תרגיל מחשב 1:

אלגברה לינארית חישובית

ביה"ס להנדסת חשמל

מבוא לשיטות חישוביות – 361.1.2251

סמסטר ב' תשפ"ב

מגיש: אסף קמבר (313390429)
kamberasaf@post.bgu.ac.il

רקע: המטריצה $\mathbf{A}$ מתארת את מדידות פוטנציאל אלקטרוסטטי מאוסף מקורות נקודתיים. במערכת ישנם M מטענים שנסמנם $q_n, n \in [1,2,...,M]$ והפוטנציאל $v_m, m \in [1,2,...,M]$ נמדד ב- M נקודות. מטרתנו לבחון שיטות לחישוב המטענים מתוך המדידות. לשם כך מנסחים את הבעיה כמערכת משוואות לינאריות בהצגה מטריצית $\mathbf{Aq} = \mathbf{v}$ כך ש:

$$\underbrace{\begin{pmatrix} a_{11} & a_{12} & \cdots & a_{1M} \\ a_{21} & a_{22} & & a_{2M} \\ \vdots & & \ddots & \vdots \\ a_{M1} & a_{M2} & \cdots & a_{MM} \end{pmatrix}}_{\mathbf{A}} \underbrace{\begin{pmatrix} q_1 \\ q_2 \\ \vdots \\ q_M \end{pmatrix}}_{\mathbf{q}} = \underbrace{\begin{pmatrix} v_1 \\ v_2 \\ \vdots \\ v_M \end{pmatrix}}_{\mathbf{v}}$$

עבור המקרה בו כל המטענים וכל נקודות המדידה מסודרים במרווח זוויתי אחיד, לאורך שתי קשתות מעגליות מקבילות שרדיוסן $\rho = 1\text{m}$ ושמרכזיהן נמצאים במרחק h כלשהו זה מזה, ניתן לכתוב את אברי המטריצה במפורש:

$$a_{mn} = \frac{1}{4\pi r_{mn}} = \frac{1}{4\pi\sqrt{\left[h + \rho\sin(m\pi/M) - \rho\sin(n\pi/M)\right]^2 + \left[\rho\cos(m\pi/M) - \rho\cos(n\pi/M)\right]^2}}$$

***הערה לגבי מבנה הקבצים של הקוד –** עקב שימוש חוזר בפונקציות (למשל בניית מטריצה A) הגדרתי קובץ נפרד לפונקציות בשם "Functions.m" שבו מוגדר class שמכיל פונקציות חוזרות במהלך הקוד, שאר הקבצים קוראים במהלך הקוד לפונקציות המופיעות שם.

נגדיר מספר משתנים שישמרו לאורך כל התוכנית:

```
%% variables
M = 18;
length = M;
Rho = 1;
q = [3,1,3,3,9,0,4,2,9,3,1,3,3,9,0,4,2,9]'; % my i
```


<u>מעבר מהיר (ניתן להשתמש עם תוכנה מתאימה)</u>

שאלה 1 - סעיף ב', סעיף ג' , סעיף ד', סעיף ה'.

שאלה 2 – סעיף ב', סעיף ג' , סעיף ד'.

שאלה 3 – סעיף ב'.

קוד מאטלב

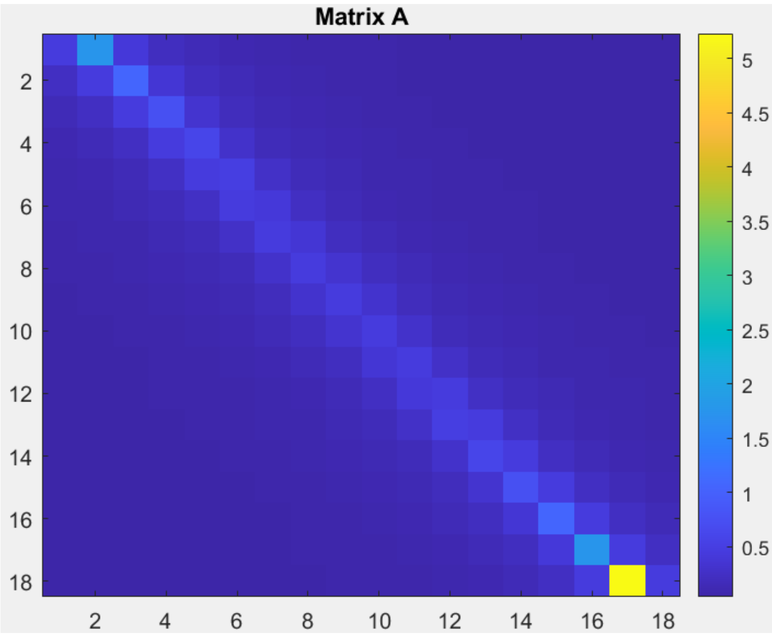
שאלה 1: אלימיניציה גאוסית ופירוק LU

א. בסעיף זה התבקשנו לבצע מספר דברים:

- **Assume** $h = \frac{\pi \cdot \rho}{M} \to h = \frac{\pi \cdot 1}{18} = \frac{\pi}{18}$.
- **Task:** Built matrix A.
  Process: I have written the following MATLAB code to build the matrix:

```
classdef Functions
    methods
        function res = Matrix_A(~,M,h)
            % create matrix A with given M (size) and h (changes values).
            res=zeros(M);
            Rho=1;
            for m=1:M
                for n=1:M
                    rad = sqrt((h+Rho*sin((m*pi)/M)-Rho*sin((n*pi)/M)).^2+ ...
                        (Rho*cos((m*pi)/M)-Rho*cos((n*pi)/M)).^2);
                    res(m,n)=1 ./ (4*pi*rad);
                end
            end
        end
```

Result:



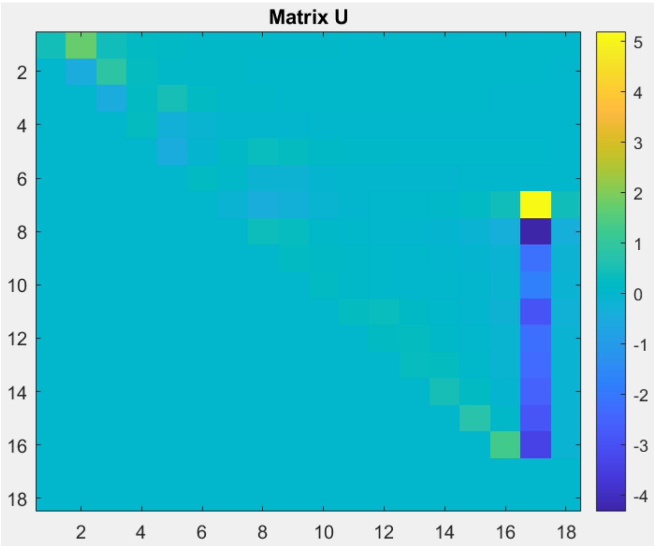
- **Task:** Compute $\bar{v} = Aq$. (MATLAB Code: v=A*q)
  Result: (transpose variable display):

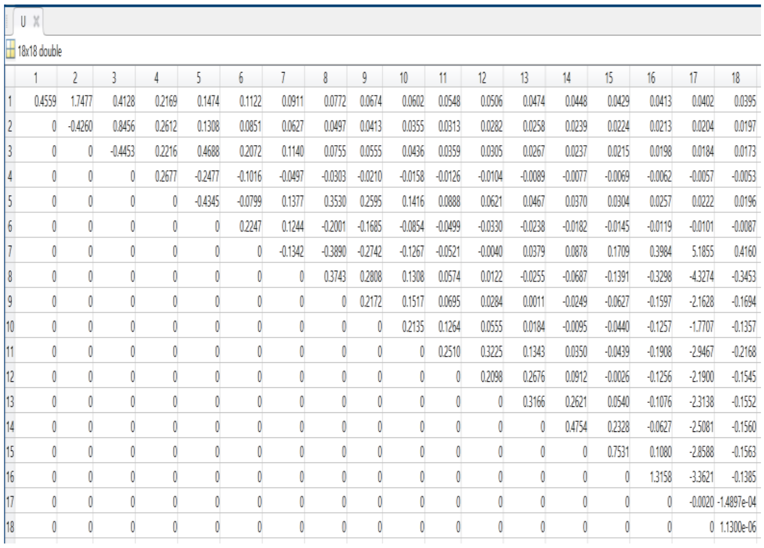
v
1x18 double

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
1	8.9900	10.1489	10.5069	11.6248	10.1541	9.2314	9.4575	9.8640	10.5759	9.8502	9.0555	9.1267	9.8979	11.3892	13.2335	10.3408	14.8078	20.3266

- **Task:** Calculate LU decomposition (with Pivoting).
  Process: MATLAB has built in function “lu()” to calculate matrices L,U and P $\boxed{\texttt{[L,U,P] = lu(A)}}$
  Results:

Matrix U:





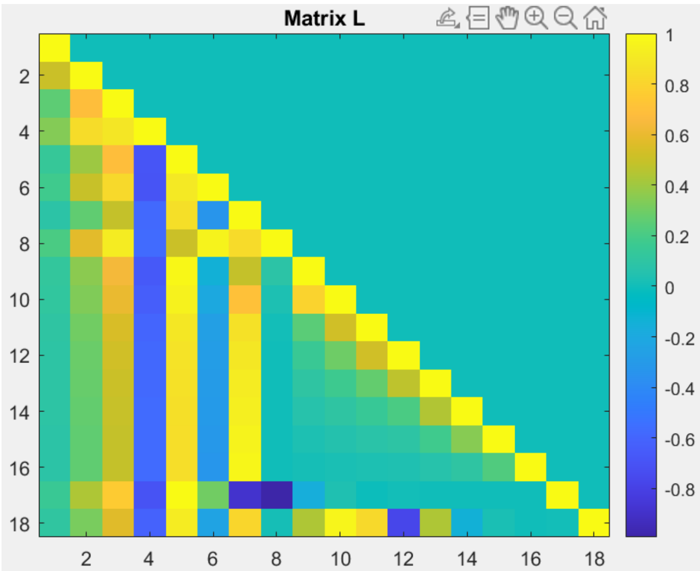
U

18x18 double

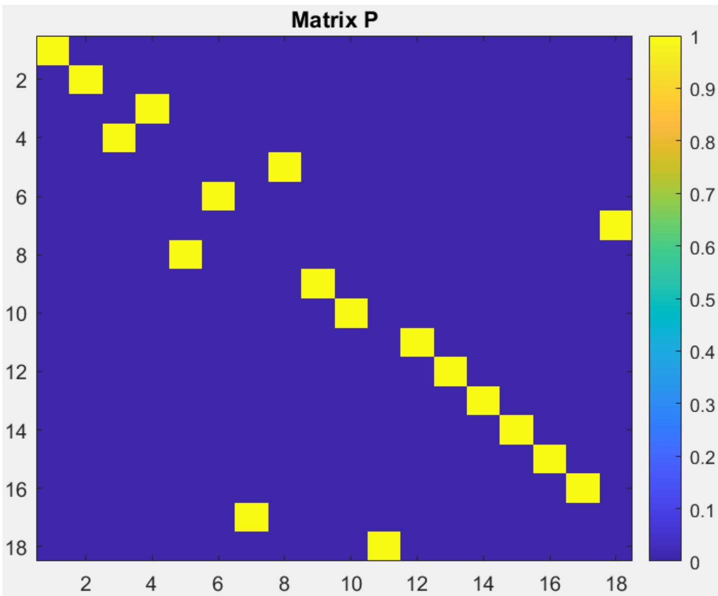
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
1	0.4559	1.7477	0.4128	0.2169	0.1474	0.1122	0.0911	0.0772	0.0674	0.0602	0.0548	0.0506	0.0474	0.0448	0.0429	0.0413	0.0402	0.0395
2	0	-0.4260	0.8456	0.2612	0.1308	0.0851	0.0627	0.0497	0.0413	0.0355	0.0313	0.0282	0.0258	0.0239	0.0224	0.0213	0.0204	0.0197
3	0	0	-0.4453	0.2216	0.4688	0.2072	0.1140	0.0755	0.0555	0.0436	0.0359	0.0305	0.0267	0.0237	0.0215	0.0198	0.0184	0.0173
4	0	0	0	0.2677	-0.2477	-0.1016	-0.0497	-0.0303	-0.0210	-0.0158	-0.0126	-0.0104	-0.0089	-0.0077	-0.0069	-0.0062	-0.0057	-0.0053
5	0	0	0	0	-0.4345	-0.0799	0.1377	0.3530	0.2595	0.1416	0.0888	0.0621	0.0467	0.0370	0.0304	0.0257	0.0222	0.0196
6	0	0	0	0	0	0.2247	0.1244	-0.2001	-0.1685	-0.0854	-0.0499	-0.0330	-0.0238	-0.0182	-0.0145	-0.0119	-0.0101	-0.0087
7	0	0	0	0	0	0	-0.1342	-0.3890	-0.2742	-0.1267	-0.0521	-0.0040	0.0379	0.0878	0.1709	0.3984	5.1855	0.4160
8	0	0	0	0	0	0	0	0.3743	0.2808	0.1308	0.0574	0.0122	-0.0255	-0.0687	-0.1391	-0.3298	-4.3274	-0.3453
9	0	0	0	0	0	0	0	0	0.2172	0.1517	0.0695	0.0284	0.0011	-0.0249	-0.0627	-0.1597	-2.1628	-0.1694
10	0	0	0	0	0	0	0	0	0	0.2135	0.1264	0.0555	0.0184	-0.0095	-0.0440	-0.1257	-1.7707	-0.1357
11	0	0	0	0	0	0	0	0	0	0	0.2510	0.3225	0.1343	0.0350	-0.0439	-0.1908	-2.9467	-0.2168
12	0	0	0	0	0	0	0	0	0	0	0	0.2098	0.2676	0.0912	-0.0026	-0.1256	-2.1900	-0.1545
13	0	0	0	0	0	0	0	0	0	0	0	0	0.3166	0.2621	0.0540	-0.1076	-2.3138	-0.1552
14	0	0	0	0	0	0	0	0	0	0	0	0	0	0.4754	0.2328	-0.0627	-2.5081	-0.1560
15	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0.7531	0.1080	-2.8588	-0.1563
16	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1.3158	-3.3621	-0.1385
17	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	-0.0020	-1.4897e-04
18	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1.1300e-06

Note: I've added the actual numerical presentation of matrix U to show that it is an upper triangular matrix.

Matrix L:



Matrix P:



- **Task:** Calculate condition number K(A).
  Process: MATLAB built in function "cond()" (code: K=cond(A)).
  Result: $K(A) = 9669241.27639153 \cong 9.6692 \cdot 10^6$.

- **Task:** Calculate the following norms: $||A||_F$, $||\bar{v}||_2$, $||\bar{q}||_2$.
  Process: MATLAB built in function "norm()".
  Code:

```
q_norm = norm(q);            % 2-norm of q (p=2).
v_norm = norm(v);            % 2-norm of V.
A_fb_norm = norm(A,"fro"); % Frobenius norm of matrix A.
```

  Results: $||A||_F = 6.912$, $||\bar{v}||_2 = 48.1744$, $||\bar{q}||_2 = 20.4939$.

ב. התבקשנו לממש שתי שגרות לפתרון מערכת מהצורה $\boldsymbol{Ly = b}$ ומהצורה $\boldsymbol{Ux = y}$, ללא קריאה לפקודות חישוב הופכי או פתרון ישיר למערכת. ניעזר בהצבה קדמית ואחורית שלמדנו בהרצאות:

Ly=b:

$$\overbrace{\begin{pmatrix} l_{1,1} & 0 & \dots & 0 \\ l_{2,1} & l_{2,2} & & \vdots \\ \vdots & \vdots & & o \\ l_{n,1} & l_{n,2} & \dots & l_{n,n} \end{pmatrix}}^{L} \cdot \overbrace{\begin{pmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{pmatrix}}^{y} = \overbrace{\begin{pmatrix} b_1 \\ \vdots \\ b_n \end{pmatrix}}^{b}$$

$$l_{1,1}y_1 = b_1 \qquad y_1 = \frac{b_1}{l_{1,1}}$$

$$l_{2,1}y_1 + l_{2,2}y_2 = b_2 \qquad y_2 = \frac{b_2 - l_{2,1}y_1}{l_{2,2}}$$

$$\vdots \qquad \vdots$$

$$l_{n,1}y_1 + \dots + l_{n,n}y_n = b_n$$

$$\boxed{y_k = \frac{b_k - \sum_{i=1}^{k-1} l_{k,i}y_i}{l_{k,k}} \quad \begin{matrix} 1 \le k \le 18 \\ k \in \mathbb{N} \end{matrix}}$$

כפי שניתן לראות בתמונה, לפתרון המערכת השתמשתי באלגוריתם של הצבה קדמית, קוד מאטלב למימוש האלגוריתם:

```
function y = Ly_b(L,b) % Solving Triangular System Ly=b.
    M = length(b);
    y = zeros(1,M);      % result vector
    y(1) = b(1)/L(1,1);
    sum = 0;
    for k = 2:M
        sum = b(k);
        for i = 1:k-1
            sum = sum - L(k,i)*y(i);
        end
        y(k)=sum/L(k,k);
        sum=0;
    end
end
```

ובאופן דומה עבור פתרון המערכת **Ux=y** השתמשתי בהצבה אחורית, כנלמד בהרצאה:

$$\begin{pmatrix} u_{1,1} & u_{1,2} & \cdots & u_{1,n} \\ 0 & u_{22} & & \\ \vdots & & \ddots & u_{n-1,n} \\ 0 & \cdots & 0 & u_{n,n} \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{pmatrix} = \begin{pmatrix} y_1 \\ \vdots \\ y_n \end{pmatrix}$$

$$\begin{cases} u_{n,n}x_n = y_n \\ u_{n-1,n-1}x_{n-1} + u_{n-1,n}x_n = y_{n-1} \\ \vdots \\ u_{1,1}x_1 + \dots + u_{1,n}x_n = y_1 \end{cases} \Rightarrow \begin{cases} x_n = \dfrac{y_n}{u_{n,n}} \\ \vdots \\ \boxed{x_k = \dfrac{y_k - \sum_{i=1}^{k-1} u_{i,n-1}y_{n-1}}{u_{k,l}}} \end{cases}$$

קוד מאטלב מתאים:

```
function x = Ux_y(U,y) % Solving Triangular System Ux=y.
    M = length(U);
    x = zeros(1,M);    % result vector
    x(M) = y(M)/U(M,M);
    sum = 0;
    for r = M-1:-1:1
        sum = y(r);
        for i = M:-1:r
            sum = sum-U(r,i)*x(i);
        end
        x(r) = sum/U(r,r);
        sum = 0;
    end
    end
```

- כעת התבקשנו לממש שגרות אלה לחישוב פתרון $\bar{q}$ למשוואה $A\bar{q} = \bar{v}$.

תוצאה עבור $\bar{q}$: (טרנספוז)

q_b

1x18 double

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
1	3.0000	1.0000	3.0000	3.0000	9.0000	-2.4845e-11	4.0000	2.0000	9.0000	3.0000	1.0000	3.0000	3.0000	9.0000	-1.5864e-10	4.0000	2.0000	9.0000

אם נשווה לq המקורי:

q = [3,1,3,3,9,0,4,2,9,3,1,3,3,9,0,4,2,9]

נוכל לראות כי קיימת **שגיאה** הנובעת מה**ייצוג הסופי** של מספר הספרות במחשב, למשל את המספר אפס (הספרה שישית), ניתן לראות כי הספרה לא חושבה כאפס אלא מספר בסדר גודל של 10^{-11}, כלומר מאוד קרובה לאפס אבל לא שווה ממש לאפס.

ניתן לראות זאת בנוסף בתצוגה המפורטת עם ייצוג של יותר ספרות:

q_b

1x18 double

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
1	2.9999999...	1.0000000...	2.9999999...	2.999999...	8.9999999...	-2.484476...	3.999999999971...	1.9999999...	8.9999999...	2.9999999...	0.9999999...	2.9999999...	2.999999...	8.9999999...	-1.586384...	3.9999999...	1.9999999...	9.000000

- **חישוב שגיאה יחסית** – על פי ההגדרה שראינו בהרצאה רשמתי פונקציה בקובץ נפרד שתממש את חישוב השגיאה היחסית (בנוסף לפונקציה לחישוב שגיאה אבסולוטית), החישוב שהתקבל:

$$\delta_{\hat{x}} = \frac{||x - \hat{x}||}{|x|} \rightarrow \delta_{\hat{q}} = \frac{||q - \bar{q}||_2}{||q||_2} = 1.310 * 10^{-10}$$

MATHLAB code (in “Functions.m” file):

```
function res = abs_error(~,e,e_t,p) % get absolute error.
    res = norm(e-e_t,p);
end

function [res,per] = rel_error(~,e,e_t,p) % get relative error.
    res = norm(e-e_t,p)/norm(e,p); % relative error.
    per = 100 * res; % percent error.
end
```

ג. כעת נוסף וקטור שגיאת מדידה δv שאבריו: $\delta v_m = 10^{-3} \cdot ||\bar{v}||_2$.
נרצה לפתור שוב בשיטת פירוק LU של הצבה קדמית ואחורית את המשוואה $A\bar{q} = \bar{v} + \delta v$.

תוצאה עבור $\bar{q}$ (טרנספוז):

q_c
1x18 double

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
1	3.0354	1.0033	3.0115	3.0149	9.0168	0.0182	4.0191	2.0195	9.0196	3.0192	1.0183	3.0170	3.0150	9.0122	0.0076	4.0008	1.9952	9.1227

תוצאה עבור השגיאה היחסית $\delta_{\bar{q}}$:

$$\delta_{\hat{q}} \cong 0.0069$$

בהשוואה לשגיאה היחסית קיבלנו בסעיף ב' $(1.310 * 10^{-10})$, ניתן להסביר את השינוי באי-שוויון שראינו בהרצאה:

$$\frac{1}{K(A)} \frac{||\Delta b||}{||b||} \leq \frac{||\Delta x||}{||x||} \leq K(A) \frac{||\Delta b||}{||b||}$$

מכיוון שנוספה לנו שגיאה בפתרון (כלומר $\delta_v = \delta_b$) נוכל לשים לב שבמקרה הזה השגיאה היחסית בפתרון קטנה מ-1 ולכן **החסם התחתון גדל** , כך שנקבל בוודאות שגיאה יחסית גדולה מהסעיף הקודם (שהייתה בסדר גודל של 10^{-10}).

$$\frac{1}{k(A)} \cdot \delta_{\hat{v}} \leq \delta_{\hat{q}} \rightarrow (9.6692 \cdot 10^6)^{-1}(0.9959) \cong 1.03 \cdot 10^{-7} \leq \delta_{\hat{q}}$$

ד. כעת נפלה שגיאה במטריצה ולכל איבר שלה נוספה שגיאה: $\delta A_{mn} = 10^{-3} \cdot ||A||_F$.

נרצה לפתור שוב בשיטת פירוק LU של הצבה קדמית ואחורית את המשוואה $(A + \delta A)\bar{q} = \bar{v}$.

תוצאה עבור $\bar{q}$:

q_d
1x18 double

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
2.6718	0.9698	2.8934	2.8623	8.8438	-0.1686	3.8232	1.8190	8.8186	2.8222	0.8300	2.8424	2.8607	8.8873	-0.0704	3.9928	2.0449	7.8629

והשגיאה היחסית:

$$\delta_{\hat{q}} \cong 0.0637$$

אם שוב נשווה לשגיאה היחסית מסעיף ב' $(1.310 * 10^{-10})$, ניתן להסביר את השינוי שוב בעזרת אי השוויון שראינו בהרצאה ובהגדרת מספר המצב:

$$\frac{1}{K(A)} \frac{||\Delta b||}{||b||} \leq \frac{||\Delta x||}{||x||} \leq K(A) \frac{||\Delta b||}{||b||}$$

$$K(A) = ||A^{-1}|| \cdot ||A||$$

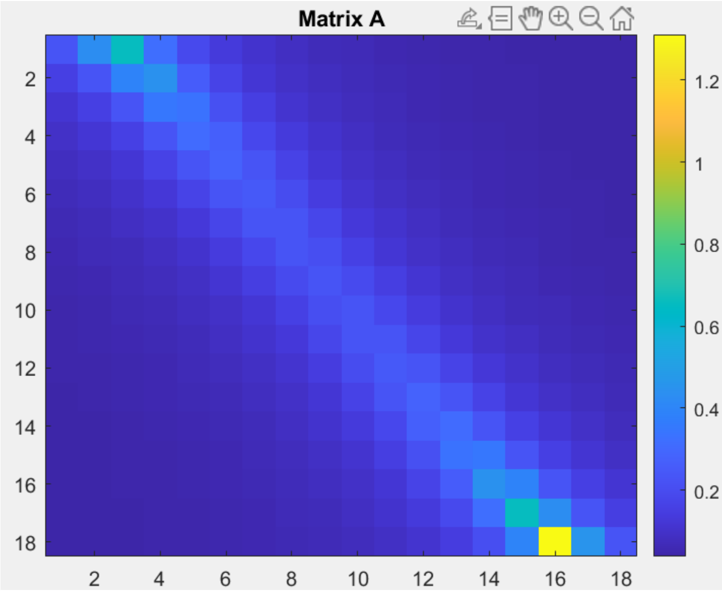
כזכור מספר המצב שקיבלנו ללא השגיאה היה: $K(A) \cong 9.6692 \cdot 10^6$
ובמקרה של שגיאה באיברי המטריצה נקבל : $K(A) \cong 9.1034 \cdot 10^{18}$.
כלומר מספר המצב גדל, לכן אי השוויון מתרחב לשני הצדדים, כלומר החסם התחתון קטן **והעליון גדל.**

ה. נחזור על הסעיפים א'-ד' ע"י לולאת *for* על ערכי h משתנים:

$h \in \{1,2,5,10,20,50\} \times \pi\rho/M$

(2) $h = \frac{2\pi\rho}{M}$:

Matrix A:



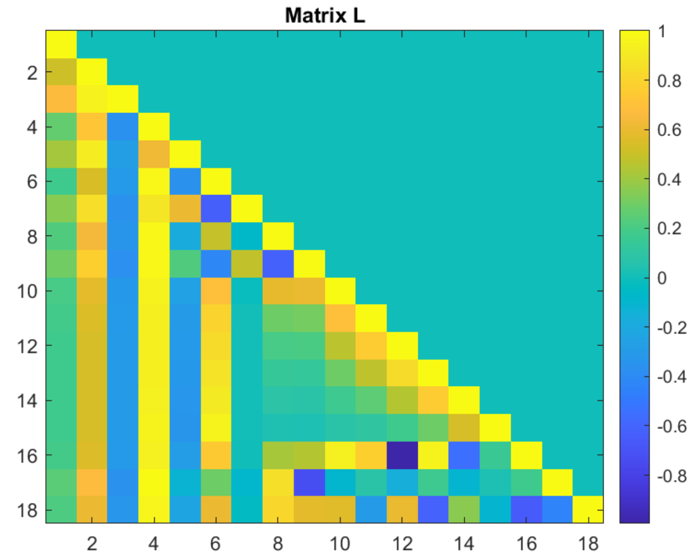
V:

V

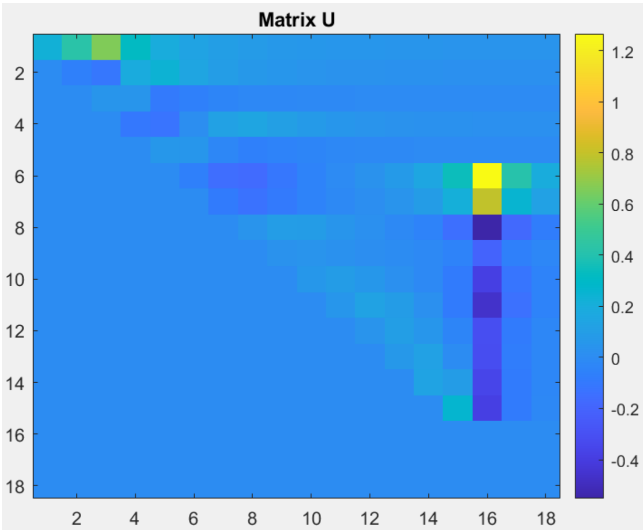
1x18 double

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
1	8.5402	8.5239	8.5360	8.0392	7.4376	7.2097	7.1800	7.2549	7.3156	7.2416	7.1317	7.1705	7.4808	8.0964	9.0998	9.9133	9.6065	12.9720

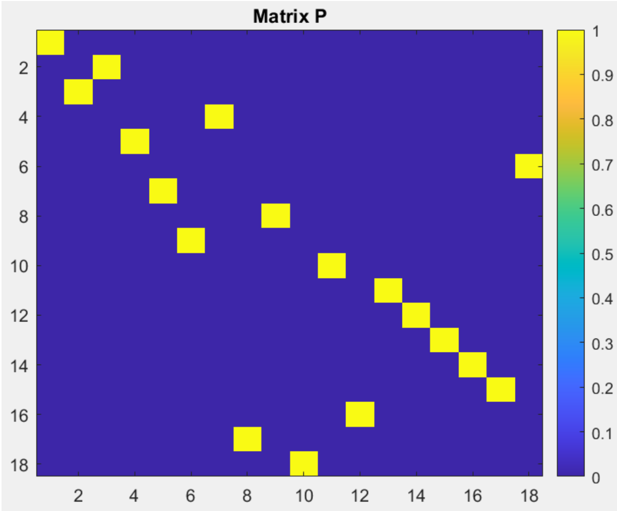
L:



U:



P:



$\boldsymbol{K(A)} \cong 3.5441 \cdot 10^{9}$

$||A||_{\boldsymbol{F}} = 3.0428\,,\ ||\bar{v}||_2 = 35.5802\,,\ ||\bar{q}||_2 = 20.4939.$

$\bar{q}$ (b):

q_b | q_c | q_d

1x18 double

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
1	3.0000	1.0000	3.0000	3.0000	9.0000	8.0712e-09	4.0000	2.0000	9.0000	3.0000	1.0000	3.0000	3.0000	9.0000	1.1019e-07	4.0000	2.0000	9.0000

$$\boldsymbol{\delta_{\hat{q}}} = \frac{||q-\bar{q}||_2}{||q||_2} = 9.9936 * 10^{-8}$$

<u>$c. A\bar{q} = \bar{v} + \delta v$</u>

$\bar{q}$ (c):

q_b | q_c | q_d

1x18 double

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
1	-2.6041	2.7247	3.6164	3.3839	9.2990	0.2613	4.2451	2.2420	9.2496	3.2686	1.3035	3.3649	3.4777	9.7156	1.3907	9.3437	-20.3229	19.0967

$\delta_{\hat{q}} \cong 1.2602$

<u>$d. (A + \delta A)\bar{q} = \bar{v}$</u>

$\bar{q}$ (d):

q_b | q_c | q_d

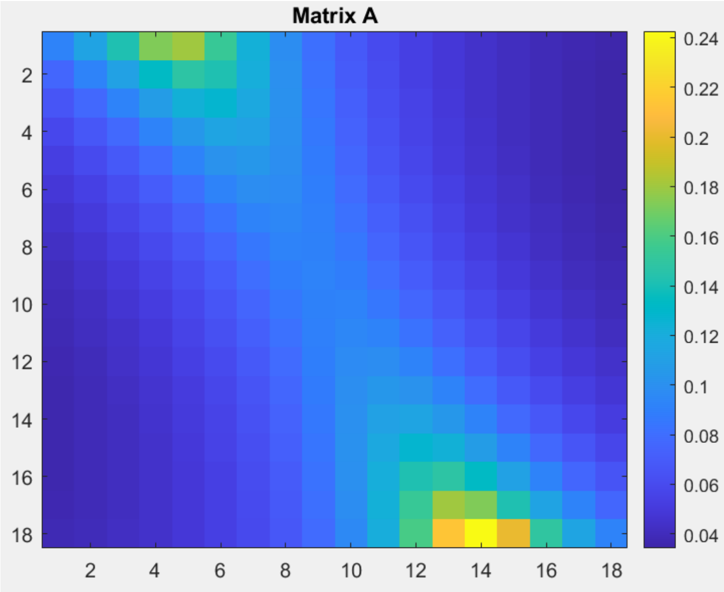
1x18 double

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
1	59.4604	-16.3760	-3.2101	-0.8675	5.9873	-2.6321	1.5311	-0.4384	6.4854	0.2939	-2.0579	-0.6767	-1.8130	1.7909	-14.0110	-49.8367	226.8980	-92.7220

$\delta_{\hat{q}} \cong 12.6965$

(3) $h = \frac{5\pi\rho}{M}$:

Matrix A:

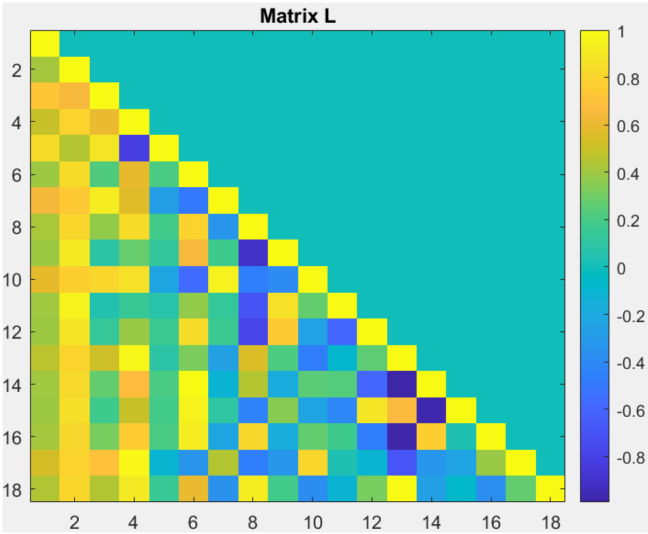


V:

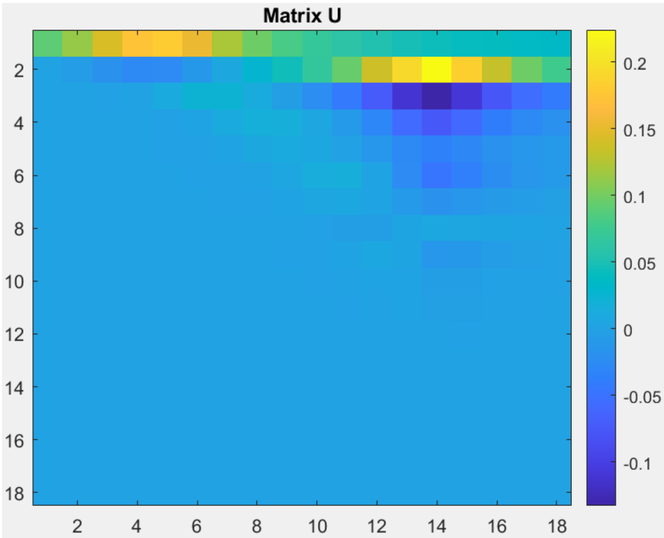
1x18 double

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
1	5.8805	5.3235	4.9321	4.6564	4.4621	4.3283	4.2422	4.1965	4.1872	4.2135	4.2774	4.3844	4.5440	4.7723	5.0955	5.5577	6.2378	7.2823

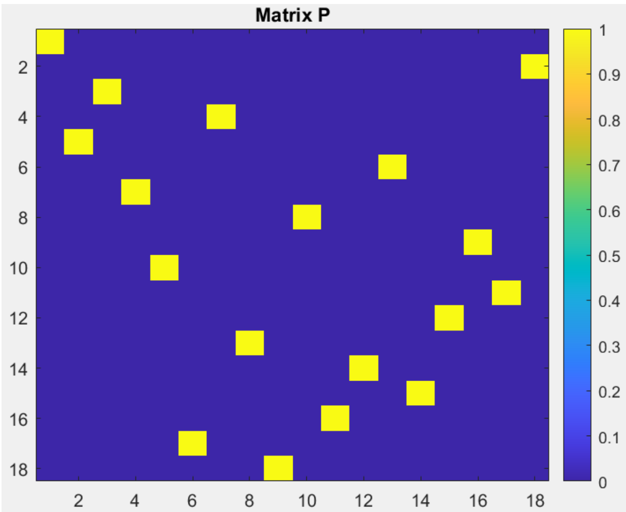
L:



U:



P:



$\boldsymbol{K}(\boldsymbol{A}) \cong 2.4849 \cdot 10^{13}$

$||A||_F = 1.4344\,,\ ||\bar{v}||_2 = 21.1707\,,\ ||\bar{q}||_2 = 20.4939.$

$\bar{q}$ (b):

q_b | q_c | q_d | V

1x18 double

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
1	3.0000	1.0000	3.0000	3.0000	9.0000	8.0712e-09	4.0000	2.0000	9.0000	3.0000	1.0000	3.0000	3.0000	9.0000	1.1019e-07	4.0000	2.0000	9.0000

$$\boldsymbol{\delta}_{\hat{q}} = \frac{||q-\bar{q}||_2}{||q||_2} = 3.0169 * 10^{-4}$$

<u>$c.\, A\bar{q} = \bar{v} + \delta v$</u>

$\bar{q}$ (c):

q_b | q_c | q_d | V

1x18 double

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
1	-2.6041	2.7247	3.6164	3.3839	9.2990	0.2613	4.2451	2.2420	9.2496	3.2686	1.3035	3.3649	3.4777	9.7156	1.3907	9.3437	-20.3229	19.0967

$\delta_{\hat{q}} \cong 207.6651$

<u>$d.\, (A + \delta A)\bar{q} = \bar{v}$</u>

$\bar{q}$ (d):

q_b | q_c | q_d | V

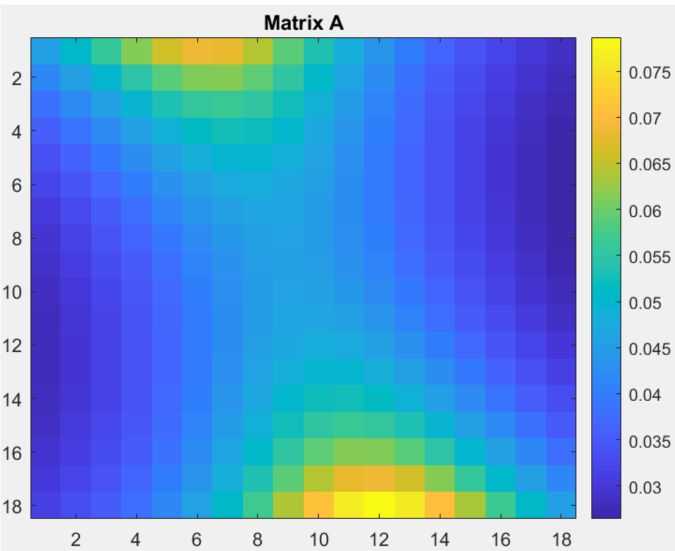
1x18 double

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
1	59.4604	-16.3760	-3.2101	-0.8675	5.9873	-2.6321	1.5311	-0.4384	6.4854	0.2939	-2.0579	-0.6767	-1.8130	1.7909	-14.0110	-49.8367	226.8980	-92.7220

$\delta_{\hat{q}} \cong 3.4034 \cdot 10^{10}$

(4) $h = \frac{10\pi\rho}{M}$:

Matrix A:



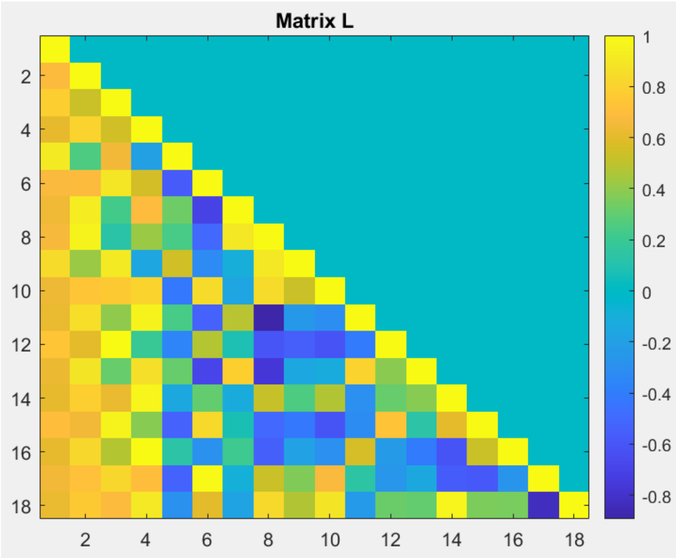
V:

q_b | q_c | q_d | V

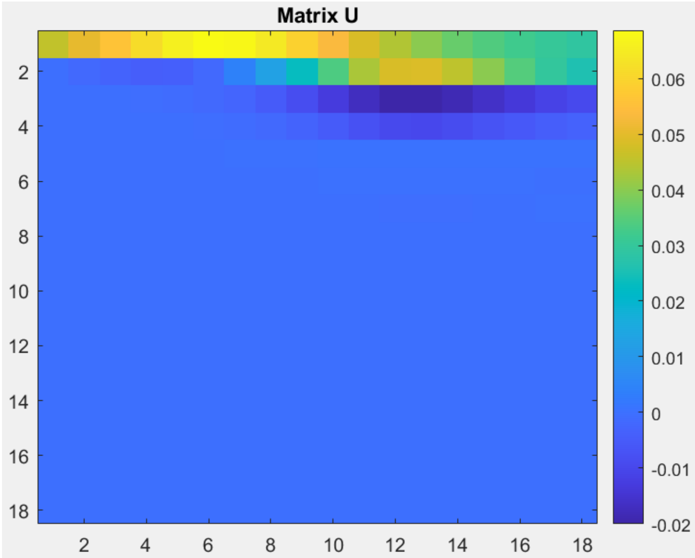
1x18 double

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
1	3.3048	3.0717	2.8940	2.7586	2.6571	2.5838	2.5352	2.5087	2.5031	2.5183	2.5546	2.6138	2.6985	2.8128	2.9629	3.1577	3.4106	3.7423

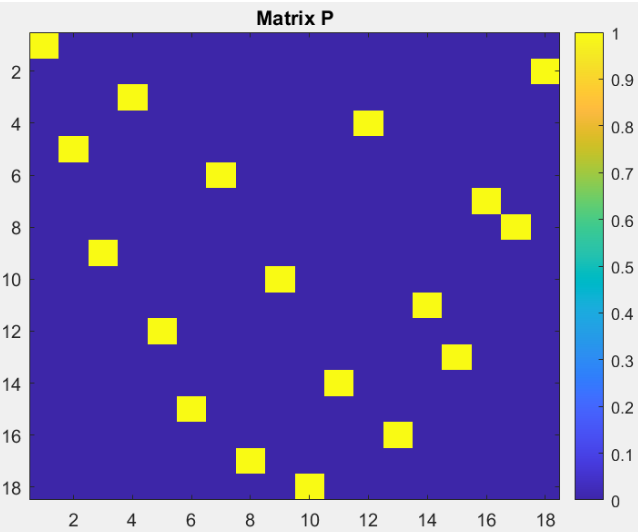
L:



U:



P:



$\boldsymbol{K(A)} \cong 3.9368 \cdot 10^{17}$

$||A||_{\boldsymbol{F}} = 0.7794,\ ||\bar{v}||_2 = 12.1797\ ,\ ||\bar{q}||_2 = 20.4939.$

$\bar{q}$ (b):

q_b | q_c | q_d | V | L

1x18 double

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
1	3.2866	1.3987	4.4109	6.7160	16.6615	4.7434	-35.1420	37.4064	-20.4733	65.3093	-82.6564	14.7008	18.3055	18.7726	2.2599	6.6364	1.6267	9.8216

$$\boldsymbol{\delta_{\hat{q}}} = \frac{||q - \bar{q}||_2}{||q||_2} = 5.9984$$

$\underline{c.\ A\bar{q} = \bar{v} + \delta v}$

$\bar{q}$ (c):

Variables - q_c

q_b | q_c | q_d | V | L

1x18 double

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
1	1.5490e+05	4.7502e+04	6.6953e+05	1.1215e+06	3.0589e+06	1.3589e+06	-1.3659e+07	1.1984e+07	-9.1208e+06	1.8080e+07	-2.3851e+07	3.2464e+06	4.4533e+06	2.7056e+06	6.9631e+05	6.8210e+05	-6.5981e+04	2.0890e+05

$\delta_{\hat{q}} \cong 1.8000 \cdot 10^6$

$d. (A + \delta A)\bar{q} = \bar{v}$

$\bar{q}$ (d):

Variables - q_d

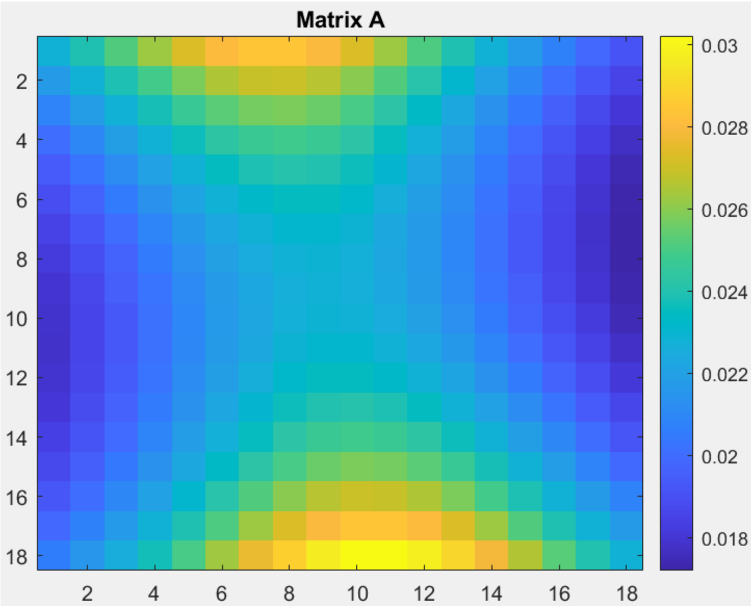
1x18 double

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
1	1.6330e+14	2.9631e+13	5.3017e+14	8.6105e+14	2.1490e+15	1.0217e+15	-9.5715e+15	6.9893e+15	-1.9362e+15	-3.3187e+15	7.5369e+15	-1.8026e+15	-6.0768e+14	-1.5663e+15	2.8745e+14	-8.9028e+14	4.5057e+14	-3.2558e+14

$\delta_{\hat{q}} \cong 7.3355 \cdot 10^{14}$

(5) $h = \frac{20\pi\rho}{M}$:

Matrix A:



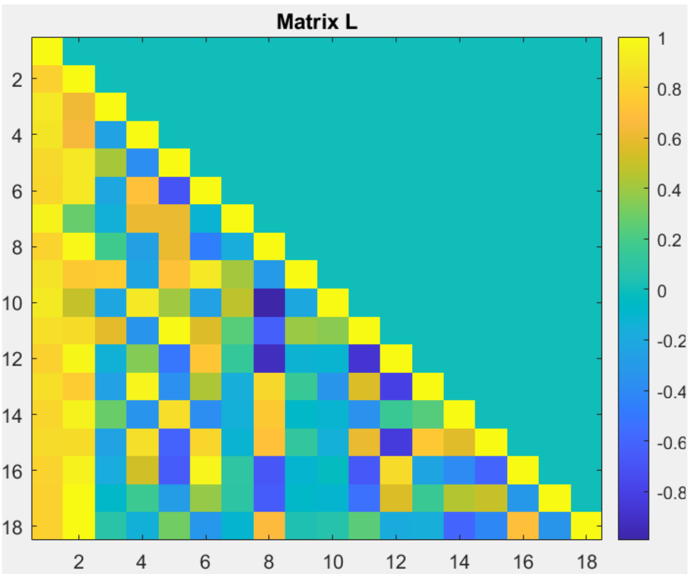
V:

Variables - V

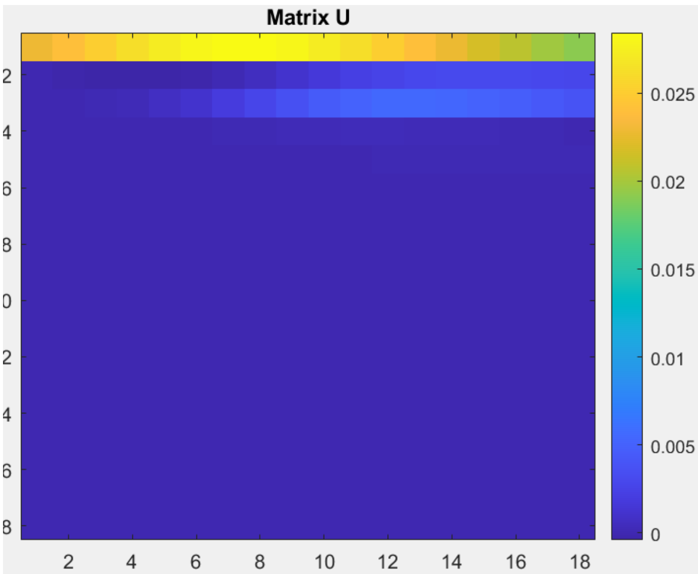
1x18 double

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
1	1.6694	1.5968	1.5370	1.4885	1.4505	1.4221	1.4026	1.3916	1.3889	1.3943	1.4080	1.4303	1.4617	1.5027	1.5544	1.6177	1.6938	1.7839

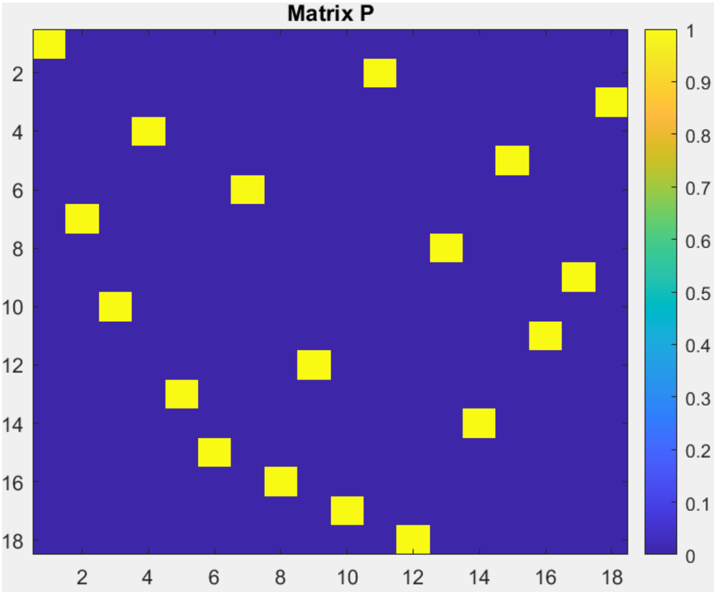
L:



U:



P:



$\boldsymbol{K(A)} \cong 3.3041 \cdot 10^{17}$

$||A||_F = 0.7794\,,\ ||\bar{v}||_2 = 12.1797\,,\ ||\bar{q}||_2 = 20.4939.$

$\bar{q}$ (d):

q_b q_c q_d V

1x18 double

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
1	3.2866	1.3987	4.4109	6.7160	16.6615	4.7434	-35.1420	37.4064	-20.4733	65.3093	-82.6564	14.7008	18.3055	18.7726	2.2599	6.6364	1.6267	9.8216

$$\boldsymbol{\delta_{\hat{q}}} = \frac{||q - \bar{q}||_2}{||q||_2} = 2.7163$$

$\underline{c.\, A\bar{q} = \bar{v} + \delta v}$

$\bar{q}$ (c):

q_b q_c q_d V

1x18 double

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
1	1.5490e+05	4.7502e+04	6.6953e+05	1.1215e+06	3.0589e+06	1.3589e+06	-1.3659e+07	1.1984e+07	-9.1208e+06	1.8080e+07	-2.3851e+07	3.2464e+06	4.4533e+06	2.7056e+06	6.9631e+05	6.8210e+05	-6.5981e+04	2.0890e+05

$\delta_{\hat{q}} \cong 2.4605 \cdot 10^6$

$\underline{d.\, (A + \delta A)\bar{q} = \bar{v}}$

$\bar{q}$ (d):

q_b q_c q_d V

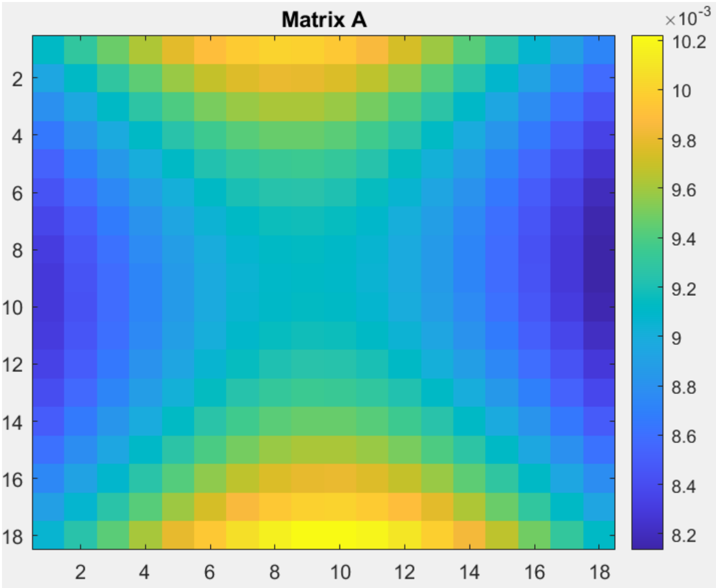
1x18 double

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
1	1.6330e+14	2.9631e+13	5.3017e+14	8.6105e+14	2.1490e+15	1.0217e+15	-9.5715e+15	6.9893e+15	-1.9362e+15	-3.3187e+15	7.5369e+15	-1.8026e+15	-6.0768e+14	-1.5663e+15	2.8745e+14	-8.9028e+14	4.5057e+14	-3.2558e+14

$\delta_{\hat{q}} \cong 1.7251 \cdot 10^{15}$

(6) $h = \frac{50\pi\rho}{M}$:

Matrix A:



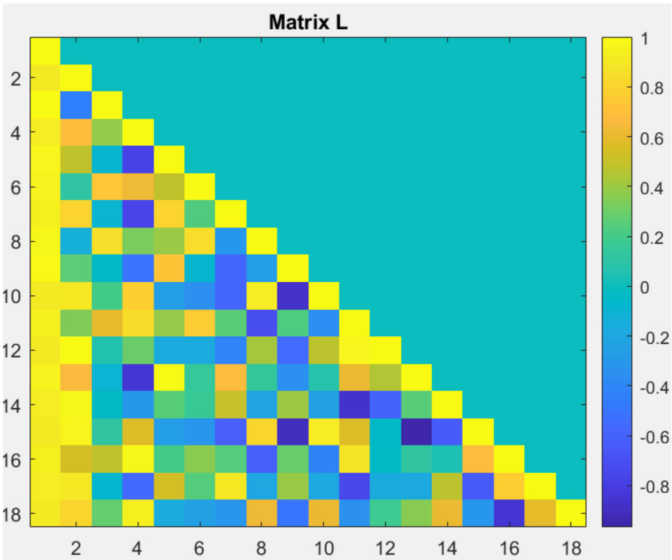
V:

q_b q_c q_d V

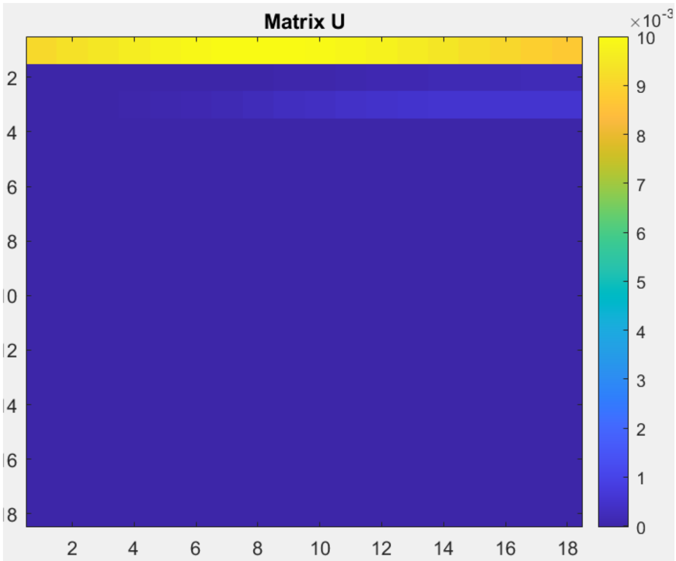
1x18 double

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
1	0.6469	0.6347	0.6239	0.6145	0.6069	0.6009	0.5966	0.5941	0.5934	0.5944	0.5972	0.6017	0.6080	0.6159	0.6255	0.6365	0.6490	0.6625

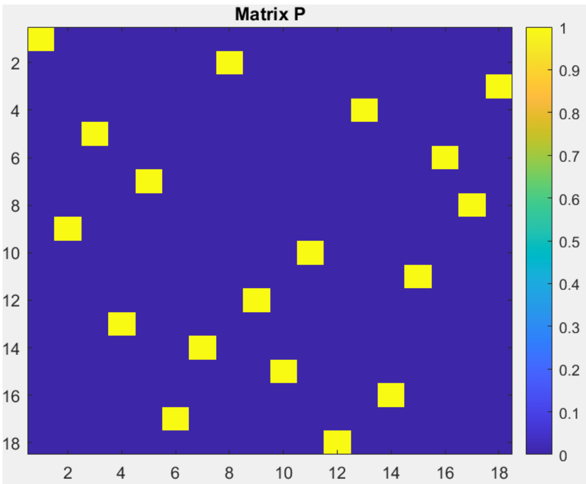
L:



U:



P:



$\boldsymbol{K(A)} \cong 2.8120 \cdot 10^{19}$

$||A||_{\boldsymbol{F}} = 0.4040\,,\ ||\bar{v}||_2 = 6.4285\,,\ ||\bar{q}||_2 = 20.4939.$

$\bar{q}$ (b):

q_b | q_c | q_d | V

1x18 double

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
1	2.4796	4.2150	-6.5800	21.3391	-16.0252	24.7521	-11.7106	2.9525	21.9499	-17.1906	20.3866	-10.1932	8.5278	9.4117	-3.0113	6.6590	0.7999	9.2375

$$\boldsymbol{\delta_{\hat{q}}} = \frac{||q-\bar{q}||_2}{||q||_2} = 78.7610$$

$c.\,A\bar{q} = \bar{v} + \delta v$

$\bar{q}$ (c):

q_b | q_c | q_d | V

1x18 double

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
1	2.8762e+05	-1.6528e+06	4.4932e+06	-7.5573e+06	8.2857e+06	-4.7724e+06	-2.2906e+06	9.0592e+06	-1.0413e+07	3.8696e+06	8.0429e+06	-1.9674e+07	2.5977e+07	-2.4867e+07	1.7887e+07	-9.2646e+06	3.0803e+06	-4.9137e+05

$\delta_{\hat{q}} \cong 2.7096 \cdot 10^6$

$d.\,(A+\delta A)\bar{q} = \bar{v}$

$\bar{q}$ (d):

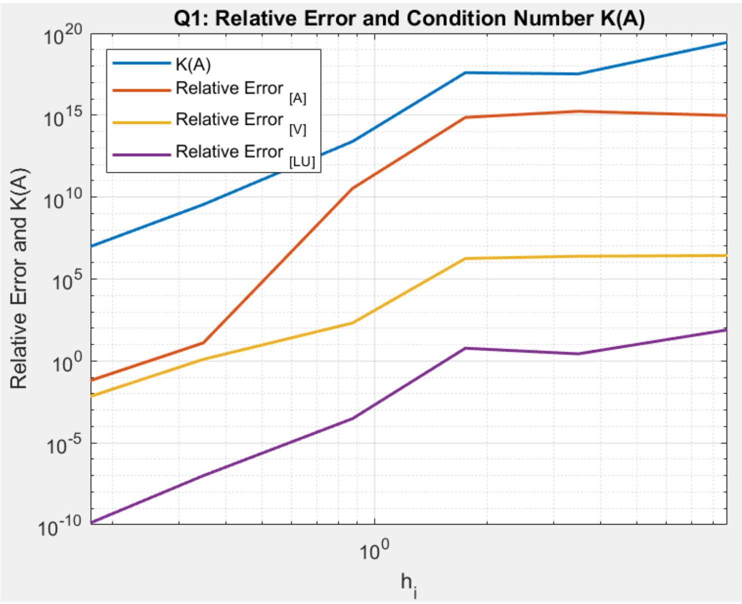
q_b | q_c | q_d | V

1x18 double

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
1	-1.1273e+13	-4.3584e+13	5.0529e+14	-1.8965e+15	4.5535e+15	-8.2657e+15	1.2042e+16	-1.4118e+16	1.2654e+16	-7.1355e+15	-8.7254e+14	8.5683e+15	-1.3380e+16	1.3951e+16	-1.0716e+16	5.8786e+15	-2.0568e+15	3.4285e+14

$\delta_{\hat{q}} \cong 9.6368 \cdot 10^{14}$

לאחר שסיימנו עם האיטרציות, נדפיס על גרף את שגיאת החישוב היחסית ואת מספר המצב כפונקציה של *h*, התוצאה שהתקבלה:



הסבר: ניתן לראות כי כשהערך של *h* גדל, אז מספר המצב *K(A)* של המטריצה *A* גדל ולכן גם השגיאה היחסית, כפי שהסברנו בסעיפים הקודמים, החסמים על השגיאה משתנים ולכן גם הערכים שנקבל עבור השגיאה יגדלו, החל מ*LU* שזו המדידה ללא שגיאת מדידה בוקטור הפתרונות או טעות באיברי המטריצה שעבורה נקבל שגיאת חישוב נמוכה משאר המדידות. לאחר מכן זו עם השגיאה בווקטור *V* ולאחר מכן המדידה עם השגיאות באיברי המטריצה *A*.

סוף שאלה 1, קוד MATLAB מצורף במלואו ב**נספח (עמ' 29).**

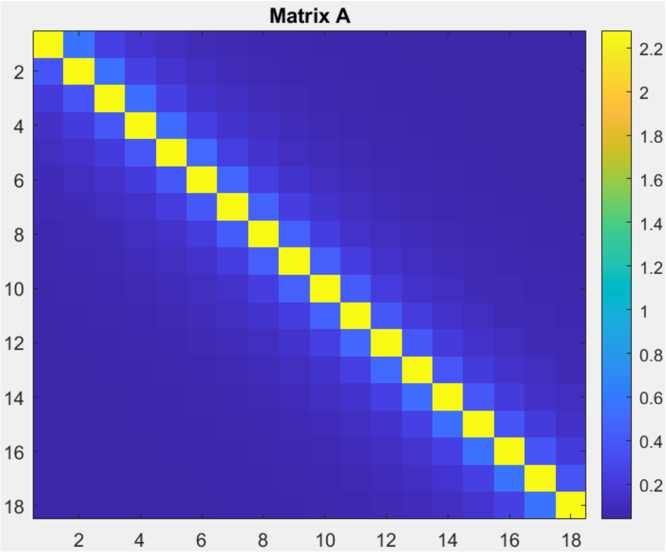
שאלה 2: פתרון מקורב בשיטות איטרטיביות

א. נרצה להגיע לפתרון המשוואה $Aq = \bar{v}$ בשיטת פתרון מקורב איטרטיבית Gauss-Seidel. ניזכר במשתנים שהגדרנו בתחילת התוכנית:

```
%% variables
M = 18;
length = M;
Rho = 1;
q = [3,1,3,3,9,0,4,2,9,3,1,3,3,9,0,4,2,9]'; % my id
```

התבקשנו להגדיר מספר משתנים:

- $h = \frac{\pi\rho}{5M} = \frac{\pi}{5\cdot18} = \frac{\pi}{90}$
- **Build matrix A:**



- Calculate $\bar{\boldsymbol{v}} = \boldsymbol{Aq}$ (transpose variable):

v

1x18 double

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
1	12.2954	10.0688	14.7722	16.9001	27.1071	11.1073	17.6863	15.5895	28.1495	17.0586	12.2257	15.6970	16.9083	27.7601	11.8064	17.0701	14.3981	26.2029

נרצה לפתור את המשוואה $A\bar{q} = \bar{v}$ בשיטת Gauss-Seidel, לשם כך:

*1. נרצה להגדיר **סיבולת שגיאה יחסית של** $\mathbf{10^{-3}}$, נוודא זאת באופן הבא -*
נגדיר משתנה וקטור עמודה בגודל מתאים q_k (18,x), בכל איטרציה חישובית שנבצע, נוסיף למשתנה זה עמודה מימין ונחשב את מרחק השגיאה היחסית בין העמודה האחרונה שחושבה לבין הפתרון האמיתי, ונוודא שגודל זה קטן מ-10^{-3}.

2. ניחוש התחלתי $\bar{\boldsymbol{q}}^{(\mathbf{0})} = \mathbf{0}$ (במקרה שלנו – וקטור עמודה באורך 18 עם כל האיברים שהם 0).

*3. (מההרצאה) על מנת שנקבל פתרון מתכנס בשיטת גאוס-זיידל עלינו לוודא כי **לפחות אחד** מהתנאים הבאים מתקיימים:*
א. למטריצה A יש אלכסון דומיננטי.
ב. המטריצה A סימטרית מוגדרת וחיובית ($x^T Ax > 0$).

<u>הערה (גם לגבי שיטת יעקובי):</u> מכיוון שלא ניתן לחלק ב-0 (ניחוש התחלתי נתון הוא וקטור אפסים), הגדרתי שגיאה יחסית בין 2 הפתרונות הצמודים הראשונים שווה ל-1.

לצורך פתירת הסעיף, בניתי את הפונקציה הבאה שתפתור את המשוואה Ax=b בשיטת גאוס-זיידל (קוד במאטלב):

```
function [iter_track,dist_error,real_error] = Gauss_Seidel(~,A,x,b)
    tol = 1e-3;
    L = tril(A,-1);    % Lower triangular matrix
    U = triu(A,1);     % Upper triangular matrix
    D = diag(diag(A)); % Diagonal matrix
    Q = L+D;
    G = -inv(Q)*U;
    c = inv(Q)*b;
    itr = 1; max_iter = 600;
    iter_track = ones;
    q_k = c;           % q^(0) = 0 , q^(1) = c
    dist_error = ones; % ||q^(1)-q^(0)|| / ||q^(0)|| (can't divide by 0)
    real_error = ones; % ||q^(1)-q|| / ||q||
    real_error(1) = norm(c-x,2)/norm(x);
    error = norm(x-q_k(:, itr),"inf")/norm(x,"inf");
    while abs(error)>tol && itr<=max_iter        % make sure we stop.
        q_k(:,itr+1) = G*q_k(:,itr) + c;                % Gauss-Seidel Algorithm.
        error = norm(x-q_k(:, itr),"inf")/norm(x,"inf"); % finding error ||q^k-q||
        dist_error(itr+1) = norm(q_k(:,itr+1) - q_k(:, itr), ...
            "inf")/norm(q_k(:, itr),"inf");
        real_error(itr+1) = norm(q_k(:,itr+1) - x, ...
            "inf")/norm(x,"inf");
        iter_track(itr+1)=itr+1;
        itr = itr + 1;
    end
end
```

הפונקציה מחזירה: (1) מס' איטרציות, (2) מרחק שגיאה בין 2 פתרונות צמודים ו-(3) מרחק שגיאה אמיתי בין העמודה האחרונה לבין הפתרון האמיתי.

לאחר הרצת הפונקציה הזו על המשוואה $A\bar{q} = \bar{v}$ למציאת הפתרון המקורב $\bar{q}$ נקבל (כל עמודה היא איטרציה):

dist_error | rel_error | q_k

18x7 double

	1	2	3	4	5	6	7
1	5.3934	1.5280	3.4822	2.8988	3.0099	3.0018	2.9986
2	3.5126	-0.0921	1.2883	0.9593	1.0016	1.0032	0.9991
3	5.3909	2.0946	3.1507	2.9712	2.9948	3.0029	2.9995
4	5.8232	2.5539	3.1576	2.9951	2.9929	3.0023	2.9998
5	9.8868	8.3856	9.0905	9.0064	8.9926	9.0015	8.9999
6	1.8325	-0.4657	0.0539	0.0160	-0.0054	0.0012	4.0478e-05
7	5.5135	3.5610	3.9922	4.0106	3.9946	4.0009	4.0002
8	4.1942	1.8805	2.0092	2.0127	1.9948	2.0004	2.0002
9	9.7091	8.8322	9.0058	9.0166	8.9959	9.0000	9.0002
10	3.6710	2.7847	2.9829	3.0173	2.9972	2.9997	3.0002
11	2.1040	0.8463	0.9657	1.0160	0.9987	0.9997	1.0002
12	4.0982	2.9039	2.9462	3.0098	2.9990	2.9995	3.0001
13	4.3847	3.1055	2.9726	3.0108	2.9999	2.9995	3.0001
14	8.9489	8.9679	8.9575	9.0076	9.0006	8.9995	9.0001
15	0.7259	0.0549	-0.0395	0.0051	0.0012	-2.7336e-04	4.5405e-05
16	4.3851	4.0568	3.9536	3.9986	4.0007	3.9998	4.0000
17	2.7950	2.2349	1.9932	1.9997	2.0006	1.9998	2.0000
18	8.2181	9.0502	8.9943	8.9994	9.0004	8.9999	9.0000

העמודה הימנית ביותר הינה הפתרון המקורב שהתקבל, כלומר $\bar{q}$:

המרחק היחסי כפונקציה של האיטרציות (שני פתרונות צמודים):

dist_error_5M

1x7 double

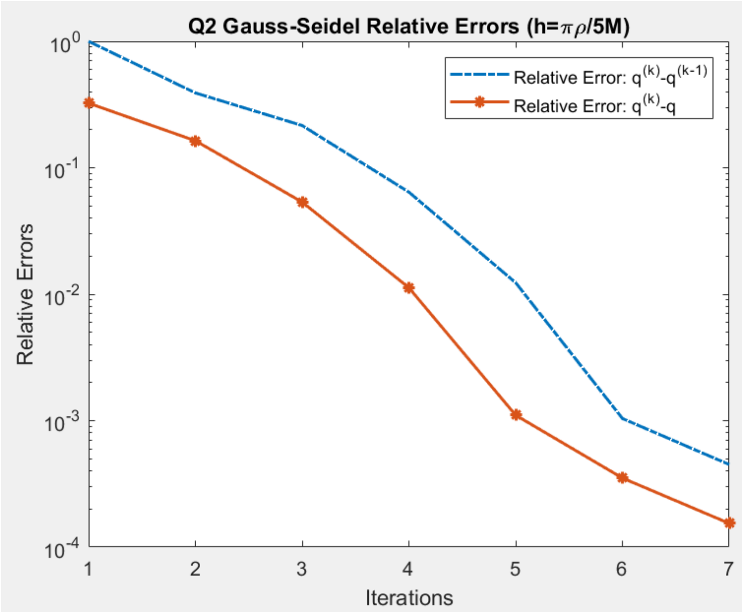
	1	2	3	4	5	6	7
1	1	0.3910	0.2159	0.0642	0.0123	0.0010	4.4952e-04

השגיאה היחסית האמיתית כפונקציה של האיטרציות (פתרון מקורב ביחס לפתרון האמיתי):

rel_error_5M

1x7 double

	1	2	3	4	5	6	7
1	0.3244	0.1636	0.0536	0.0112	0.0011	3.5147e-04	1.5393e-04



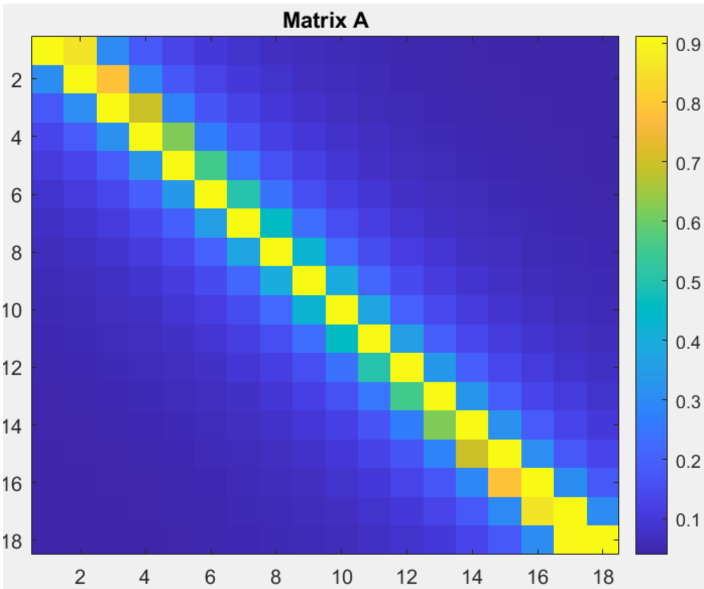
השגיאה היחסית *של* $\bar{q}$ *ביחס ל-*q *:* $1.5393 \cdot 10^{-4}$.
מספר האיטרציות *שנדרש להתכנסות: 7.*

ניתן לראות כי השגיאה היחסית הסופית קטנה מ-10^{-3} *בהתאם לדרישה,*
בנוסף לכך שבכל איטרציה השגיאה היחסית קטנה עוד יותר מהקודמת לה, כלומר מונוטונית יורדת.

ב. חזרה על סעיף א' כאשר $h=\frac{\pi\rho}{2M}, h=\frac{\pi\rho}{M}$:

$\boldsymbol{h=\frac{\pi\rho}{2M}}$**:**

Matrix A:



V=Aq (Transpose variable):

v

17	3.0001	0.9999	3.0000	3.0000	9.0000	-8.2411e-06	4.0000	2.0000	9.0000	3.0000	1.0000	3.0000	3.0000	9.0000	1.2880e-05	4.0000	2.0000	9.0000

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
8.7929	9.4409	11.3122	13.4359	14.5700	10.4631	11.9443	12.1827	15.4333	12.4203	10.4341	11.1933	12.2643	15.6645	12.9103	11.6747	12.3807	14.9190

Gauss-Seidel $\bar{\mathbf{q}}$ (Transpose variable):

14	2.9972	1.0017	3.0001	2.9999	8.9999	-4.5247e-05	4.0000	2.0000	9.0000	3.0000	1.0000	3.0000	3.0000	9.0000	1.1129e-04	3.9998	2.0004	8.9997

Relative Distance Error ($q^{(k)}-q^{(k-1)}$) as a function of iterations:

dist_error_2M

1x14 double

	1	2	3	4	5	6	7	8	9	10	11	12	13	14
1	1	1.6808	1.5640	0.4962	0.1814	0.0949	0.0321	0.0366	0.0336	0.0141	0.0026	0.0020	6.4944e-04	5.2327e-04

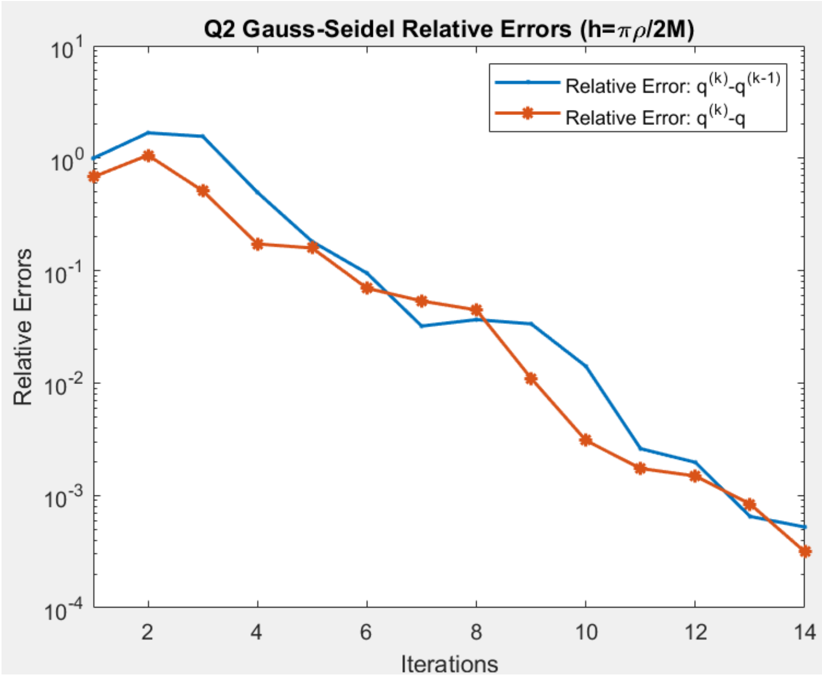
Relative Error $(q^{(k)} - q)$ as a function of iterations:

rel_error_2M

1x14 double

	1	2	3	4	5	6	7	8	9	10	11	12	13	14
1	0.6814	1.0628	0.5123	0.1726	0.1589	0.0697	0.0537	0.0446	0.0110	0.0031	0.0017	0.0015	8.3838e-04	3.1511e-04

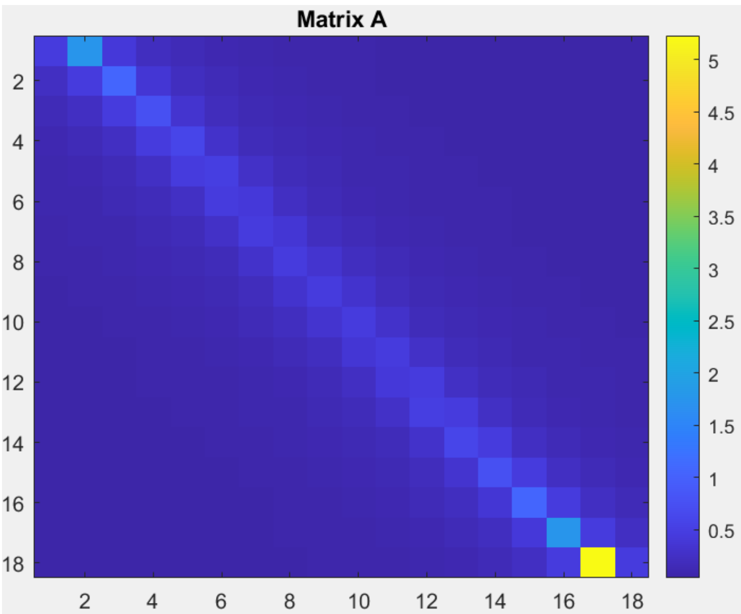
Relative Error: $3.1511 \cdot 10^{-4}$.
Number of iterations: *14*.
Figure:



ניתוח: הפתרון שחושב קודם התקבל עבור 8 איטרציות, הפעם עבור $h = \frac{\pi\rho}{2M}$, (כלומר h גדול יותר) , קיבלנו פתרון עבור 14 איטרציות. ניתן להסביר זאת ע"י כך ש-h מתאר את המרחק שבין מרכז של שתי קשתות מעגליות מקבילות, לכן ככל שהמרחק גדול יותר, כלומר השגיאות בחישוב המטענים יהיו גדולות יותר ולכן נדרש לבצע יותר מדידות על מנת לקבל את הפתרון המתכנס.

$\boldsymbol{h = \frac{\pi\rho}{M}}$:

Matrix A (same as Q1-a):



Variables - A

A

18x18 double

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
1	0.4559	1.7477	0.4128	0.2169	0.1474	0.1122	0.0911	0.0772	0.0674	0.0602	0.0548	0.0506	0.0474	0.0448	0.0429	0.0413	0.0402	0.0395
2	0.2301	0.4559	1.0539	0.3707	0.2052	0.1417	0.1087	0.0886	0.0753	0.0659	0.0590	0.0537	0.0497	0.0465	0.0441	0.0422	0.0407	0.0396
3	0.1546	0.2337	0.4559	0.7586	0.3298	0.1926	0.1355	0.1049	0.0860	0.0733	0.0643	0.0577	0.0526	0.0487	0.0457	0.0433	0.0415	0.0401
4	0.1171	0.1573	0.2392	0.4559	0.5961	0.2942	0.1803	0.1293	0.1010	0.0834	0.0714	0.0628	0.0564	0.0516	0.0479	0.0449	0.0427	0.0409
5	0.0948	0.1191	0.1611	0.2469	0.4559	0.4940	0.2646	0.1689	0.1233	0.0973	0.0808	0.0695	0.0613	0.0553	0.0506	0.0470	0.0442	0.0420
6	0.0801	0.0964	0.1219	0.1663	0.2572	0.4559	0.4246	0.2404	0.1586	0.1177	0.0938	0.0784	0.0677	0.0600	0.0542	0.0497	0.0463	0.0436
7	0.0699	0.0815	0.0986	0.1255	0.1730	0.2705	0.4559	0.3747	0.2207	0.1497	0.1127	0.0907	0.0762	0.0661	0.0587	0.0532	0.0489	0.0456
8	0.0623	0.0710	0.0832	0.1012	0.1300	0.1814	0.2875	0.4559	0.3377	0.2047	0.1420	0.1083	0.0878	0.0742	0.0646	0.0576	0.0523	0.0482
9	0.0567	0.0634	0.0725	0.0853	0.1045	0.1354	0.1918	0.3094	0.4559	0.3094	0.1918	0.1354	0.1045	0.0853	0.0725	0.0634	0.0567	0.0516
10	0.0523	0.0576	0.0646	0.0742	0.0878	0.1083	0.1420	0.2047	0.3377	0.4559	0.2875	0.1814	0.1300	0.1012	0.0832	0.0710	0.0623	0.0559
11	0.0489	0.0532	0.0587	0.0661	0.0762	0.0907	0.1127	0.1497	0.2207	0.3747	0.4559	0.2705	0.1730	0.1255	0.0986	0.0815	0.0699	0.0615
12	0.0463	0.0497	0.0542	0.0600	0.0677	0.0784	0.0938	0.1177	0.1586	0.2404	0.4246	0.4559	0.2572	0.1663	0.1219	0.0964	0.0801	0.0689
13	0.0442	0.0470	0.0506	0.0553	0.0613	0.0695	0.0808	0.0973	0.1233	0.1689	0.2646	0.4940	0.4559	0.2469	0.1611	0.1191	0.0948	0.0791
14	0.0427	0.0449	0.0479	0.0516	0.0564	0.0628	0.0714	0.0834	0.1010	0.1293	0.1803	0.2942	0.5961	0.4559	0.2392	0.1573	0.1171	0.0936
15	0.0415	0.0433	0.0457	0.0487	0.0526	0.0577	0.0643	0.0733	0.0860	0.1049	0.1355	0.1926	0.3298	0.7586	0.4559	0.2337	0.1546	0.1157
16	0.0407	0.0422	0.0441	0.0465	0.0497	0.0537	0.0590	0.0659	0.0753	0.0886	0.1087	0.1417	0.2052	0.3707	1.0539	0.4559	0.2301	0.1530
17	0.0402	0.0413	0.0429	0.0448	0.0474	0.0506	0.0548	0.0602	0.0674	0.0772	0.0911	0.1122	0.1474	0.2169	0.4128	1.7477	0.4559	0.2283
18	0.0401	0.0409	0.0420	0.0436	0.0456	0.0482	0.0515	0.0558	0.0614	0.0688	0.0789	0.0933	0.1152	0.1520	0.2261	0.4470	5.2292	0.4559

V=Aq *(transpose variable):*

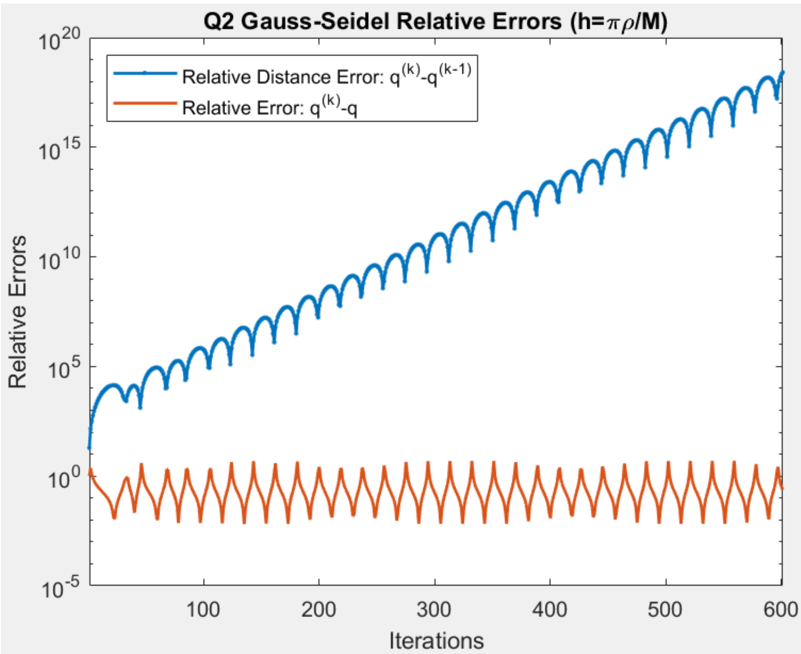
v

1x18 double

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
1	8.9900	10.1489	10.5069	11.6248	10.1541	9.2314	9.4575	9.8640	10.5759	9.8502	9.0555	9.1267	8.9979	11.3892	13.2335	10.3408	14.8078	20.3266

אם נתבונן במטריצה A נוכל לראות כי אין לה אלכסון דומיננטי (למשל האיבר במקום (1,1) קטן מהאיבר במקום (1,2)), לכן לא נוכל לקבל פתרון מתכנס במצב זה, כלומר השגיאה היחסית האמיתית מתבדרת.

לאחר 600 איטרציות נקבל את הגרף:



ג. כעת נרצה לפתור את המשוואה $Aq = v$ בשיטת פתרון מקורב איטרטיבית *Jacobi*. נחזור על סעיף א' – המשתנים *A,v,h* הינם **זהים** למה שחישבנו בסעיף א', רק שיטת הפתרון משתנה.
על מנת לקבל פתרון מתכנס בשיטת יעקובי:

$$\underbrace{\left\| \underline{\underline{I}} - \underline{\underline{Q}}^{-1}\underline{\underline{A}} \right\|_\infty}_{\|\underline{\underline{G}}\|_\infty} = \max\Big\{ \overbrace{\left|\frac{a_{12}}{a_{11}}\right| + \left|\frac{a_{13}}{a_{11}}\right| + \dots + \left|\frac{a_{1n}}{a_{11}}\right|}^{\text{שורה 1}}, \overbrace{\left|\frac{a_{21}}{a_{22}}\right| + \left|\frac{a_{23}}{a_{22}}\right| + \dots + \left|\frac{a_{2n}}{a_{22}}\right|}^{\text{שורה 2}}, \dots, \underbrace{\left|\frac{a_{n1}}{a_{nn}}\right| + \left|\frac{a_{n2}}{a_{nn}}\right| + \dots + \left|\frac{a_{n,n-1}}{a_{nn}}\right|}_{\text{שורה n}} \Big\}$$

נורמת ∞ של מטריצה: סכום הערכים המוחלטים של איברים בשורה ואז max מבין כל השורות

$$= \underbrace{\max_{1\le i\le n} \sum_{\substack{j=1 \\ j\ne i}}^{n} \left|\frac{a_{ij}}{a_{ii}}\right|}_{\|\underline{\underline{G}}\|_\infty} \underbrace{< 1}_{\text{לקבלת התכנסות}}$$

<u>משפט</u> (מההרצאה) – אם למטריצה A **אלכסון דומיננטי**, אזי סדרת הוקטורים $\{x^{(k)}\}$ המתקבלת מהפעלת איטרצית יעקובי **מתכנסת** לפתרון המערכת *Ax=b* **מכל** תנאי התחלה $x^{(0)}$.

מצורפת תמונה של פונקציה שרשמתי שתפתור את המשוואה Ax=b בשיטת יעקובי (קוד במאטלב):

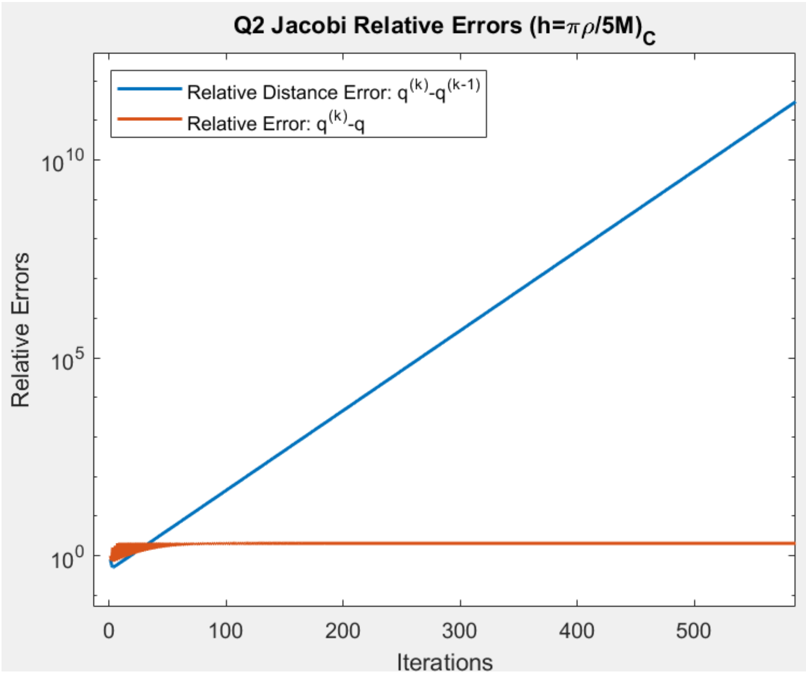
```
function [iter_track,dist_error,real_error] = Jacobi(~,A,x,b) %Aq=v,Ax=b
    tol = 1e-3;
    D = diag(diag(A));        % Diagonal matrix
    Q = D;
    I = eye(length(x));
    G = I-inv(Q)*A;
    norm_G = norm(G,"inf");   % Check for || G || < 1,
    c = inv(Q)*b;
    itr = 1; max_iter = 600;
    iter_track = ones;
    q_k = c;              % q^(0)=0 , q^(1)=c
    dist_error = ones; real_error = ones;  % same as G-S
    real_error(1) = norm(c-x,2)/norm(x);    % ".
    error = norm(x-q_k(:, itr),"inf")/norm(x,"inf"); % just a start value.
    while abs(error) > tol && itr<=max_iter % make sure we stop.
        q_k(:,itr+1) = G*q_k(:,itr) + c;         % Jacobi Algorit.
        error = norm(x-q_k(:, itr),"inf")/norm(x,"inf");   % finding error
        dist_error(itr+1) = norm(q_k(:,itr+1) - q_k(:, itr), ...
            "inf")/norm(q_k(:, itr),"inf");
        real_error(itr+1) = norm(q_k(:,itr) - x,"inf")/norm(x,"inf");
        iter_track(itr+1)=itr+1;
        itr = itr + 1;
    end
end
```

הפונקציה מחזירה: (1) מס' איטרציות, (2) מרחק שגיאה בין 2 פתרונות צמודים ו-(3) מרחק שגיאה אמיתי בין הפתרון האחרון שנמדד לבין הפתרון האמיתי.

נשים לב שעבור הרצת פונקציה זו על המשוואה $A\bar{q} = \bar{v}$ למציאת הפתרון המקורב $\bar{q}$ נקבל:

$$||G|| = \left\|I - Q^{-1}A\right\|_\infty = 1.103 > 1$$

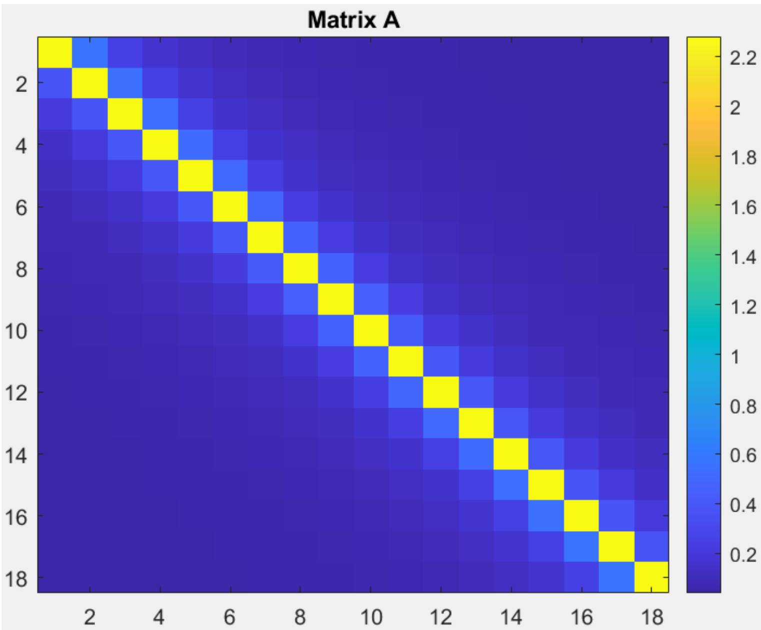
כלומר תנאי הכרחי לקבלת התכנסות פתרון יעקובי לא מתקיים, לכן הפתרון שמחושב בשיטה זו לא מתכנס לפתרון מתאים למשוואה, והשגיאה היחסית האמיתית מתבדרת. לאחר 600 איטרציות, הגרף המתקבל הינו (בעמוד הבא):



ד. נחזור על סעיף ג', נרצה לפתור את המשוואה $Aq = v$ בשיטת פתרון מקורב איטרטיבית *Jacobi*, הפעם עם שינוי איברי המטריצה *A*:

$$a_{mn} = \frac{1}{4\pi r_{mn}^2} = \frac{1}{4\pi\left\{\left[h + \rho\sin(m\pi / M) - \rho\sin(n\pi / M)\right]^2 + \left[\rho\cos(m\pi / M) - \rho\cos(n\pi / M)\right]^2\right\}}$$

ניעזר בפונקציה שבנינו בשאלה 1 סעיף א' (הפעם האיבר שכופל את 4*פאי הוא בריבוע, קוד בנספח), נחשב את מטריצה *A* ונקבל :



(נשים לב לאלכסון המטריצה שהתקבל בסעיף זה ביחס לאלכסון המטריצה שהתקבל בתמונה של הסעיף הקודם – כמו של סעיף א')

V=Aq (Transpose variable):

v

1x18 double

	1	2	3	4	5	6	7	8	9
1	206.7975	89.9169	220.1742	238.0655	602.7541	39.6712	282.5506	171.1359	608.3365

	10	11	12	13	14	15	16	17	18
1	230.6573	94.0693	218.4221	230.3011	607.5105	49.4790	280.1758	168.3891	602.7181

$\bar{q}$ $(Transpose\ variable)$:

4	2.9995	0.9994	2.9992	2.9993	8.9992	-6.9760e-04	3.9992	1.9993	8.9992	2.9993	0.9992	2.9992	2.9992	8.9992	-7.5935e-04	3.9991	1.9993

Relative Distance Error $(q^{(k)} - q^{(k-1)})$:

dist_error_d

1x4 double

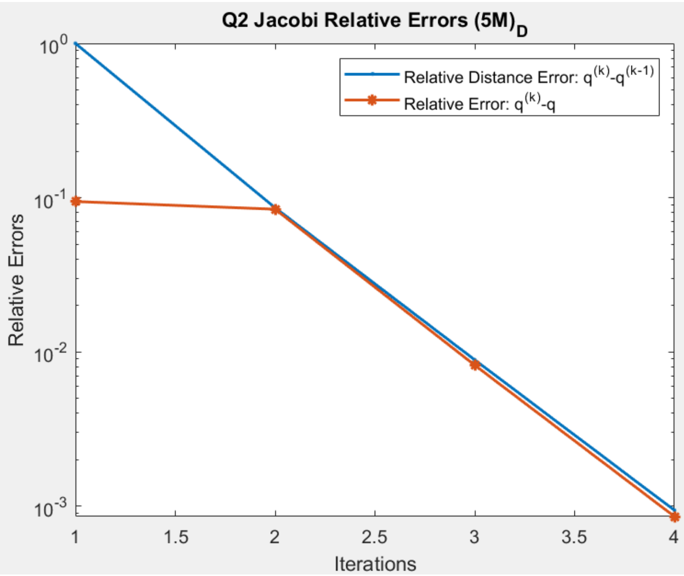
	1	2	3	4
1	1	0.0860	0.0089	9.4106e-04

Relative Error $(q^{(k)} - q)$:

rel_error_d

1x4 double

	1	2	3	4
1	0.0944	0.0842	0.0082	8.5732e-04



הפעם קיבלנו פתרון מתכנס **ב-4 איטרציות.**

שגיאה יחסית סופית: $8.5732 \cdot 10^{-4}$.

קיבלנו פתרון מתכנס עבור מטריצה A שונה שעבורה מתקיים:

$$||G|| = ||I - Q^{-1}A||_{\infty} = 0.1244 < 1$$

לכן התנאים להתכנסות הפתרון בשיטת יעקובי מתקיימים.

נשים לב, שבשיטת גאוס-זיידל (סעיף א') קיבלנו פתרון מתכנס באותו מצב בדיוק שבו שיטת יעקובי (סעיף ג') לא הביאה פתרון מתכנס, כלומר שיטת גאוס-זיידל מתכנסת תחת הנחות חלשות יותר ולכן יותר עמידה תחת מצבים שונים.

סוף שאלה 2, קוד MATLAB מצורף במלואו ב<u>נספח</u> (עמ' 31).

שאלה 3: פתרון בשיטת Least Squares עבור מטריצה ריבועית

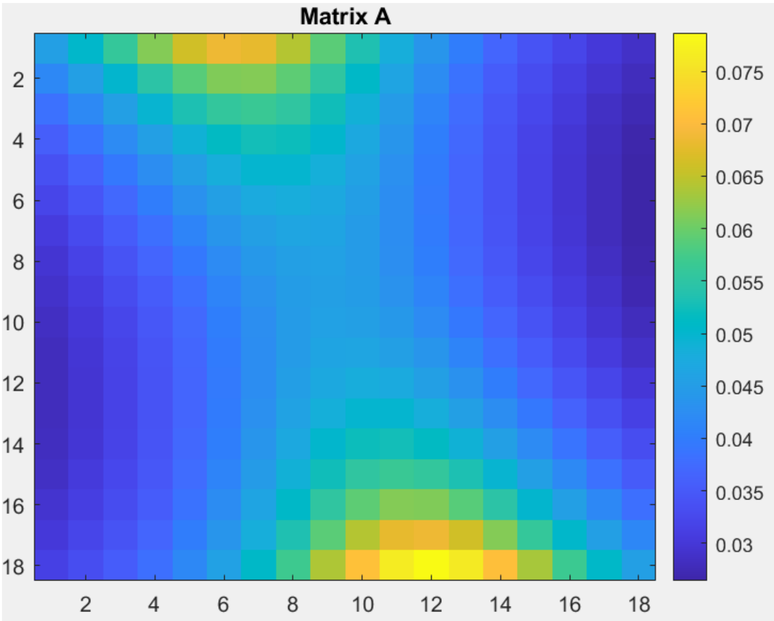
עבור המשוואה $Aq = v$ נרצה למצוא את הפתרון $\bar{q}$ בשיטת Least Squares, ראינו בהרצאה כי עבור $\bar{\boldsymbol{q}} = (\boldsymbol{A}^T\boldsymbol{A})^{-1}\boldsymbol{A}^T\boldsymbol{v}$ נקבל פתרון מקורב המביא למינימום את השגיאה הריבועית $||A\bar{q} - v||_2$.

נשאר עם אותם המשתנים שהגדרנו בתחילת התוכנית:

```
%% variables
M = 18;
length = M;
Rho = 1;
q = [3,1,3,3,9,0,4,2,9,3,1,3,3,9,0,4,2,9]'; % my id
```

א. עבור $h = \frac{10\pi\rho}{M}$, נבנה את המטריצה A (עם אותה הפונקציה משאלה 1 סעיף א'):

Matrix A:



Absolute Determinant** Value of matrix A:* $-7.5197 \cdot 10^{-147}$ ***(Real det. value is negative)

***Vector** $\bar{\boldsymbol{v}}$ (Transpose):*

v

1x18 double

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
1	3.3048	3.0717	2.8940	2.7586	2.6571	2.5838	2.5352	2.5087	2.5031	2.5183	2.5546	2.6138	2.6985	2.8128	2.9629	3.1577	3.4106	3.7423

$\bar{\boldsymbol{q}}$ *as computed* $(A^TA)^{-1}A^T$ *(Transpose):*

rel_q

1x18 double

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
1	-202.9551	-1.0656	10.2309	49.1130	-250.7383	-247.5511	-92.9086	-349.8924	129.4765	84.3405	-117.5221	-19.9790	-81.3248	-163.7294	-260.4655	89.3378	-161.0124	98.7454

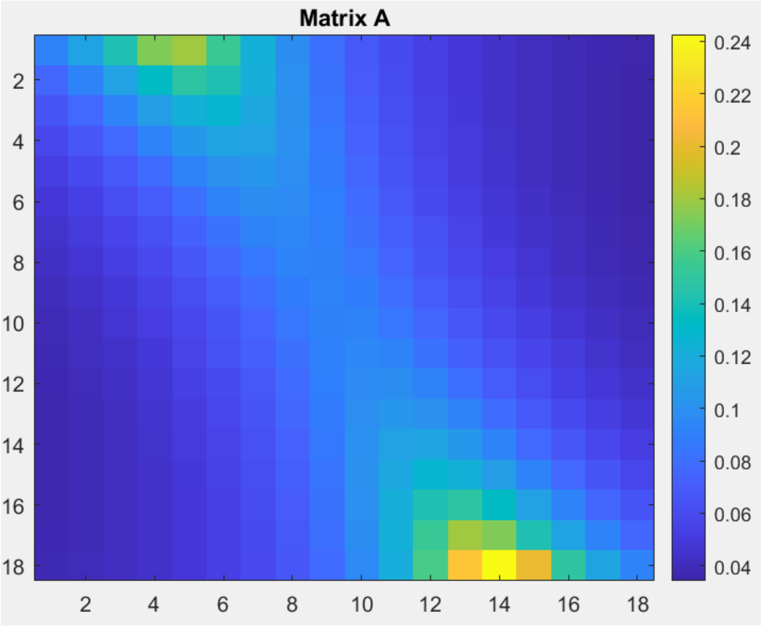
***Relative Error for** $\bar{\boldsymbol{q}}$: 34.1205*

כפי שראינו בהרצאה, אם A הייתה הפיכה היינו מקבלים את הפתרון המדויק של המערכת והשגיאה הייתה 0, ככל שנקטין את h המטריצה תשאף להיות מטריצה הפיכה והפתרון יהיה יותר מדויק.

ב. כעת בעזרת לולאה נרוץ על ערכי h כך ש- $h \in \left\{10, 5, 2, \frac{1}{2}, \frac{1}{5}\right\} \times \frac{\pi\rho}{M}$.
נצייר על גרף את הדטרמיננטה (בערך מוחלט) ואת שגיאת החישוב היחסית בפתרון כפונקציה של h.

$h = \frac{5\pi\rho}{M}$:

Matrix A:



Absolute Determinant** Value of matrix A:* $2.5204 \cdot 10^{-89}$ ***(Real det value is negative)

***Vector** $\overline{v}$ (Transpose):*

v

1x18 double

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
1	5.8805	5.3235	4.9321	4.6564	4.4621	4.3283	4.2422	4.1965	4.1872	4.2135	4.2774	4.3844	4.5440	4.7723	5.0955	5.5577	6.2378	7.2823

$\overline{q}$ *as computed* $(A^T A)^{-1} A^T$ *(Transpose):*

rel_q

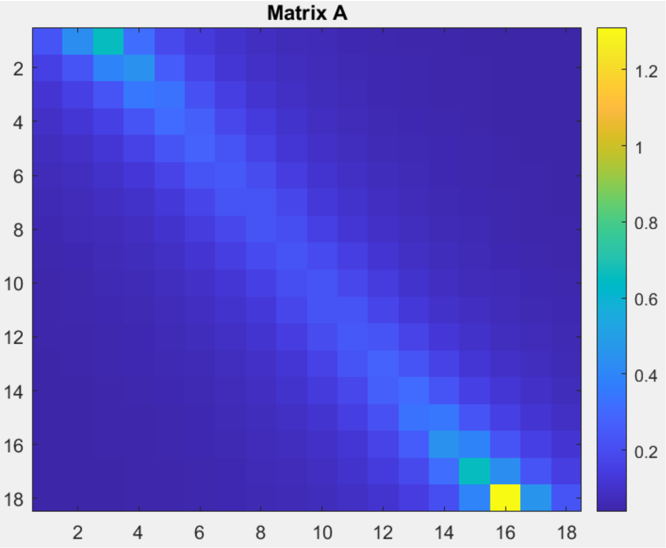
1x18 double

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
1	460.8148	2.9922e+03	-351.8082	-138.8071	-69.0729	-37.9222	5.3427	-5.4897	4.5001	-10.2695	-7.0519	-1.6665	9.2919	-25.8542	-462.2608	-182.2021	-795.4549	137.8024

***Relative Error for** $\overline{q}$: 155.9390*

$h = \frac{2\pi\rho}{M}$:

Matrix A:



Absolute Determinant** Value of matrix A:* $3.4450 \cdot 10^{-36}$ ***(Real det. value is negative)

***Vector** $\overline{v}$ (Transpose):*

rel_err | rel_q | A | det_A_plot | v

1x18 double

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
1	8.5402	8.5239	8.5360	8.0392	7.4376	7.2097	7.1800	7.2549	7.3156	7.2416	7.1317	7.1705	7.4808	8.0964	9.0998	9.9133	9.6065	12.9720

$\overline{q}$ *as computed* $(A^T A)^{-1} A^T$ *(Transpose):*

rel_err | rel_q | A | det_A_plot | v

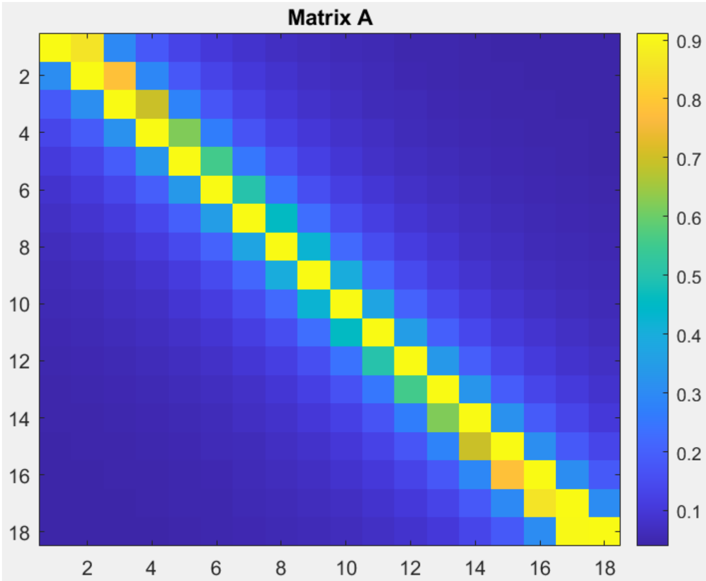
1x18 double

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
1	-23.5748	5.3987	3.1509	3.3244	8.7676	-0.2827	4.2555	1.6761	9.2890	2.9265	0.9994	2.8220	2.6035	9.4588	0.0842	3.1328	23.3503	-4.3534

Relative Error for $\overline{q}$: *1.8004*

$h = \frac{\frac{1}{2}\pi\rho}{M}$:

Matrix A:



***Absolute Determinant** Value of matrix A:* 0.0019

***Vector** $\overline{v}$ (Transpose):*

rel_err | rel_q | A | det_A_plot | v

1x18 double

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
1	8.7929	9.4409	11.3122	13.4359	14.5700	10.4631	11.9443	12.1827	15.4333	12.4203	10.4341	11.1933	12.2643	15.6645	12.9103	11.6747	12.3807	14.9190

$\overline{q}$ *as computed* $(A^T A)^{-1} A^T$ *(Transpose):*

rel_err | rel_q | A | det_A_plot | v

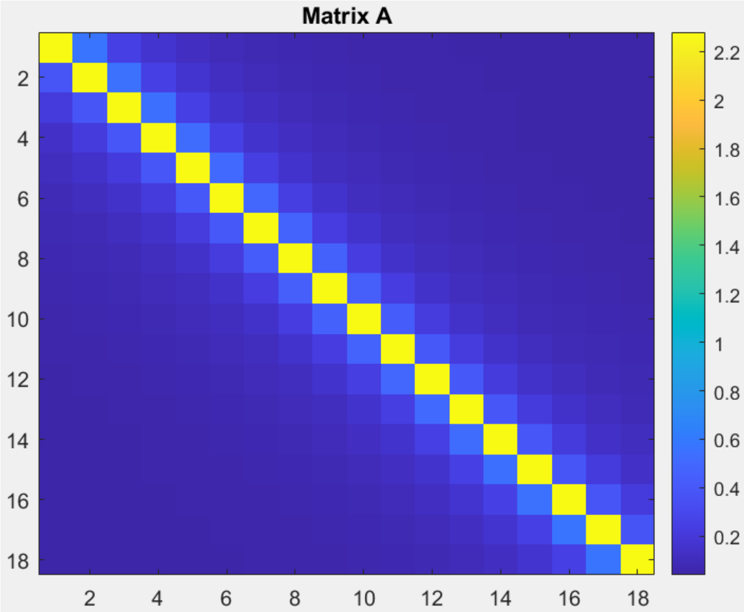
1x18 double

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
1	3.0000	1.0000	3.0000	3.0000	9.0000	2.6118e-14	4.0000	2.0000	9.0000	3.0000	1.0000	3.0000	3.0000	9.0000	-2.0452e-15	4.0000	2.0000	9.0000

Relative Error for $\overline{q}$: $4.054 \cdot 10^{-15}$

$h = \frac{\frac{1}{5}\pi\rho}{M}$:

Matrix A:



Absolute Determinant *Value of matrix A:* $1.2392 \cdot 10^{6}$

Vector $\overline{\boldsymbol{v}}$ *(Transpose):*

rel_err | rel_q | A | det_A_plot | v

1x18 double

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
1	12.2954	10.0688	14.7722	16.9001	27.1071	11.1073	17.6863	15.5895	28.1495	17.0586	12.2257	15.6970	16.9083	27.7601	11.8064	17.0701	14.3981	26.2029

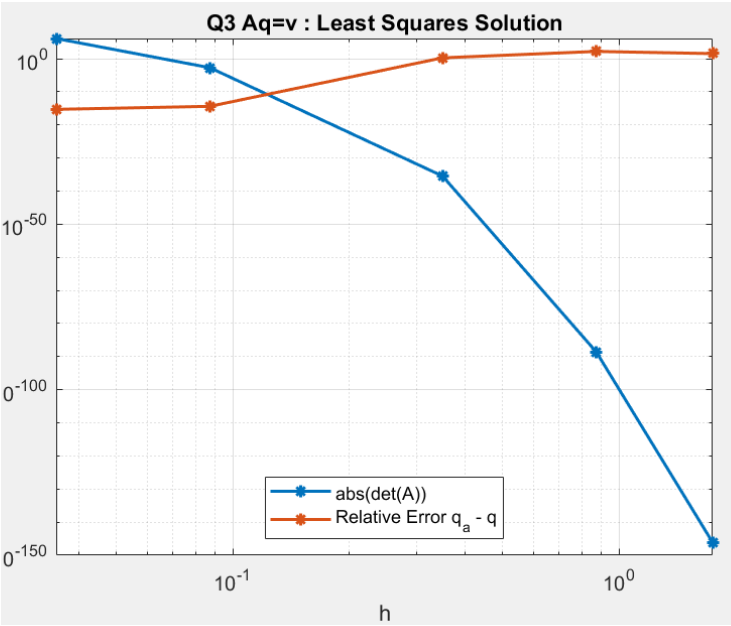
$\overline{\boldsymbol{q}}$ *as computed* $(A^{T}A)^{-1}A^{T}$ *(Transpose):*

rel_err | rel_q | A | det_A_plot | v

1x18 double

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
1	3.0000	1.0000	3.0000	3.0000	9	3.3602e-16	4.0000	2.0000	9.0000	3.0000	1.0000	3.0000	3.0000	9.0000	-5.4351e-16	4	2.0000	9.0000

Relative Error for $\overline{\boldsymbol{q}}$: $4.9571 \cdot 10^{-16}$



נוכל לראות כי ערך הדטרמיננטה של *A* שלילי עבור $h > \frac{\frac{1}{2}\pi\rho}{M}$, לכן ניקח אותו בערכו המוחלט.

נשים לב שקיבלנו כפי שצפינו שנקבל, במצב שבו **נגדיל את *h*** נקבל שכל איבר במטריצה **קטן** והערך המוחלט של הדטרמיננטה **שואף לאפס**, כלומר למטריצה **לא הפיכה**.

מצד שני ככל **שנקטין** את הערך של ***h***, **נגדיל** את ערכו של כל איבר במטריצה, המטריצה *A* שואפת להיות מטריצה **הפיכה** והפתרון שואף **לפתרון המדויק** של המשוואה (שמתקבל עבור *A* סינגולרית בשיטת *Least Squares* כפי שראינו בהרצאה), ניתן לראות זאת גם לפי ייצוג המטריצה *A* בתמונות שקיבלנו.

סוף שאלה 3, קוד MATLAB מצורף במלואו בנספח (עמ' 34).

נספח – קוד מאטלב

Task 1 - Gaussian Elimination & LU Decomposition (MATLAB code)

```
% Task 1

funcs = Functions;
%% section e
num_values = [1,2,5,10,20,50];
h_values = zeros(1,6);
cond_list = zeros(1,6);          % condition number for task a
rel_error_list_b = zeros(1,6); % relative errors for task b
rel_error_list_c = zeros(1,6); % relative errors for task c
rel_error_list_d = zeros(1,6); % relative errors for task d

for k = 1:6
    %% section a
    h=(num_values(k)*pi*Rho)/M;
    h_values(k)=h;
    A = funcs.Matrix_A(M,h);
    V = A*q ;
    [L,U,P] = lu(A);
    K=cond(A);                       % condition number K(A)
    cond_list(k)=cond(A);
    q_norm = norm(q);                % 2-norm of q (p=2).
    v_norm = norm(V);                % 2-norm of V.
    A_fb_norm = norm(A,"fro"); % Frobenius norm of matrix A.
    %% section b
    % we have Aq=V -> LUq=V -> (2) Uq=y -> (1) Ly=V
    y_b = Ly_b(L,P*V);    % calculate y in Ly=V (1)
    q_b = Ux_y(U,y_b);    % calculate q in Uq=y (2)
    rel_q_b_error = funcs.rel_error(q,q_b',2); % calculate relative error.
    rel_error_list_b(k) = rel_q_b_error; % save for graph.

    %% section c
    delta_v=1e-3.*v_norm;
    new_v = V + delta_v;

    y_c = Ly_b(L,P*new_v);    % calculate y in Ly=V (1)
    q_c = Ux_y(U,y_c);        % calculate q in Uq=y (2)
    rel_q_c_error=funcs.rel_error(q,q_c',2); % calculate relative error.
    rel_error_list_c(k) = rel_q_c_error;

    %% section d
    delta_A=A_fb_norm.*1e-3;
    A_d = delta_A + A;
    [L_d,U_d,P_d] = lu(A_d);

    y_d = Ly_b(L_d,P*V);
    q_d = Ux_y(U_d,y_d);
    rel_q_d_error=funcs.rel_error(q,q_d',2);
    rel_error_list_d(k) = rel_q_d_error;
end

% graphs for section e
figure(1);
subplot(2,1,1);
loglog(h_values,rel_error_list_d,LineWidth=1.5); % A
hold on;
loglog(h_values,rel_error_list_c,LineWidth=1.5); % V
loglog(h_values,rel_error_list_b,LineWidth=1.5); % LU
legend('Relative Error _[_A_]','Relative Error _[_V_]','Relative Error
_[_L_U_]',Location='northwest')
hold off
xlabel('h_i')
```

```
ylabel('Relative Error')
grid on;
title('Relative Error');

subplot(2,1,2);
loglog(h_values,cond_list,LineWidth=1.5,Color='#7E2F8E');
xlabel('h_i')
ylabel('K(A)')
legend('K(A)',Location='northwest')
grid on;
title('Condition Number K(A)');

movegui(1,"southwest");
sgtitle('Question 1: Gaussian Elimination & LU Decomposition')

%% Functions
function y = Ly_b(L,b)  % Solving Lower Triangular System Ly=b.
M = length(b);
y = zeros(1,M);     % result vector
y(1) = b(1)/L(1,1);
sum = 0;            % temp for sigma
for k = 2:M
    sum = b(k);
    for i = 1:k-1
        sum = sum - L(k,i)*y(i);
    end
    y(k)=sum/L(k,k);
    sum=0;
end
end

function x = Ux_y(U,y) % Solving Upper Triangular System Ux=y.
M = length(U);
x = zeros(1,M);    % solution vector
x(M) = y(M)/U(M,M);
sum = 0;           % temp for sigma
for r = M-1:-1:1
    sum = y(r);
    for i = M:-1:r
        sum = sum-U(r,i)*x(i);
    end
    x(r) = sum/U(r,r);
    sum = 0;
end
end
```

Task 2: Iteration Solutions (MATLAB code)

```
%% Task 2

funcs = Functions;
Rho=1;

iter_list_5M = zeros;
dist_error_5M = zeros;
rel_error_5M = zeros;
iter_list_2M = zeros;
dist_error_2M = zeros;
rel_error_2M = zeros;
iter_list_1M = zeros;
dist_error_1M = zeros;
rel_error_1M = zeros;

%% task a + b
num_values_2 = [5,2,1];
for i = 1:3
    h = (pi*Rho)/(num_values_2(i)*M);
    A = funcs.Matrix_A(M,h);
    v = A*q;
    [iter_list,dist_error,rel_error] = funcs.Gauss_Seidel(A,q,v);

    if i==1
        iter_list_5M = iter_list;
        dist_error_5M = dist_error;
        rel_error_5M = rel_error;
    end

    if i==2
        iter_list_2M = iter_list;
        dist_error_2M = dist_error;
        rel_error_2M = rel_error;
    end
    if i ==3
        iter_list_1M = iter_list;
        dist_error_1M = dist_error;
        rel_error_1M = rel_error;
    end

end

%% task c
h=(pi*Rho)/(5*M);
A=funcs.Matrix_A(M,h);
v=A*q;
[iter_list_c,dist_error_c,rel_error_c] = funcs.Jacobi(A,q,v);



%% task d
A_d = funcs.Matrix_A_task2(M,h);
v=A_d*q;
[iter_list_d,dist_error_d,rel_error_d] = funcs.Jacobi(A_d,q,v);

Q2 = figure('Renderer', 'painters', 'Position', [13 11 700 550]);
hold on
subplot(13,11,[1,38]) % Gauss-Seidel 5M
semilogy(iter_list_5M,dist_error_5M,iter_list_5M,rel_error_5M);
legend('q^(^k^)-q^(^k^-^1^)','q^(^k^)-q',Location='southwest');
title('G-S Relative Errors (5M)');
```

```
ylabel('Relative Errors');
xlabel('Iterations');
xlim([1,iter_list_5M(end)])

subplot(13,11,[7,44]) % Gauss-Seidel 2M
semilogy(iter_list_2M,dist_error_2M,iter_list_2M,rel_error_2M);
legend('q^(^k^)-q^(^k^-^1^)','q^(^k^)-q',Location='southwest');
ylabel('Relative Errors');
xlabel('Iterations');
title('G-S Relative Errors (2M)');
xlim([1,iter_list_2M(end)])

subplot(13,11,[59,85]) % Gauss-Seidel 1M
semilogy(iter_list_1M,rel_error_1M,iter_list_1M,dist_error_1M);
ylabel('Relative Errors');
xlabel('Iterations');
legend('q^(^k^)-q','q^(^k^)-q^(^k^-^1^)',Location='east');
title('G-S Relative Errors (1M)')


subplot(13,11,[100,137]) % Jacobi (c)
semilogy(iter_list_c,rel_error_c,iter_list_c,dist_error_c);
ylabel('Relative Errors');
xlabel('Iterations');
legend('q^(^k^)-q','q^(^k^)-q^(^k^-^1^)',Location='northwest');
title('Jacobi Relative Errors (5M)_C')


subplot(13,11,[106,143]) % Jacobi (d)
semilogy(iter_list_d,dist_error_d,iter_list_d,rel_error_d);
legend('q^(^k^)-q^(^k^-^1^)','q^(^k^)-q',Location='northeast');
title('Jacobi Relative Errors (5M)_D');
ylabel('Relative Errors');
xlabel('Iterations');

sgtitle('Question 2: Iteration Solutions')
movegui(Q2,"north");
```

Task 3: Least Squares Solutions (MATLAB code)

```
%% Task 3

funcs = Functions;

%% task a+b
num_values = [10,5,2,1/2,1/5];
det_A_plot=zeros(1,5);
rel_err=zeros(1,5);
h_values = zeros(1,5);
for i = 1:5
    h=(num_values(i)*pi*Rho)/M;
    h_values(i)=h;
    A = funcs.Matrix_A(M,h);
    det_A_plot(i) = abs(det(A));
    v=A*q;
    A_T = transpose(A);
    pseudo_inv=inv(A_T*A)*A_T;
    rel_q = pseudo_inv * v;
    rel_err(i)=funcs.rel_error(q,rel_q,2);
end

figure(3);
subplot(2,1,1);
loglog(h_values,det_A_plot,'*-',LineWidth=1.5,Color='#77AC30');
title('Q3 Least Squares Solution: Abs(Det(A))');
ylabel('Abs(Det(A))')
xlabel('h')
legend('Relative Error q_a - q',Location='south')
grid on;

subplot(2,1,2);
loglog(h_values,rel_err,'*-',LineWidth=1.5,Color='#D95319');
title('Q3 Least Squares Solution: Relative Error');
xlabel('h')
ylabel('Relative Error')
legend('Relative Error q_a - q',Location='south')
grid on;

movegui(3,"southeast");
sgtitle('Question 3: Least Squares Solutions')
```

Functions (MATLAB code)

```
classdef Functions
    % This file is for functions used in all 3 Tasks.
    methods
        function res = Matrix_A(~,M,h)
            % create matrix A with given M (size) and h (changes values).
            res = zeros(M);
            Rho = 1;
            for m = 1:M
                for n = 1:M
                    rad = sqrt((h+Rho*sin((m*pi)/M)-Rho*sin((n*pi)/M)).^2+
...
                        (Rho*cos((m*pi)/M)-Rho*cos((n*pi)/M)).^2);
                    res(m,n) = 1 ./ (4*pi*rad);
                end
            end
        end

        function res = Matrix_A_task2(~,M,h) % create matrix A with given M
(size) and h (changes values).
            res = zeros(M);
            Rho = 1;
            for m = 1:M
                for n=1:M
                    rad = (h+Rho*sin((m*pi)/M)-
Rho*sin((n*pi)/M)).^2+(Rho*cos((m*pi)/M)-Rho*cos((n*pi)/M)).^2;
                    res(m,n)=1 ./ (4*pi*rad);
                end
            end
        end

        function res = abs_error(~,e,e_t,p) % get absolute error.
            res = norm(e-e_t,p);
        end

        function [res,per] = rel_error(~,e,e_t,p) % get relative error.
            res = norm(e-e_t,p)/norm(e,p); % relative error.
            per = 100 * res; % percent error.
        end

        function [iter_track,dist_error,real_error] = Gauss_Seidel(~,A,x,b)
            tol = 1e-3;
            L = tril(A,-1);     % Lower triangular matrix
            U = triu(A,1);      % Upper triangular matrix
            D = diag(diag(A)); % Diagonal matrix
            Q = L+D;
            G = -inv(Q)*U;
            c = inv(Q)*b;
            itr = 1; max_iter = 600;
            iter_track = ones;
            q_k = c;            % q^(0) = 0 , q^(1) = c
            dist_error = ones; % ||q^(1)-q^(0)|| / ||q^(0)|| (can't divide
by 0)
            real_error = ones; % ||q^(1)-q|| / ||q||
            real_error(1) = norm(c-x,2)/norm(x);
            error = norm(x-q_k(:, itr),"inf")/norm(x,"inf");
            while abs(error)>tol && itr<=max_iter        % make sure we
stop.
                q_k(:,itr+1) = G*q_k(:,itr) + c;                  % Gauss-
Seidel Algorithm.
                error = norm(x-q_k(:, itr),"inf")/norm(x,"inf"); % finding
error ||q^k-q||
```