ECSE 543 Assignment 2 Report

Q1

Q1

$$\begin{bmatrix} U_1 \\ U_2 \\ U_3 \end{bmatrix} = \begin{bmatrix} 1 & x_1 & y_1 \\ 1 & x_2 & y_2 \\ 1 & x_3 & y_3 \end{bmatrix} \begin{bmatrix} a \\ b \\ c \end{bmatrix}, \quad U = \sum_{i=1}^{3} U_i \alpha_i(x,y), \text{ where:}$$

$$\left.\begin{aligned} \alpha_1 &= \frac{1}{2A}\left[(x_2y_3 - x_3y_2) + (y_2 - y_3)x + (x_3 - x_2)y\right] \\ \alpha_2 &= \frac{1}{2A}\left[(x_3y_1 - x_1y_3) + (y_3 - y_1)x + (x_1 - x_3)y\right] \\ \alpha_3 &= \frac{1}{2A}\left[(x_1y_2 - x_2y_1) + (y_1 - y_2)x + (x_2 - x_1)y\right] \end{aligned}\right\} A = \text{Area of Triangle.}$$

Disjoint

$$S^{(e)} = \int_{\Delta e} \nabla\alpha_i \cdot \nabla\alpha_j \, ds = A(\nabla\alpha_i \cdot \nabla\alpha_j) \rightarrow S^{(e)} = \begin{bmatrix} S_{11} & S_{12} & S_{13} \\ S_{21} & S_{22} & S_{23} \\ S_{31} & S_{32} & S_{33} \end{bmatrix}$$

$$S_{11} = \frac{1}{4A}\left[(y_2 - y_3)^2 + (x_3 - x_2)^2\right]$$

$$S_{12} = \frac{1}{4A}\left[(y_2 - y_3)(y_3 - y_1) + (x_3 - x_2)(x_1 - x_3)\right]$$

$$S_{13} = \frac{1}{4A}\left[(y_2 - y_3)(y_1 - y_2) + (x_3 - x_2)(x_2 - x_1)\right]$$

$$S_{21} = S_{12}$$

$$S_{22} = \frac{1}{4A}\left[(y_3 - y_1)^2 + (x_1 - x_3)^2\right]$$

$$S_{23} = \frac{1}{4A}\left[(y_3 - y_1)(y_1 - y_2) + (x_1 - x_3)(x_2 - x_1)\right]$$

$$S_{31} = S_{13}$$

$$S_{32} = S_{23}$$

$$S_{33} = \frac{1}{4A}\left[(y_1 - y_2)^2 + (x_2 - x_1)^2\right]$$

For Triangle (1, 2, 3):

$(x_1, y_1) = (0.00, 0.02)$

$(x_2, y_2) = (0.00, 0.00)$

$(x_3, y_3) = (0.02, 0.00)$

For Triangle (4, 5, 6):

$(x_1, y_1) = (0.02, 0.02)$

$(x_2, y_2) = (0.00, 0.02)$

$(x_3, y_3) = (0.02, 0.00)$

$A = \frac{1}{2} \times (0.02)^2 = 0.0002$

$$S^{(1,2,3)} = \begin{bmatrix} \frac{1}{2} & -\frac{1}{2} & 0 \\ -\frac{1}{2} & 1 & -\frac{1}{2} \\ 0 & -\frac{1}{2} & \frac{1}{2} \end{bmatrix} \qquad S^{(4,5,6)} = \begin{bmatrix} 1 & -\frac{1}{2} & -\frac{1}{2} \\ -\frac{1}{2} & \frac{1}{2} & 0 \\ -\frac{1}{2} & 0 & \frac{1}{2} \end{bmatrix}$$

$$U_{dis} = \begin{bmatrix} U_1 \\ U_2 \\ U_3 \\ U_4 \\ U_5 \\ U_6 \end{bmatrix}, \quad U_{con} = \begin{bmatrix} U_1 \\ U_2 \\ U_3 \\ U_4 \end{bmatrix}, \quad U_{dis} = C U_{con}, \quad S_{con} = C^T S_{dis} C$$

For the given triangle configuration:

$$C = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix}, \quad S_{dis} = \begin{bmatrix} S^{(1,2,3)} & 0 \\ 0 & S^{(4,5,6)} \end{bmatrix} = \begin{bmatrix} \frac{1}{2} & -\frac{1}{2} & 0 & 0 & 0 & 0 \\ -\frac{1}{2} & 1 & -\frac{1}{2} & 0 & 0 & 0 \\ 0 & -\frac{1}{2} & \frac{1}{2} & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & -\frac{1}{2} & -\frac{1}{2} \\ 0 & 0 & 0 & -\frac{1}{2} & \frac{1}{2} & 0 \\ 0 & 0 & 0 & -\frac{1}{2} & 0 & \frac{1}{2} \end{bmatrix}$$

$$S_{cont} = C^T S_{dis} C = \begin{bmatrix} 1 & -\frac{1}{2} & 0 & -\frac{1}{2} \\ -\frac{1}{2} & 1 & -\frac{1}{2} & 0 \\ 0 & -\frac{1}{2} & 1 & -\frac{1}{2} \\ -\frac{1}{2} & 0 & -\frac{1}{2} & 1 \end{bmatrix}$$

Q2

a)

The two-element mesh configuration of the lower-left quarter of the co-axial cable is shown below in Figure 1. The nodes, including the boundary nodes, are labelled with numbers. The blue boundary has the potential of 0 volt, the red boundary has the potential of 110 volts, and the green boundaries are boundaries of symmetry. There are in total of 34 nodes and 46 triangles. Thus, the input data file has 34 entries for the node locations, and 46 entries for the triangle mesh configurations, as shown in Figure 2 and 3. Figure 4 shows the boundary conditions, where the nodes on the blue boundary (in Figure 1) are set to 0 volts, and the nodes on red boundaries (in Figure 1) are set to 110 volts.

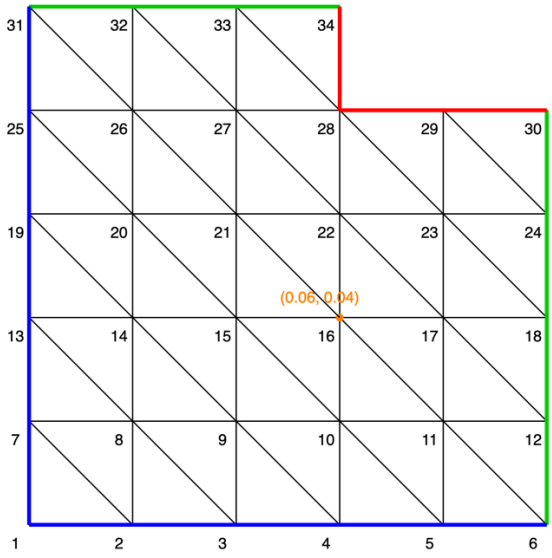

Figure 1 - Q1 Mesh Configuration

1	0.00	0.00
2	0.02	0.00
3	0.04	0.00
4	0.06	0.00
5	0.08	0.00
6	0.10	0.00
7	0.00	0.02
8	0.02	0.02
9	0.04	0.02
10	0.06	0.02
11	0.08	0.02
12	0.10	0.02
13	0.00	0.04
14	0.02	0.04
15	0.04	0.04
16	0.06	0.04
17	0.08	0.04
18	0.10	0.04
19	0.00	0.06
20	0.02	0.06
21	0.04	0.06
22	0.06	0.06
23	0.08	0.06
24	0.10	0.06
25	0.00	0.08
26	0.02	0.08
27	0.04	0.08
28	0.06	0.08
29	0.08	0.08
30	0.10	0.08
31	0.00	0.10
32	0.02	0.10
33	0.04	0.10
34	0.06	0.10

Figure 2 - Q2 Node Numbering

1	2	7	0.00
2	7	8	0.00
2	3	8	0.00
3	8	9	0.00
3	4	9	0.00
4	9	10	0.00
4	5	10	0.00
5	10	11	0.00
5	6	11	0.00
6	11	12	0.00
7	8	13	0.00
8	13	14	0.00
8	9	14	0.00
9	14	15	0.00
9	10	15	0.00
10	15	16	0.00
10	11	16	0.00
11	16	17	0.00
11	12	17	0.00
12	17	18	0.00
13	14	19	0.00
14	19	20	0.00
14	15	20	0.00
15	20	21	0.00
15	16	21	0.00
16	21	22	0.00
16	17	22	0.00
17	22	23	0.00
17	18	23	0.00
18	23	24	0.00
19	20	25	0.00
20	25	26	0.00
20	21	26	0.00
21	26	27	0.00
21	22	27	0.00
22	27	28	0.00
22	23	28	0.00
23	28	29	0.00
23	24	29	0.00
24	29	30	0.00
25	26	31	0.00
26	31	32	0.00
26	27	32	0.00
27	32	33	0.00
27	28	33	0.00
28	33	34	0.00

Figure 3 - Q2 Triangle Mesh

1	0.00
2	0.00
3	0.00
4	0.00
5	0.00
6	0.00
7	0.00
13	0.00
19	0.00
25	0.00
31	0.00
34	110.0
28	110.0
29	110.0
30	110.0

Figure 4 – Q2 Boundary Conditions

b)

The output of the **SIMPLE2D** program is shown in Figure 5. The potential at the location of **(x,y) = (0.06, 0.04)** equals to **40.5365 volts**.

1	0	0	0
2	0.0200	0	0
3	0.0400	0	0
4	0.0600	0	0
5	0.0800	0	0
6	0.1000	0	0
7	0	0.0200	0
8	0.0200	0.0200	7.0186
9	0.0400	0.0200	13.6519
10	0.0600	0.0200	19.1107
11	0.0800	0.0200	22.2643
12	0.1000	0.0200	23.2569
13	0	0.0400	0
14	0.0200	0.0400	14.4223
15	0.0400	0.0400	28.4785
16	0.0600	0.0400	40.5265
17	0.0800	0.0400	46.6897
18	0.1000	0.0400	48.4989
19	0	0.0600	0
20	0.0200	0.0600	22.1921
21	0.0400	0.0600	45.3132
22	0.0600	0.0600	67.8272
23	0.0800	0.0600	75.4690
24	0.1000	0.0600	77.3592
25	0	0.0800	0
26	0.0200	0.0800	29.0330
27	0.0400	0.0800	62.7550
28	0.0600	0.0800	110
29	0.0800	0.0800	110
30	0.1000	0.0800	110
31	0	0.1000	0
32	0.0200	0.1000	31.1849
33	0.0400	0.1000	66.6737
34	0.0600	0.1000	110

Figure 5 - Q2 Potential Results

c)

The total energy per unit length of the capacitor (W) can be calculated with the following equations:

$$W = \frac{1}{2} {U_{con}}^T \, S \, U_{con} \qquad (1)$$

The capacitance per unit length (C_{unit}) can then be calculated with the following equation:

$$W = \frac{1}{2} \, C_{unit} \, V^2 \qquad (2)$$

The values of U_{con} and S have already been calculated in the provided MATLAB code. The modified MATLAB code to extract these values is attached in the Appendix. The energy per unit length of the capacitor can then be calculated with Equation (1). However, since the S and U_{con} is only for one-quarter of the capacitor, the actual energy to be used in Equation (2) needs to be **four times** the calculated one with Equation (1). The voltage difference V is given, which equals to 110 volts. The capacitance per unit length (C_{unit}) is calculated to be 52.136 pF/m, as shown in Figure 6. The program for capacitance per unit length calculation is attached in the Appendix.

```
xiangyun@Xiangyuns-MacBook-Pro Assignment 2 % swift capacitance.swift
5.2136328078687805e-11
```

Figure 6 - Q2 Capacitance Result

Q3

a)

The Conjugate Gradient method is implemented, which is attached in the Appendix. The equation to be solved for Q3 has the form of $A\,x = b$. The node numbering for Q3 is shown in Figure 7, and the corresponding matrix A is shown in Figure 8. The idea for the matrix generation is similar to the Q3 in Assignment 1, where the potential at one node is related to the nearby nodes. The nodes on the blue and red boundaries, as shown in Figure 7, are not numbered, since these boundary conditions are not included in the matrix but are used to form the right-hand-side (RHS) vector b of the matrix equation, as shown in Figure 9. The values are negative as they are on the RHS of the equation. The formed matrix A shown in Figure 7 is clearly not symmetric positive definite (SPD), which is also confirmed with Cholesky Decomposition, as shown in Figure 10. To modify the equation, one rule can be utilized, where **for any n-by-n, non-singular matrix $\boldsymbol{M}$, $\boldsymbol{N = MM^T}$ is SPD**. The formed matrix A is non-singular, where all rows of A are different, so that A^TA would be SPD. Therefore, the matrix equation can be modified to $A^TA\,x = A^Tb$, and x would still be the original solution to the problem. The matrix modification code is shown in Figure 11.

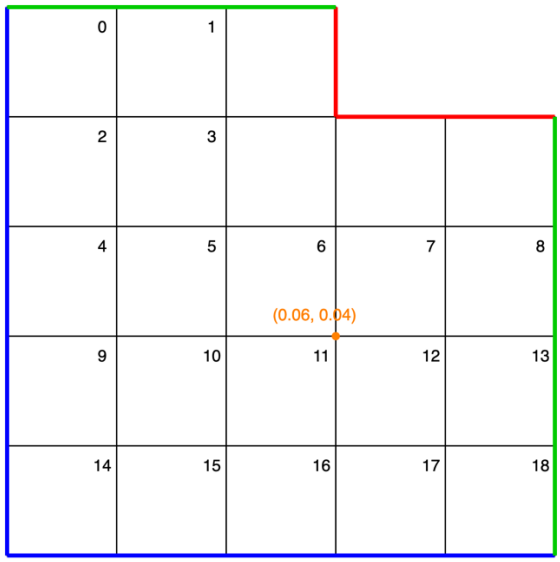

Figure 7 - Q3 Node Numbering

```
[[-4.0, 1.0, 2.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0],
 [1.0, -4.0, 0.0, 2.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0],
 [1.0, 0.0, -4.0, 1.0, 1.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0],
 [0.0, 1.0, 1.0, -4.0, 0.0, 1.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0],
 [0.0, 0.0, 1.0, 0.0, -4.0, 1.0, 0.0, 0.0, 0.0, 1.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0],
 [0.0, 0.0, 0.0, 1.0, 1.0, -4.0, 1.0, 0.0, 0.0, 0.0, 1.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0],
 [0.0, 0.0, 0.0, 0.0, 0.0, 1.0, -4.0, 1.0, 0.0, 0.0, 0.0, 1.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0],
 [0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 1.0, -4.0, 1.0, 0.0, 0.0, 0.0, 1.0, 0.0, 0.0, 0.0, 0.0, 0.0],
 [0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 2.0, -4.0, 0.0, 0.0, 0.0, 0.0, 1.0, 0.0, 0.0, 0.0, 0.0],
 [0.0, 0.0, 0.0, 0.0, 1.0, 0.0, 0.0, 0.0, 0.0, -4.0, 1.0, 0.0, 0.0, 0.0, 1.0, 0.0, 0.0, 0.0],
 [0.0, 0.0, 0.0, 0.0, 0.0, 1.0, 0.0, 0.0, 0.0, 1.0, -4.0, 1.0, 0.0, 0.0, 0.0, 1.0, 0.0, 0.0],
 [0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 1.0, 0.0, 0.0, 0.0, 1.0, -4.0, 1.0, 0.0, 0.0, 0.0, 1.0, 0.0],
 [0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 1.0, 0.0, 0.0, 0.0, 1.0, -4.0, 1.0, 0.0, 0.0, 0.0, 1.0],
 [0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 1.0, 0.0, 0.0, 0.0, 2.0, -4.0, 0.0, 0.0, 0.0, 0.0],
 [0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 1.0, 0.0, 0.0, 0.0, 0.0, -4.0, 1.0, 0.0, 0.0],
 [0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 1.0, 0.0, 0.0, 0.0, 1.0, -4.0, 1.0, 0.0],
 [0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 1.0, 0.0, 0.0, 0.0, 1.0, -4.0, 1.0],
 [0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 1.0, 0.0, 0.0, 0.0, 2.0, -4.0]]
```

Figure 8 - Q3 Matrix

```
[0.0, -110.0, 0.0, -110.0, 0.0, 0.0, -110.0, -110.0, -110.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0]
```

Figure 9 - Q3 RHS Vector

```
xiangyun@Xiangyuns-MBP Assignment 2 % swift A2P3.swift
Error: Matrix not SPD!!
```

Figure 10 - Q3 Cholesky Failed for Non-SPD Matrix

```
var cholesky_ans: [Double]? = choleskySolver_Optimized(A: dotproduct_matrix(A: transpose(A: grid), B: grid),
    b: dotproduct_vector(A: transpose(A: grid), b: b))
```

Figure 11 - Matrix Modification for SPD

b)

The solutions to the matrix equations using Cholesky Decomposition and Conjugate Gradient are shown in Figure 12, where the sequence of the potential at each node follows the node numbering in Figure 7.

```
xiangyun@Xiangyuns-MacBook-Pro Assignment 2 % swift A2P3.swift
Cholesky Decomposition:
[31.184935941368693, 66.67372442302754, 29.033009671223596, 62.754980875370705, 22.1921218681549, 45.31318940723157, 67.82717752884213,
 75.46901809691111, 77.35922364524426, 14.42228839416434, 28.478477356558376, 40.52650261122577, 46.68967121355802, 48.498858387154804,
 7.018554351944022, 13.651929013611722, 19.110684345944467, 22.264305758940345, 23.25686747625887]
Conjugate Gradient:
[31.184935941438166, 66.67372442295772, 29.033009671125196, 62.7549808754696, 22.192121868225577, 45.313189407158546, 67.82717752885274
, 75.46901809694691, 77.35922364520964, 14.422288394120862, 28.478477356598926, 40.52650261123335, 46.68967121349636, 48.49885838720863
, 7.018554351962819, 13.651929013597345, 19.11068434593021, 22.26430575898954, 23.256867476217728]
```

Figure 12 - Cholesky Decomposition & Conjugate Gradient Results

c)

The infinity norm and the 2-norm of the residual vector versus the number of iterations for the conjugate gradient program is shown in Figure 13. The norms of the residual vector gradually decrease, indicating that the solution gradually converges as expected. The implementation of the infinity norm and the 2-norm are attached in the Appendix.

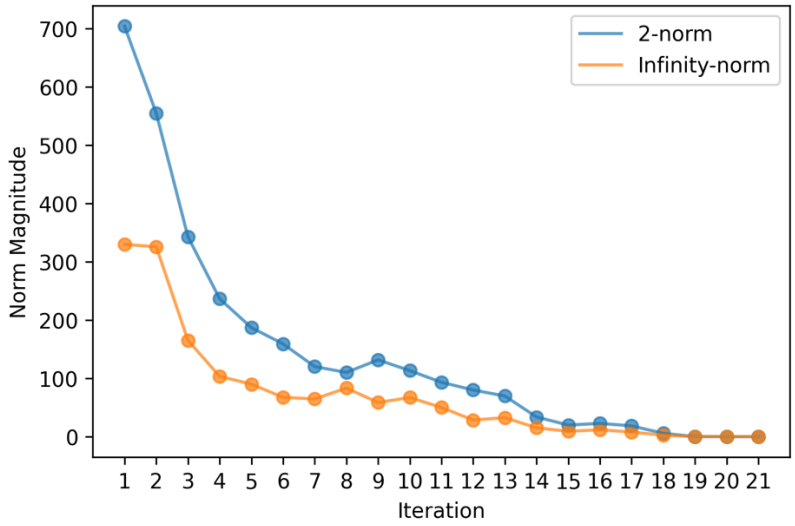

Figure 13 - Infinity Norm & 2-Norm versus Number of Iterations

d)

Table 1 shows the potentials at (x, y) = (0.06, 0.04), calculated with SOR (h=0.02) in Assignment 1, Finite Element in Q2, Cholesky Decomposition in Q3 and Conjugate Gradient in Q3. The potential results at (0.06, 0.04) for all the methods are **extremely close** to each other.

Table 1 - Potential at (0.06, 0.04) with Different Methods

Method / Location	A1 – SOR (volts)	A2Q2 (volts)	A2Q3 – Cholesky (volts)	A2Q3 – CG (volts)
(0.06, 0.04)	40.526502245388045	40.526502611225640	40.52650261122577	40.52650261123335

e)

As the potential at each node is calculated with CG method, the same strategy for capacitance per unit length calculation can be used as the **part c)** of Q2. The U_{con} can be formed with the calculated potentials and the boundary conditions, and the S matrix needs to be adjusted according to the node numbering as shown in Figure 7.

Appendix

MATLAB File Modification

(Extra output **Sglob** is added for the capacitance calculation)

```
function [Potential,Sglob] = SIMPLE2D_M(filename1,filename2,filename3)
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% This code is solely developed to solve the ECSE 543 assignments.
% Send your questions and feedbacks to ali.akbarzadehsharbaf@mail.mcgill.ca
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

Capacitance Calculation

```
218
219  let epsilon: Double = 8.854e-12
220  let total_energy: Double = 0.5 * dotproduct_matrix(A: dotproduct_matrix(A: transpose(A: potential),
221      B: S), B: potential)[0][0];
222  let V: Double = 110.0
223  let C: Double = 2.0*total_energy*epsilon/(V*V) * 4
224  print(C);
```

Conjugate Gradient Program

```
226  func CG_solve(A:[[Double]],b:[Double], threshold: Double) -> ([Double],[Double],[Double]){
227      var x: [Double] = Array(repeating: 0.0, count: b.count)
228      var r: [Double] = subtract(A: b, B: dotproduct_vector(A: A, b:x))
229      var p: [Double] = r
230      var two_norm_array: [Double] = []
231      var infinity_norm_array: [Double] = []
232      var infinity_error: Double = infinity_norm(A: r)
233      var error: Double = two_norm(A: r)
234      two_norm_array.append(error)
235      infinity_norm_array.append(infinity_error)
236      var alpha: Double = 0.0
237      var beta: Double = 0.0
238      var pt: [[Double]] = [p]
239      while error > threshold {
240          pt = [p]
241          alpha = dotproduct_vector(A: pt, b: r)[0] / dotproduct_vector(A: dotproduct_matrix(A: pt, B: A), b: p)[0]
242          x = addition(A: multiplication_vector(a: p, c: alpha), B: x)
243          r = subtract(A: b, B: dotproduct_vector(A: A, b:x))
244          beta = (-1) * dotproduct_vector(A: dotproduct_matrix(A: pt, B: A), b: r)[0] /
245              dotproduct_vector(A: dotproduct_matrix(A: pt, B: A), b: p)[0]
246          p = addition(A: multiplication_vector(a: p, c: beta), B: r)
247          error = two_norm(A: r)
248          infinity_error = infinity_norm(A: r)
249          two_norm_array.append(error)
250          infinity_norm_array.append(infinity_error)
251
252      }
253      return (x,two_norm_array,infinity_norm_array)
254  }
```

Infinity Norm & 2-Norm

```
256  func two_norm(A: [Double]) -> Double{
257      var norm: Double = 0.0
258      for element: Double in A{
259          norm = norm + element*element
260      }
261      return sqrt(norm)
262  }
263
264  func infinity_norm(A: [Double]) -> Double{
265      var max: Double = 0.0
266      for element: Double in A {
267          if max < element * element{
268              max = element * element
269          }
270      }
271      return sqrt(max)
272  }
```

Calculated Potentials for all Nodes

Method / Location	A2Q2	A2Q3 - Cholesky	A2Q3 - CG
(0.02, 0.02)	7.018554351943976	7.018554351944022	7.018554351962819
(0.02, 0.04)	14.422288394164253	14.42228839416434	14.422288394120862
(0.02, 0.06)	22.192121868154793	22.1921218681549	22.192121868225577
(0.02, 0.08)	29.033009671223510	29.033009671223596	29.033009671125196
(0.02, 0.1)	31.184935941368620	31.184935941368693	31.184935941438166
(0.04, 0.02)	13.651929013611650	13.651929013611722	13.651929013597345
(0.04, 0.04)	28.478477356558248	28.478477356558376	28.478477356598926
(0.04, 0.06)	45.313189407231420	45.31318940723157	45.313189407158546
(0.04, 0.08)	62.754980875370606	62.754980875370705	62.7549808754696
(0.04, 0.1)	66.673724423027450	66.67372442302754	66.67372442295772
(0.06, 0.02)	19.110684345944392	19.110684345944467	19.11068434593021
(0.06, 0.04)	40.526502611225640	40.52650261122577	40.52650261123335
(0.06, 0.06)	67.827177528842010	67.82717752884213	67.82717752885274
(0.08, 0.02)	22.264305758940290	22.264305758940345	22.26430575898954
(0.08, 0.04)	46.689671213557915	46.68967121355802	46.68967121349636
(0.08, 0.06)	75.469018096911030	75.46901809691111	75.46901809694691
(0.1, 0.02)	23.256867476258822	23.25686747625887	23.256867476217728
(0.1, 0.04)	48.498858387154710	48.498858387154804	48.49885838720863
(0.1, 0.06)	77.359223645244190	77.35922364524426	77.35922364520964