

# HW 6 Yokoi Connectivity Number

## Yokoi Connectivity Number

---

### Source code

Please refer to the file “hw6.py” within the same folder as this report document.  
There is a class “RegionalOperator” I defined for representing a regional operator.  
An instance of this class can be initiated with two function:  $h()$  and  $f()$ .

```
10 class RegionalOperator:
11     def __init__(self, func1, func2):
12         self.f = func1
13         self.h = func2
14     def getValue(self, *x):
15         a1 = self.h(x[0], x[1], x[6], x[2])
16         a2 = self.h(x[0], x[2], x[7], x[3])
17         a3 = self.h(x[0], x[3], x[8], x[4])
18         a4 = self.h(x[0], x[4], x[5], x[1])
19         return self.f(a1, a2, a3, a4)
```

After implementing the two functions for Yokoi, a Yokoi operator can be initialized.

```
49 Yokoi = RegionalOperator(YokoiF, YokoiH)
```

By the way, for downsampling, there's a function for doing it.

```
84 def downSample(src, block=(8, 8)):
85     # calculate the size of the down-sampled image
86     newImageSize = [0, 0]
87     newImageSize[0] = int(math.ceil(float(src.size[0]) / block[0]))
88     newImageSize[1] = int(math.ceil(float(src.size[1]) / block[1]))
89     # create the new image
90     newImage = Image.new(src.mode, newImageSize)
91     newImagePixels = newImage.load()
92     # down-sampling
93     srcPixels = src.load()
94     for i in range(newImageSize[0]):
95         for j in range(newImageSize[1]):
96             newImagePixels[i, j] = srcPixels[i*block[0], j*block[1]]
97     return newImage
```

## Result

(the resulted 64x64 matrix is as followed, and it's saved as "yokoi\_result.txt" as well.)

```

11111111 12111111111122322221 1111111111111 1155555555511 21
15555551 1155555555511 2 11 11 1555555555511 1
15555551 1 2115555112 21112221 155555555551 1
15555551 1 2 155112 22221511 1555555555511 1
15555551 22 2112 22 121 1555555555511
15555551 1 2 21 2 1 1555555555551
15555551 12 1 121111 1321 15555555555511
15111551 1322 1155551111 15555555555551
111 1551 1 121555555511 15555555555511
11 1551 21155555511 155115555511
21 1551 2 15555555111 1551 11555511
1 1551 2 155555555511 1551 115551
1551 11211555555551 1551 15511 12
1551 155555555555511 1551 1111 111
1551 1 22211555555555511 1151 11 1151
1551 2 22 1 155555555555511 151 11111 1551
1551 2 1 1155555555555551 151 115551 11551
1551 2 1155555555555555111511155511 115551
1551 12 115555555555555555555551 155551
1551 11 221555555555555555555555112 1155551
1551 111 22 15555555555555555555551 1 1555551
1551 1511 1 125112111111211155555555111 1155551
1551 15521 1 121 1 11 1 15555555111 1555551
1551 1151 132 2 11555555111 11555551
1551 151 322 115555111 121 15555551
1551 1221 2 1555551 131 115555551
1551 2 1 115555511 1 115555551
1551 2 115555551 1 155555551
1551 2 1155555551 2115555551
1551 1 11555555551 1555555551
1551 1 11511115555521 1 11555555551
1551 1 1 11111 1155511 2 15555555551
1551 131 111 15111 2 15555555551
1551 121 1121 1 111 1 2 115555555551
1551 11 111 1 221 11 1 2 155555555551
1551 12 1 21 121 11 1111 2 155555555551
1551 1 12 22 151111111551 2 1155555555551
1551 1 2 1555551115511 1 1555555555551
1551 2 22 12555551 15551 1 1555555555551
1551 1 1555511 11511 2 11555555555551
1551 21 155551 1 151 2 15555555555551
1551 2 15555112 151 2 15555555555551
1551 1 1 1155555511111 2 15555555555551
1551 2 22 111511111212 2115555555555551
1551 1 12 151 2 1 1555555511155551
1551 1111 121 15555551 1555551
1551 11111111 15555551 1555551
1551 115551 15555551 1555511
1551 15551 211111111 155511
11521 1 12 122155511 2 11 115511
1 151 1 15555111 2111 15511
22 1511 1 15555555111 155111 1511
22 1511 1 15555555551 155551 1151
2 151 11155555555511 155511 1511
2 1521 155555555555511 15551 12151
2 151 155555555555551 155511 1551
2 1511 155555555555551 115551 1511
21 1511 155555555555551 111111151
11 151 11555555555555511 111511
11 151 15555555555555551 151
11 151 11555555555555551 211
11 151 11555555555555511 1
11 151 1555555555555551
12111111111111111111

```