

# NRCH

September 29, 2022

```
[ ]: import numpy as np
from numpy.linalg import eigvals
import matplotlib.pyplot as plt

from matplotlib import cm
from matplotlib.colors import LogNorm
from matplotlib.animation import FuncAnimation as FA

plt.rc("font", family="serif", size=16)
plt.rc("mathtext", fontset="cm")
plt.rc("lines", lw=2)
```

```
[ ]: D11 = lambda phi, a, x, c: \
    -( 
        + 2*(phi[0] - c[0, 0])**2
        + 2*(phi[0] - c[0, 1])**2
        + 8*(phi[0] - c[0, 0])*(phi[0] - c[0, 1])
        + 2*x[1]*phi[1]**2
    )+0j
D22 = lambda phi, a, x, c: \
    -( 
        + 2*(phi[1] - c[1, 0])**2
        + 2*(phi[1] - c[1, 1])**2
        + 8*(phi[1] - c[1, 0])*(phi[1] - c[1, 1])
        + 2*x[1]*phi[0]**2
    )+0j
D12 = lambda phi, a, x, c: -((x[0] + a) + 4*x[1]*phi[0]*phi[1])+0j
D21 = lambda phi, a, x, c: -((x[0] - a) + 4*x[1]*phi[0]*phi[1])+0j

def D0(phi, a, x, c):
    D = np.matrix([
        [D11(phi, a, x, c), D12(phi, a, x, c)],
        [D21(phi, a, x, c), D22(phi, a, x, c)]])
    return D
```

```
[ ]: c = np.matrix([[.2, .5], [.1, .5]])
x = np.array([-2., 2.])

n = 200
k = 1.
phi1 = np.linspace(-k, k, n)
phi2 = np.linspace(-k, k, n)
p = np.array(np.meshgrid(phi1, phi2))

aa = np.linspace(0, .5, 8)
print('c_11 = {}'.format(c[0, 0]))
print('c_12 = {}'.format(c[0, 1]))
print('c_21 = {}'.format(c[1, 0]))
print('c_22 = {}'.format(c[1, 1]))
print('chi = {}'.format(x[0]))
print('chi\' = {}'.format(x[1]))
```

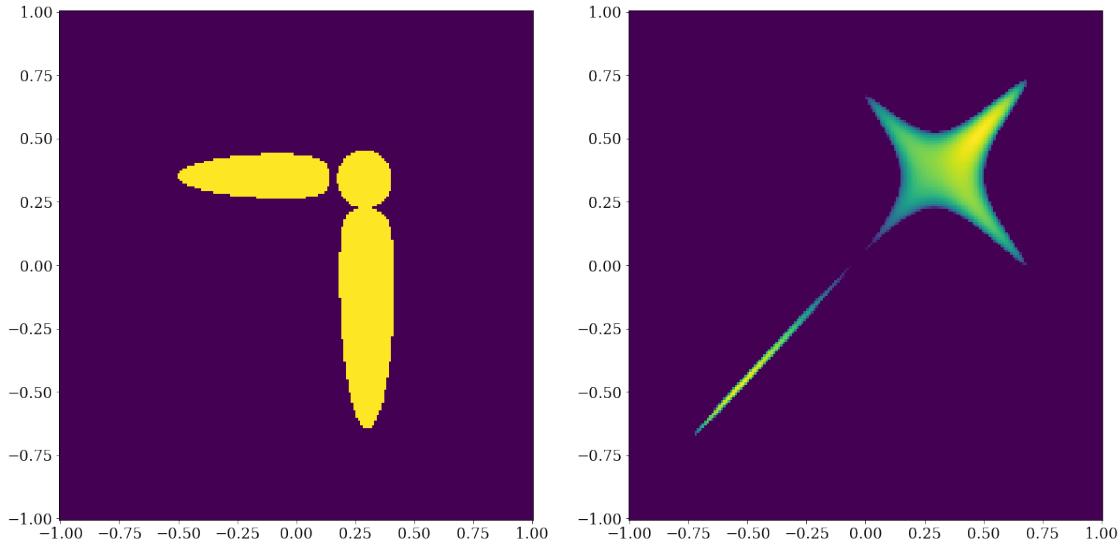
```
c_11 = 0.2
c_12 = 0.5
c_21 = 0.1
c_22 = 0.5
chi = -0.2
chi' = 0.2
```

```
[ ]: a = .2
fig, ax = plt.subplots(1, 2, figsize=(20, 10))
D = lambda p: D0(p, a, x, c)
l = np.array([[eigvals(D(p[:, i, j])) for i in range(n)] for j in range(n)])

print("a = {}".format(a))
Cr = np.logical_or(l[:, :, 0].real>0, l[:, :, 1].real>0)
Ci = l[:, :, 0].imag
ax[0].pcolormesh(p[0], p[1], Cr)
ax[1].pcolormesh(p[0], p[1], Ci)

plt.show()
```

```
a = 0.2
```



```
[ ]: n = 1000
k = 1.
phi1 = np.linspace(-k, k, n)
phi2 = np.linspace(-k, k, n)
p = np.array(np.meshgrid(phi1, phi2))

l1 = lambda phi, a : 1/2 * ( (D11(phi, a, x, c) + D22(phi, a, x, c)) + np.
    ↪sqrt((D11(phi, a, x, c) - D22(phi, a, x, c))**2 + 4*D12(phi, a, x, ↪
    ↪c)*D21(phi, a, x, c) ) )
l2 = lambda phi, a : 1/2 * ( (D11(phi, a, x, c) + D22(phi, a, x, c)) - np.
    ↪sqrt((D11(phi, a, x, c) - D22(phi, a, x, c))**2 + 4*D12(phi, a, x, ↪
    ↪c)*D21(phi, a, x, c) ) )

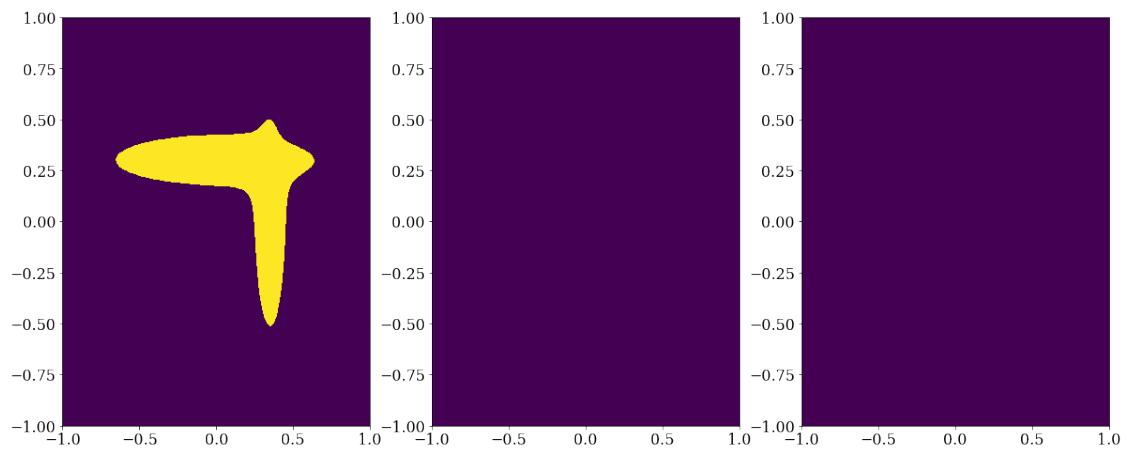
aa = np.linspace(0, 1., 15)

for a in aa:
    print(a)
    fig, ax = plt.subplots(1, 3, figsize=(20, 8))
    Cr1 = (l1(p, a).real>0)
    Cr2 = (l2(p, a).real>0)
    Ci = l2(p, a).imag

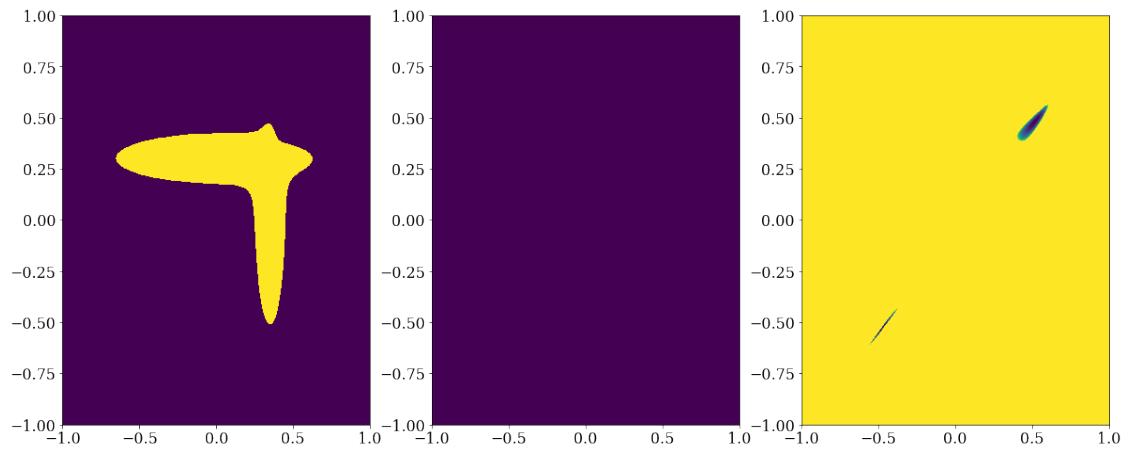
    ax[0].pcolormesh(p[0], p[1], Cr1)
    ax[1].pcolormesh(p[0], p[1], Cr2)
    ax[2].pcolormesh(p[0], p[1], Ci)

    plt.show()
```

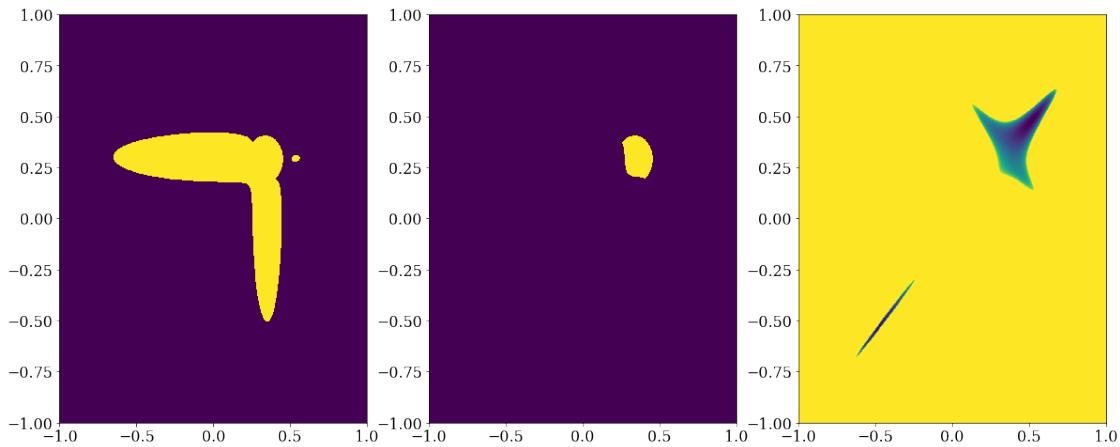
0.0



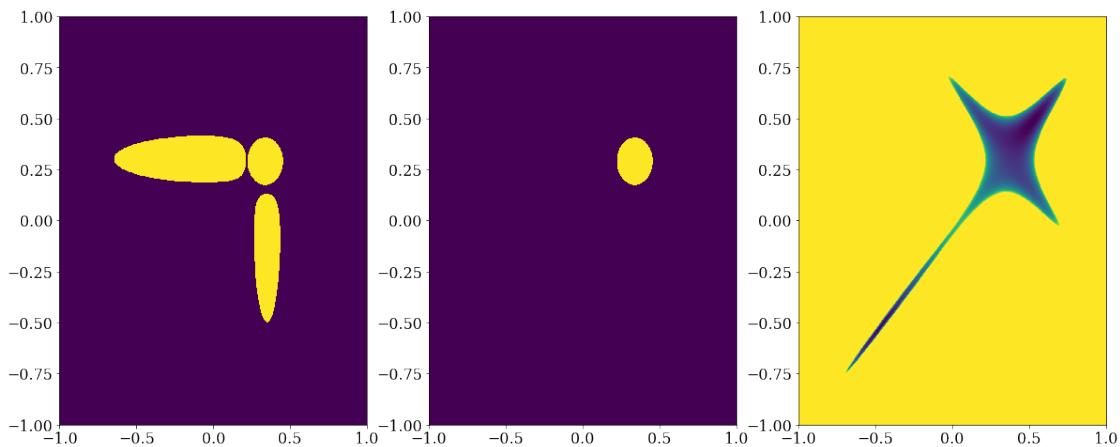
0.07142857142857142



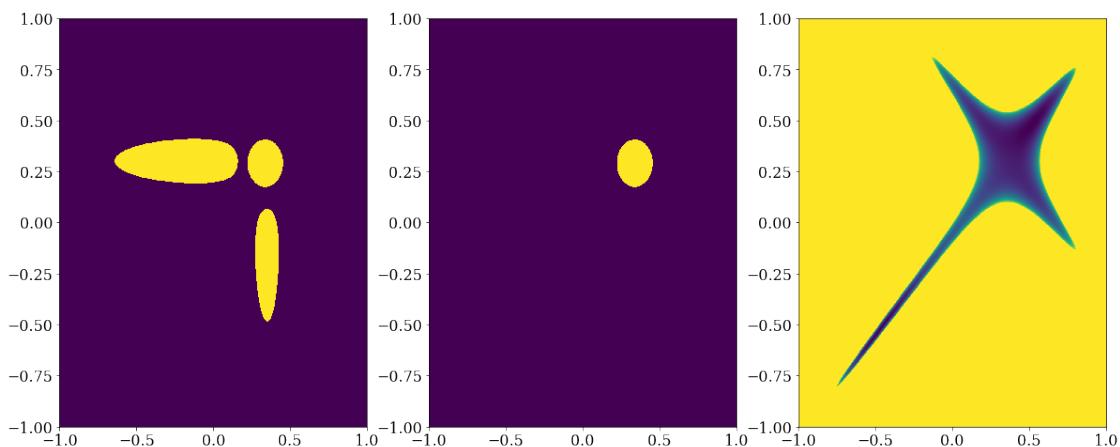
0.14285714285714285



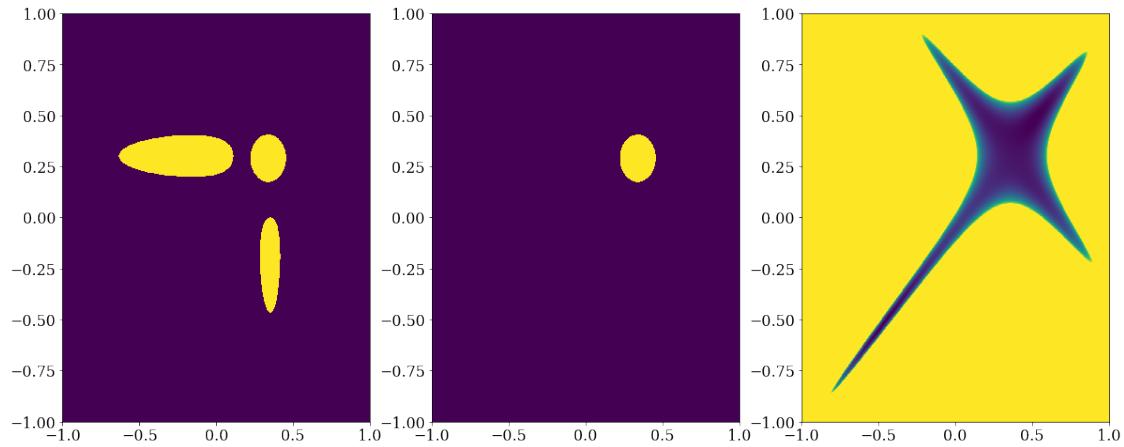
0.21428571428571427



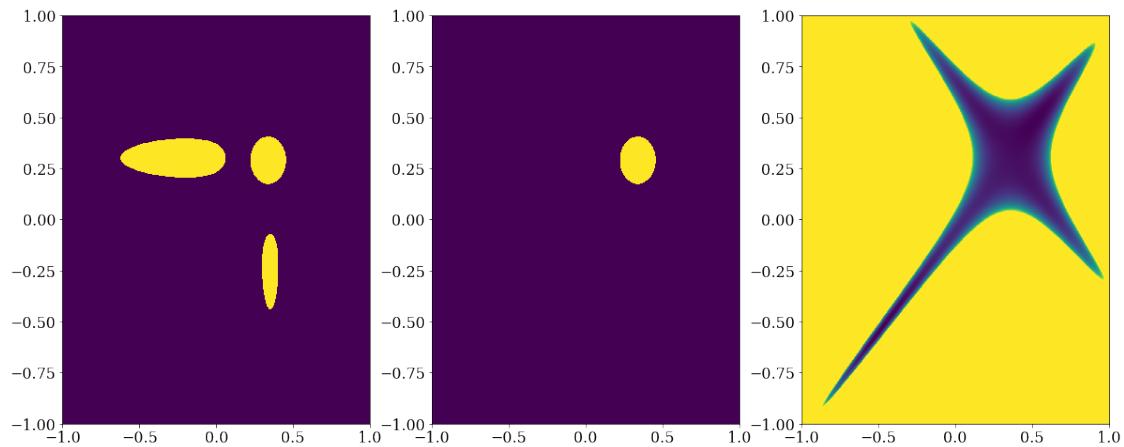
0.2857142857142857



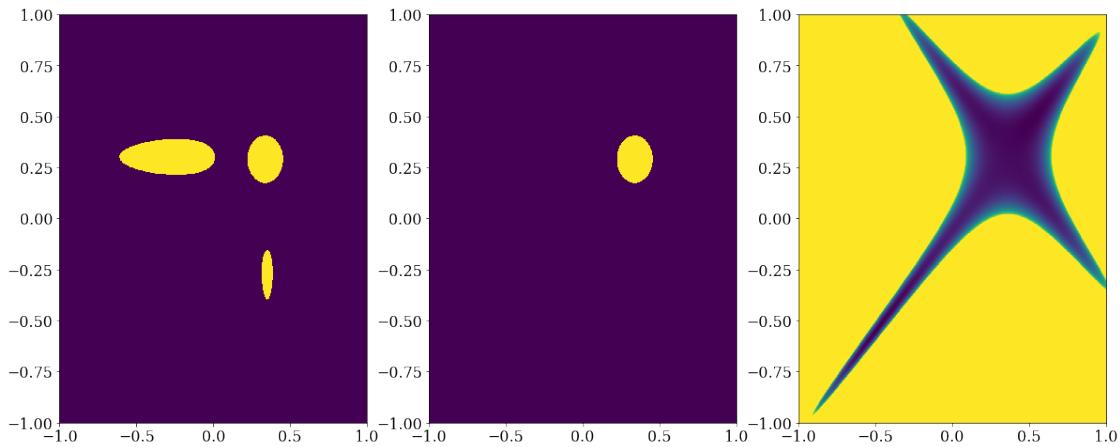
0.3571428571428571



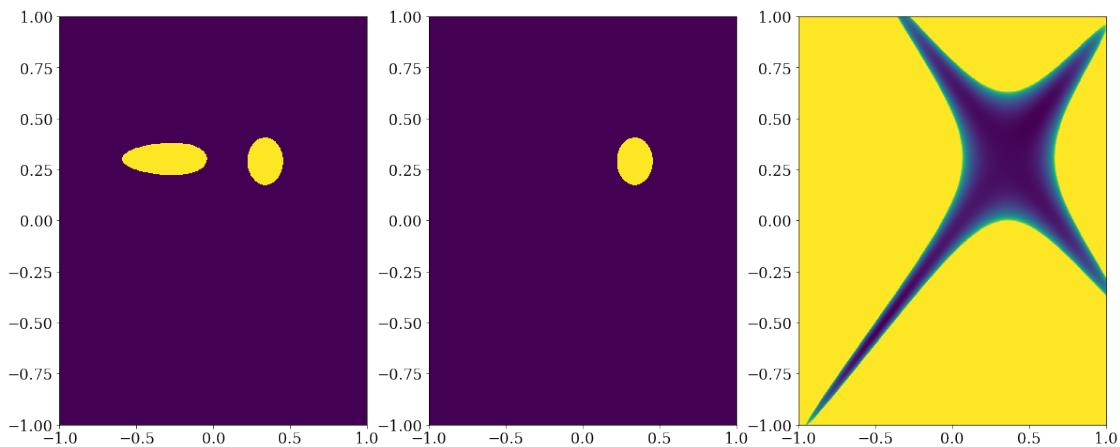
0.42857142857142855



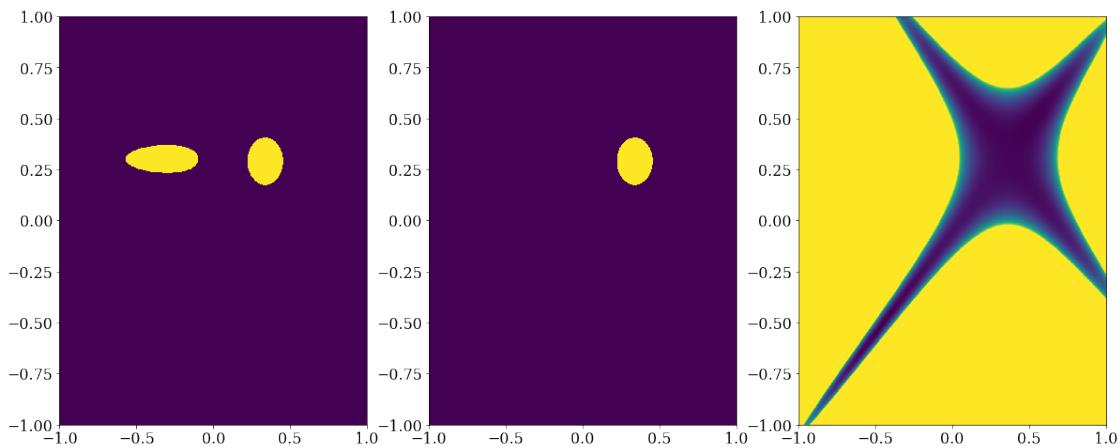
0.5



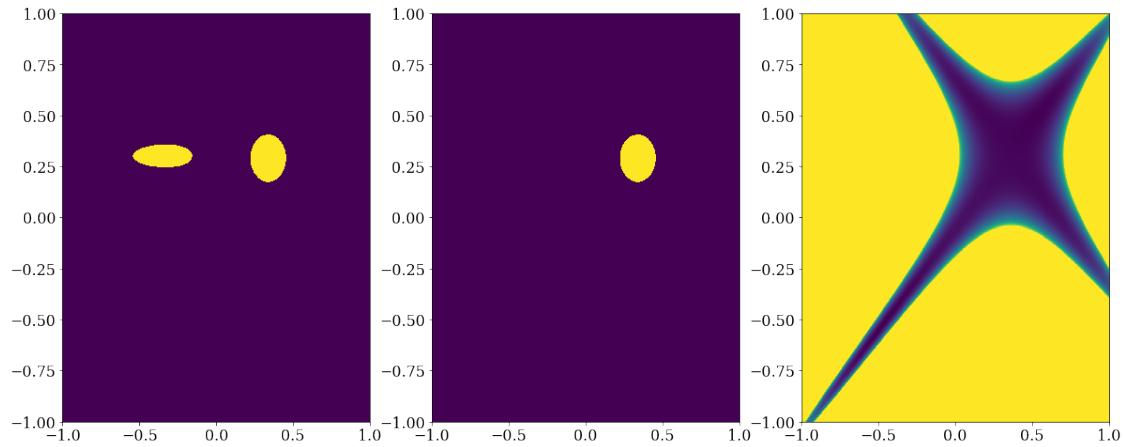
0.5714285714285714



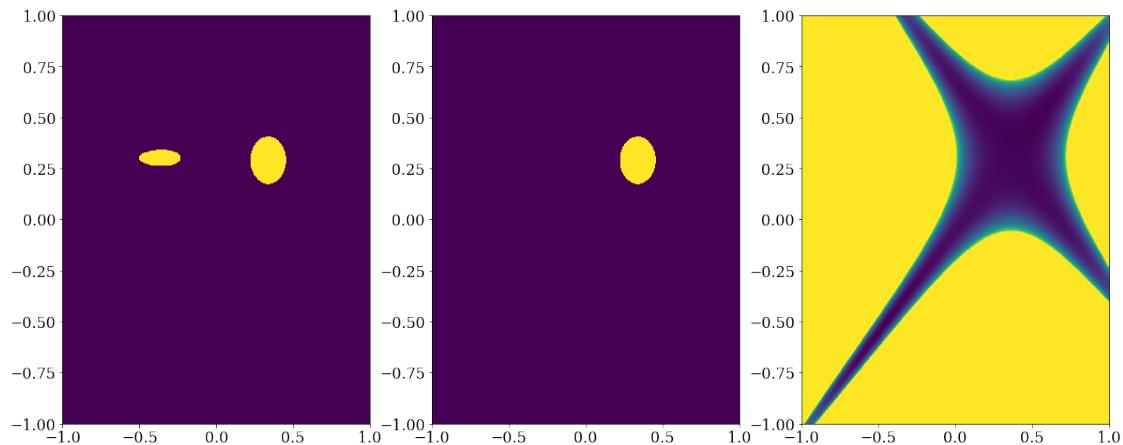
0.6428571428571428



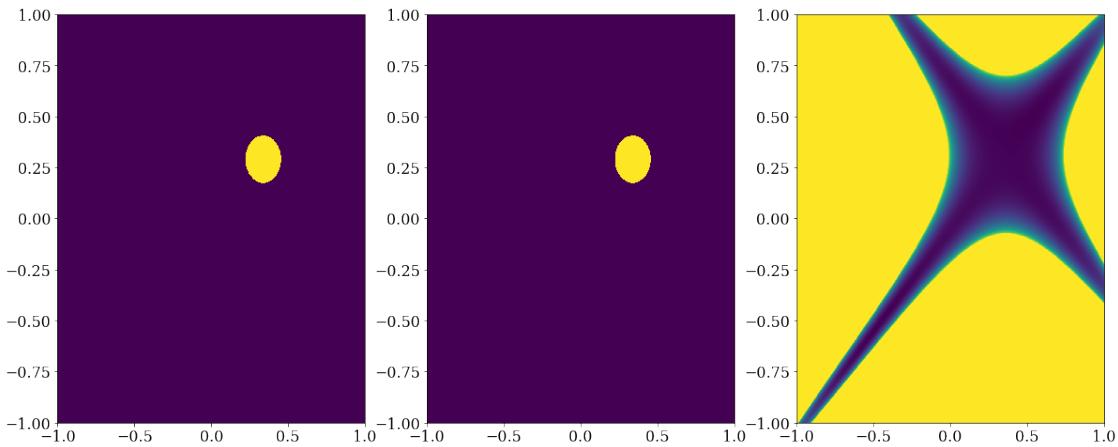
0.7142857142857142



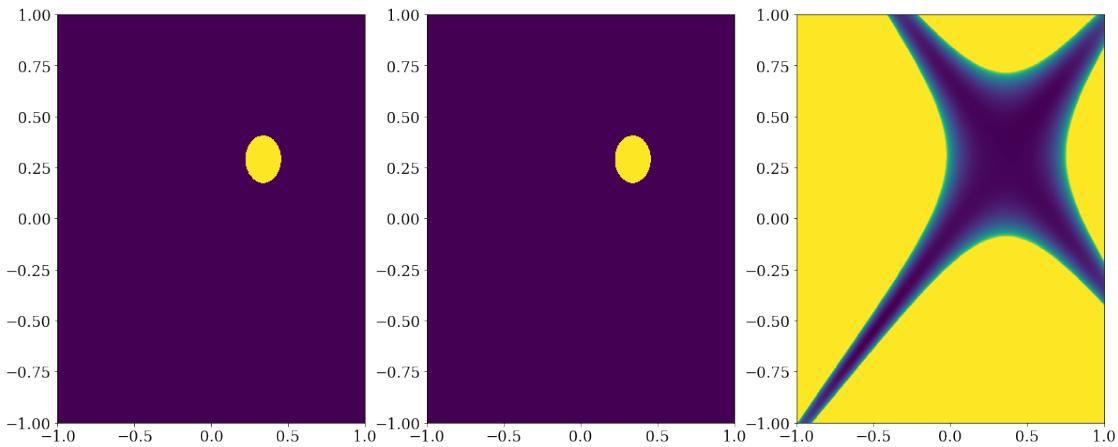
0.7857142857142857



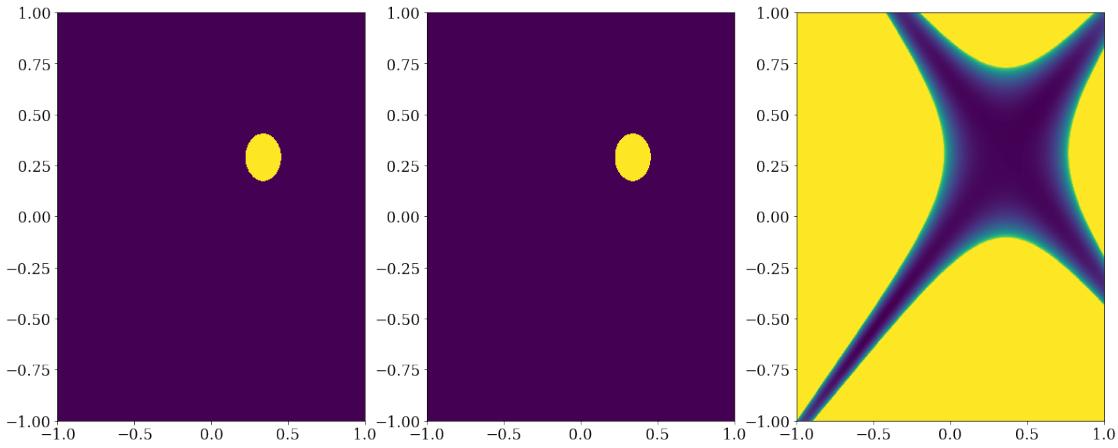
0.8571428571428571



0.9285714285714285



1.0



```
[ ]: cii = .1
cij = .5
x0 = .5

c = np.matrix([[cii, cij], [cij, cii]])
x = np.array([-x0, x0])

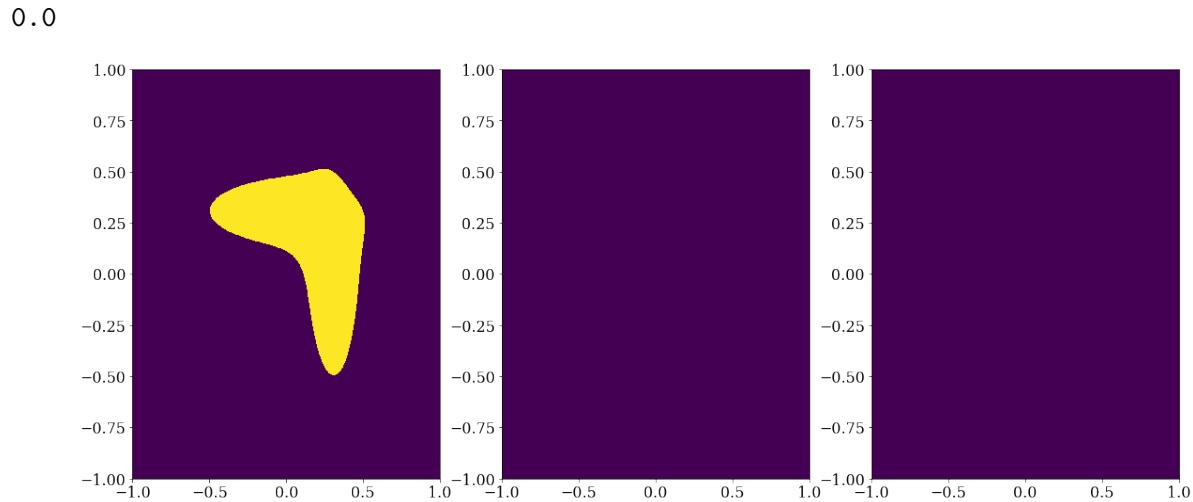
l1 = lambda phi, a : 1/2 * ( (D11(phi, a, x, c) + D22(phi, a, x, c)) + np.
    ↪sqrt((D11(phi, a, x, c) - D22(phi, a, x, c))**2 + 4*D12(phi, a, x, c)*D21(phi, a, x, c) ) )
l2 = lambda phi, a : 1/2 * ( (D11(phi, a, x, c) + D22(phi, a, x, c)) - np.
    ↪sqrt((D11(phi, a, x, c) - D22(phi, a, x, c))**2 + 4*D12(phi, a, x, c)*D21(phi, a, x, c) ) )

aa = np.linspace(0, 1., 15)

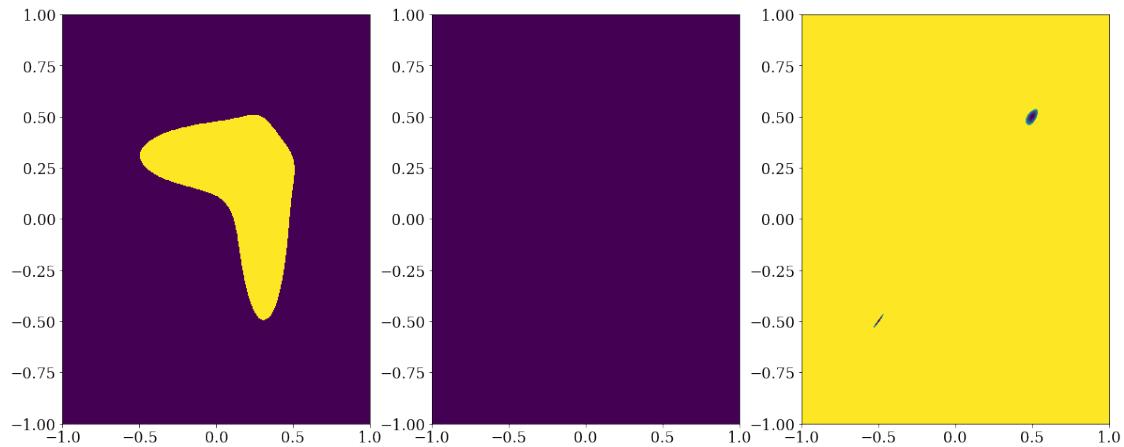
for a in aa:
    print(a)
    fig, ax = plt.subplots(1, 3, figsize=(20, 8))
    Cr1 = (l1(p, a).real>0)
    Cr2 = (l2(p, a).real>0)
    Ci = l2(p, a).imag

    ax[0].pcolormesh(p[0], p[1], Cr1)
    ax[1].pcolormesh(p[0], p[1], Cr2)
    ax[2].pcolormesh(p[0], p[1], Ci)

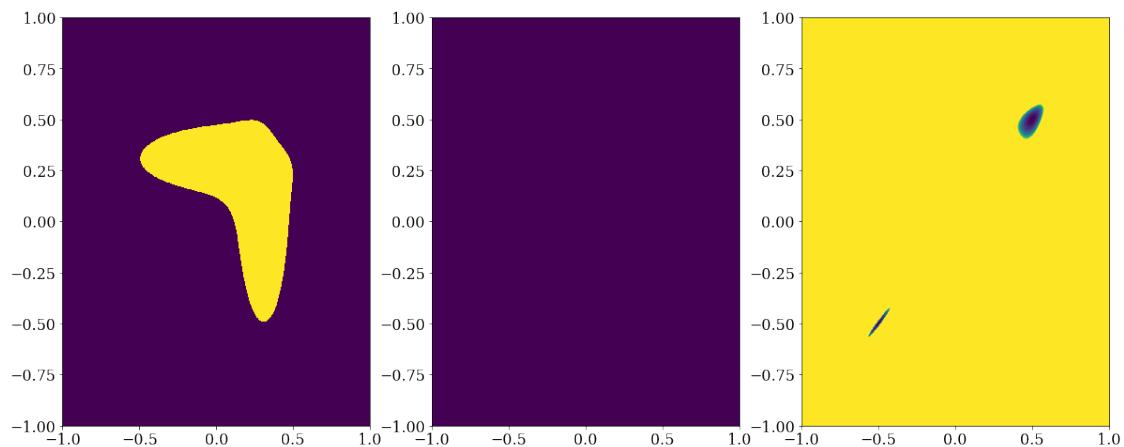
    plt.show()
```



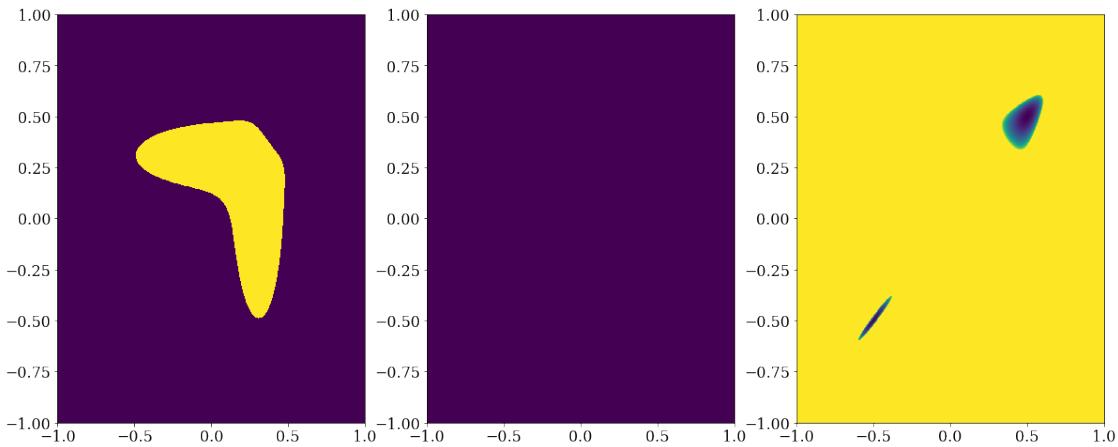
0.07142857142857142



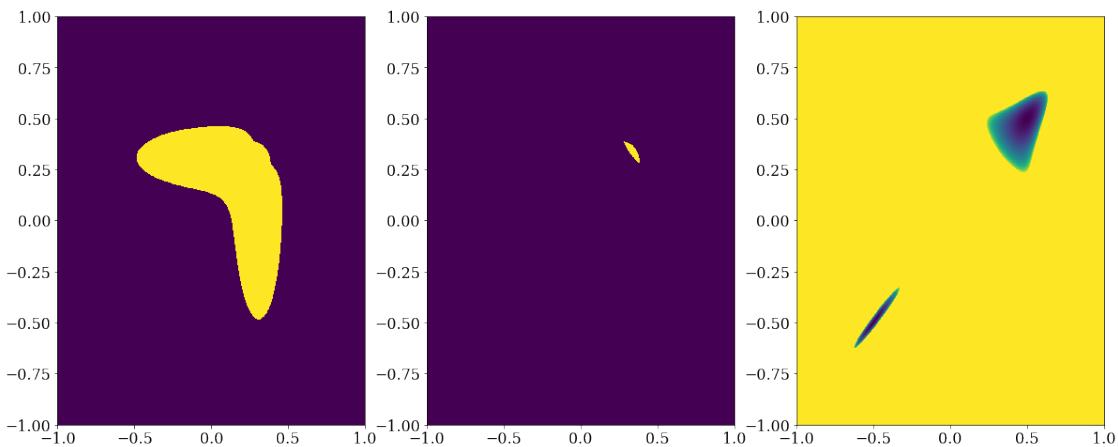
0.14285714285714285



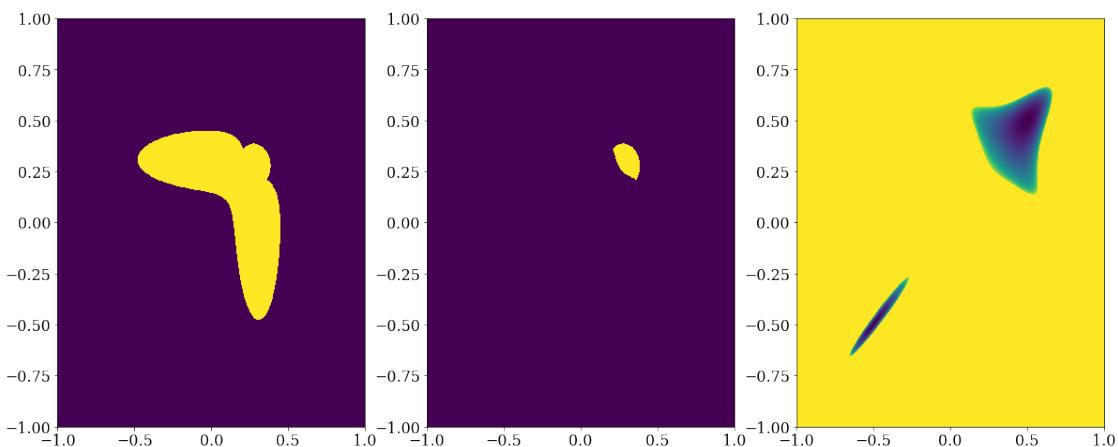
0.21428571428571427



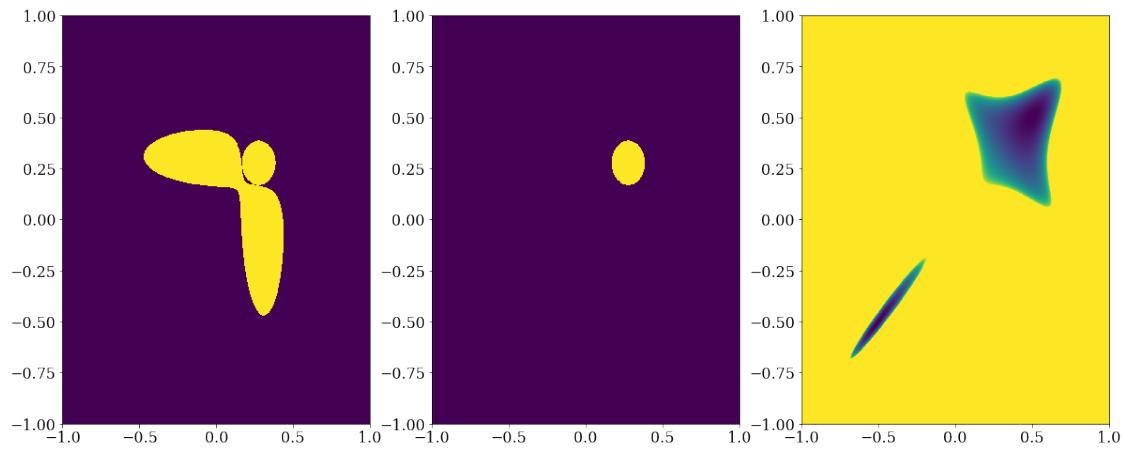
0.2857142857142857



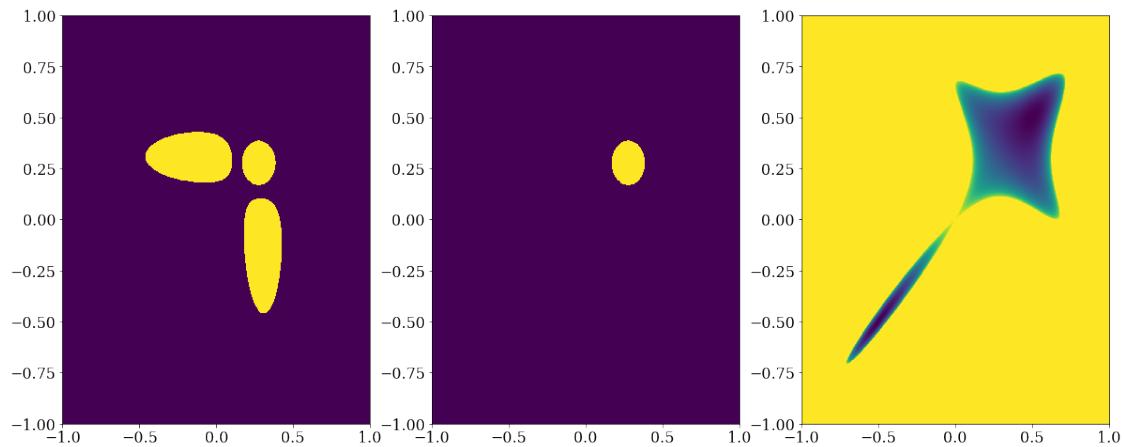
0.3571428571428571



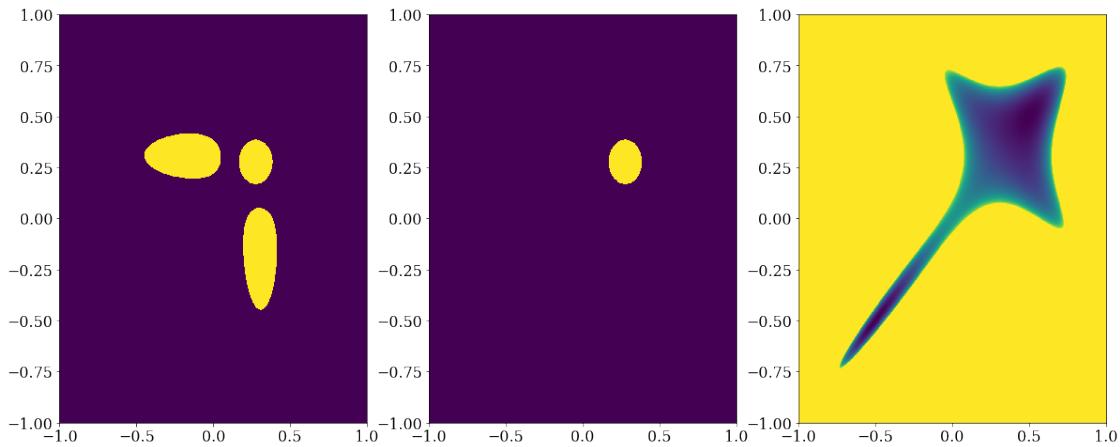
0.42857142857142855



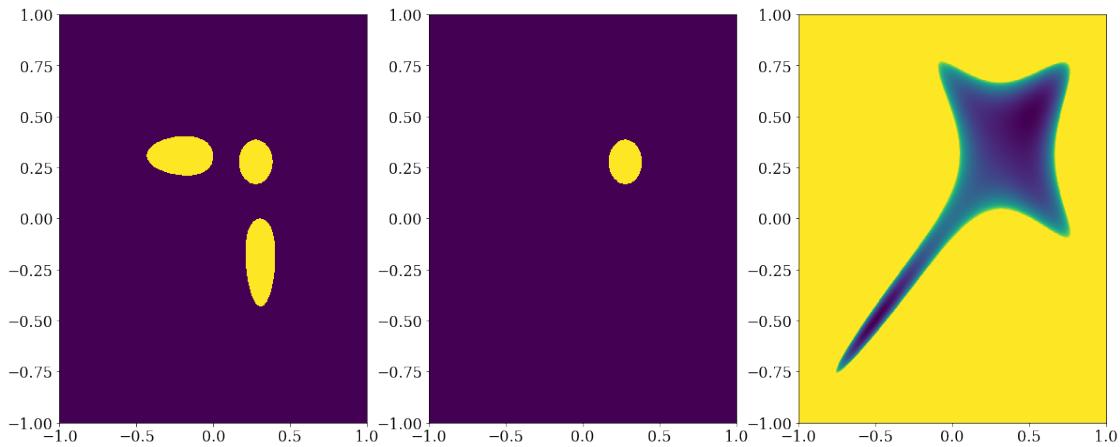
0.5



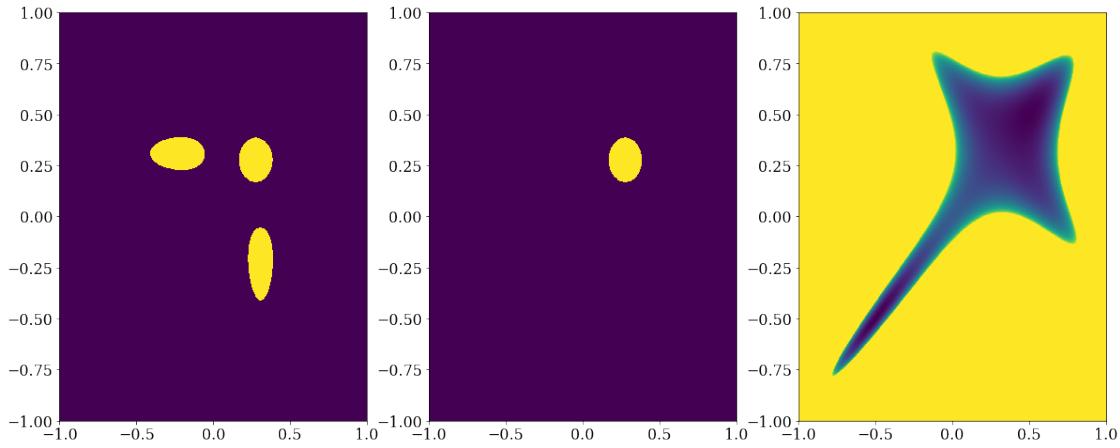
0.5714285714285714



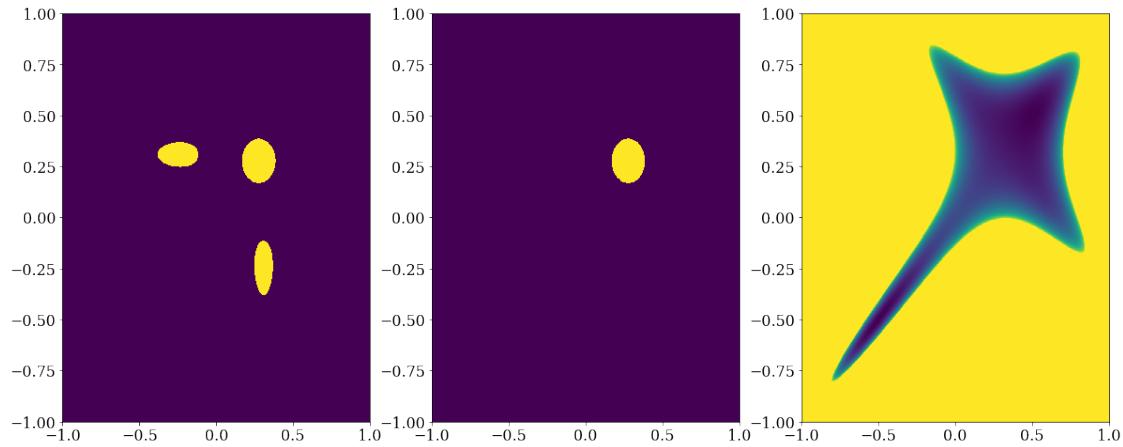
0.6428571428571428



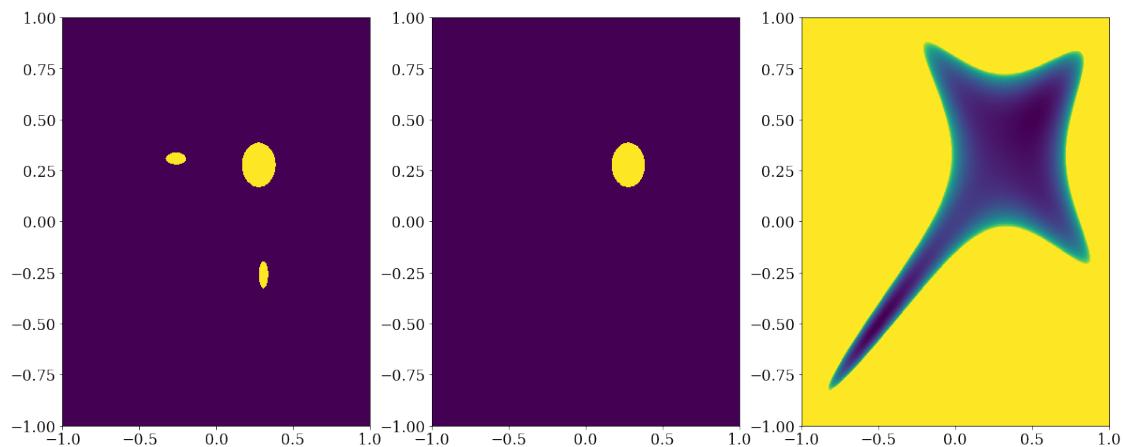
0.7142857142857142



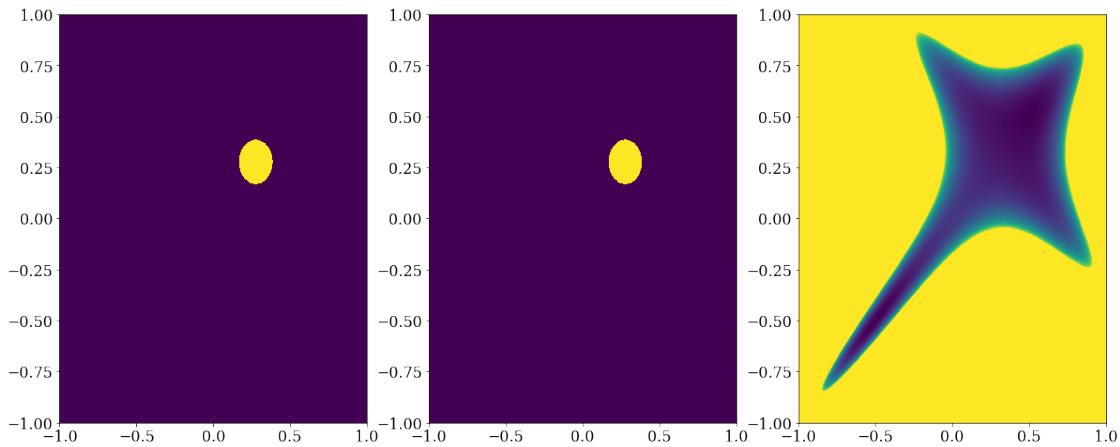
0.7857142857142857



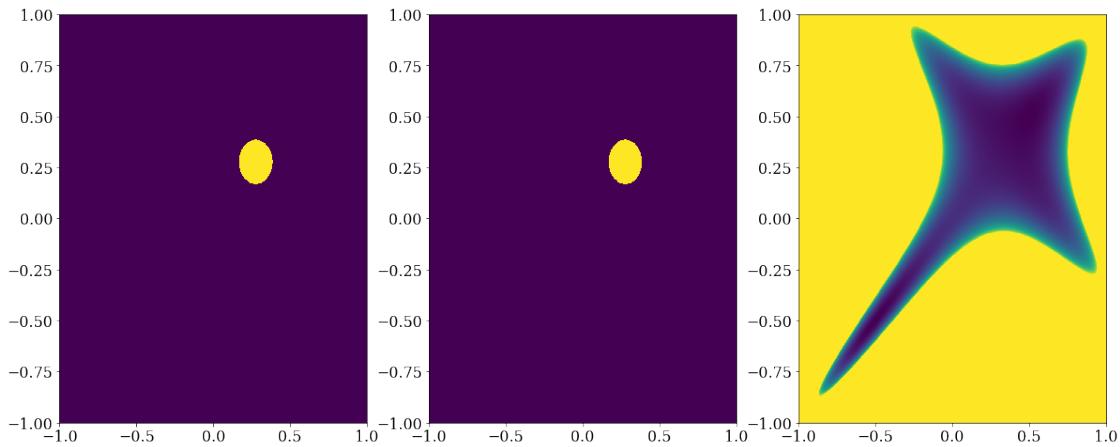
0.8571428571428571



0.9285714285714285



1.0



```
[ ]: n = 2000
k = 1.
phi1 = np.linspace(-k, k, n)
phi2 = np.linspace(-k, k, n)
p = np.array(np.meshgrid(phi1, phi2))

aa = np.linspace(0, .5, 15)

for a in aa:
    fig, ax = plt.subplots( figsize=(30, 30))

    print(a)
    Ci = l1(p, a).imag
    Cr = ( l1(p, a).real>0) * l1(p, a).real
```

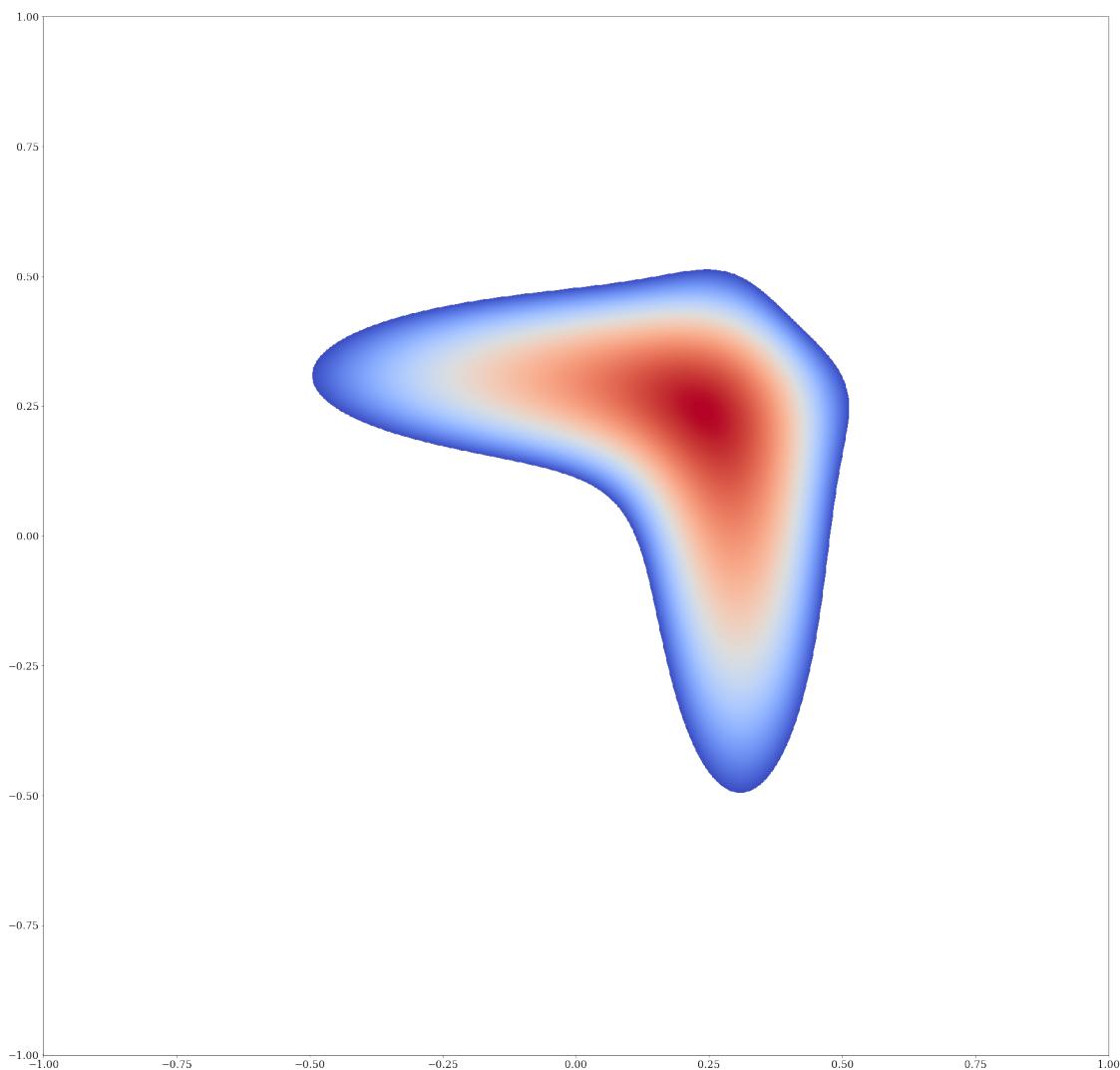
```

# Cr = np.ma.masked_where(Ci, Cr)

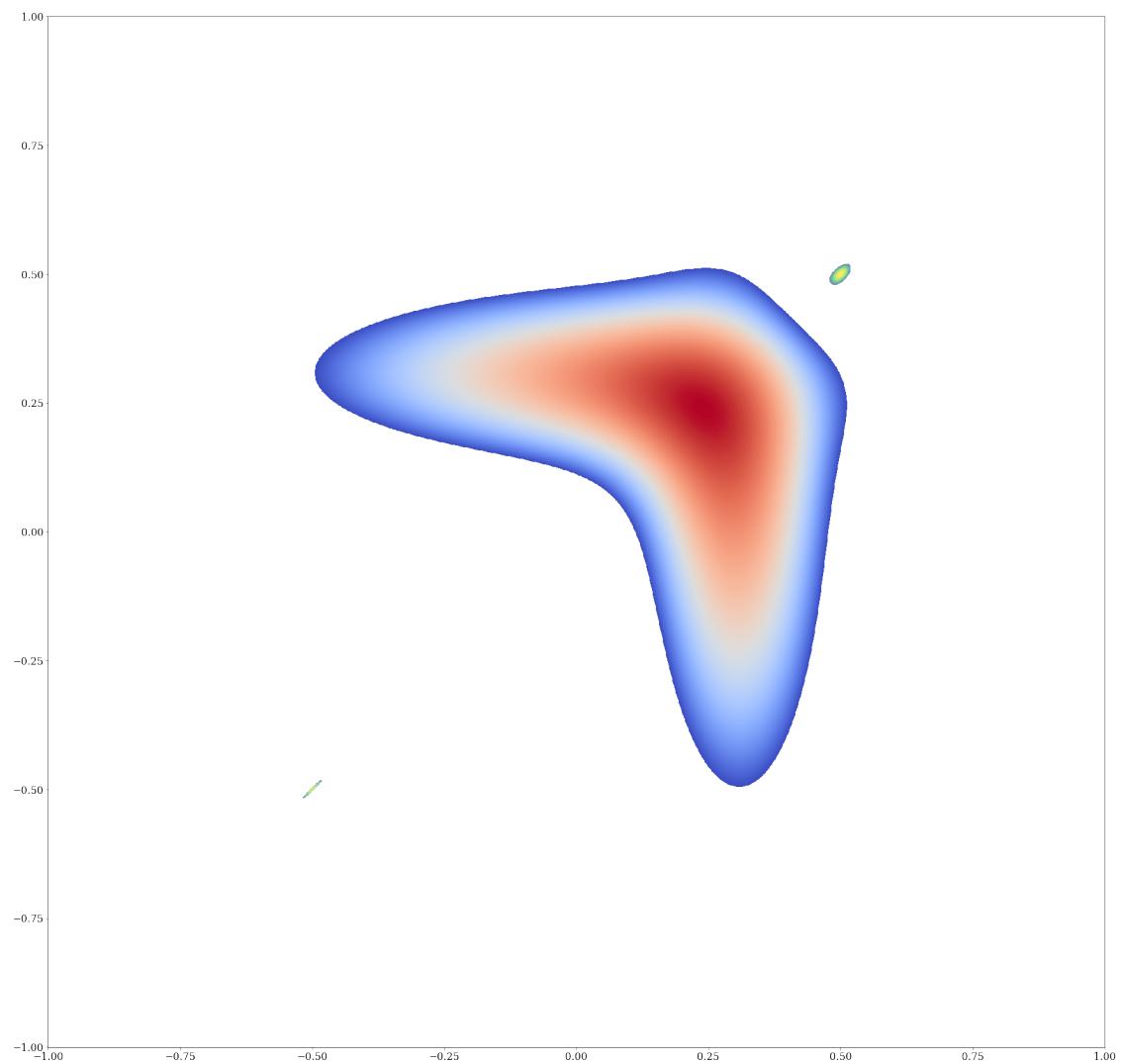
Ci = np.ma.masked_where(Ci==0, Ci)
Cr = np.ma.masked_where(Cr==0, Cr)
ax.pcolormesh(p[0], p[1], Ci, alpha=0.7)
ax.pcolormesh(p[0], p[1], Cr, cmap="coolwarm")
plt.show()

```

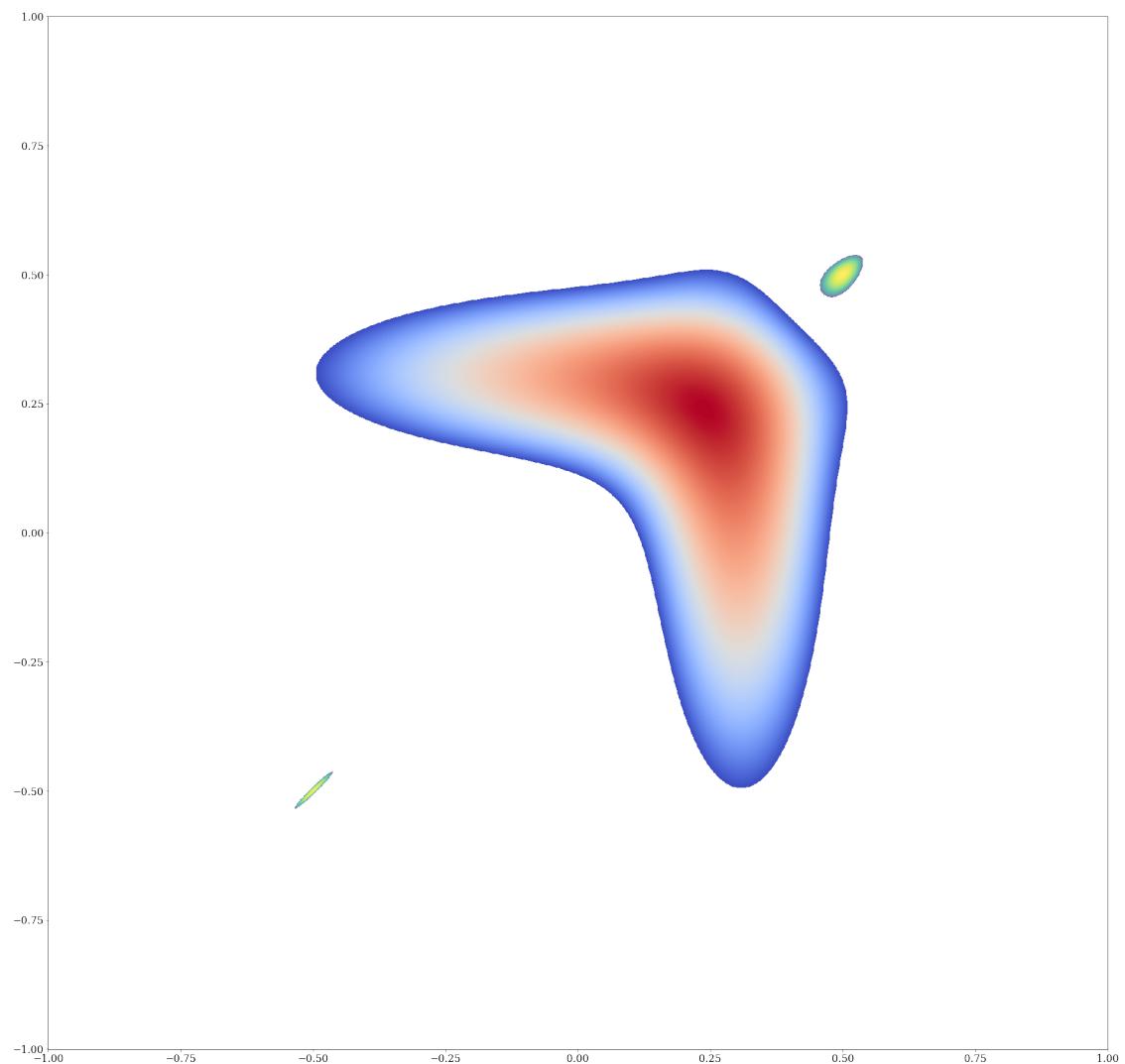
0.0



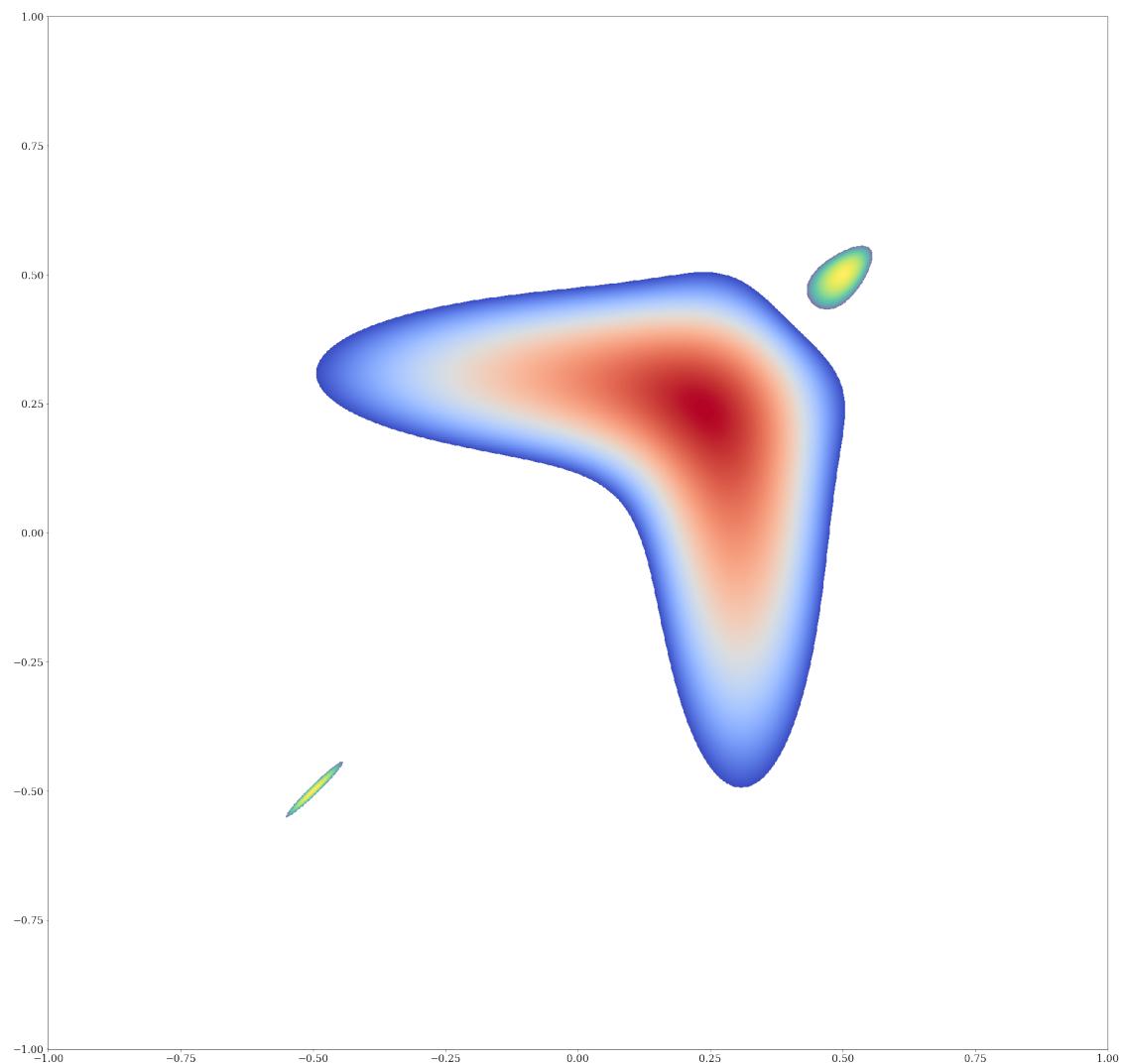
0.03571428571428571



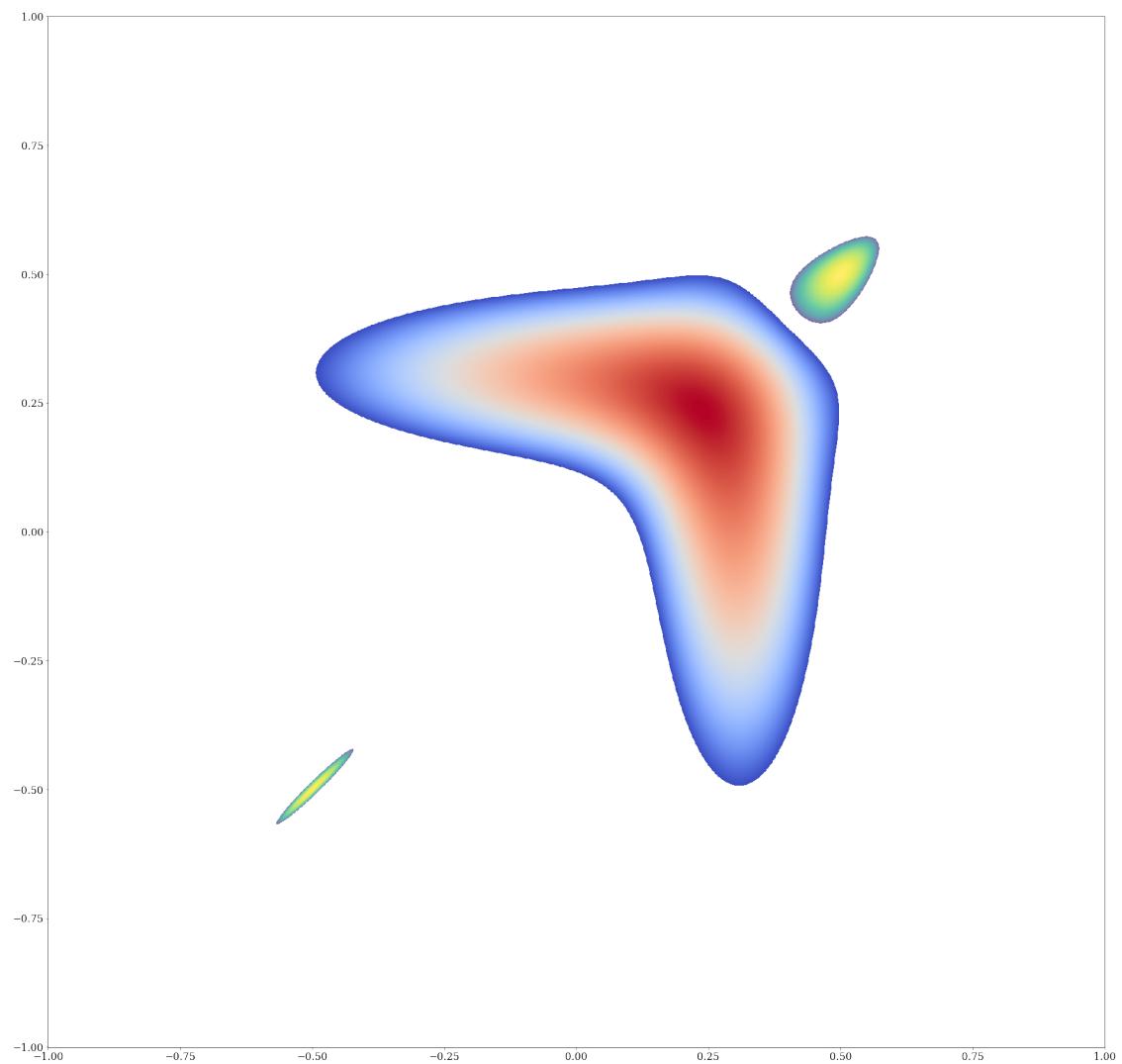
0.07142857142857142



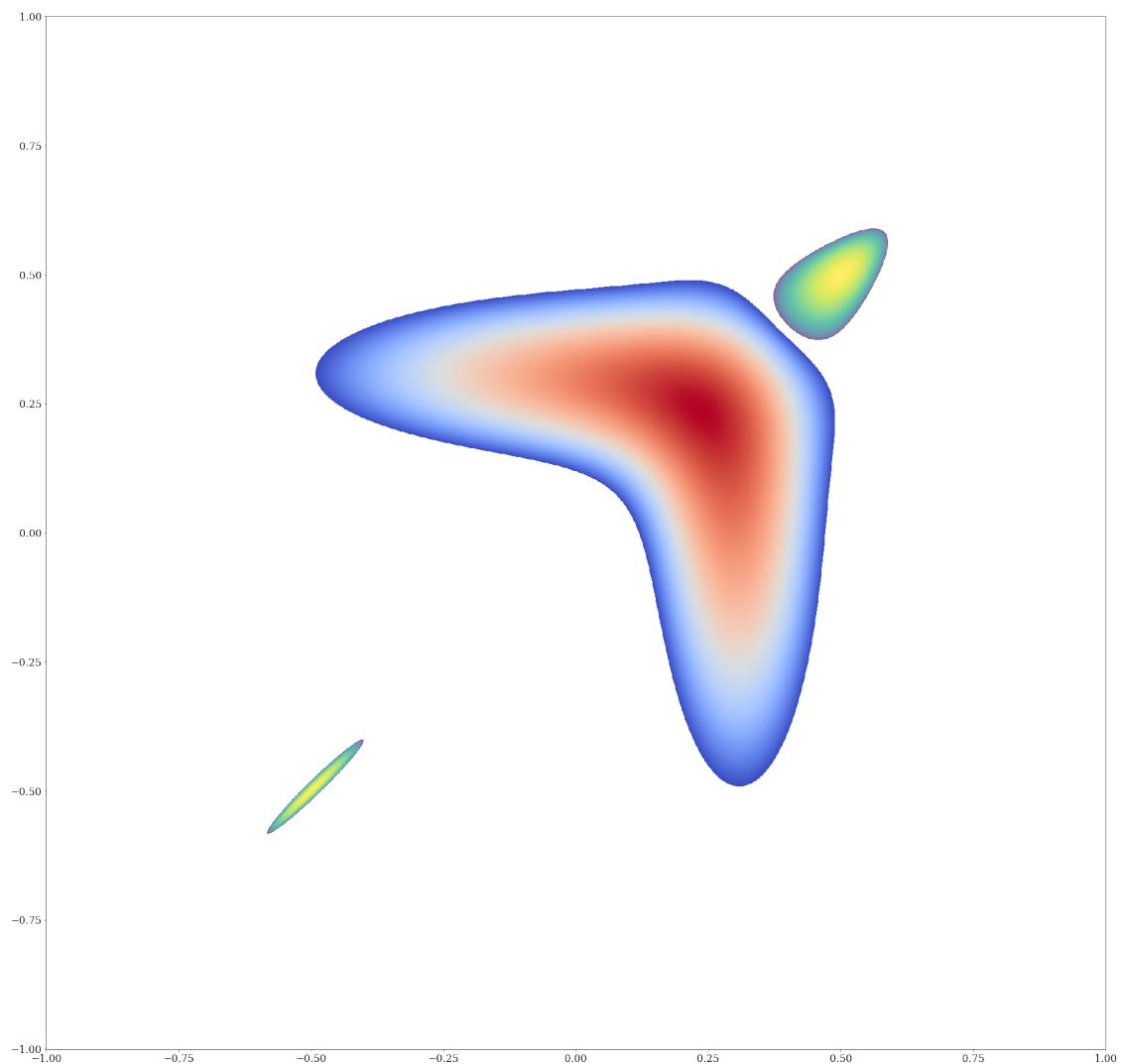
0.10714285714285714



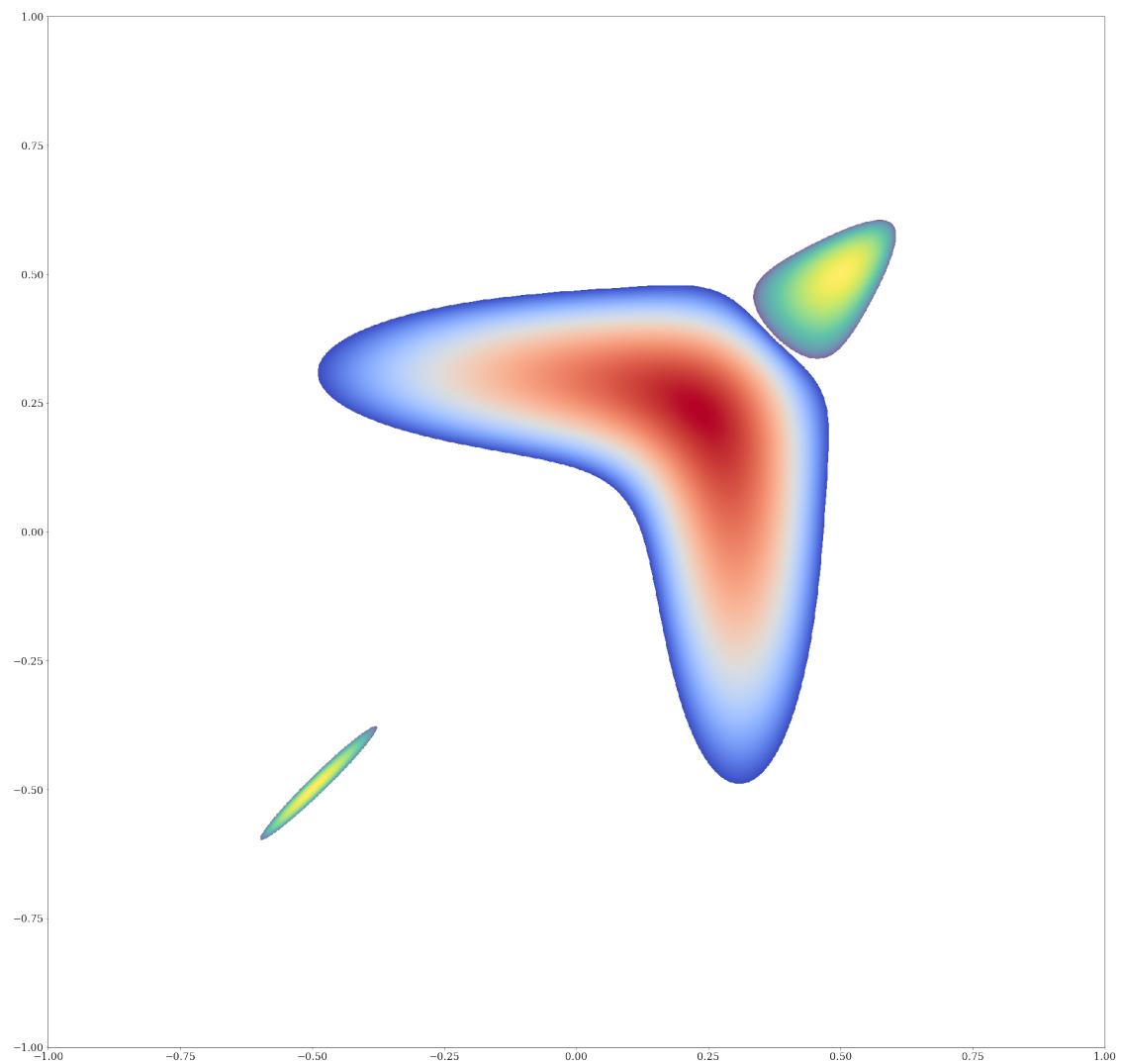
0.14285714285714285



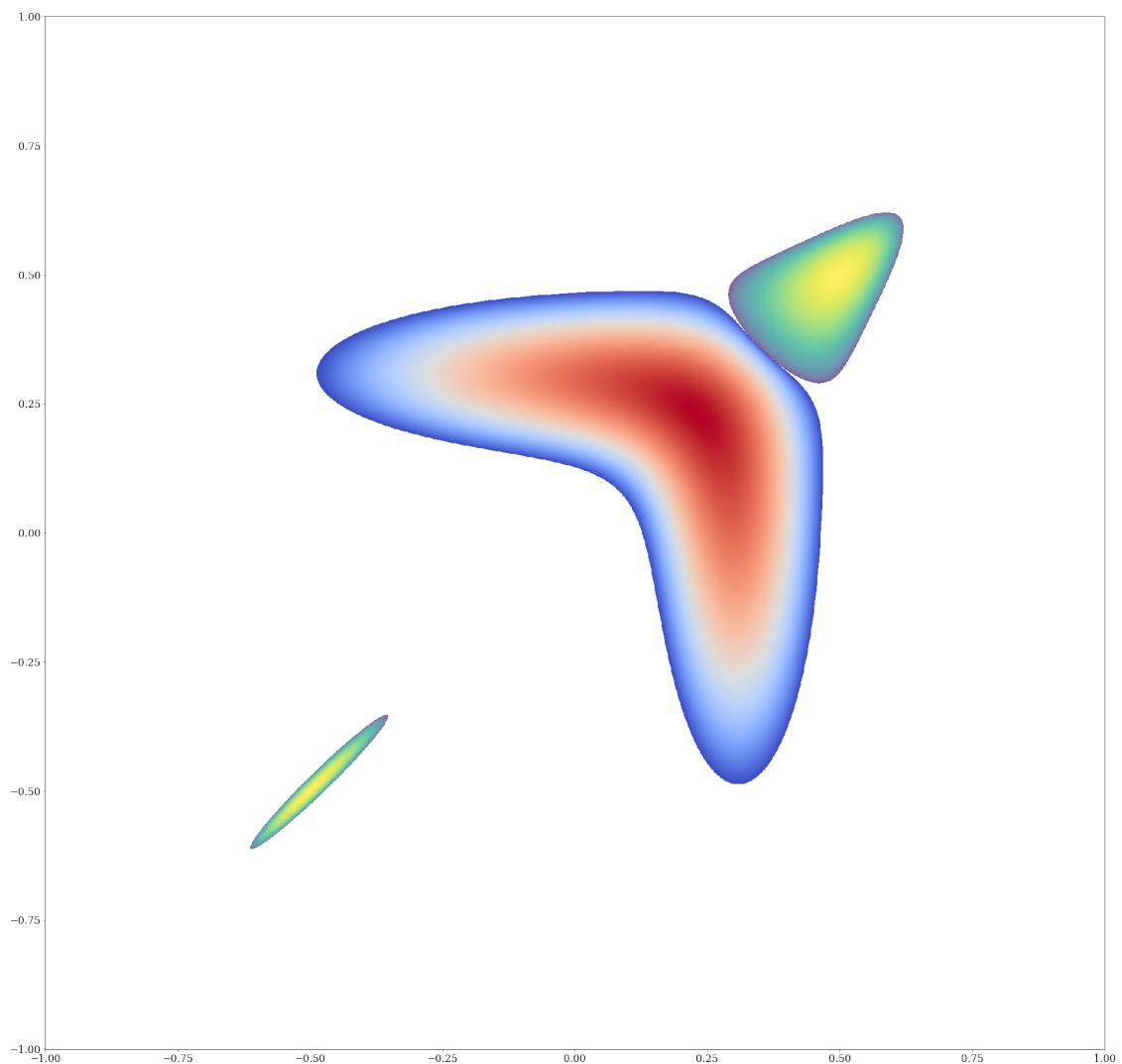
0.17857142857142855



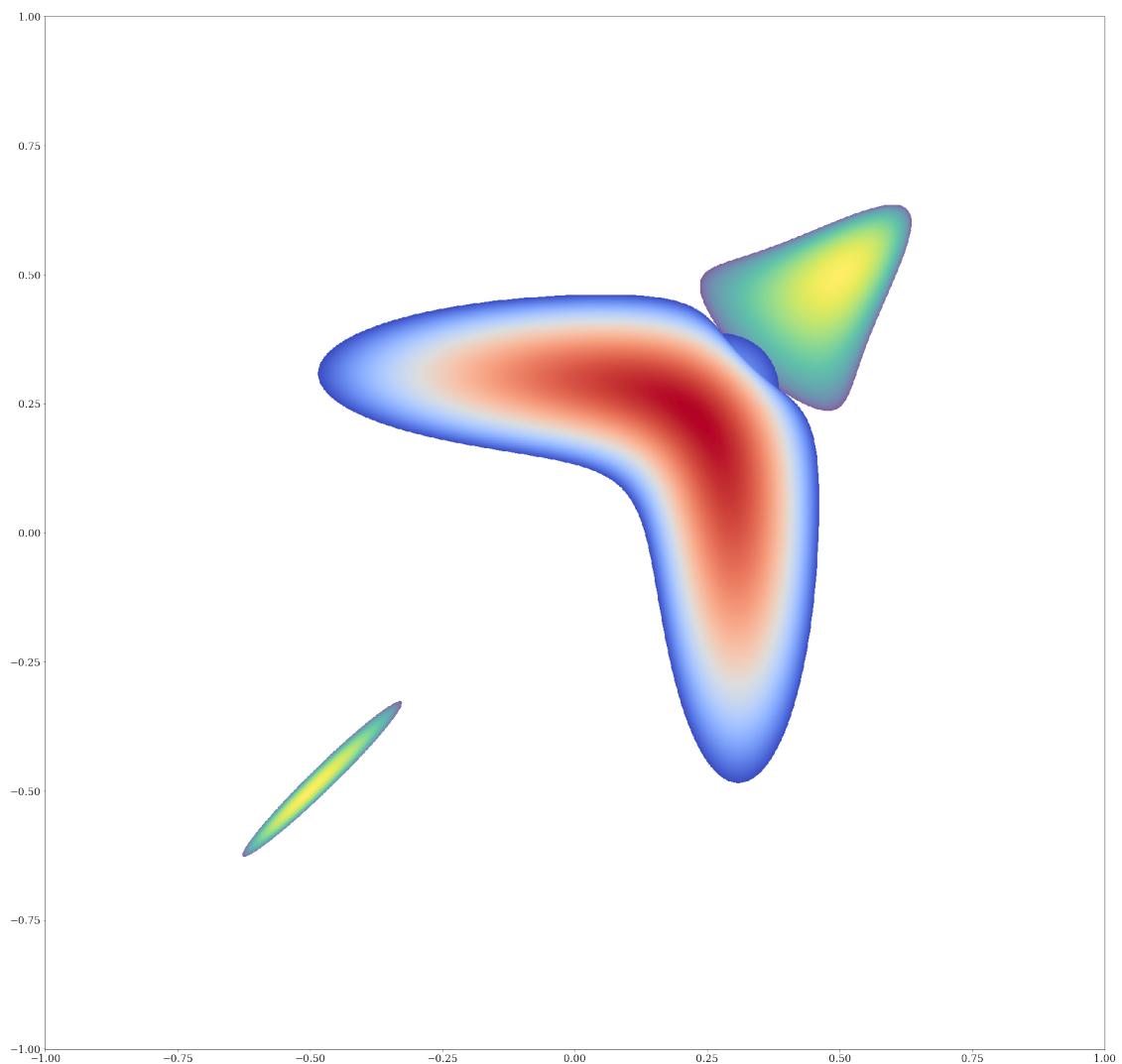
0.21428571428571427



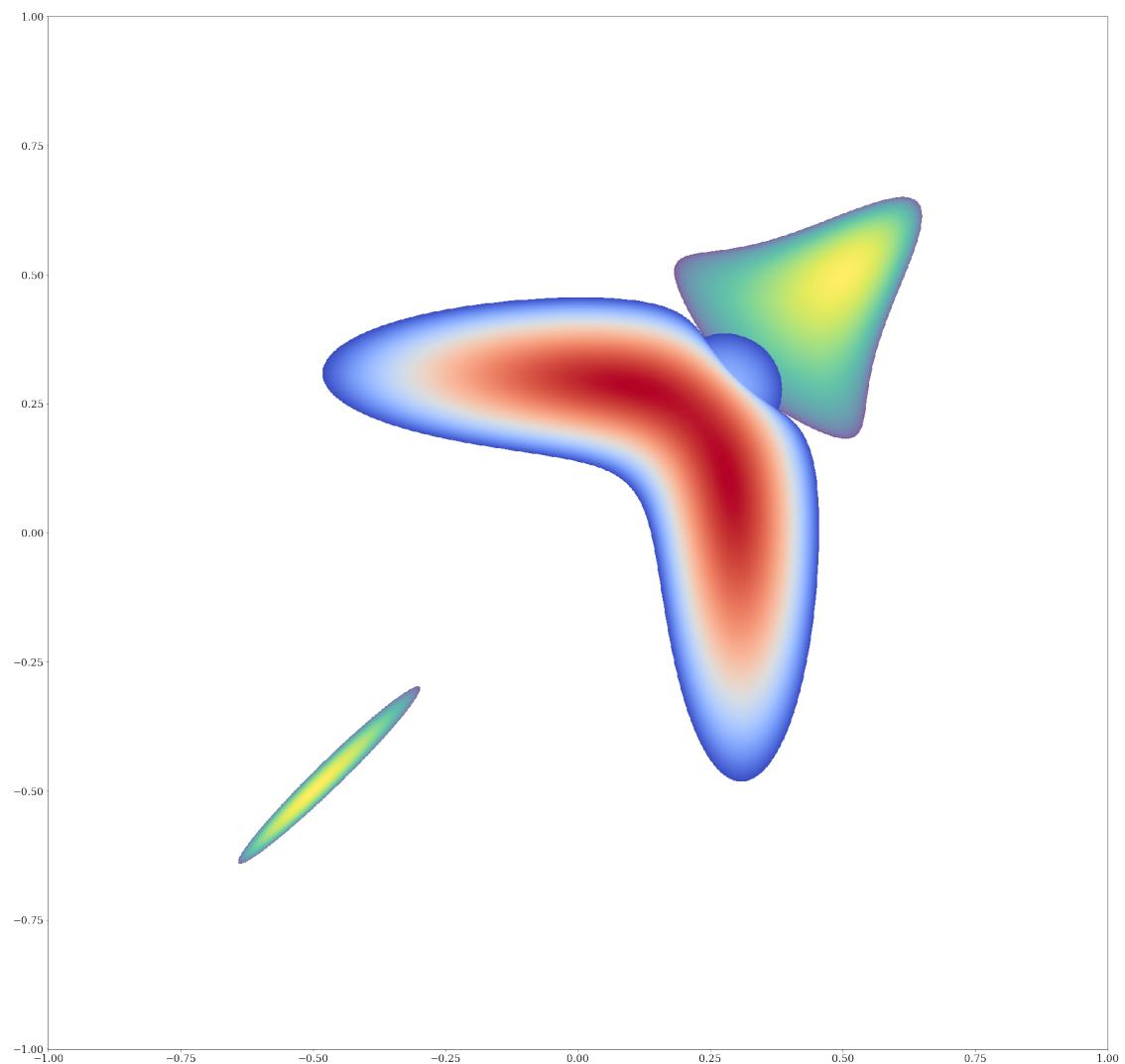
0.25



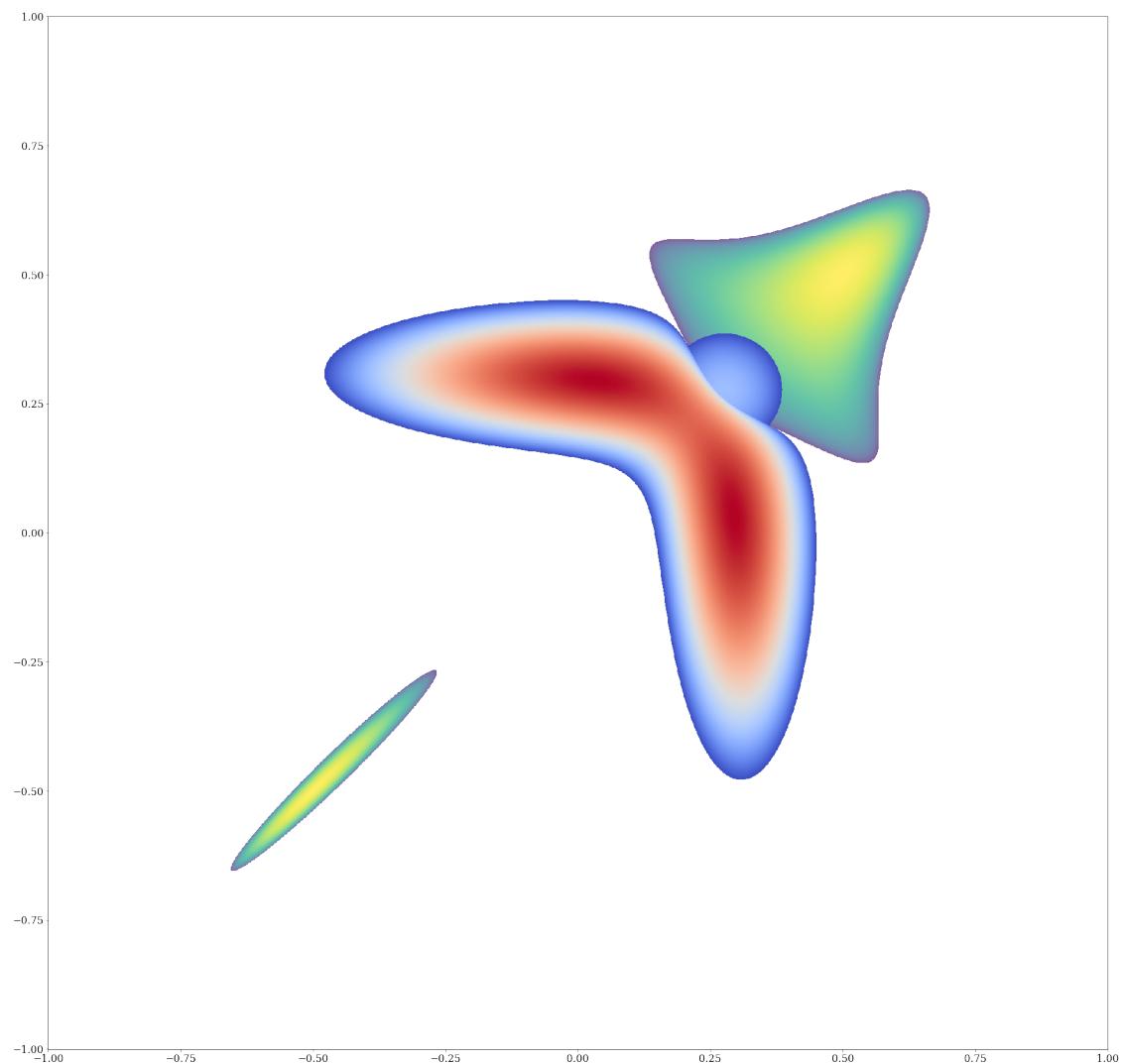
0.2857142857142857



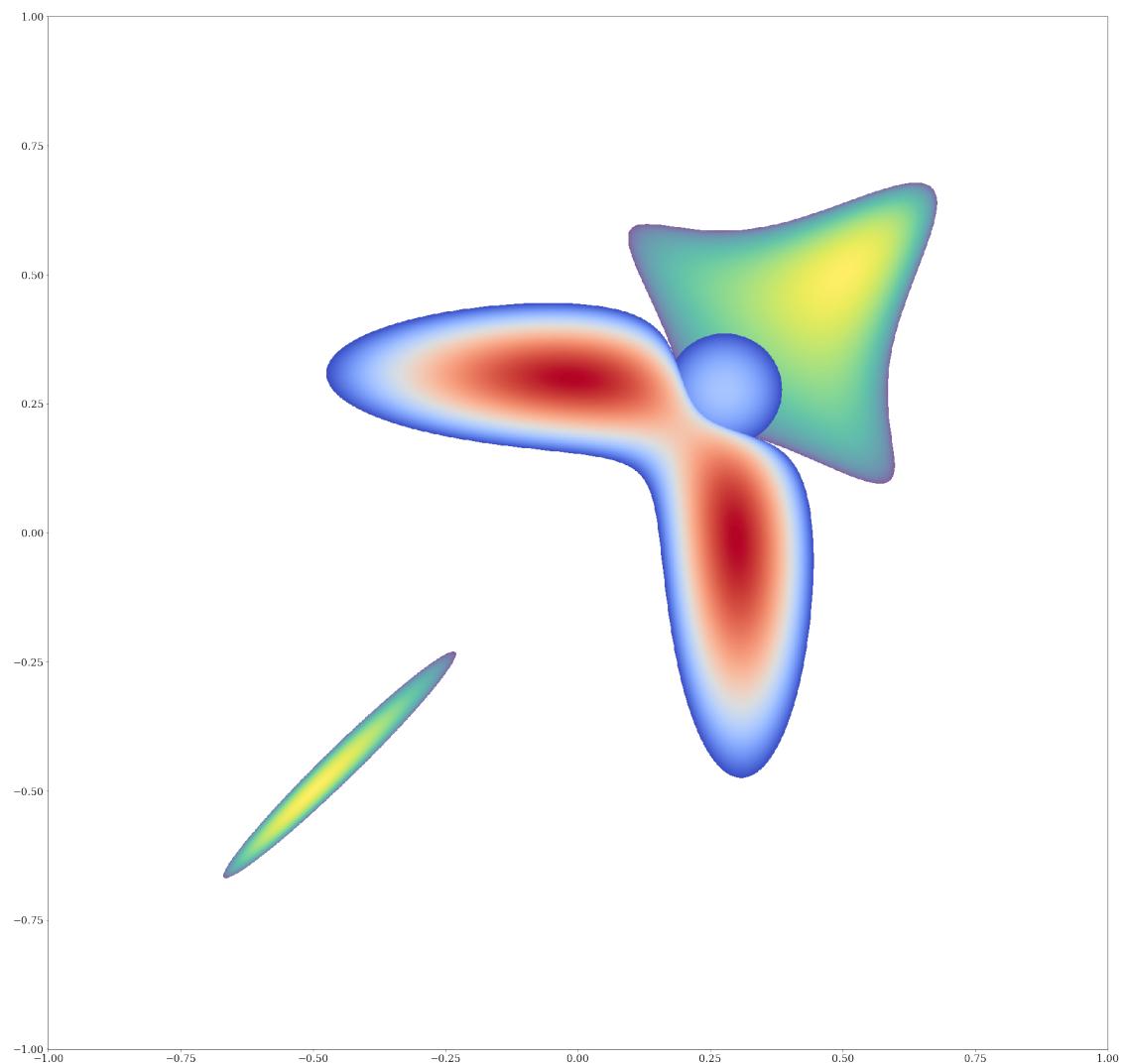
0.3214285714285714



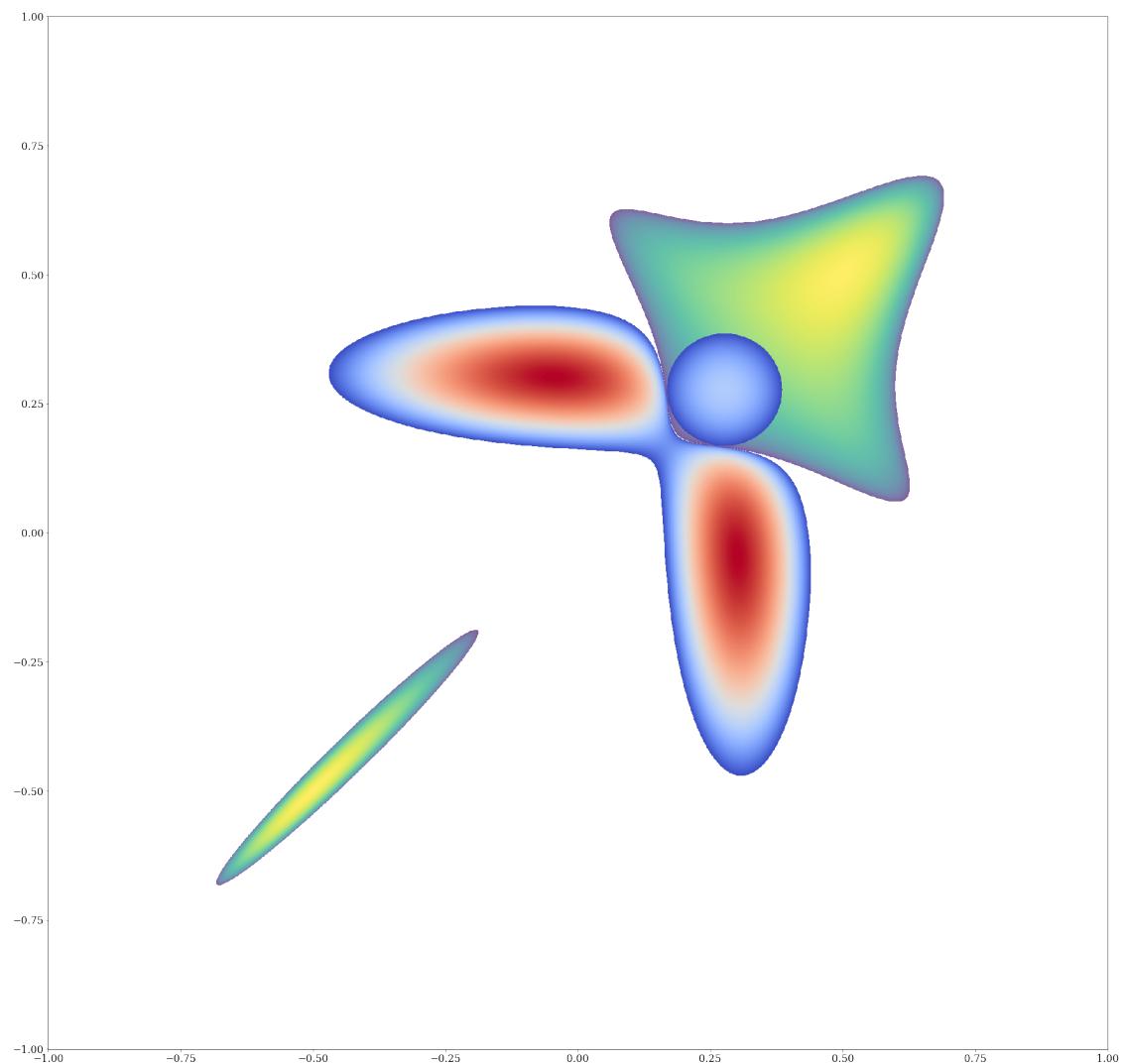
0.3571428571428571



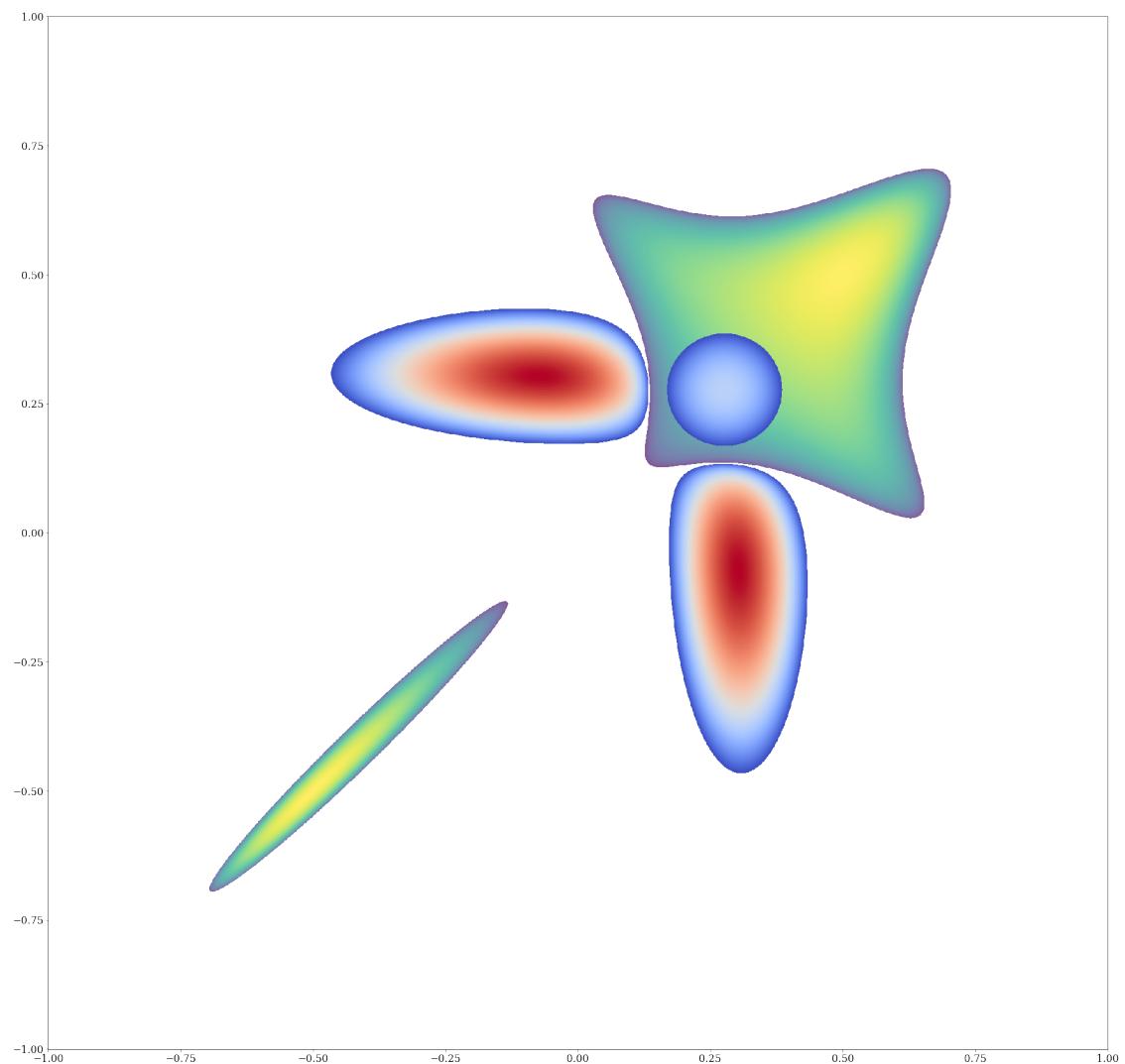
0.39285714285714285



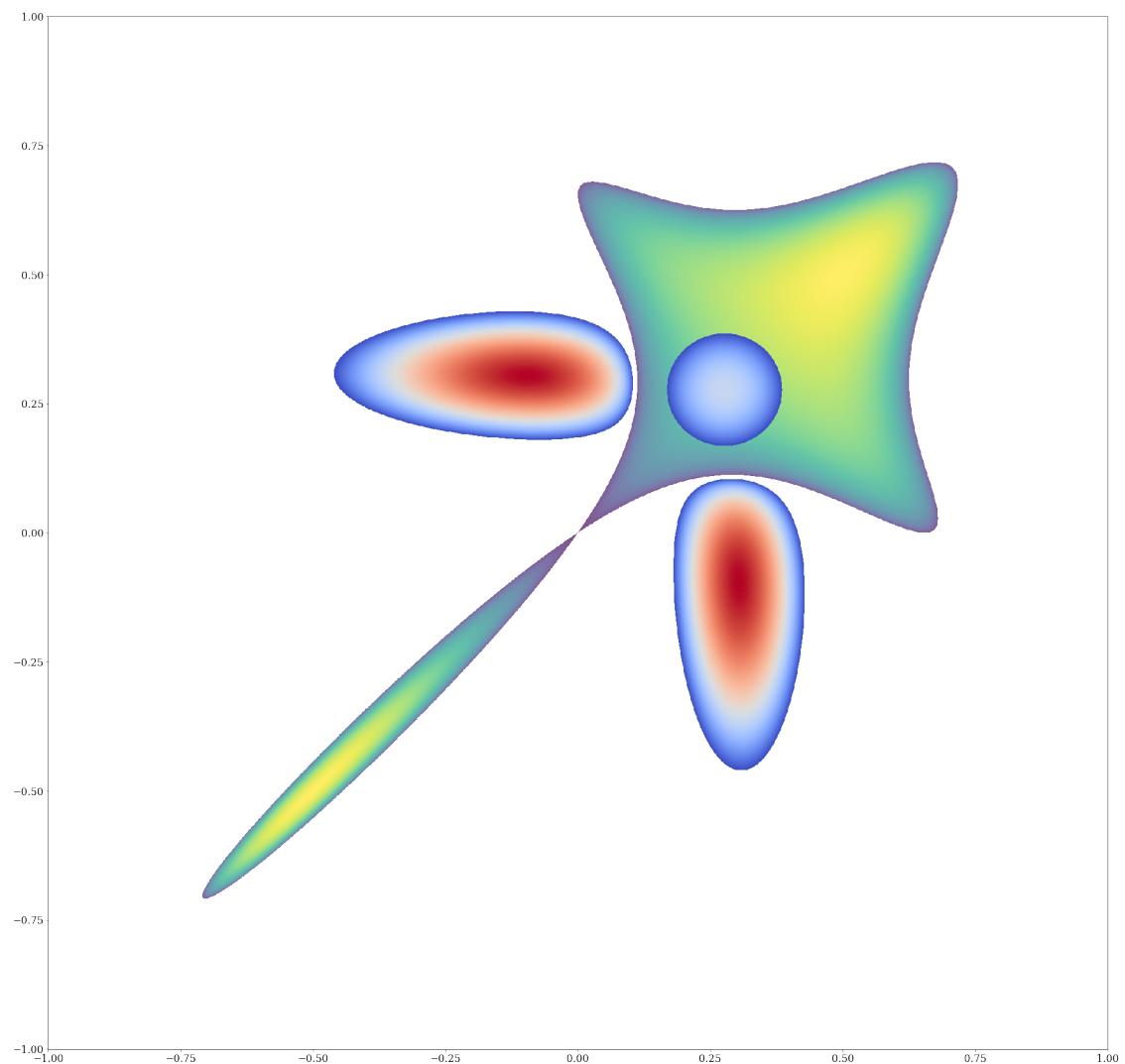
0.42857142857142855



0.46428571428571425



0.5



[ ]:   
[ ]:   
[ ]:   
[ ]: