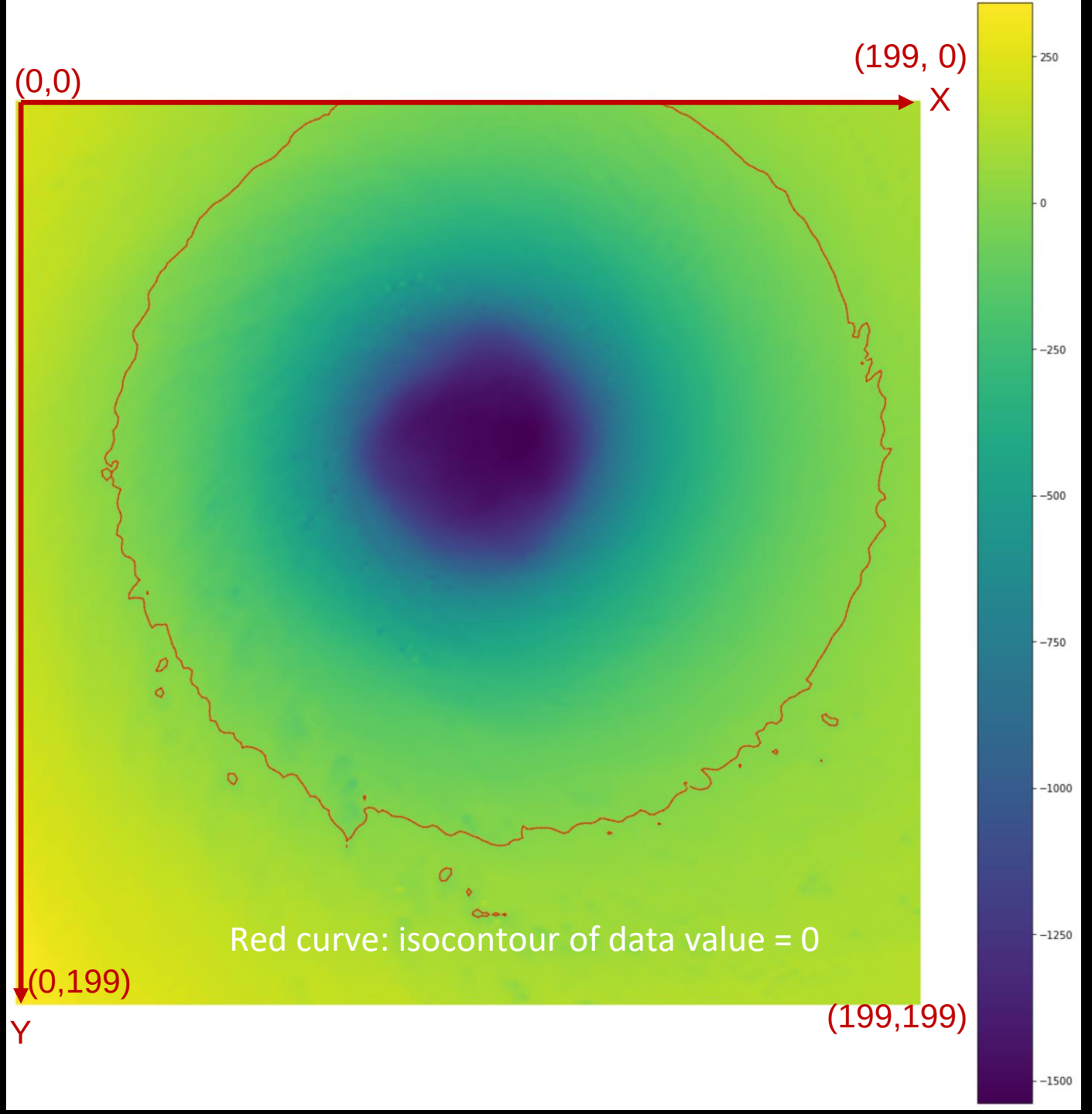
Isocontour (2D)

(A slice of hurricane pressure dataset)

This is a 200X200 2D dataset
Data value is the data pressure
This image is visualization of this 2D pressure dataset
The center circle hurricane eye

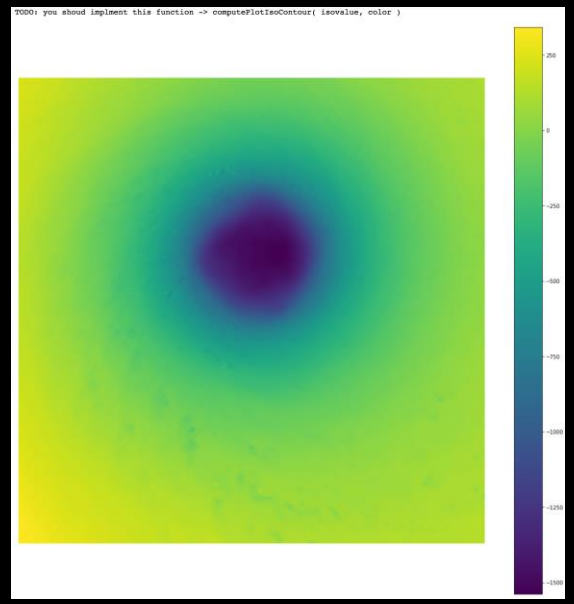


Files

- Isocontour.ipynb: code template and you should complete this homework in this file and submit this file
- plotData.npy and rawData.npy: data set (you need them in the working folder)

Template

You can directly run the template. But it only show the data image without any isocontour




```python
import numpy as np
import matplotlib.pyplot as plt

data2D = 0
data2DPlot = 0

##### draw a line segment between [x0, y0] -> [x1, y1].
##### color: color of the line segment
##### DO NOT modify this function and you also do not understand the code inside this function
def plotOneEdge(x0, y0, x1, y1, color="white"):
    plt.plot([y0*5,y1*5], [x0*5,x1*5], linewidth=1, color=color)

##### Get a data value at (x, y)
##### x, y: location.
##### Return: data value at (x,y)
##### DO NOT modify this function and you also do not understand the code inside this function
def getDataValue(x, y):
    return data2D[x, y]

##### data loading and setup/plot image
##### DO NOT modify this function and you also do not understand the code inside this function
def Initialize():
    global data2D
    global data2DPlot
    data2D = np.load("rawData.npy")
    data2DPlot = np.load("plotData.npy")

    plt.axis('off')
    plt.rcParams['figure.figsize'] = [20, 20]
    plt.imshow(data2DPlot)
    plt.colorbar()

##### (TODO) WORK on this function
#####compute and draw the isocontour of the given datavalue ("isovalue")
#####color: isocontour color
##### you should use "getDataVlue()" to get the data you want and use "plotOneEdge()" to a segment of the isocontour
##### I do not mind the computation is efficnet or not
def computePlotIsoContour( isovalue, color ):
    print("TODO: you shoud implment this function -> computePlotIsoContour( isovalue, color )")


##### main
Initialize()

##### You can modify this function call to test your program on different isovalues
computePlotIsoContour(0, "red")

plt.show()
```

Template

Main procedure. You can change the argument in “computePlotIsoContour” to test different isovalues

(Do not remove Initialize() and plt.show())

```python
import numpy as np
import matplotlib.pyplot as plt

data2D = 0
data2DPlot = 0

##### draw a line segment between [x0, y0] -> [x1, y1].
##### color: color of the line segment
##### DO NOT modify this function and you also do not understand the code inside this function
def plotOneEdge(x0, y0, x1, y1, color="white"):
    plt.plot([y0*5,y1*5], [x0*5,x1*5], linewidth=1, color=color)

##### Get a data value at (x, y)
##### x, y: location.
##### Return: data value at (x,y)
##### DO NOT modify this function and you also do not understand the code inside this function
def getDataValue(x, y):
    return data2D[x, y]

##### data loading and setup/plot image
##### DO NOT modify this function and you also do not understand the code inside this function
def Initialize():
    global data2D
    global data2DPlot
    data2D = np.load("rawData.npy")
    data2DPlot = np.load("plotData.npy")

    plt.axis('off')
    plt.rcParams['figure.figsize'] = [20, 20]
    plt.imshow(data2DPlot)
    plt.colorbar()

##### (TODO) WORK on this function
#####compute and draw the isocontour of the given datavalue ("isovalue")
#####color: isocontour color
##### you should use "getDataVlue()" to get the data you want and use "plotOneEdge()" to a segment of the isocontour
##### I do not mind the computation is efficnet or not
def computePlotIsoContour( isovalue, color ):
    print("TODO: you shoud implment this function -> computePlotIsoContour( isovalue, color )")
```

```python
##### main
Initialize()

##### You can modify this function call to test your program on different isovalues
computePlotIsoContour(0, "red")

plt.show()
```

Template

```python
import numpy as np
import matplotlib.pyplot as plt

data2D = 0
data2DPlot = 0

##### draw a line segment between [x0, y0] -> [x1, y1].
##### color: color of the line segment
##### DO NOT modify this function and you also do not understand the code inside this function
def plotOneEdge(x0, y0, x1, y1, color="white"):
    plt.plot([y0*5,y1*5], [x0*5,x1*5], linewidth=1, color=color)

##### Get a data value at (x, y)
##### x, y: location.
##### Return: data value at (x,y)
##### DO NOT modify this function and you also do not understand the code inside this function
def getDataValue(x, y):
    return data2D[x, y]
```

```python
##### data loading and setup/plot image
##### DO NOT modify this function and you also do not understand the code inside this function
def Initialize():
    global data2D
    global data2DPlot
    data2D = np.load("rawData.npy")
    data2DPlot = np.load("plotData.npy")

    plt.axis('off')
    plt.rcParams['figure.figsize'] = [20, 20]
    plt.imshow(data2DPlot)
    plt.colorbar()
```

```python
##### (TODO) WORK on this function
#####compute and draw the isocontour of the given datavalue ("isovalue")
#####color: isocontour color
##### you should use "getDataVlue()" to get the data you want and use "plotOneEdge()" to a segment of the isocontour
##### I do not mind the computation is efficnet or not
def computePlotIsoContour( isovalue, color ):
    print("TODO: you shoud implment this function -> computePlotIsoContour( isovalue, color )")


##### main
Initialize()

##### You can modify this function call to test your program on different isovalues
computePlotIsoContour(0, "red")

plt.show()
```

Template

This is the function you should complete. (I do not mind the efficiency of your implementation)

```python
import numpy as np
import matplotlib.pyplot as plt

data2D = 0
data2DPlot = 0

##### draw a line segment between [x0, y0] -> [x1, y1].
##### color: color of the line segment
##### DO NOT modify this function and you also do not understand the code inside this function
def plotOneEdge(x0, y0, x1, y1, color="white"):
    plt.plot([y0*5,y1*5], [x0*5,x1*5], linewidth=1, color=color)

##### Get a data value at (x, y)
##### x, y: location.
##### Return: data value at (x,y)
##### DO NOT modify this function and you also do not understand the code inside this function
def getDataValue(x, y):
    return data2D[x, y]

##### data loading and setup/plot image
##### DO NOT modify this function and you also do not understand the code inside this function
def Initialize():
    global data2D
    global data2DPlot
    data2D = np.load("rawData.npy")
    data2DPlot = np.load("plotData.npy")

    plt.axis('off')
    plt.rcParams['figure.figsize'] = [20, 20]
    plt.imshow(data2DPlot)
    plt.colorbar()

##### (TODO) WORK on this function
#####compute and draw the isocontour of the given datavalue ("isovalue")
#####color: isocontour color
##### you should use "getDataVlue()" to get the data you want and use "plotOneEdge()" to a segment of the isocontour
##### I do not mind the computation is efficnet or not
def computePlotIsoContour( isovalue, color ):
    print("TODO: you shoud implment this function -> computePlotIsoContour( isovalue, color )")


##### main
Initialize()

##### You can modify this function call to test your program on different isovalues
computePlotIsoContour(0, "red")

plt.show()
```

Template

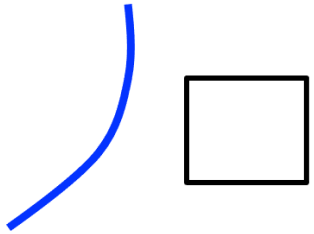
```python
import numpy as np
import matplotlib.pyplot as plt

data2D = 0
data2DPlot = 0

##### draw a line segment between [x0, y0] -> [x1, y1].
##### color: color of the line segment
##### DO NOT modify this function and you also do not understand the code inside this function
def plotOneEdge(x0, y0, x1, y1, color="white"):
    plt.plot([y0*5,y1*5], [x0*5,x1*5], linewidth=1, color=color)

##### Get a data value at (x, y)
##### x, y: location.
##### Return: data value at (x,y)
##### DO NOT modify this function and you also do not understand the code inside this function
def getDataValue(x, y):
    return data2D[x, y]

##### data loading and setup/plot image
##### DO NOT modify this function and you also do not understand the code inside this function
def Initialize():
    global data2D
    global data2DPlot
    data2D = np.load("rawData.npy")
    data2DPlot = np.load("plotData.npy")

    plt.axis('off')
    plt.rcParams['figure.figsize'] = [20, 20]
    plt.imshow(data2DPlot)
    plt.colorbar()

##### (TODO) WORK on this function
#####compute and draw the isocontour of the given datavalue ("isovalue")
#####color: isocontour color
##### you should use "getDataVlue()" to get the data you want and use "plotOneEdge()" to a segment of the isocontour
##### I do not mind the computation is efficnet or not
def computePlotIsoContour( isovalue, color ):
    print("TODO: you shoud implment this function -> computePlotIsoContour( isovalue, color )")


##### main
Initialize()

##### You can modify this function call to test your program on different isovalues
computePlotIsoContour(0, "red")

plt.show()
```

Template

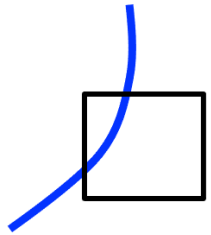
```python
import numpy as np
import matplotlib.pyplot as plt

data2D = 0
data2DPlot = 0

##### draw a line segment between [x0, y0] -> [x1, y1].
##### color: color of the line segment
##### DO NOT modify this function and you also do not understand the code inside this function
def plotOneEdge(x0, y0, x1, y1, color="white"):
    plt.plot([y0*5,y1*5], [x0*5,x1*5], linewidth=1, color=color)

##### Get a data value at (x, y)
##### x, y: location.
##### Return: data value at (x,y)
##### DO NOT modify this function and you also do not understand the code inside this function
def getDataValue(x, y):
    return data2D[x, y]

##### data loading and setup/plot image
##### DO NOT modify this function and you also do not understand the code inside this function
def Initialize():
    global data2D
    global data2DPlot
    data2D = np.load("rawData.npy")
    data2DPlot = np.load("plotData.npy")

    plt.axis('off')
    plt.rcParams['figure.figsize'] = [20, 20]
    plt.imshow(data2DPlot)
    plt.colorbar()

##### (TODO) WORK on this function
#####compute and draw the isocontour of the given datavalue ("isovalue")
#####color: isocontour color
##### you should use "getDataVlue()" to get the data you want and use "plotOneEdge()" to a segment of the isocontour
##### I do not mind the computation is efficnet or not
def computePlotIsoContour( isovalue, color ):
    print("TODO: you shoud implment this function -> computePlotIsoContour( isovalue, color )")


##### main
Initialize()

##### You can modify this function call to test your program on different isovalues
computePlotIsoContour(0, "red")

plt.show()
```

Unique Topological Cases

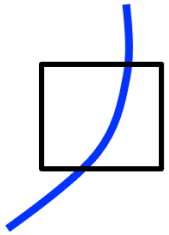
- There are only four unique topological cases



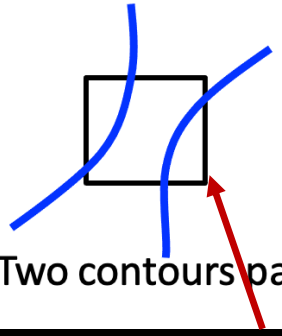
(1) No intersection



(2) Intersect with two adjacent edges



(3) Intersect with two opposite cases



(4) Two contours pass through the cell

- In this homework, you can ignore this case (do not have to implement it). This data rarely has this case.