

Homework #4

20221059 정상목

1. 문제

1. Solve the Laplace equation for a circle. In this time, write a *.contact file (your own format). By using this file, you can specify the edges belonging to the terminals.

2. Write a *.element file (your own format) to specify the yin and yang pattern.

Visualize the read structure. Calculate the interface edges.

2. Contact 구성

*.contact 파일로 정의된 contact 정보를 이용해서 Dirichlet boundary condition을 정해줄 것이다. Homework #3에서 사용한 코드를 활용한다. *.contact을 이용해 결과를 확인한다. *.contact를 수정해 손쉽게 결과를 확인했다.

2.1. 생성할 Mesh 확인

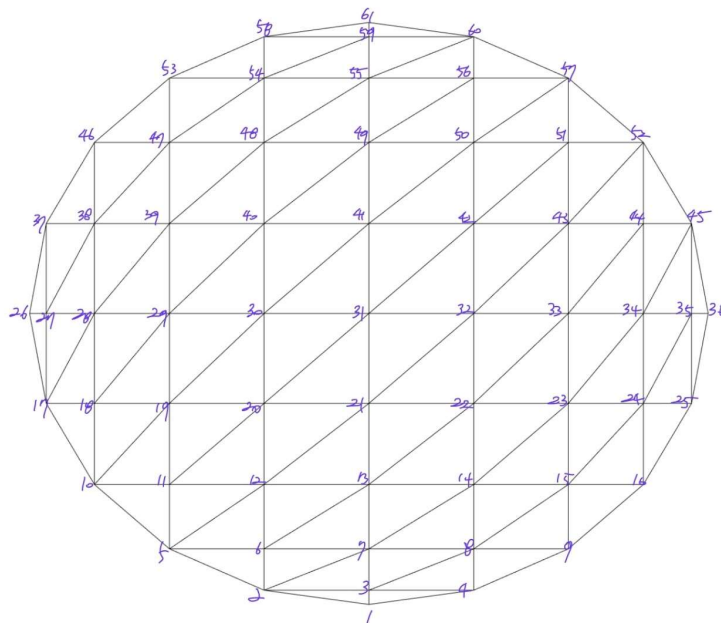


Fig 1. 생성할 Mesh (과제 1).

Homework #3에서 만든 vertex를 수정해 좀 더 조밀하게(N=20) 만들었다. 각 vertex의 index는 Fig 1에 보이는 대로 작성했다. Homework #3에서는 top vertex와 bottom vertex인 index 1, 61 만 Dirichlet boundary condition을 정했다. 이번 과제에서는 index 1, 2, 4, 5, 7은 0V, index 53, 57, 58, 60, 61은 1V로 설정했다.

2.2. *.contact 파일 작성.

```

Dirichlet_BC.txt
파일(F) 편집(E)
1,0
2,0
4,0
5,0
9,0
61,1
60,1
58,1
57,1
53,1

```

Fig 2. *.contact 파일

'Dirichlet_BC.txt'에 contact에 관한 정보를 저장했다. 1열은 index이고 2열은 potential로 지정했다. 행을 기준으로 구분했다. 'Dirichlet_BC.txt'를 Matlab으로 읽어와 Dirichlet boundary condition으로 활용했다. 이 정보에 따라 A, b matrix가 수정되도록 코드를 작성했다. 다른 vertex에 contact를 만들고 싶다면 이 파일을 수정해 손쉽게 결과를 확인할 수 있을 것이다.

2.3. 결과 확인

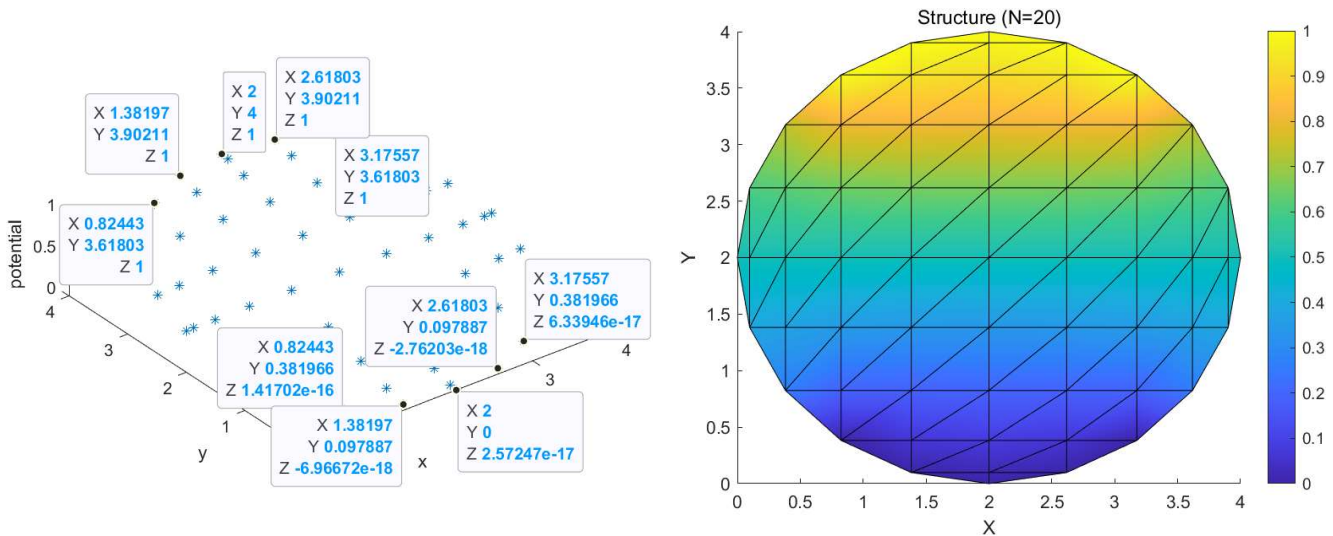


Fig 3. 결과 1.

결과를 확인하면 'Dirichlet_BC.txt'에서 설정한 대로 index 1, 2, 4, 5, 7은 0V, index 53, 57, 58, 60, 61은 1V가 나옴을 확인할 수 있다. Fig 3의 우측 그림을 통해 나머지 지점에서도 결과가 잘 나왔음을 확인할 수 있다.

2.4. 추가적인 contact 지정

```
*Dirichlet_BC.txt
파일(F) 편집(E) 서
1,0
2,0
4,0
5,0
9,0
61,1
60,1
58,1
57,1
53,1
17,-1
26,-1
37,-1
```

Fig 4. 추가적인 Contact를 지정한 *.contact 파일

'Dirichlet_BC.txt'에서 지정한 contact이 잘 작동하는지 확인하기 위해 추가로 index 17, 26, 37에 해당하는 vertex(원의 좌측 부분)에서 -1V의 contact를 진행했다. 다음과 같은 결과가 나왔다.

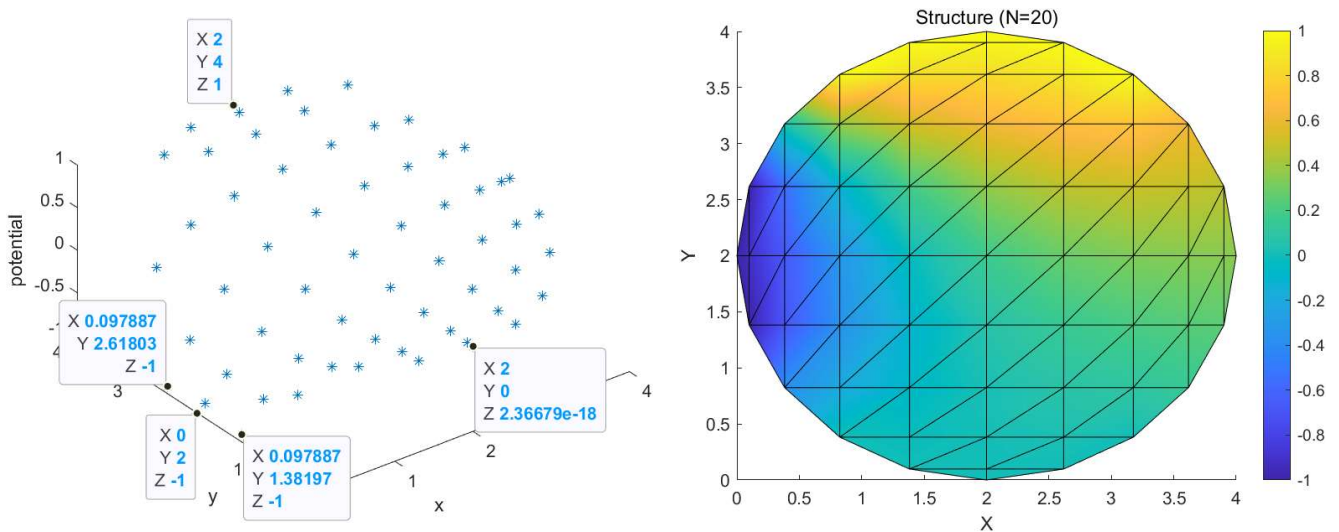


Fig 5. 결과 2.

'Dirichlet_BC.txt'에서 설정한 대로 원의 좌측 부분인 index 17, 26, 37에서 -1이 나와 contact이 잘 되었음을 확인할 수 있다. 우측 그림을 통해 다른 부분도 잘 계산되었음을 확인할 수 있다.

3. Interface 찾기

태극 문양의 Region을 구성한다. 편의상 태극 문양의 위쪽을 ox 부분, 태극 문양의 아래쪽을 Si라고 하였다. 각 Region에 해당하는 element를 작성하고 두 Region 간의 interface에 있는 edge를 계산할 것이다.

3.1. Vertex, element 작성

Vertex는 앞선 과제 1에서 사용한 vertex를 활용했다. 태극 문양을 그리기 쉽게 element를 작성했다. 각 Region에 맞는 element를 작성했다. 생성한 mesh와 각 Region의 모습은 다음과 같다.

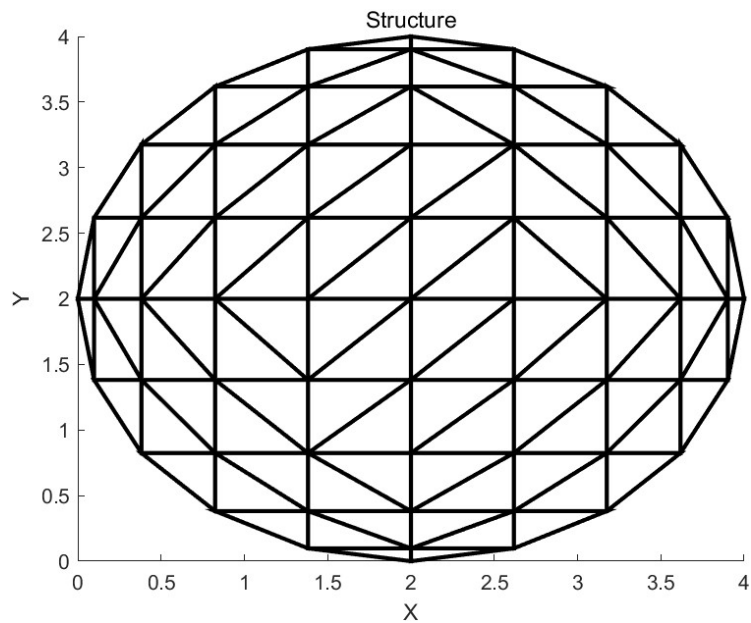


Fig 6. 사용한 mesh (과제 2).

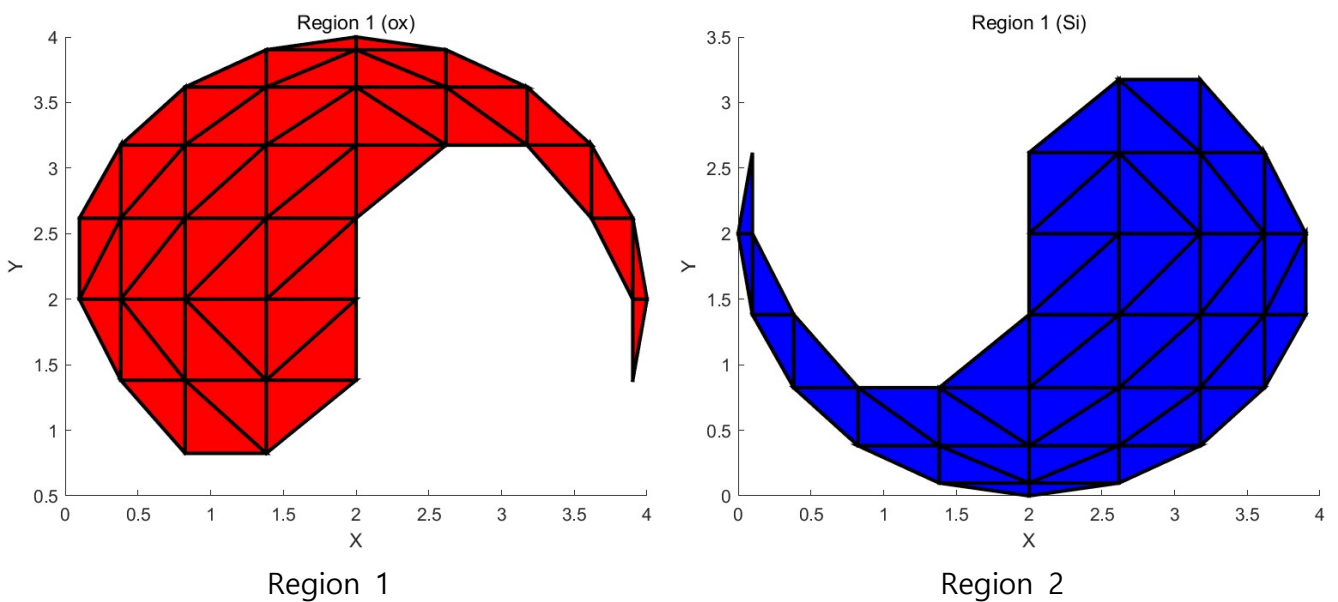


Fig 7. 각 Region

3.2. Interface 계산

Region 1과 Region 2의 interface를 계산하기 위해서 각 Region 별로 작성된 element 파일을 이용했다. element 파일을 이용해 각 Region에 있는 edge 정보를 알 수 있었다. 이 edge 정보들을 비교하여 interface를 계산할 수 있었다. Interface는 다음과 같은 순서로 계산했다.

- 1) 각 Region에 해당하는 element 정보를 이용해서 Region_Si, Region_ox를 구성하는 edge_Si, edge_ox의 정보를 얻는다.
- 2) edge_Si, edge_ox 중 겹치는 edge가 있는지 확인한다. 이때 양쪽 Region에 모두 존재하는 edge가 interface가 될 것이다.

위와 같은 과정을 통해 interface를 구할 수 있었다. 결과는 다음과 같다. Interface 부분은 초록색으로 표시하였다.

edge_interface		
12x2 double		
	1	2
1	11	12
2	12	21
3	18	11
4	21	31
5	27	18
6	31	41
7	35	25
8	37	27
9	41	50
10	44	35
11	50	51
12	51	44

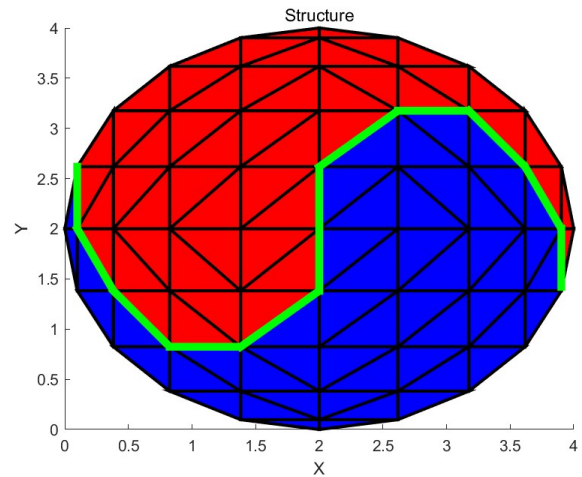


Fig 8. Interface 1.

그림과 같이 결과가 잘 나옴을 확인할 수 있었다.

그러나 총 Vertex는 61개이지만 각 Region 당 edge의 수는 각각 450개로 두 개의 edge를 비교하여 겹치는 부분을 interface로 찾으면 $450 \times 450 = 202,500$ 개의 경우의 수가 있음을 확인할 수 있다. 물론 컴퓨터는 충분히 이를 계산할 수 있지만 경우의 수를 줄이기 위해 Region에서 중복된 edge를 제거했다.





중복 edge 제거 전		중복 edge 제거 후	
	edge_ox		edge_ox
	edge_si		edge_si
	450x2 double		86x2 double
	450x2 double		86x2 double

Fig 9. 중복 edge 제거.

각 Region 별로 edge 수가 450개에서 86개로 줄어들었음을 확인할 수 있다. 따라서 각 edge를 비교할 때 경우의 수가 $20,250$ 개에서 $86 \times 86 = 7,396$ 개로 0.365배로 줄어들었음을 확인할 수 있다.

중복된 edge를 제거하고 계산한 interface가 기존의 interface와 같은지 확인했다.

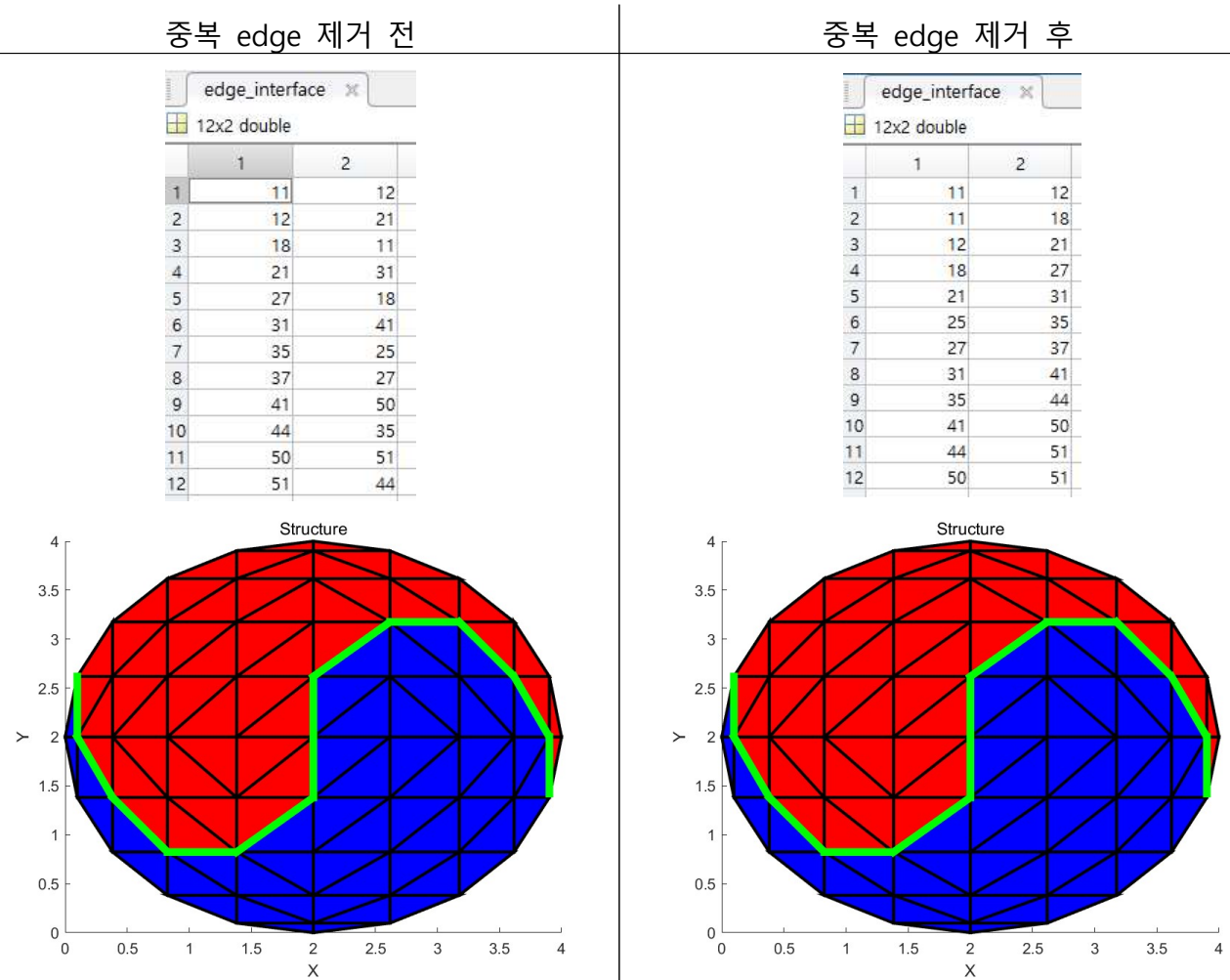


Fig 10. 중복 edge 제거.

결과를 살펴보면 중복된 값 제거 전과 제거 후가 결과가 같음을 확인했다.