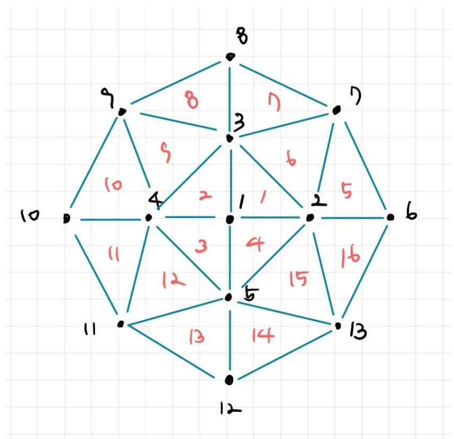
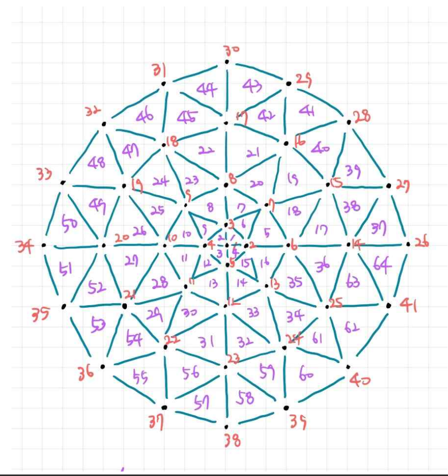
HW3

Seong-Min, Han (20221060)

Due: AM 08:00, March 15, 2022

1. Solve the Laplace equation for a circle.
At the top point, the solution is 1.
At the bottom point, the solution is 0.
For other boundary nodes, apply the homogenous Neumann boundary condition.

- 구상

circle 2	circle 4
	

각 점은 vertex, vertex를 element로 가지는 면을 face, vertex 사이의 거리를 len, 외심원의 반지름과 vertex 사이의 거리의 절반으로 구한 edge를 edge라고 정의하겠습니다.
원의 개수에 따라 mesh의 개수가 달라지고, 중요 vertex의 값이 달라지기 때문에 중요 vertex(top, bottom vertex 등)는 계차수열을 사용하여 일반항을 구했습니다.
구상과 결과에 대한 설명은 circle 2(원 2개)를 기준으로 하겠습니다.

circle 2의 경우, 반지름을 다르게 해서 2개의 원을 구성하고 각을 다르게 지정해서 내부와 boundary vertex를 정했습니다. 내부 원은 내부 vertex, 외부 원은 boundary vertex를 나타냅니다. 각 vertex는 right hand method로 face를 구성하는 element로 저장하였습니다. (ex) element(1) = [V(1) V(2) V(3)]로 구성

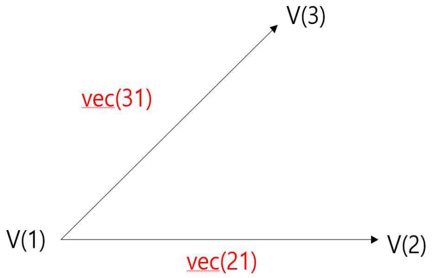
내부 원 : $(x,y) = (rcos\theta, rsin\theta)$, $r = 3$, $\theta = \frac{\pi}{2}$ 씩 증가

외부 원 : $(x,y) = (rcos\theta, rsin\theta)$, $r = 6$, $\theta = \frac{\pi}{4}$ 씩 증가

원의 개수를 numc라고 지정하면, 원이 늘어날 때마다 해당 원의 분할 각은 $\theta = \frac{\pi}{2*numc}$ 로 정의할 수 있습니다.
이 경우 top point인 $\phi_8 = 1$이 되고, $\phi_{12} = 0$ 이 됩니다. 위 좌측 그림과 같이 vertex와 element를 구성했습니다,

vertex의 개수는 원이 1개일 경우 5개, 2개일 경우 13개, 3개일 경우 25개, 4개일 경우 61개입니다. vertex의 개수를 numv라고 지정하면, vertex의 개수는 계차수열을 이용하여 $a_n = a_1 + \sum_{k=1}^{n-1} b_n = 5 + \sum_{k=1}^{n-1}(4n+4) = 2n^2 + 2n + 1$로 나타낼 수 있고 $numc$를 사용하면 $numv = 2(numc)^2 + 2numc + 1$로 나타낼 수 있습니다.
vertex의 index는 각 원의 vertex 시작점과 원 하나에 들어있는 vertex의 개수를 고려해서 구성했습니다.

- **vector, area, R**

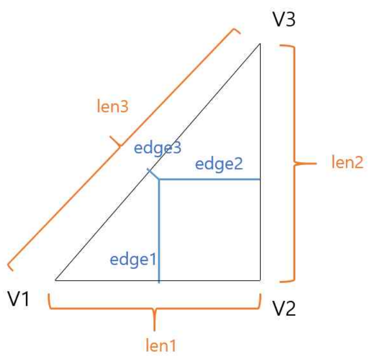


HW2와 동일하게 1 → 3으로 향하는 vec31, 2 → 1로 향하는 vec21을 만들고, 각 vector의 norm 값을 배열에 저장했습니다.

$area = \frac{1}{2}\sqrt{|\vec{a}|^2|\vec{b}|^2 - (\vec{a}\cdot\vec{b})^2}$ 를 이용하여 각 face의 area를 구하고

$R = \frac{L_1L_2L_3}{4 \times area}$ 를 이용하여 외접원의 반지름 구했습니다.

- **len, edge**



(좌측 그림 참고)
HW2와 동일하게 len(3,1), edge(3,1) matrix를 만들고

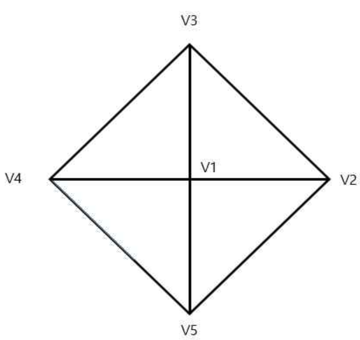
$edge = \sqrt{R^2 - (\frac{len}{2})^2}$ 를 계산하여 matrix에 값을 저장했습니다.

edge 1	edge 2	edge 3
V1 - V2	V2 - V3	V3 - V1

len 1	len 2	len 3
V1 - V2	V2 - V3	V3 - V1

- **A, b, phi**

이후 A, b, phi matrix를 만들어 potential 값을 계산했습니다.
A matrix의 경우 기준 vertex와 인접 vertex와의 관계를 고려해 구성했습니다. vertex 1을 기준으로 설명하겠습니다.



V1의 경우 인접노드와의 관계를 FVM으로 나타내면 다음과 같습니다,

$$\frac{edge1_{12}}{len_{12}}(\phi_2-\phi_1) + \frac{edge_{13}}{len_{13}}(\phi_3-\phi_1) + \frac{edge14}{len_{14}}(\phi_4-\phi_1) + \frac{edge_{15}}{len_{15}}(\phi_5-\phi_1)$$

해당 식을 1과 1의 인접 vertex로 나눌 경우

$\frac{edge1_{12}}{len_{12}}\phi_2 + \frac{edge_{13}}{len_{13}}\phi_3 + \frac{edge14}{len_{14}}\phi_4 + \frac{edge_{15}}{len_{15}}\phi_5$ 와

$-(\frac{edge1_{12}}{len_{12}} + \frac{edge_{13}}{len_{13}} + \frac{edge14}{len_{14}} + \frac{edge_{15}}{len_{15}})\phi_1$로 나눌 수 있습니다.

A matrix의 대각행렬 즉, 위 그림에서는 A(1,1)에서는 인접 vertex 2, 3, 4, 5의 $\frac{edge(1)}{len(1)}$ 성분의 합의 음수가 들어가게 되고, A(1,j) …… (j=2,3,4,5) 에는 각 $\frac{edge(j)}{len(j)}$ 성분이 들어가게 됩니다.
boundary vertex의 경우, homogenous Neumann boundary condition을 가정합니다.
Dirichlet boundary condition에 의해 가장 top vertex인 V(8)의 potential은 1이므로 A(8,:) = 0, A(8,8) = 1이 되고, 가장 bottom vertex인 V(12)의 potential은 0이므로 A(12,:) = 0, A(12,12) = 1이 됩니다.

b matrix의 경우, 우선 모든 행을 0으로 대입하고 Dirichlet boundary condition에 의해 top vertex인 V(8)을 고려해 b(8)에 1을 대입해주었습니다.

추가로 원의 개수가 바뀔 경우, top vertex와 bottom vertex가 바뀌기에 vertex의 값을 지정해주기 위해 계차수열을 통해 일반항을 구했습니다.

기준은 원점을 기준으로 가장 좌측 vertex로 지정하였으며, 해당 vertex는 $a_n = a_1 + \sum_{k=1}^{n-1} b_n = 4 + \sum_{k=1}^{n-1}(4n+2) = 2n^2 + 2$ 이고, 원의 개수인 numc로 나타내면 $vertex = 2(numc)^2 + 2$ 입니다.
따라서 top vertex와 bottom vertex를 일반항으로 정리하면 아래와 같습니다.

$vertex(top) = 2(numc)^2 - numc + 2$

$vertex(bottom) = 2(numc)^2 + numc + 2$

A matrix (cirlce 2)

변수 - A

A

13x13 double

	1	2	3	4	5	6	7	8	9	10	11	12	13
1	-2	0.5000	0.5000	0.5000	0.5000	0	0	0	0	0	0	0	0
2	0.5000	-1.9223	0	0	0.3204	0.6213	0.2735	0	0	0	0	0	0.2071
3	0.5000	0.3204	-1.9223	0	0	0	0.2071	0.6213	0.2735	0	0	0	0
4	0.5000	0	0.3204	-1.9223	0	0	0	0	0.2071	0.6213	0.2735	0	0
5	0.5000	0	0	0.3204	-1.9223	0	0	0	0	0	0.2071	0.6213	0.2735
6	0	0.6213	0	0	0	-0.7678	0.1464	0	0	0	0	0	0
7	0	0.2071	0.2735	0	0	0	-0.6270	0.1464	0	0	0	0	0
8	0	0	0	0	0	0	0	1	0	0	0	0	0
9	0	0	0.2071	0.2735	0	0	0	0	-0.6270	0.1464	0	0	0
10	0	0	0	0.6213	0	0	0	0	0	-0.7678	0.1464	0	0
11	0	0	0	0.2071	0.2735	0	0	0	0	0	-0.6270	0.1464	0
12	0	0	0	0	0	0	0	0	0	0	0	1	0
13	0	0.2735	0	0	0.2071	0.1464	0	0	0	0	0	0	-0.6270

b matrix

b

13x1 double

	1
1	0
2	0
3	0
4	0
5	0
6	0
7	0
8	1
9	0
10	0
11	0
12	0
13	0

phi matrix

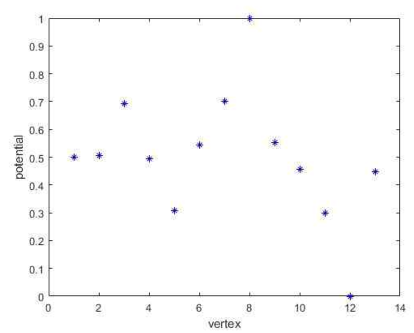
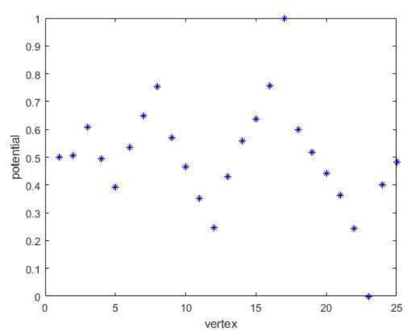
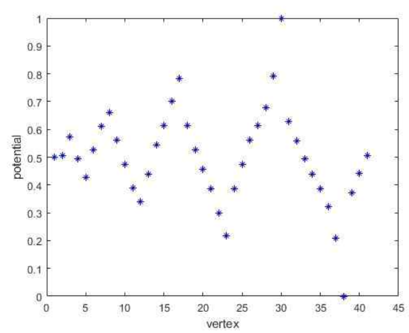
phi

13x1 double

	1
1	0.5000
2	0.5051
3	0.6915
4	0.4949
5	0.3085
6	0.5427
7	0.7020
8	1
9	0.5511
10	0.4573
11	0.2980
12	0
13	0.4489
14	

2번 행을 보면 vertex 2와 vertex 3은 연결되어있습니다. 하지만 해당 연결부분은 직각 삼각형의 빗변 부분이고, egde의 값이 0이기에 해당 위치의 값은 0인 것을 알 수 있습니다.

정의한 A matrix와 b matrix를 이용하여 phi matrix를 계산했습니다.
phi = A \ b를 이용하여 구했으며, 구한 potential의 값은 circle의 개수에 따라 다음과 같았습니다.

circle 2	circle 3	circle 4
		

- circle 2

원의 개수	vertex 개수	face 개수	top vertex	bottom vertex
2	13	16	8	12

- circle 3

원의 개수	vertex 개수	face 개수	top vertex	bottom vertex
3	25	36	17	23

- circle 4

원의 개수	vertex 개수	face 개수	top vertex	bottom vertex
4	41	64	30	38

* 마지막 장에 첨부한 일반항 식을 통해 계산가능

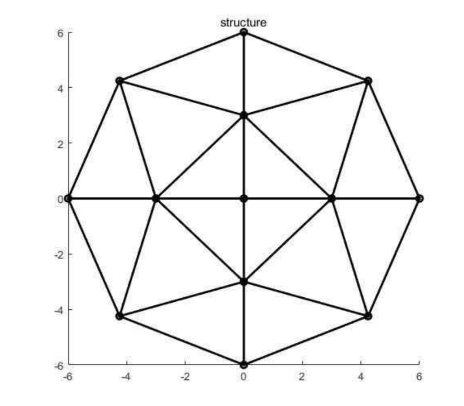
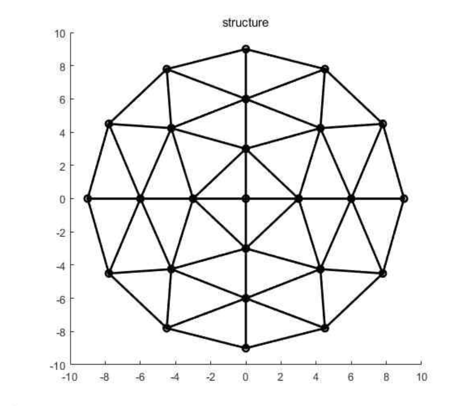
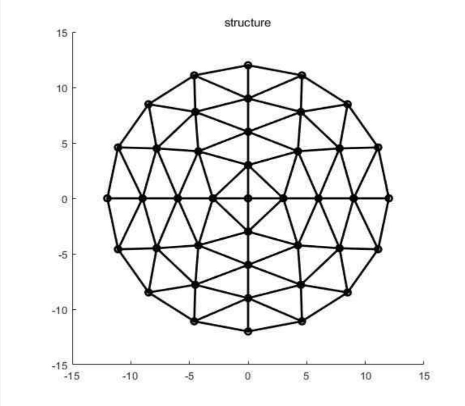
2. Read a structure specified by a *.vertex file and a *.element file.
Visualize the read structure.

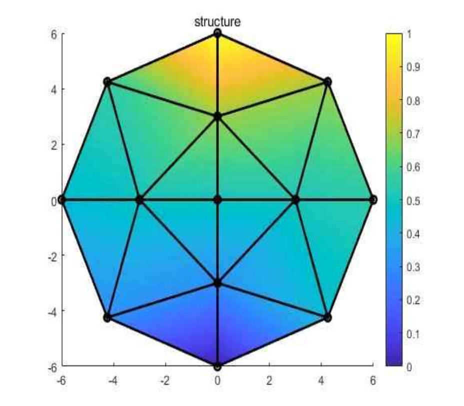
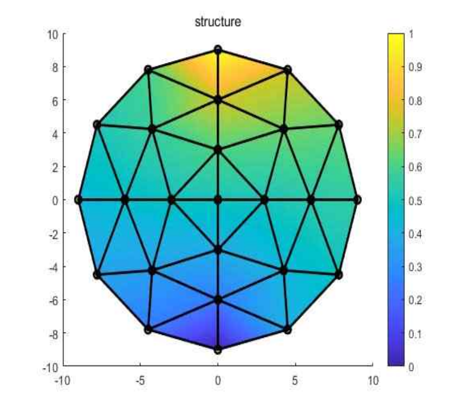
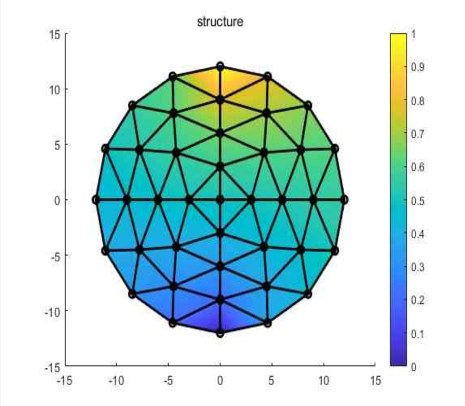
첨부한 HW3_vertexmaker.m를 통해 vertex.text를 만들고, element.text로 해당 vertex를 element로 가지는 face를 구성했습니다.
Hw3_vertexmaker에서 위에서 설명한 방식으로 각 원에서 vertex를 지정하고

```
writematrix(Vertex,'Vertex.txt');
type Vertex.txt
```

위 코드를 이용하여 vertex file로 저장했습니다.
element의 경우 저장한 vertex의 index를 각 vertex의 number로 설정하여 element의 각 행이 3개의 vertex로 구성되게 했습니다. 예를 들어 element(1)의 경우 [1 2 3]으로 구성했습니다.

Result

circle 2	circle 3	circle 4
		

circle 2	circle 3	circle 4
		

visualizing의 경우, MATLAB의 Patch를 이용하여 표현했습니다.

```
figure
patch('Faces',F,'Vertices',V,'FaceVertexCData',phi,'EdgeColor','black','FaceColor','interp','LineWidth',2,'Marker','o');
```

코드를 사용하여 각 element의 vertex를 이어 face를 구성하고, 이를 Visualize 했습니다.
아래 color가 있는 그림의 경우, 각 vertex의 potential 값을 고려한 그림입니다.

*** 일반항 식**

vertex 개수 : $2(numc)^2+2(numc)+1$

face 개수 : $(2\times numc)^2$

Top vertex : $2(numc)^2-numc+2$

bottom vertex : $2(numc)^2+numc+2$