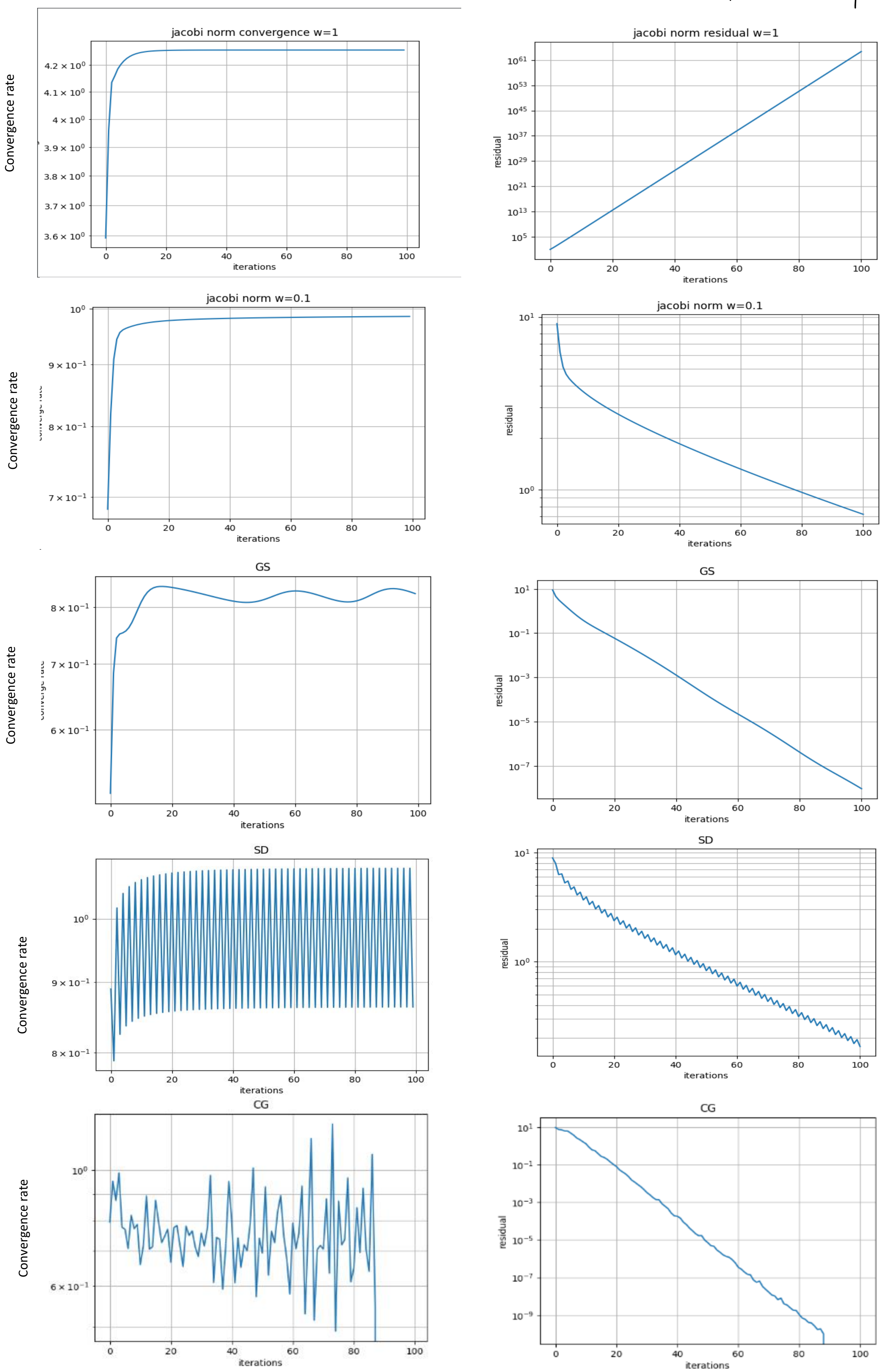
א. קוד מצורף

ב. להלן הגרפים שהתקבלו:

1.



2

a. נראה שלכל A חיובית ממש ש- $x^{(k+1)} = x^{(k)} + \frac{1}{\|A\|}(b - Ax^{(k)})$

מתכנס לפתרון עבור $Ax=b$.

ראינו שבשיטה מתכנסת כאשר $\lim_{k\to\infty} \frac{\|e^{(k+1)}\|}{\|e^{(k)}\|} \to 0$ וזה קורה

כאשר היחס קטן מ-1. למדנו בכיתה ש- $\rho(I-\alpha A) = \lim_{k\to\infty} \frac{\|e^{(k+1)}\|}{\|e^{(k)}\|}$

בשיטת ריצ'רדסון $\alpha = \frac{1}{\|A\|}$, כלומר נראה $1 > \rho(I - \frac{1}{\|A\|}A)$.

$$\rho(I - \frac{1}{\|A\|}A) = \max\{|1 - \frac{1}{\|A\|}\lambda_{min}|, |1 - \frac{1}{\|A\|}\lambda_{max}|\}$$

מכיוון ש-A spd אז כל הע"ע חיוביים כלומר נקבל $\rho(I - \frac{1}{\|A\|}A) < 1$.

b. באותו אופן כמו מקודם נגיע ל-:

$$\rho(I - \frac{1}{\|A\|}A) = \max\{|1 - \frac{1}{\|A\|}\lambda_{min}|, |1 - \frac{1}{\|A\|}\lambda_{max}|\}$$

אך כאשר ל-A יש ע"ע שליליים נקבל $\max\{,\}$ שגדול מ-1 [illegible]

ולכן השיטה לא תתכנס.

c. I. נראה תחילה ש- $f(x^{(k)}) - \frac{1}{2}\frac{\langle r^{(k)}, Ae^{(k)}\rangle^2}{\langle r^{(k)}, Ar^{(k)}\rangle} < f(x^{(k)})$

כלומר נראה ש- $\frac{1}{2}\frac{\langle r^{(k)}, Ae^{(k)}\rangle^2}{\langle r^{(k)}, Ar^{(k)}\rangle} > 0$, נשים לב כי המונה

בחזקה שנייה כלומר חיובי, כמו כן המכנה חיובי מכיוון ש- $A \succ 0$.

נכון ש- $f(x^{(k+1)}) = f(x^{(k)}) - \frac{1}{2}\frac{\langle r^{(k)}, Ae^{(k)}\rangle^2}{\langle r^{(k)}, Ar^{(k)}\rangle}$

$$f(x^k) = \frac{1}{2}\|x^* - x^k\|_A^2 = \frac{1}{2}\|e^k\|_A^2 = \frac{1}{2}e^{k^T}Ae^k$$

כמו כן, ראינו בכיתה:

$$f(x^{k+1}) = f(x^k + \alpha^k r^k) = \dots = \frac{1}{2}e^{k^T}Ae^k - \alpha^k r^{k^T}Ae^k + \frac{1}{2}(\alpha^k)^2 \cdot r^{k^T}Ar^k$$

עבור $\alpha^k = \alpha^k_{opt} = \frac{r^{k^T}Ae^k}{r^{k^T}Ar^k} = \frac{\langle r^k, Ae^k\rangle}{\langle r^k, Ar^k\rangle}$

נציב α^k במשוואה ממקודם ונקבל:

$$f(x^{k+1}) = \frac{1}{2}e^{k^T}Ae^k - \frac{\langle r^k, Ae^k\rangle}{\langle r^k, Ar^k\rangle}\langle r^k, Ae^k\rangle + \frac{1}{2}\left(\frac{\langle r^k, Ae^k\rangle}{\langle r^k, Ar^k\rangle}\right)^2 \cdot \langle r^k, Ar^k\rangle$$

$$= \underbrace{\frac{1}{2}e^{k^T}Ae^k}_{f(x^k)} - \frac{\langle r^k, Ae^k\rangle^2}{\langle r^k, Ar^k\rangle} + \frac{1}{2}\frac{\langle r^k, Ae^k\rangle^2}{\langle r^k, Ar^k\rangle}$$

$$= f(x^k) - \frac{1}{2}\frac{\langle r^k, Ae^k\rangle^2}{\langle r^k, Ar^k\rangle}.$$

II

$$f(x^{k+1}) = c^k f(x^k)$$

$$c^k = \frac{f(x^{k+1})}{f(x^k)} = \frac{f(x^k) - \frac{1}{2}\frac{\langle r^k, Ae^k\rangle^2}{\langle r^k, Ar^k\rangle}}{f(x^k)} = 1 - \frac{1}{2}\frac{\langle r^k, Ae^k\rangle^2}{f(x^k)\cdot\langle r^k, Ar^k\rangle}$$

$$= 1 - \frac{\langle r^k, Ae^k\rangle^2}{e^{k^T}Ae^k\langle r^k, Ar^k\rangle}$$

כי נכון ש- $0 < \frac{\langle r^k, Ae^k\rangle^2}{\langle e^k, Ae^k\rangle\langle r^k, Ar^k\rangle}$ ולכן $c^k < 1$).

שוב המונה בחזקה שנייה לכן חיובי, A spd לכן $\|x\|_A$ חיובי לכל x.

III. בהינתן שלכל A סימטרית מתקיים: $\forall v \in \mathbb{R}^n : \lambda_{min} \le \frac{v^T A v}{v^T v} \le \lambda_{max}$

$$C^k = 1 - \frac{\langle r^k, Ae^k \rangle^2}{\langle e^k, Ae^k \rangle \langle r^k, Ar^k \rangle} = 1 - \frac{\left(r^{k^T} A e^k\right)^2}{r^{k^T} A r^k e^{k^T} A e^k} = 1 - \frac{\left(r^{k^T} r^k\right)^2}{\left(r^{k^T} A r^k\right)\left(e^{k^T} A e^k\right)}$$

$$= 1 - \frac{r^{k^T} r^k r^{k^T} r^k}{r^{k^T} A r^k e^{k^T} A e^k} \overset{\substack{\text{* נניח כי} \\ \text{A הפיכה}}}{\underset{e^k = A^{-1} r^k}{=\!=}} 1 - \frac{r^{k^T} r^k r^{k^T} r^k}{r^{k^T} A r^k \left(A^{-1} r^k\right)^T r^k} = 1 - \frac{r^{k^T} r^k}{r^{k^T} A r^k} \cdot \frac{r^{k^T} r^k}{r^{k^T} (A^{-1})^T r^k}$$

לכן:

$$C^k \le 1 - \frac{\lambda_{min}}{\lambda_{max}} \iff \frac{r^{k^T} r^k}{r^{k^T} A r^k} \cdot \frac{r^{k^T} r^k}{r^{k^T} (A^{-1})^T r^k} \ge \frac{\lambda_{min}}{\lambda_{max}}$$

$$\iff \frac{r^{k^T} A r^k}{r^{k^T} r^k} \cdot \frac{r^{k^T} (A^{-1})^T r^k}{r^{k^T} r^k} \le \frac{\lambda_{max}}{\lambda_{min}}$$

נניח כי A סימטרית, אז מתקיים: $\frac{r^{k^T} A r^k}{r^{k^T} r^k} \le \lambda_{max}$

A סימטרית $\Leftarrow$ A^{-1} סימטרית, לכן: $\frac{r^{k^T} (A^{-1})^T r^k}{r^{k^T} r^k} \le \underset{\text{ע״ע מקס׳ של } A^{-1}}{\lambda'_{max}} = \frac{1}{\lambda_{min}}$

לכן $C^k \le 1 - \frac{\lambda_{max}}{\lambda_{min}} < 1 \Leftarrow \frac{r^{k^T} A r^k}{r^{k^T} r^k} \cdot \frac{r^{k^T} (A^{-1})^T r^k}{r^{k^T} r^k} \le \frac{\lambda_{max}}{\lambda_{min}}$.

iv. נראה ש- $\lim_{k\to\infty} f(x^k) = 0$

$$f(x^{k+1}) = c^k f(x^k) \xrightarrow[k\to\infty]{0<c^k<1} 0$$

$$\lim_{k\to\infty} f(x^k) = \lim_{k\to\infty} \frac{1}{2}\|x^* - x^k\|_A^2 = 0 \iff x^* - x^k = 0$$

$x^* = x^k$ כאשר $k\to\infty$

$$x^{k+1} = x^k + \alpha^k (b - Ax^k) = x^k + \alpha^k r^k$$ ③

אנחנו מחפשים α^k כך ש- $\|r^k\|_2$ מינימלי.

נגדיר $f = \underset{\alpha^k}{\text{argmin}} \|r^{k+1}\|_2$:

$$\|r^{k+1}\|_2 = \|b - Ax^{k+1}\|_2 = \|b - A(x^k + \alpha^k r^k)\|_2$$

$$= \|b - Ax^k - A\alpha^k r^k\|_2 = \|r^k - \alpha^k A r^k\|_2 = (r^k - \alpha^k A r^k)^T (r^k - \alpha^k A r^k)$$

$r^{k^T} - \alpha^k r^{k^T} A^T$

$$= r^{k^T} r^k - \alpha^k r^{k^T} A r^k - \alpha^k r^{k^T} A^T r^k + (\alpha^k)^2 r^{k^T} A^T A r^k$$

נבצע מינימיזציה לביטוי שקיבלנו:

$$\frac{\partial f}{\partial \alpha^k} = -r^{k^T} A r^k - r^{k^T} A^T r^k + 2\alpha^k r^{k^T} A^T A r^k = 0$$

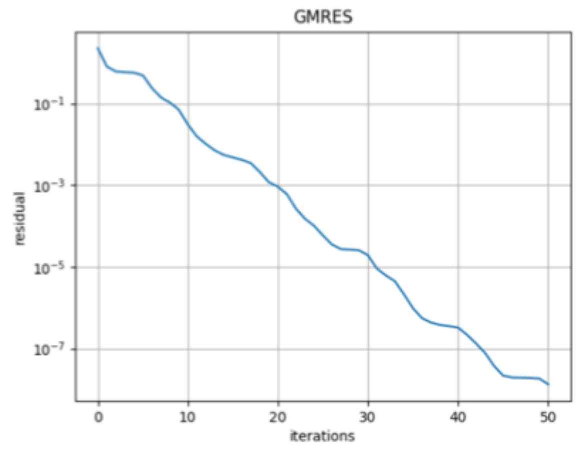
$$r^{k^T} A r^k + r^{k^T} A^T r^k = 2\alpha^k r^{k^T} A^T A r^k$$

$$\langle r^k, A r^k \rangle + \langle A r^k, r^k \rangle = 2\alpha^k r^{k^T} A^T A r^k$$

$$\langle r^k, A r^k \rangle = \alpha^k r^{k^T} A^T A r^k$$

$$\alpha^k = \frac{r^{k^T} A r^k}{r^{k^T} A^T A r^k}$$

ב. להלן הגרף שהתקבל לאחר 50 איטרציות:



ג. באופן לא מפתיע קיבלנו גרף מונוטוני יורד, תופעה זו מתרחשת מכיוון שב- GMRES בכל איטרציה אנחנו בוחרים x^k כך שיבצע מינימיזציה לפי השארית, כמו כן בגרף אנחנו מציגים את השארית במהלך האיטרציות, ולכן כצפוי הגרף מונוטוני יורד.

ב. $$X^{(K+1)} = X^{(K)} + \alpha_1^{(K)} r^K + \alpha_2^{(K)} r^K$$

נסמן $\vec{\alpha}^K = [\alpha_1^K, \alpha_2^K]$, נבצע מינימיזציה באופן הק-1

$$r^{K+1} = b - Ax^{(K+1)} = (b - A(x^{(K)} + \alpha_1^{(K)} r^K + \alpha_2^{(K)} r^K)$$

$$= b - A(x^{(K)} + R^{(K)}\vec{\alpha}^K) = b - Ax^K - AR^K\vec{\alpha}^K$$

$$f(\vec{\alpha}^K) = \| b - Ax^K - AR^K\vec{\alpha}^K \|_2^2 = \| r^K - AR^K\vec{\alpha}^K \|_2^2$$

$$= (r^{K^T} - \vec{\alpha}^{K^T} R^{K^T} A^T)(r^K - AR^K\vec{\alpha}^K) =$$

$$r^{K^T} r^K - r^{K^T} AR^K \vec{\alpha}^K - \vec{\alpha}^{K^T} R^{K^T} A^T r^K + \vec{\alpha}^{K^T} R^{K^T} A^T A R^K \vec{\alpha}^K$$

$$\langle r^K, AR^K \rangle \quad \langle AR^K, r^K \rangle$$

$$f'(\vec{\alpha}^K) = -r^{K^T} AR^K - R^{K^T} A^T r^K + 2R^{K^T} A^T A R^K \vec{\alpha}^K =$$

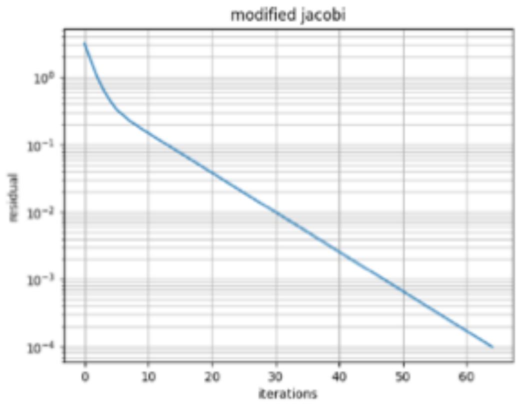
נחפש מינימום:

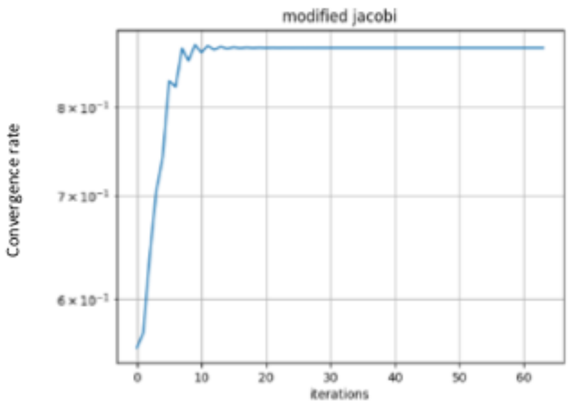
$$-2r^{K^T} AR^K + 2R^{K^T} A^T A R^K \vec{\alpha}^K = 0$$

$$r^{K^T} AR^K = R^{K^T} A^T A R^K \vec{\alpha}^K$$

$$\vec{\alpha}^K = (R^{K^T} A^T A R^K)^{-1} r^{K^T} AR^K$$

.4א



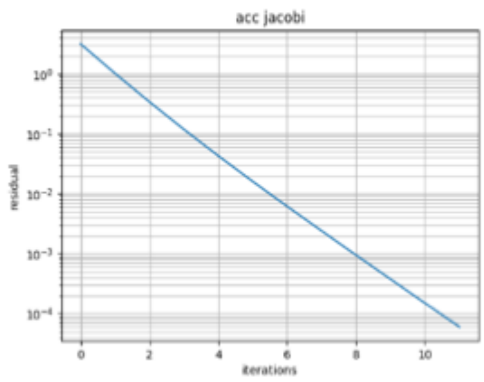


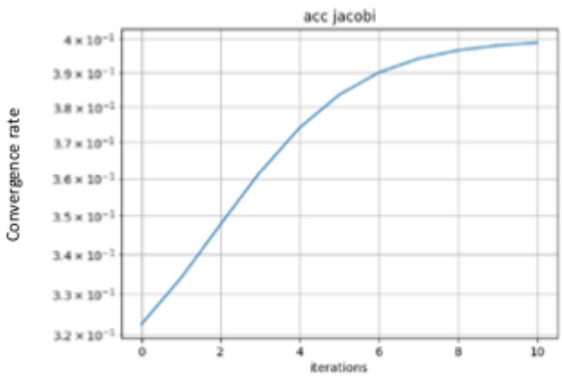
הפתרון:

```
[ 1.24949057  0.5828239   0.91620166 -0.08365973 -0.09119432 -0.5930786
 -0.13854332 -0.54952048  0.27052652 -0.37529417]
```

נדרשו 64 איטרציות להגיע לפתרון זה.

.4ב



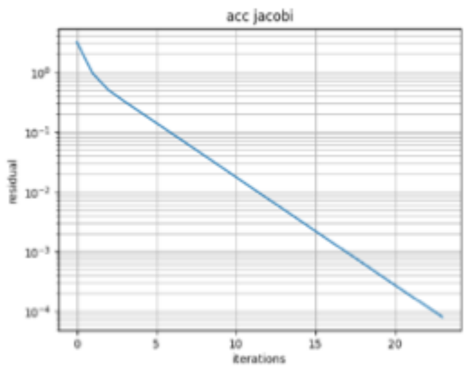


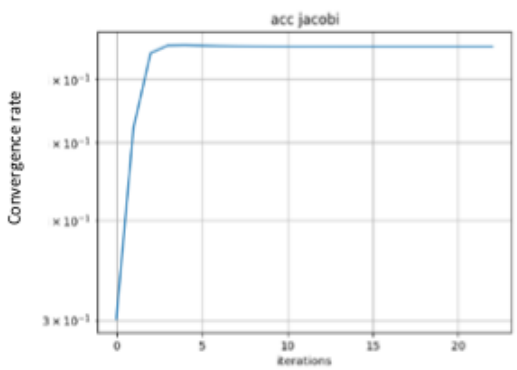
הפתרון:

```
[ 0.83335371  0.16668823  0.50002097 -0.50002097 -0.50759672 -1.00948977
 -0.55494512 -0.96592924 -0.14585493 -0.79168712]
```

נדרשו 11 איטרציות להגיע לפיתרון זה.

.4ג





הפיתרון:

```
[ 1.42285529  0.75618862  1.08952195  0.08953865  0.08199349 -0.4198928
  0.03464168 -0.37633266  0.44370399 -0.20211398]
```

נדרשו 23 איטרציות להגיע לפיתרון זה.

:קוד מצורף

```
import numpy
import numpy as np
import matplotlib.pyplot as plot
import scipy.sparse.linalg
from scipy.sparse import random
import scipy.sparse as sparse
from scipy.linalg import block_diag

# 1
def Jacobi_method(A, b, N, x,w):
    # start x_0
    if x is None:
        x = np.zeros(len(A[0]))
    nr2 = np.zeros(N + 1)
    nr2[0] = np.linalg.norm(A @ x - b)

    # create D
    D = np.diag(A)
    LU = A - np.diagflat(D)

    for i in range(N):
        x = w*(b - (LU) @ x) / D + (1-w)*x
        nr2[i + 1] = np.linalg.norm(A @ x - b)

    return x, nr2, convergence_factor(nr2)


def GS(A, b, N, x):
    # start x_0
    if x is None:
        x = np.zeros(len(A[0]))
    nr2 = np.zeros(N + 1)
    nr2[0] = np.linalg.norm(A @ x - b)

    # create L+D
    LD = numpy.tril(A)

    for i in range(N):
        x = x + np.linalg.inv(LD) @ (b - A @ x)
        nr2[i + 1] = np.linalg.norm(A @ x - b)

    return x, nr2, convergence_factor(nr2)


def SD(A, b, N, x):
    if x is None:
        x = np.zeros(np.shape(A[0]))

    nr2 = np.zeros(N + 1)
    nr2[0] = np.linalg.norm(b)

    for i in range(N):
        r = b - A @ x
        if np.linalg.norm(r) <= pow(10, -10):
            break
        alpha = np.dot(r, r) / np.dot(r, A @ r)
        x = x + alpha * r
        nr2[i + 1] = np.linalg.norm(A @ x - b)

    return x, nr2, convergence_factor(nr2)
```

```python
def CG(A, b, N, x):
    if x is None:
        x = np.zeros(np.shape(A[0]))

    nr2 = np.zeros(N + 1)
    nr2[0] = np.linalg.norm(b)
    r = b - A @ x
    p = r.copy()
    for i in range(N):
        Ap = A.dot(p)
        alpha = np.dot(p, r) / np.dot(p, Ap)
        x = x + alpha * p
        nr2[i + 1] = np.linalg.norm(A @ x - b)
        r = b - A.dot(x)
        if np.linalg.norm(r) <= pow(10, -10):
            break
        beta = -np.dot(r, Ap) / np.dot(p, Ap)
        p = r + beta * p
    return x, nr2, convergence_factor(nr2)


def show_results(xlabel, ylabel, title, aray):
    plot.grid(True, which="both")
    plot.semilogy(np.array(range(0, len(aray))), aray)
    plot.title(title)
    plot.ylabel(ylabel)
    plot.xlabel(xlabel)
    plot.show()


def convergence_factor(nr2):
    con_fac = np.zeros((len(nr2)) - 1)
    for i in range(len(con_fac)):
        if nr2[i] == 0:
            break
        con_fac[i] = nr2[i + 1] / nr2[i]

    return con_fac


# 3 c
def GMRES (A, b, N, x):

    nr2 = np.zeros(N + 1)
    nr2[0] = np.linalg.norm(b)

    for i in range(N):
        r = b - A @ x
        a1 = np.transpose(r)@A@r
        a2 = np.transpose(r)@np.transpose(A)@A@r
        alpha = a1/a2
        x = x + alpha * r
        nr2[i + 1] = np.linalg.norm(A @ x - b)

    return x, nr2
```

```python
# 4 a
def mod_Jacobi_method(A, b, e,x):
    # start x_0
    if x is None:
        x = np.zeros(len(A[0]))
    nr2 = []
    nr2.append(np.linalg.norm(A @ x - b))

    # create D
    D = np.diag(A)
    LU = A - np.diagflat(D)
    r = np.linalg.norm(A @ x - b)
    counter = 0
    while np.linalg.norm(r) >= e:
        x = (b - (LU) @ x) / D
        r = np.linalg.norm(A @ x - b)
        nr2.append(r)
        counter = counter +1
        print(np.linalg.norm(r))

    return x, nr2,  convergence_factor(nr2), counter

# 4 b
def acc_jacobi(A, b,M, e,x,w):
    # start x_0
    if x is None:
        x = np.zeros(len(A[0]))
    nr2 = []
    nr2.append(np.linalg.norm(A @ x - b))

    # create D
    M_inv = np.linalg.inv(M)
    r = np.linalg.norm(A @ x - b)
    counter = 0
    while np.linalg.norm(r) >= e:
        x = x + w* M_inv @ (b-A@x)
        r = np.linalg.norm(A @ x - b)
        nr2.append(r)
        counter = counter +1
        print(np.linalg.norm(r))

    return x, nr2,  convergence_factor(nr2), counter
```

```python
if __name__ == '__main__':
    # 1
    n = 256
    A = random(n, n, 5 / n, dtype=float)
    v = np.random.rand(n)
    v = sparse.spdiags(v, 0, v.shape[0], v.shape[0], 'csr')
    A = A.transpose() * v * A + 0.1 * sparse.eye(n)
    b = np.random.rand(n)
    A = A.toarray()

    sol, nr2,fr = GS(A, b, 100, None)
    show_results("iterations", "residual", "GS", nr2)
    show_results("iterations", "converge rate", "GS", fr)

    sol, nr2, fr = Jacobi_method(A, b, 100, None, 0.1)
    show_results("iterations", "residual", "jacobi norm w=0.1", nr2)
    show_results("iterations", "converge rate", "jacobi norm w=0.1",
fr)

    sol, nr2, fr = SD(A, b, 100, None)
    show_results("iterations", "residual", "SD", nr2)
    show_results("iterations", "converge rate", "SD", fr)

    sol, nr2, fr = CG(A, b, 100, None)
    show_results("iterations", "residual", "CG", nr2)
    show_results("iterations", "converge rate", "CG", fr)


    # 3
    A = [[5,4,4,-1,0], [3,12,4,-5,-5], [-4,2,6,0,3] , [4,5,-7,10,2] ,
[1,2,5,3,10]]
    b = [1,1,1,1,1]
    x = np.zeros(5)

    sol,nr2 = GMRES(A,b,50,x)

    show_results("iterations", "residual", "GMRES", nr2)
    print(sol)

    # 4 a

    A=[[2,-1,-1,0,0,0,0,0,0,0] , [-1,2,-1,0,0,0,0,0,0,0] , [-1,-1,3,-
1,0,0,0,0,0,0] , [0,0,-1,5,-1,0,-1,0,-1,-1] , [0,0,0,-1,4,-1,-1,-
1,0,0] ,[0,0,0,0,-1,3,-1,-1,0,0] , [0,0,0,-1,-1,-1,5,-1,0,-1]
,[0,0,0,0,-1,-1,-1,4,0,-1] , [0,0,0,-1,0,0,0,0,2,-1] , [0,0,0,-
1,0,0,-1,-1,-1,4]]
    b = [1,-1,1,-1,1,-1,1,-1,1,-1]

    sol,nr2,fr,it = mod_Jacobi_method(A,b,10e-5,None)

    show_results("iterations", "residual", "modified jacobi", nr2)
    show_results("iterations", "convergence factor", "modified
jacobi", fr)
    print(sol)
    print(it)

    # 4 b
    M1 = [[2,-1,-1] , [-1,2,-1] , [-1,-1,3]]
    M2 = [[5,-1,0, -1,0,-1,-1] , [-1,4,-1,-1,-1,0,0] ,[0,-1,3,-1,-
1,0,0] ,[-1,-1,-1,5,-1,0,-1] , [0,-1,-1,-1,4,0,-1] ,[-1,0,0,0,0,2,-1]
,[-1,0,0,-1,-1,-1,4]]
```

```
    M = block_diag(M1,M2)


    sol,nr2,fr,it = mod_Jacobi_method(A,b,10e-5,None)

    show_results("iterations", "residual", "modified jacobi", nr2)
    show_results("iterations", "convergence factor", "modified 
jacobi", fr)
    print(sol)
    print(it)


    sol,nr2,fr,it = acc_jacobi(A,b,M,10e-5,None , 0.7)

    show_results("iterations", "residual", "acc jacobi", nr2)
    show_results("iterations", "convergence factor", "acc jacobi", 
fr)
    print(sol)
    print(it)



    # 4c
    M1 = [[2,-1,-1] , [-1,2,-1] , [-1,-1,3]]
    M2 = [[5,-1,0,-1] , [-1,4,-1,-1] ,[0,-1,3,-1] ,[-1,-1,-1,5]]
    M3 = [[4,0,-1] , [0,2,-1] , [-1,-1,4]]
    M = block_diag(M1, M2 , M3)
    sol,nr2,fr,it = acc_jacobi(A,b,M,10e-5,None , 0.7)

    show_results("iterations", "residual", "acc jacobi", nr2)
    show_results("iterations", "convergence factor", "acc jacobi", 
fr)
    print(sol)
    print(it)
```