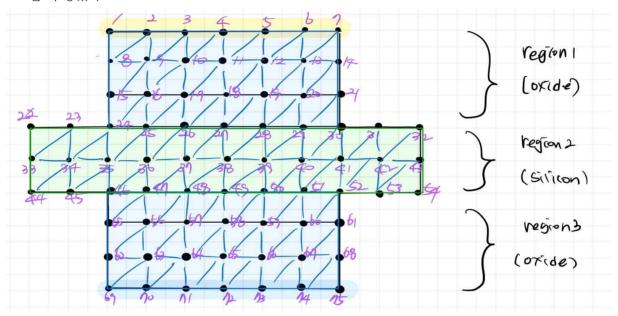
Design

이번 과제는 interface의 vertex를 region에 따라 고려하여 jacobian matrix를 재구성하는 것이었다. 기존의 jacobian matrix를 구성할 때는 region에 따라 vertex($\phi_V^{region1}$, $\phi_V^{region2}$)를 나누지 않았다. 하지만 이번 case에서는 각 region에 따라 vertex를 고려하여 preference에 따라 값을 다르게 대입하여 jacobian matrix를 구성했다.



구조의 경우 HW8에서 사용한 plus 형태의 구조를 동일하게 사용하였다. 후에 구조는 새로운 vertex와 element를 설정하여 다른 구조를 만들어볼 것이다.위 구조에서 interface는 2개가 존재하며 interface1은 vertex 24~30, interface2는 vertex 46~52에 해당한다.

| region | vertex 개수 | interface vertex 개수 |
|-------------|-----------|---------------------|
| 1 (silicon) | 28 | 7 |
| 2 (oxide) | 33 | |
| 3 (silicon) | 28 | 7 |
| | 89 | 14 |

potential을 구하기 위해 기존의 jacobian matrix는 각 행, 열의 번호와 vertex의 번호가 일치하는 matrix였다. 따라서 matrix size는 75X75였다. 하지만 이번에 재구성한 matrix의 경우 각 region을 순서 대로 stack 하여 matrix를 구성했다.

| フ | 존 | 재- | 구성 | |
|--------|--------------|---------|----------|--|
| | | index | vertex | |
| index | vertex | 1 ~ 28 | region 1 | |
| 1 ~ 75 | total region | 29 ~ 61 | region 2 | |
| | | 62 ~ 89 | region 3 | |

따라서 index를 re-indexing 할 필요가 존재했고, 위의 표처럼 각 region에 해당하는 vertex를 순서대로 stack하여 index를 구성했다. 기존의 matrix보다 interface vertex들을 한번 더 고려해주어야 했고, matrix size는 (75+14) X (75+14)로 89 X 89 matrix를 만들었다.

| region | 1 | | re | egion 2 | | re | gion 3 | |
|--------|-------|----------|----|---------|----------|----|--------|----------|
| V | ertex | re-index | | vertex | re-index | | vertex | re-index |
| | 1 | 1 | | 22 | 29 | | 46 | 62 |
| | 2 | 2 | | 23 | 30 | | 47 | 63 |
| | 3 | 3 | | 24 | 31 | | 48 | 64 |
| | 4 | 4 | | 25 | 32 | | 49 | 65 |
| | 5 | 5 | | 26 | 33 | | 50 | 66 |
| | 6 | 6 | | 27 | 34 | | 51 | 67 |
| | 7 | 7 | | 28 | 35 | | 52 | 68 |
| | 8 | 8 | | 29 | 36 | | 55 | 69 |
| | 9 | 9 | | 30 | 37 | | 56 | 70 |
| | 10 | 10 | | 31 | 38 | | 57 | 71 |
| | 11 | 11 | | 32 | 39 | | 58 | 72 |
| | 12 | 12 | | 33 | 40 | | 59 | 73 |
| | 13 | 13 | | 34 | 41 | | 60 | 74 |
| | 14 | 14 | | 35 | 42 | | 61 | 75 |
| | 15 | 15 | | 36 | 43 | | 62 | 76 |
| | 16 | 16 | | 37 | 44 | | 63 | 77 |
| | 17 | 17 | | 38 | 45 | | 64 | 78 |
| | 18 | 18 | | 39 | 46 | | 65 | 79 |
| | 19 | 19 | | 40 | 47 | | 66 | 80 |
| | 20 | 20 | | 41 | 48 | | 67 | 81 |
| | 21 | 21 | | 42 | 49 | | 68 | 82 |
| | 24 | 22 | | 43 | 50 | | 69 | 83 |
| | 25 | 23 | | 44 | 51 | | 70 | 84 |
| | 26 | 24 | | 45 | 52 | | 71 | 85 |
| | 27 | 25 | | 46 | 53 | | 72 | 86 |
| | 28 | 26 | | 47 | 54 | | 73 | 87 |
| | 29 | 27 | | 48 | 55 | | 74 | 88 |
| | 30 | 28 | | 49 | 56 | | 75 | 89 |
| | | | | 50 | 57 | | | |
| | | | | 51 | 58 | | | |
| | | | | 52 | 59 |] | | |
| | | | | 53 | 60 | 1 | | |
| | | | | 54 | 61 |] | | |

purple box : interface1 vertex (region1 - region2)
green box : interface2 vertex (region2 - region3)

위 interface를 보면 각 vertex는 2개의 state을 가지며, 각 state는 preference를 고려하여 preference한 경우에는 preference한 행과 non-preference 행의 jacobian matrix 값들을 몰아서 가지고, 아닌 경우에는 boundary condition을 나타내는 값을 가지도록 설정해야한다.

예를 들어 B.C : $\phi_V^{region1} - \phi_V^{region2} = 0$ 이므로 region2를 vertex V를 preference 한다고 했을 때, region1의 vertex V의 jacobian matrix 값은 region2의 vertex V가 몰아서 가지고, region1의 vertex V 행은 B.C을 나타낸다.

Matlab code

먼저, 전체 region에 대해 vertex를 stack 한 89X1 matrix VFT 를 만들었다. 다음으로 각 region에 대한 vertex를 re-indexing 한 결과 matrix인 (28 X 1 VF1_2nd, 33 X 1 VF2_2nd, 28 X 1 VF3_2nd)를 만들었다.

- 1) building separated row (Wrong Jacobian) find문을 이용하여, 각 region의 vertex와 re-indexing 한 matrix의 중복 vertex를 찾아 region 별로 element에 대해 loop를 실행하여 jacobian matrix를 구성했다.
- 2) merge rows by preference (sum of 2 fluxes) & impose B.C preference 하지 않은 region과 interface와의 중복 vertex를 찾고, jacobian matrix에서 해당 vertex 의 행의 값을 preference 한 region의 vertex 행에 더한다. 그리고 non-preference한 행을 모두 0으로 바꾸고 B.C에 해당하는 값 1, -1을 non-preference 한 행의 preference vertex와 non-preference vertex에 대입한다.

Result

- re-indexing matrix

| gior | n 1 | | region | 2 | | region | 3 | | | |
|----------|------------|-------|--------|------------|----------|---------------|-------------|----|--|--|
| <u> </u> | 8x2 double | | | 3x2 double | | === 28 | 28x2 double | | | |
| | 1 | 2 | | 1 | 2 | | 1 | 2 | | |
| 1 | 1 | 1 | 1 | 22 | 29 | 1 | 46 | 62 | | |
| 200 | 2 | 2 | 2 | 23 | 30 | 2 | 47 | 63 | | |
| 2 | 3 | 3 | 3 | 24 | 31 | 3 | 48 | 64 | | |
| 3 4 | 4 | 4 | 4 | 25 | 32 | 4 | 49 | 65 | | |
| 5 | | | 5 | 26 27 | 33 34 | 5 | 50 | 66 | | |
| | 5 | 5 | 7 | 28 | 35 | 1100 | | 67 | | |
| 6 | 6 | 6 | 8 | 29 | 36 | 6 | 51 | | | |
| 7 | 7 | 7 | 9 | 30 | 37 | 7 | 52 | 68 | | |
| 8 | 8 | 8 | 10 | 31 | 38 | 8 | 55 | 69 | | |
| 9 | 9 | 9 | 11 | 32 | 39 | 9 | 56 | 70 | | |
| 10 | 10 | 10 | 12 | 33 | 40 | 10 | 57 | 71 | | |
| 11 | 11 | 11 | 13 | 34 | 41 | 11 | 58 | 72 | | |
| 12 | 12 | 12 | 14 | 35 | 42 | 12 | 59 | 73 | | |
| 13 | 13 | 13 | 15 | 36 | 43 | 13 | 60 | 74 | | |
| 14 | 14 | 14 | 16 | 37 | 44 | 14 | 61 | 75 | | |
| 15 | 15 | 15 | 17 | 38 | 45 | 15 | 62 | 76 | | |
| 16 | 16 | 16 | 18 | 39 | 46 | 16 | 63 | 77 | | |
| DO-COC 1 | 2208 | 0.000 | 19 | 40 | 47 | 17 | 64 | 78 | | |
| 17 | 17 | 17 | 20 | 41 | 48 49 | 91.53 | | | | |
| 18 | 18 | 18 | 22 | 43 | 50 | 18 | 65 | 79 | | |
| 19 | 19 | 19 | 23 | 44 | 51 | 19 | 66 | 80 | | |
| 20 | 20 | 20 | 24 | 45 | 52 | 20 | 67 | 81 | | |
| 21 | 21 | 21 | 25 | 46 | 53 | 21 | 68 | 82 | | |
| 22 | 24 | 22 | 26 | 47 | 54 | 22 | 69 | 83 | | |
| 23 | 25 | 23 | 27 | 48 | 55 | 23 | 70 | 84 | | |
| 24 | 26 | 24 | 28 | 49 | 56 | 24 | 71 | 85 | | |
| 25 | 27 | 25 | 29 | 50 | 57 | 25 | 72 | 86 | | |
| 26 | 28 | 26 | 30 | 51 | 58 | 26 | 73 | 87 | | |
| 27 | 29 | 27 | 31 | 52 | 59 | 27 | 74 | 88 | | |
| 28 | 30 | 28 | 32 | 53 | 60 | 28 | 75 | 89 | | |
| 20 | 30 | 28 | 33 | 54 | 61 | 20 | /3 | 09 | | |

위 design part에서 계획한대로 re-indexing을 하기 위한 matrix를 만들고 각 matrix의 2열에 해당하는 값을 index로 하여 jacobian matrix를 구성했습니다.

- building separated row (Wrong Jacobian)
- : interface 1에 있는 24번 vertex를 예시로 들면 region1의 24번 vertex의 index는 22이고, region2의 24번 vertex의 index는 31이다. preference 한 vertex를 region 2의 vertex라고 가정한다.

| 77.7 | | 30 | | | | | | | | | | | - | -76 | | | | | | |
|------|---|----|---|--------|---------|---------|---------|---------|---------|--------|----------|----------|----------|----------|----------|----------|----------|---------|---------|---------|
| 22 | 0 | 0 | 0 | 5,8500 | 0 | 0 | 0 | 0 | 0 | 0 | -11,7000 | 5.8500 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 23 | 0 | 0 | 0 | 0 | 11.7000 | 0 | 0 | 0 | 0 | 0 | 5.8500 | -23,4000 | 5.8500 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 24 | 0 | 0 | 0 | 0 | 0 | 11.7000 | 0 | 0 | 0 | 0 | 0 | 5.8500 | -23.4000 | 5.8500 | 0 | 0 | 0 | 0 | 0 | 0 |
| 25 | 0 | 0 | 0 | 0 | 0 | 0 | 11.7000 | 0 | 0 | 0 | 0 | 0 | 5.8500 | -23.4000 | 5.8500 | 0 | 0 | 0 | 0 | 0 |
| 26 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 11.7000 | 0 | 0 | 0 | 0 | 0 | 5.8500 | -23.4000 | 5.8500 | 0 | 0 | 0 | 0 |
| 27 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 11.7000 | 0 | 0 | 0 | 0 | 0 | 5.8500 | -23.4000 | 5.8500 | 0 | 0 | 0 |
| 28 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 5.8500 | 0 | 0 | 0 | 0 | 0 | 5.8500 | -11.7000 | 0 | 0 | 0 |
| 29 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | -3.9000 | 1.9500 | 0 |
| 30 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1.9500 | -7.8000 | 1.9500 |
| 31 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1.9500 | -7.8000 |

2) merge rows by preference (sum of 2 fluxes) & impose B.C

| 22 | 0 | 0 | 0 | 0 | 0 | .0 | 0 | 0 | 0 | 0 | -1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | - 1 |
|----|---|---|---|--------|---|----|---|---|---|---|----------|--------|----|----|----|----|----|---------|---------|---------|
| 23 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | -1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 24 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | -1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 25 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | -1 | 0 | 0 | 0 | 0 | 0 | 0 |
| 26 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | -1 | 0 | 0 | 0 | 0 | 0 |
| 27 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | -1 | 0 | 0 | 0 | 0 |
| 28 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | -1 | 0 | 0 | 0 |
| 29 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | -3.9000 | 1.9500 | 0 |
| 30 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1.9500 | -7.8000 | 1.9500 |
| 31 | 0 | 0 | 0 | 5.8500 | 0 | 0 | 0 | 0 | 0 | 0 | -11.7000 | 5.8500 | 0 | 0 | 0 | 0 | 0 | 0 | 1.9500 | -7.8000 |

index 22의 행의 값들이 index 31의 행에 다 더해지고, index 22의 행은 jac(22,22)=-1, jac(22,31)=1이 대입된 것을 확인할 수 있다.

- potential result

| 1 | 1,0000 | 21 | 0.8043 | 41 | 0.5000 | 61 | 0.4735 | | |
|----|--------|----|--------|----|--------|----|--------|------|--------|
| 2 | 1.0000 | 22 | 0.6857 | 42 | 0.5000 | 62 | 0.3143 | | |
| 3 | 1,0000 | 23 | 0.7229 | 43 | 0.5000 | 63 | 0.2771 | | |
| 4 | 1 | 24 | 0.7354 | 44 | 0.5000 | 64 | 0.2646 | | |
| 5 | 1,0000 | 25 | 0.7385 | 45 | 0.5000 | 65 | 0.2615 | | |
| 6 | 1.0000 | 26 | 0.7354 | 46 | 0.5000 | 66 | 0.2646 | | |
| 7 | 1,0000 | 27 | 0.7229 | 47 | 0.5000 | 67 | 0.2771 | | |
| 8 | 0.9045 | 28 | 0.6857 | 48 | 0.5000 | 68 | 0.3143 | | |
| 9 | 0.9068 | 29 | 0.5265 | 49 | 0.5000 | 69 | 0.1957 | 1000 | |
| 10 | 0.9093 | 30 | 0.5531 | 50 | 0.5000 | 70 | 0.1865 | 81 | 0.0932 |
| 11 | 0.9102 | 31 | 0.6857 | 51 | 0.4735 | 71 | 0.1799 | 82 | 0.0955 |
| 12 | 0.9093 | 32 | 0.7229 | 52 | 0.4469 | 72 | 0.1778 | 83 | 0 |
| 13 | 0.9068 | 33 | 0.7354 | 53 | 0.3143 | 73 | 0.1799 | 84 | 0 |
| 14 | 0.9045 | 34 | 0.7385 | 54 | 0.2771 | 74 | 0.1865 | 85 | 0 |
| 15 | 0.8043 | 35 | 0.7354 | 55 | 0.2646 | 75 | 0.1957 | 86 | 0 |
| 16 | 0.8135 | 36 | 0.7229 | 56 | 0.2615 | 76 | 0.0955 | 87 | 0 |
| 17 | 0,8201 | 37 | 0.6857 | 57 | 0.2646 | 77 | 0.0932 | 88 | 0 |
| 18 | 0.8222 | 38 | 0.5531 | 58 | 0.2771 | 78 | 0.0907 | 89 | |
| 19 | 0.8201 | 39 | 0,5265 | 59 | 0.3143 | 79 | 0.0898 | 03 | 0 |
| 20 | 0.8135 | 40 | 0.5000 | 60 | 0.4469 | 80 | 0.0907 | | |

potential result를 보면 dirichlet boundary condition이 잘 적용된 것을 확인할 수 있으며, Vertex 24 의 값을 비교해보면 다음과 같다.

| 22 | 0.6857 | |
|----|--------|--|
| 23 | 0.7229 | |
| 24 | 0.7354 | |
| 25 | 0.7385 | index 22(region1 vertex 24)의 potential 값은 0.6857이고 |
| 26 | 0.7354 | |
| 7 | 0.7229 | index 31(region2 vertex 24)의 potential 값도 동일하게 0.6857인 것을 확인 |
| 3 | 0.6857 | 할 수 있다. |
| | 0.5265 | |
| | 0.5531 | |
| 31 | 0.6857 | |