In [2]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
from sklearn.linear_model import LinearRegression
from sympy import symbols, solve

data = pd.read_csv('m02_bonus.csv')
x = data['x'].values
y = data['y'].values
sigma = data['sigma'].values
```

In [3]:
```
X = np.column_stack([x, y])
model = LinearRegression().fit(X, sigma)

print('Fitted equation: x * (', model.coef_[0], ') + y * (', model.coef_[1], ') + (
```

```
Fitted equation: x * ( 0.13851116932147264 ) + y * ( -0.22081029743649447 ) + ( 9.08
1655840821126 ) = sigma
```

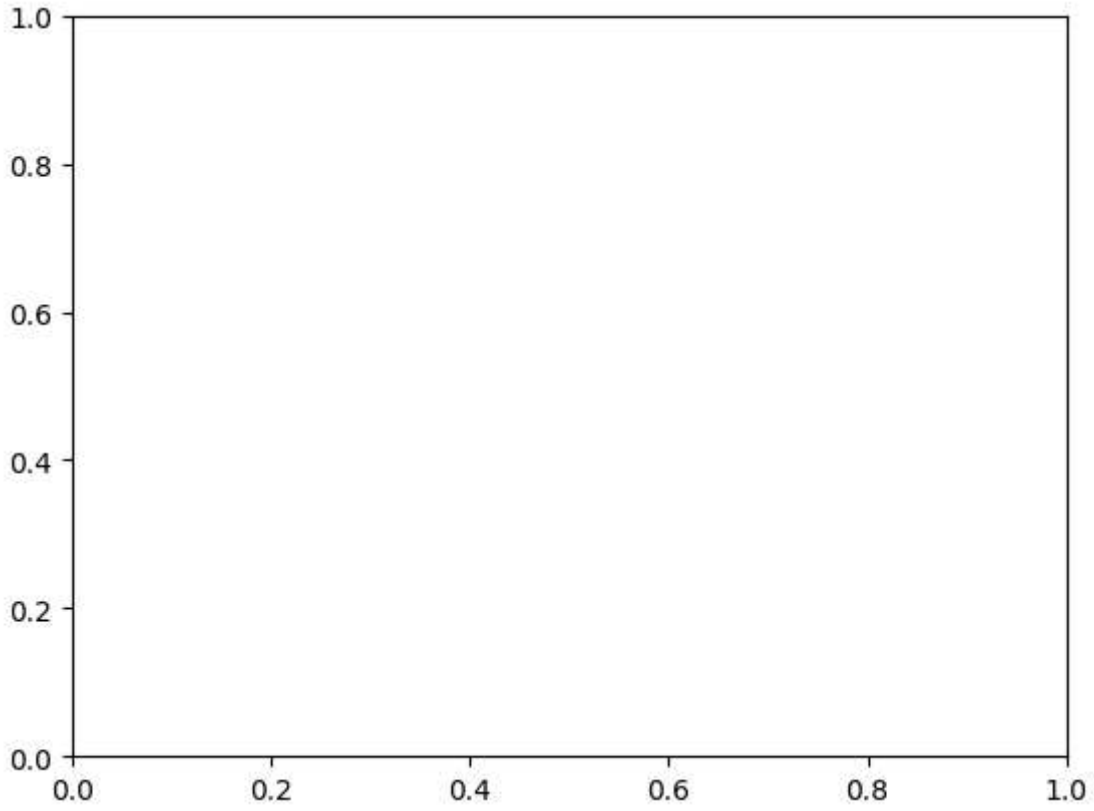
In [4]:
```
# using two data points to infer the values of sigma, R

# for x[0], y[0], calculating r and theta, we get
r0 = np.sqrt(x[0] ** 2 + y[0] ** 2)
theta0 = np.arctan2(y[0], x[0])

# for x[1], y[1]
r1 = np.sqrt(x[0] ** 2 + y[0] ** 2)
theta1 = np.arctan2(y[1], x[1])

R = symbols('R')

expr = sigma[1] * (1 - R ** 2 / r1 ** 2 + np.cos(2 * theta1) * (1 - 4 * R ** 2 / r1
R_sol = solve(expr)
R_vals = np.array([R_sol[2], R_sol[3]])

sigma_sol = sigma[0] * 2 / (1 - np.power(R_vals, 2) / r1 ** 2 + np.cos(2 * theta1)
print('R: ', R_vals.T, '\nsigma_0: ', sigma_sol.T)
```

```
R:  [4.63643150995978 6.09582686727525]
sigma_0:  [-0.137888698292661 -0.0450673227364446]
```

In [32]:
```
# for R = 4.64
x = np.linspace(-5 * 4.64, 5 * 4.64, 101).reshape(-1, 1)
y = np.linspace(-5 * 4.64, 5 * 4.64, 101).reshape(-1, 1)
X = np.hstack([x, y])

preds = np.zeros([len(x), len(y)])
preds = model.predict(X)

predictions = np.meshgrid(preds)
plt.pcolormesh(X, predictions)
```

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
Cell In[32], line 10
      7 preds = model.predict(X)
      9 predictions = np.meshgrid(preds)
---> 10 plt.pcolormesh(X, predictions)

File c:\Users\barat\AppData\Local\Programs\Python\Python312\Lib\site-packages\matplo
tlib\pyplot.py:3697, in pcolormesh(alpha, norm, cmap, vmin, vmax, shading, antialias
ed, data, *args, **kwargs)
   3684 @_copy_docstring_and_deprecators(Axes.pcolormesh)
   3685 def pcolormesh(
   3686     *args: ArrayLike,
   (...)
   3695     **kwargs,
   3696 ) -> QuadMesh:
-> 3697     __ret = gca().pcolormesh(
   3698         *args,
   3699         alpha=alpha,
   3700         norm=norm,
   3701         cmap=cmap,
   3702         vmin=vmin,
   3703         vmax=vmax,
   3704         shading=shading,
   3705         antialiased=antialiased,
   3706         **({"data": data} if data is not None else {}),
   3707         **kwargs,
   3708     )
   3709     sci(__ret)
   3710     return __ret

File c:\Users\barat\AppData\Local\Programs\Python\Python312\Lib\site-packages\matplo
tlib\__init__.py:1473, in _preprocess_data.<locals>.inner(ax, data, *args, **kwargs)
   1470 @functools.wraps(func)
   1471 def inner(ax, *args, data=None, **kwargs):
   1472     if data is None:
-> 1473         return func(
   1474             ax,
   1475             *map(sanitize_sequence, args),
   1476             **{k: sanitize_sequence(v) for k, v in kwargs.items()})
   1478     bound = new_sig.bind(ax, *args, **kwargs)
   1479     auto_label = (bound.arguments.get(label_namer)
   1480                   or bound.kwargs.get(label_namer))

File c:\Users\barat\AppData\Local\Programs\Python\Python312\Lib\site-packages\matplo
tlib\axes\_axes.py:6428, in Axes.pcolormesh(self, alpha, norm, cmap, vmin, vmax, sha
ding, antialiased, *args, **kwargs)
   6425 shading = shading.lower()
   6426 kwargs.setdefault('edgecolors', 'none')
-> 6428 X, Y, C, shading = self._pcolorargs('pcolormesh', *args,
   6429                                      shading=shading, kwargs=kwargs)
   6430 coords = np.stack([X, Y], axis=-1)
   6432 kwargs.setdefault('snap', mpl.rcParams['pcolormesh.snap'])

File c:\Users\barat\AppData\Local\Programs\Python\Python312\Lib\site-packages\matplo
tlib\axes\_axes.py:5953, in Axes._pcolorargs(self, funcname, shading, *args, **kwarg
```

```
s)
   5951     nrows, ncols = C.shape[:2]
   5952 else:
-> 5953     raise _api.nargs_error(funcname, takes="1 or 3", given=len(args))
   5955 Nx = X.shape[-1]
   5956 Ny = Y.shape[0]

TypeError: pcolormesh() takes 1 or 3 positional arguments but 2 were given
```



In [ ]: