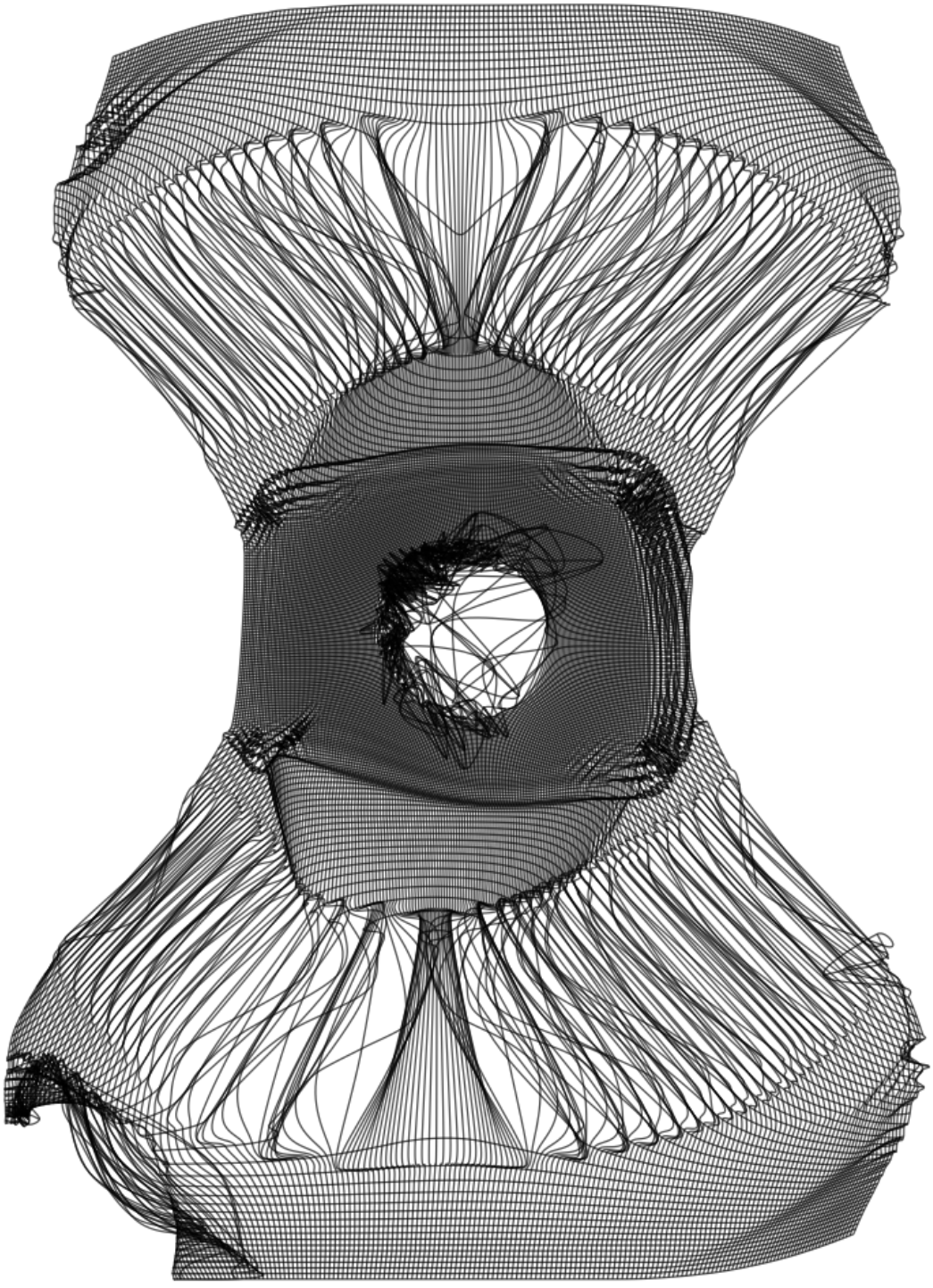
Out[10]:



In [13]:

```
picture.make_canvas()
picture.plot_shapes([grid], **dp)
picture.display_overlays(False)
picture.fig.savefig('glitchy_hourglass.svg')
#picture.add_grid(11,14)
picture.fig
```

Out[13]:



In [ ]: