

Homework #3

20221059 정상목

1. 문제

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.

2) Read a structure specified by a *.vertex file and a *.element file.

Visualize the read structure.

2. Mesh 구성

Mesh를 구하기 위해 어떠한 mesh를 생성할 것인지 확인한 후, vertex와 element를 만든 방법을 설명할 것이다.

2.1. 생성할 Mesh 확인

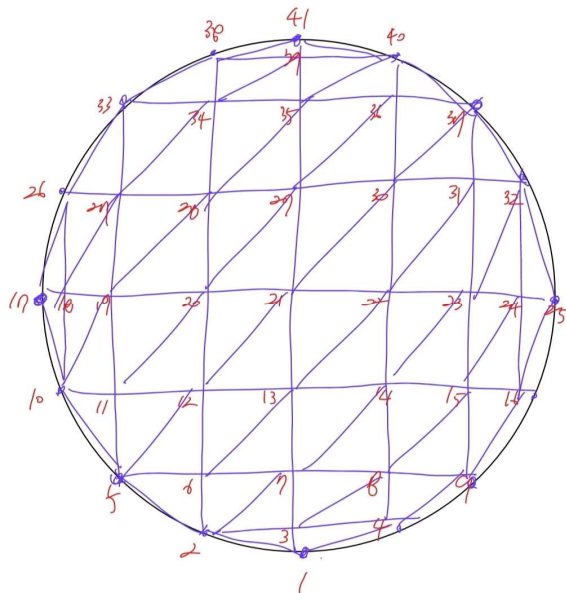


Fig 1. 생성할 Mesh.

Vertex 구성을 위해 반지름이 r 인 원을 N 개의 node로 쪼개줄 것이다. 원의 중심은 node의 x, y 좌표를 모두 양수로 만들기 위해 (r, r) 에 위치시켰다. 이 점들을 기준으로 좌표를 만들고 그중 원 내부에 있는 점만 사용하여 Vertex를 만들 예정이다.

2.2. Vertex 구성

1) 기준 원 이산화하기

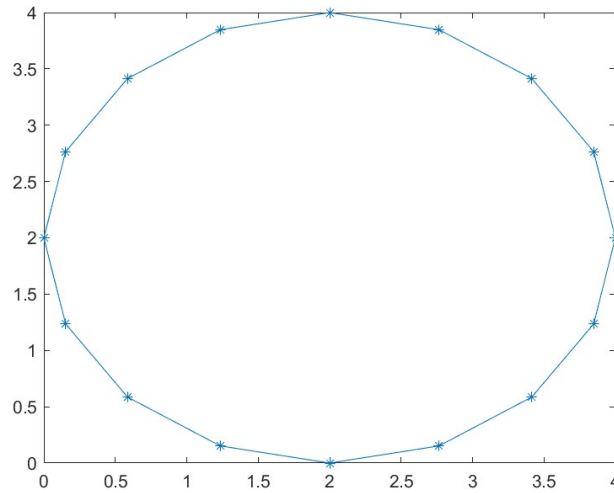


Fig 2. 기준 원 이산화.

반지름이 2이고 중점이 (2, 2)에 있는 원을 이용했다. 이 원을 16개의 점으로 나누어 주었다. 원의 상하좌우 지점에 vertex가 위치시키기 위해 N은 4의 배수인 16으로 설정했다.

2) 좌표 설정

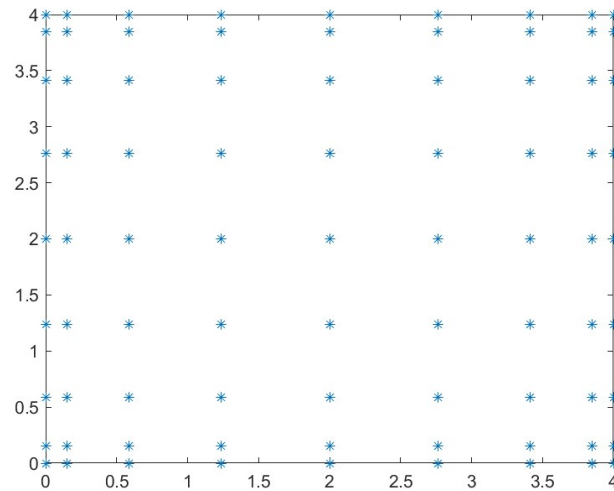


Fig 3. 좌표 설정.

앞서 원을 16개로 쪼개서 얻은 점들의 x, y 좌표를 기준으로 좌표를 설정해 주었다. 이렇게 좌표를 얻음으로써 앞으로 얻을 삼각형 mesh가 모두 직각삼각형의 형태를 가지고 있음을 예상할 수 있다.

3) 원 내부의 점인지 확인

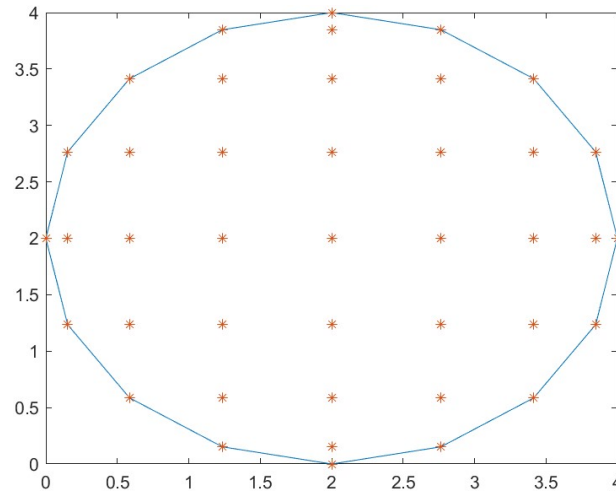


Fig 4. Vertex.

앞선 과정에서 얻은 좌표들이 원 내부에 있는지 확인한다. 원은 $(x-2)^2 + (y-2)^2 = 2^2$ 임을 이용하여 $(x-2)^2 + (y-2)^2 \leq 2^2$ 을 만족하는 점들만 선택하여 Vertex를 확정하였다.

원의 하단 좌측부터 순서대로 index를 매겨주었다. Matlab 내에서는 vertex를 저장한 행을 index로 활용했다.

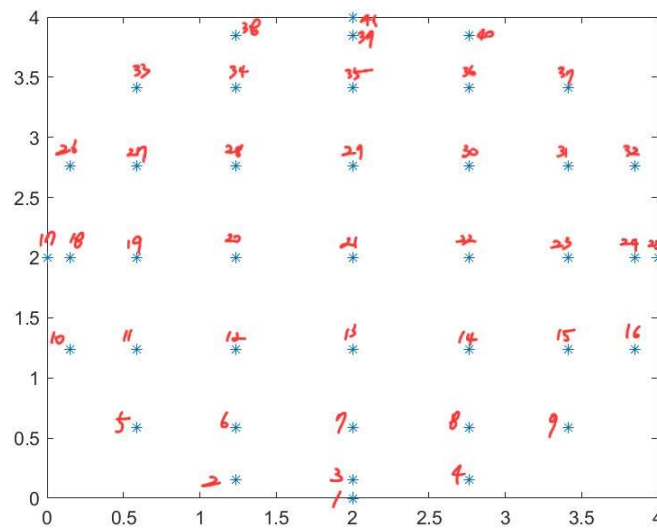


Fig 5. Index.

2.3. Element 구성

2.3.1. 원의 경계에 있는 vertex의 index 구하기.

Element를 자동으로 구성하기 위해 원의 경계에 있는 vertex의 index를 구할 필요가 있었다. 원의 경계에 있는 vertex는 다음과 같고 각 vertex에 매겨진 index의 차이가 그림과 같음을 확인할 수 있다.

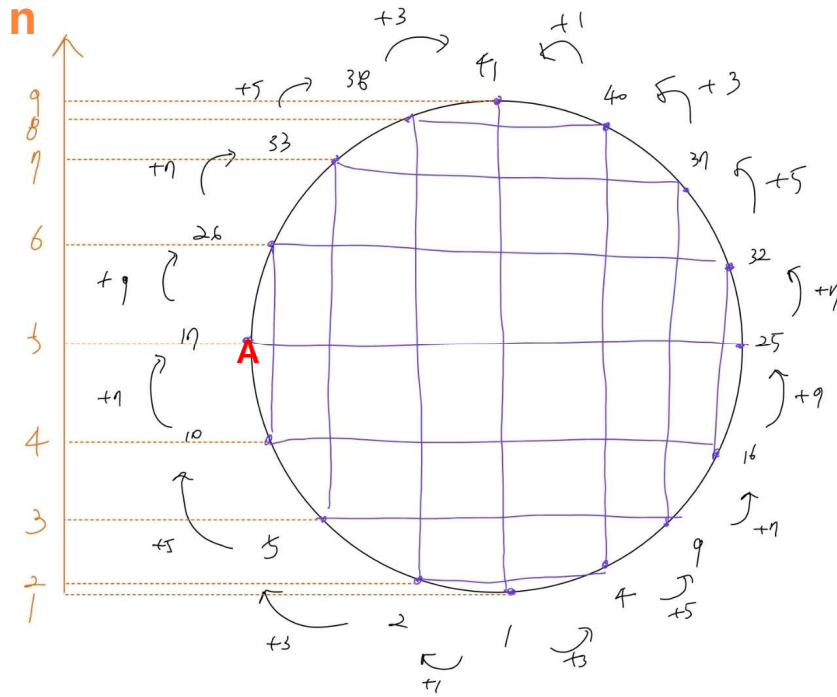


Fig 6. 원 경계에 있는 vertex의 index.

원의 경계에 있는 index는 계차수열의 관계가 있음을 확인하고 원을 왼쪽과 오른쪽으로 구분해 각각의 계차수열의 일반식을 구해 원의 경계에 있는 vertex의 index를 확인할 수 있도록 했다. 원의 왼쪽과 오른쪽으로 나누어 보더라도 원의 중간지점을 지나면서 증가하는 비율이 다시 달라지므로 이것 또한 범위를 나누어 일반식을 구했다.

또 위치를 쉽게 확인하기 위해 높이 index인 n 을 설정했다. n 의 최대 값은 $n = \frac{N}{2} + 1$ 을 이용해 쉽게 예측할 수 있다. 예를 들어 index 1과 index 41 중앙에는 index 17이 있는데 처음 N 을 설정할 때 4의 배수가 되도록 설정했으므로 index 17인 A지점은 다음과 같은 식으로 구할 수 있다.

$$A = \frac{N}{4} + 1$$

$N=16$ 인 경우 A 지점은 17임을 확인 할 수 있다. 이제 왼쪽과 오른쪽 index의 일반식을 구해보자.

1) 왼쪽 index의 일반식 ($n \leq 6$)

$a_n = a_1 + \sum_{k=1}^{n-1} b_k$ 식을 이용하자. $b_k = 2n - 1$ 이고 $a_1 = 1$ 이므로 $a_n = 1 + (n-1)^2$ 임을 확인할 수 있다.

2) 왼쪽 index의 일반식 ($n > 6$)

$b_1 = a_6 - a_5$ 이고 $b_k = -2n$ 이므로 $b_k = (a_6 - a_5) - 2n$ 임을 확인할 수 있다. 이 경우 $a_1 = a_6$ 이므로 $a_n = a_6 + (a_6 - a_5 - (n-6+1))(n-6)$ 이 나옴을 확인할 수 있었다.

3) 왼쪽 index의 일반식 ($n \leq 5$)

이 경우에는 계차수열이 아니라 등비수열임을 확인할 수 있다. 따라서 일반식은 $a_n = n^2$ 이다.

4) 왼쪽 index의 일반식 ($n > 5$)

$b_1 = a_5 - a_4$ 이고 $b_k = -2n$ 이므로 $b_k = (a_5 - a_4) - 2n$ 임을 확인할 수 있다. 이 경우 $a_1 = a_5$ 이므로 $a_n = a_6 + (a_6 - a_5 - (n - 5 + 1))(n - 5)$ 이 나옴을 확인할 수 있었다.

- 왼쪽 반원의 index

$$L_n = \begin{cases} 1 + (n-1)^2, & n \leq m+1 \\ L_{m+1} + (L_{m+1} - L_m - (n-m)(n-m-1)), & n > m+1 \end{cases}, \text{ 단 } m=N/4+1$$

- 오른쪽 반원의 index

$$R_n = \begin{cases} 1 + (n-1)^2, & n \leq m \\ R_m + (R_m - R_{m-1} - (n-m-1)(n-m-2)), & n > m \end{cases}, \text{ 단 } m=N/4+1$$

$N=16$ 인 경우 좌, 우측 반원의 인덱스가 예상값과 같게 나옴을 확인했다. 이 값은 ind_circle에 저장했다. 행은 높이 index n 에 해당하고 열은 1의 경우 원의 왼쪽, 2의 경우 원의 오른쪽 위에 있는 vertex의 index이다.

ind_circle		
9x2 double		
	1	2
1	1	1
2	2	4
3	5	9
4	10	16
5	17	25
6	26	32
7	33	37
8	38	40
9	41	41

Fig 6. 원 경계에 있는 vertex의 index (Matlab).

2.3.2. Element 구성하기.

Element를 구성하기 위해 앞서 구한 원 경계의 index를 활용했다.

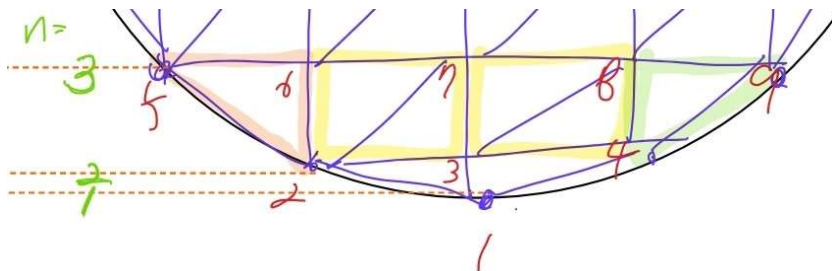


Fig 7. 아래쪽 원에서의 element 구성.

Fig 7을 보면 각 mesh의 형태가 좌측에 삼각형 1개, 중앙에 삼각형 2개로 이루어진 사각형, 우측에 삼각형 1개로 이루어져 있음을 확인할 수 있다. 이를 이용해서 element는 다음과 같은 순서를 통해 작성된다. 앞서 정의한 높이 index n 과 원 경계 위에 있는 vertex의 index인 ind_circle 을 사용한다.

- Matlab 내에서 Element가 작성되는 순서 (Fig 7을 예시로)

높이 index n 이 1 증가하면

1. 분홍색 삼각형인 좌측 삼각형에 해당하는 element [2 6 5] 을 작성해야 한다. $n=2$ 일 때 원 경계에 있는 vertex의 index는 $ind_circle(2,1) = 2$ 이다. index 6은 $ind_circle(2+1,1)+1 = ind_circle(3,1)+1 = 6$ 이고, index 5는 $ind_circle(2+1,1)=ind_circle(3,1)=5$ 로 좌측 삼각형의 element를 작성할 수 있었다.
2. $ind_circle(2,1)$ 의 값을 변수 p 에 저장한다. 이후 p 가 오른쪽 경계에 있는 vertex의 index와 동일하면 1번과 같은 방법으로 Fig 7상의 초록색 삼각형을 그린다.
3. p 가 오른쪽 경계에 있는 vertex의 index와 같지 않다면 노란색 네모 안에 있는 삼각형을 그리게 된다. 이때 필요한 index는 p 를 기준으로 p 바로 위, p 우측, p 대각 우측인데, 이 index는 원 경계의 index의 차이를 이용해 쉽게 유추할 수 있다.
4. 원의 절반을 넘어가게 되면 원 아래와 다르게 element를 구하는 순서가 약간 달라지게 되지만 1~3의 방법을 이용해 손쉽게 규칙을 작성할 수 있다.

위와 같은 방법으로 element를 작성할 수 있었다.

	1	2	3
1	1	3	2
2	1	4	3
3	2	6	5
4	2	7	6
5	2	3	7
6	3	8	7
7	3	4	8
8	4	9	8
9	5	11	10

Fig 8. Element (Matlab).

3. Matrix 구하기

Matrix를 구하기 위해서는 각 mesh를 구성하는 삼각형의 길이, 삼각형의 면적, 삼각형 외접원의 반지름, 외심에서 삼각형 각 변까지의 최단 거리가 필요하다. 앞서 구한 element를 이용해 계산할 수 있다.

3.1. Vector 구하기

Vector는 앞서 구한 element를 이용해서 구할 수 있었다. Element에 저장된 vertex의 index를 이용해 각 vector를 mesh마다 1->2, 2->3, 3->1 순서로 cell 기능을 이용해 저장했다.

3.2. 삼각형 길이/면적/외접의 반지름/외심까지의 거리 구하기

- 삼각형 길이

앞서 구한 vector를 이용했다. cell에 저장된 vector와 norm 명령어를 이용해 삼각형의 길이를 구할 수 있었다.

- 삼각형의 면적

삼각형의 면적은 $Area = \frac{1}{2} |\vec{v12} \times \vec{v13}|$ 을 이용해 계산했다. 앞서 vector를 구할 때 $\vec{v13}$ 이 아닌 $\vec{v31}$ 을 구했으므로 $Area = \frac{1}{2} |\vec{v12} \times (-\vec{v31})|$ 식과 norm 명령어를 이용해 삼각형의 면적을 계산한다.

- 외접의 반지름

외접의 반지름은 $R = \frac{L1 L2 L3}{4 \times Area}$ 을 이용해 계산했다.

- 외심까지의 거리

외심까지의 거리는 $A = \sqrt{R^2 - \left(\frac{L_i}{2}\right)^2}$ 을 이용하여 계산했다.

3.3. A , b matrix 구하기

각 mesh 마다 matrix를 만들고 이 matrix를 합치는 방법을 사용했다.

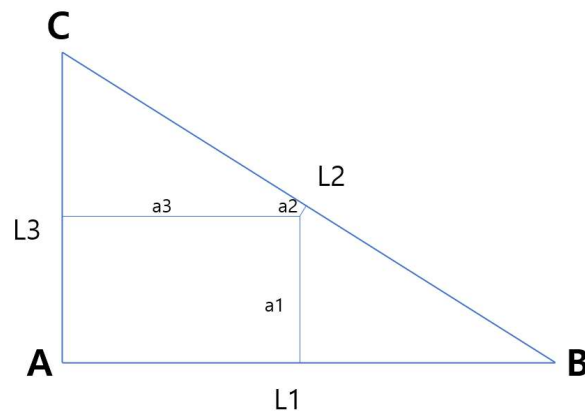


Fig 9. Mesh.

다음과 같은 mesh가 있을 때 각 vertex의 potential은 다음과 같다.

- vertex 1

$$\frac{\phi_B - \phi_A}{L_1} \frac{A_1}{2} + \frac{\phi_C - \phi_A}{L_3} \frac{A_3}{2} = - \left(\frac{A_1}{2L_1} + \frac{A_3}{2L_3} \right) \phi_1 + \frac{A_1}{2L_1} \phi_2 + \frac{A_3}{2L_3} \phi_3$$

- vertex 2

$$\frac{\phi_C - \phi_B}{L_2} \frac{A_2}{2} + \frac{\phi_A - \phi_B}{L_1} \frac{A_1}{2} = + \frac{A_1}{2L_1} \phi_1 - \left(\frac{A_1}{2L_1} + \frac{A_2}{2L_2} \right) \phi_2 + \frac{A_2}{2L_2} \phi_3$$

- vertex 3

$$\frac{\phi_A - \phi_C}{L_3} \frac{A_3}{2} + \frac{\phi_B - \phi_C}{L_2} \frac{A_2}{2} = \frac{A_3}{2L_3} \phi_1 + \frac{A_2}{2L_2} \phi_2 - \left(\frac{A_2}{2L_2} + \frac{A_3}{2L_3} \right) \phi_3$$

따라서 다음과 같은 행렬을 얻을 수 있다.

$$\begin{pmatrix} -\left(\frac{A_1}{2L_1} + \frac{A_3}{2L_3}\right) & \frac{A_1}{2L_1} & \frac{A_3}{2L_3} \\ \frac{A_1}{2L_1} & -\left(\frac{A_1}{2L_1} + \frac{A_2}{2L_2}\right) & \frac{A_2}{2L_2} \\ \frac{A_3}{2L_3} & \frac{A_2}{2L_2} & -\left(\frac{A_2}{2L_2} + \frac{A_3}{2L_3}\right) \end{pmatrix} \begin{pmatrix} \phi_A \\ \phi_B \\ \phi_C \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix}$$

다음과 같은 과정을 모든 mesh에서 수행하면 potential을 구할 수 있다. 단 dirichlet boundary condition인 부분에서는 그 값이 나오도록 마지막에 matrix를 수정해 주었다.

b matrix는 dirichlet boundary condition 조건인 index 1인 지점은 0, index 41인 지점은 1이 되도록 입력해 주었고 phi=Awb 를 통해 potential 값을 구할 수 있었다.

구해진 potential은 그림과 같다.

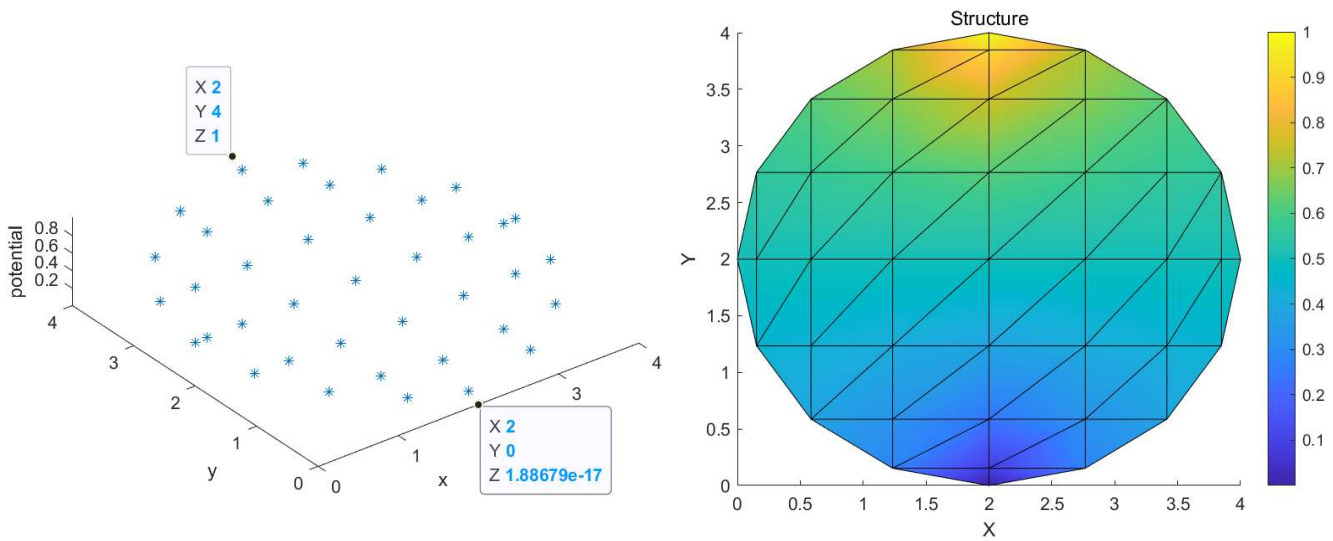


Fig 10. potential. (N=16).

그림을 살펴보면 index 1인 (2,0)은 potential이 0, index 41인 (2,4)는 potential이 1임을 확인 할 수 있다.

writematrix 명령어를 이용해 vertex와 element 정보를 Vertex.txt 와 Element.txt 파일로 저장했다. 이 파일들을 바탕으로 visualize 할 것이다.

4. Visualize

앞서 얻은 Vertex.txt 와 Element.txt 파일을 이용해 visualize 했다. Patch 명령어를 이용해 시각화 했다. 시각화 한 결과는 다음과 같다.

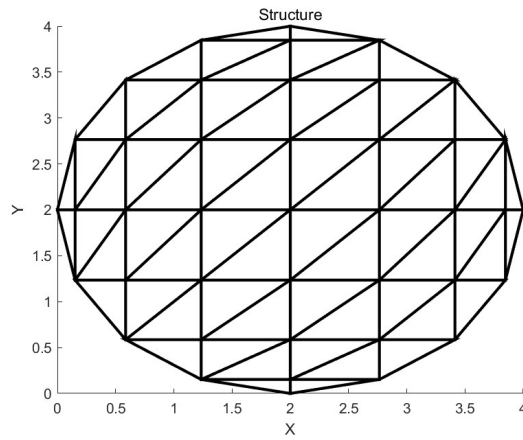


Fig 11. Visualize 1.

시각화가 잘 되었음을 확인할 수 있다.

추가적으로 N=20, 60, 100인 경우는 다음과 같이 나온다.

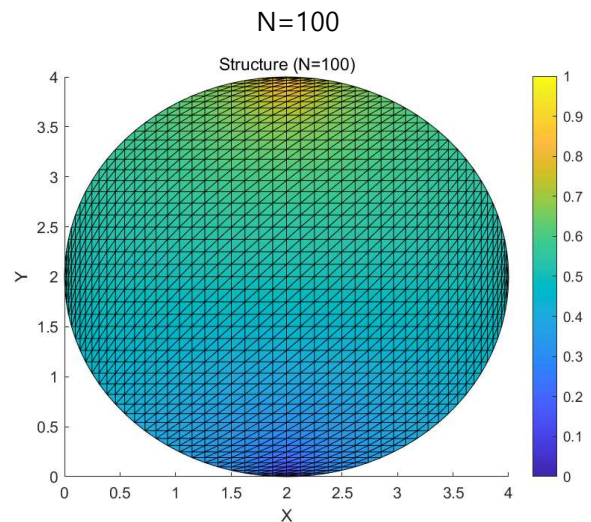
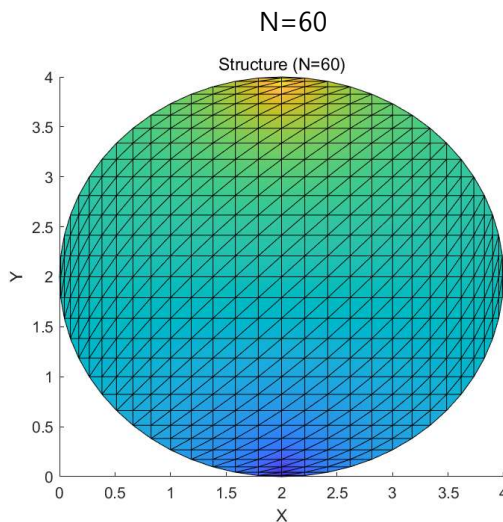
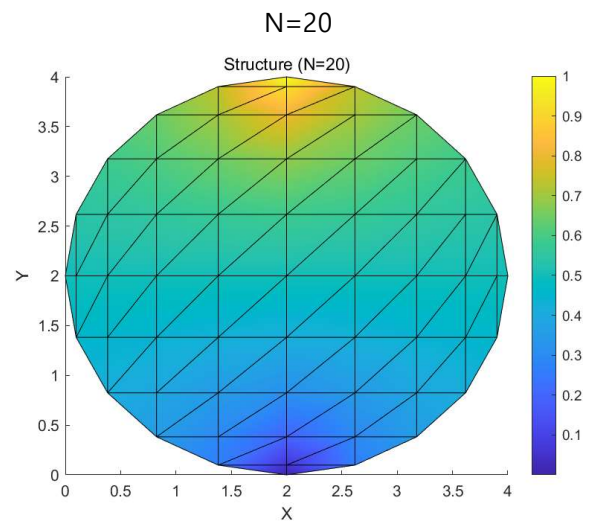
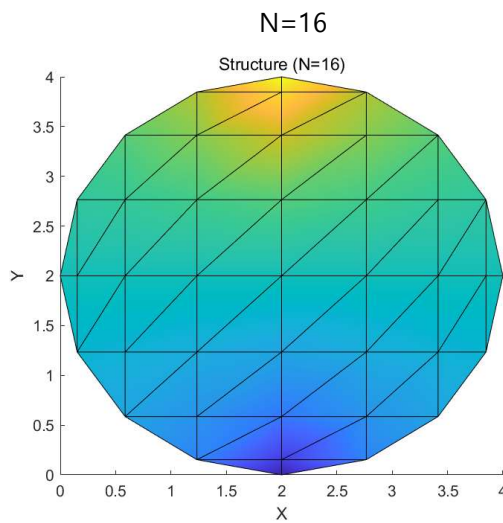


Fig 12. Visualize 2.