N1. functions:

```python
def generic_bisect(f, df, l, u, eps, k):
    fv = [f(u)]
    iter = 0
    while abs(u - l) > eps and iter < k:
        m = (l + u) / 2
        if df(m) * df(l) < 0:
            u = m
        else:
            l = m
        fv.append(f(m))
        iter += 1
    return m, fv
```

```python
def generic_newton(f, df, ddf, x0, k):
    fv = [f(x0)]
    iter = 0
    while iter < k:
        x0 = x0 - (df(x0) / ddf(x0))
        fv.append(f(x0))
        iter += 1
    return x0, fv
```

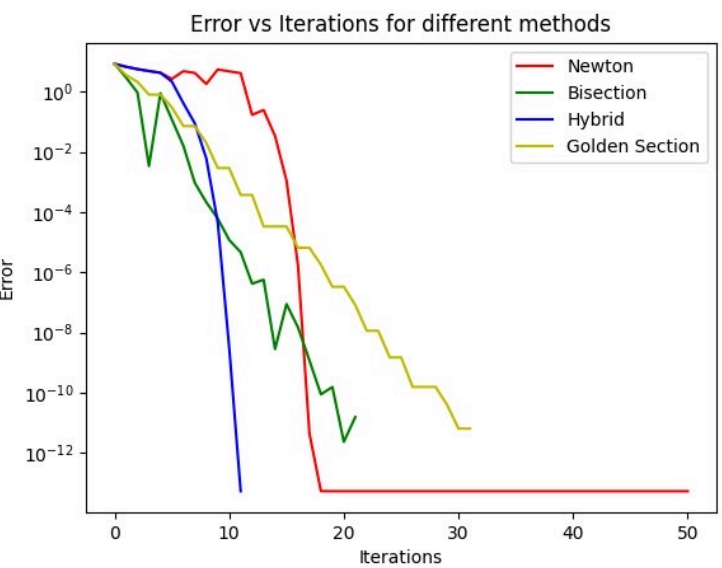
```python
def generic_hybrid(f, df, ddf, l, u, x0, eps, k):
    fv = [f(x0)]
    iter = 0
    x = x0
    while iter < k and abs(u - l) > eps and abs(df(x)) > eps:
        x = x0 - df(x0) / ddf(x0)
        if (x < l or x > u) or abs(df(x)) >= 0.99 * abs(df(x0)):
            x = (l + u) / 2
        x0 = x

        if df(x) * df(l) < 0:
            u = x
        else:
            l = x

        fv.append(f(x))
        iter += 1
    return x, fv
```

```python
def generic_gs(f, l, u, eps, k):
    tau = (3 - 5 ** 0.5) / 2
    fv = [f(u)]
    iter = 0
    x2 = l + (u - l) * tau
    x3 = l + (u - l) * (1 - tau)

    while abs(u - l) > eps and iter < k:
        if f(x2) < f(x3):
            u = x3
            x3 = x2
            x2 = l + (u - l) * tau
        else:
            l = x2
            x2 = x3
            x3 = l + (u - l) * (1 - tau)
        fv.append(f(u))
        iter += 1
    return l, fv
```

main code:

```python
def main():
    k = 50
    l = -0.999
    u = 0.9
    x0 = u
    eps = 1e-6

    def f(x):
        return ((x - 1) ** 3) + (1 / (1 - x ** 2))
    def df(x):
        return 3 * (x - 1) ** 2 + 2 * x / (1 - x ** 2) ** 2
    def ddf(x):
        return 6 * (x - 1) + 2 * (3 * x ** 2 + 1) / (1 - x ** 2) ** 3
    x_star = -3.08891254695156

    fv_newton = generic_newton(f, df, ddf, x0, k)[1]
    fv_newton = [val - x_star for val in fv_newton]
    fv_bisect = generic_bisect(f, df, l, u, eps, k)[1]
    fv_bisect = [val - x_star for val in fv_bisect]
    fv_hybrid = generic_hybrid(f, df, ddf, l, u, x0, eps, k)[1]
    fv_hybrid = [val - x_star for val in fv_hybrid]
    fv_gs = generic_gs(f, l, u, eps, k)[1]
    fv_gs = [val - x_star for val in fv_gs]

    plt.semilogy(range(len(fv_newton)), fv_newton, color='r', label='Newton')
    plt.semilogy(range(len(fv_bisect)), fv_bisect, color='g', label='Bisection')
    plt.semilogy(range(len(fv_hybrid)), fv_hybrid, color='b', label='Hybrid')
    plt.semilogy(range(len(fv_gs)), fv_gs, color='y', label='Golden Section')
    plt.xlabel('Iterations')
    plt.ylabel('Error')
    plt.legend()
    plt.title('Error vs Iterations for different methods')

    plt.show()
```

plot:



1.אנו יכולים לראות ששיטה ההיברידית היא המהירה ביותר.
2.אכן יש הבדל מהותי בין שיטת החצייה לבין שיטת החתך הזהה. שיטת החצייה מתכנסת לפתרון ב10 איטרציות פחות.
3.אפשר גם לראות מהגרף שקצב ההתכנסות של שיטת הניוטון ושיטה ההיברידית הוא הרבה יותר מהיר. מאשר שיטת החתך הזהה ושיטת החצייה.
4.עד איטרציה מסוימת שיטת הניוטון ושיטה ההיברידית אכן מתלכדות וזה משום שעד האיטרציה הזאת שיטת הניוטון נתנה עדכון "טוב יותר" מאשר שיטת החצייה אבל באיזה שהן שלב בחרנו לעדכן את הצעד עם שיטת החצייה וזה הסיבה לפרידה הראשונה.

שאלה 2

נתונה מטריצה $\mathbf{Q}\in\mathbb{R}^{n\times n}$ סימטרית ומוגדרת אי־שלילית ונתון וקטור $\mathbf{0}_n\neq\mathbf{a}\in\mathbb{R}^n$. תהי $f:\mathbb{R}\to\mathbb{R}$ המוגדרת ע"י

$$f(\mu)=\mathbf{a}^T\left(\mathbf{Q}+\mu\mathbf{I}_n\right)^{-1}\mathbf{a}.$$

$$Q=U\Delta U^T$$

א. הוכיחו כי f מוגדרת לכל $\mu>0$.

$$f(\mu)=a^T(Q+\mu I_n)^{-1}a=a^T(U\Delta U^T+\mu\cdot I_n)^{-1}\cdot a=a^T(U(\mu+\Delta)U^T)^{-1}\cdot a$$

$Q\succeq 0 \Leftarrow$ $\forall i: \lambda_i\geq 0$ $\overset{\mu>0}{\Leftarrow}$ $\forall i: \mu+\lambda_i>0$ $\Leftarrow$ $\Delta+\mu$ מדרגה מלאה $\Leftarrow$ $Q+\mu I_n$ מדרגה מלאה

$\Leftarrow$ ה) כי Q מדרגה מלאה ולכן לפי משפט זה הפיכה

$\Leftarrow$ $\exists(Q')^{-1}\in\mathbb{R}^{n\times n}=(Q+\mu I_n)^{-1}$ $\Leftarrow$ $f(\mu)$ ק"מ ■

ב)

ב. הוכיחו כי $f(\mu)>0$ לכל $\mu>0$.

נניח כי $\mu>0$, $Q\succeq 0$ בפרט $Q+\mu I_n\succ 0$ מוגדרת חיובית כפי שהוכחנו בסעיף 4 הקודם.

$\overset{\text{Coe}}{\Leftarrow}$ $Q+\mu I_n$ סימטרית.

נגדיר $y=(Q+\mu I_n)x$, $y\in\mathbb{R}^n$ עבור $x\in\mathbb{R}^n$ כלשהו.

$$y^T(Q+\mu\cdot I_n)^{-1}y=x^T\cdot(Q+\mu\cdot I_n)^T\underbrace{(Q+\mu\cdot I_n)^{-1}\cdot(Q+\mu I_n)}_{I}\cdot x$$

$$\Rightarrow x^T\underbrace{(Q+\mu\cdot I_n)}_{\text{מוגדרת חיובית}}\cdot x>0$$

בפרט נגיע למצב $\exists x\in\mathbb{R}: y=a\in\mathbb{R}^n$

$\Leftarrow$ עבור $\mu>0$ אזי $f(\mu)>0$ ■

ג)

ג. הוכיחו כי f יורדת בתחום $\mu > 0$.

u ו"ע עבור ה-ע"ע של $Q+\mu\cdot I_n$ $\Leftarrow$ $Qu=\lambda u \Rightarrow (Q+\mu\cdot I_n)u = Qu+\mu\cdot u = (\lambda+\mu)u$

$$f(\mu) = a^T\circ u\circ\left(\sum_{i=1}^n \frac{1}{\lambda(Q)+\mu}\right)\circ u^T a$$

$$f(\mu+\alpha) = a^T\circ u\circ\left(\sum_{i=1}^n \frac{1}{\lambda(Q)+\mu+\alpha}\right)\circ u^T$$

$\Leftarrow$ $\begin{cases} \underbrace{\lambda(Q+\mu\cdot I_n)^{-1} = \frac{1}{\lambda(Q+\mu\cdot I_n)} = \frac{1}{\lambda(Q)+\mu}} \\ (Q+\mu\cdot I_n)^{-1} \overset{\text{ש}}{=} u\cdot\sum_{i=1}^n \frac{1}{\lambda(Q)+\mu}\cdot u^T \end{cases}$

u מטריצת הו"ע המתאימים ל-Q

כי המכנה הגדול גדול מהימני כי שניהם חיוביים

$$f(\mu+\alpha)-f(\mu) = a^T\circ u\circ\left(\sum_{i=1}^n \frac{1}{\lambda(Q)+\mu+\alpha}\right)\circ u^T - a^T\circ u\circ\left(\sum_{i=1}^n \frac{1}{\lambda(Q)+\mu}\right)\circ u^T = a^T\cdot \underbrace{u\left(\sum_{i=1}^n \frac{1}{\lambda(Q)+\mu+\alpha} - \frac{1}{\lambda(Q)+\mu}\right)u^T}_{f'} a$$

$\Leftarrow$ מכאן שה-ע"ע של f' שהוא מטריצה סימטרית כך שגדלם קטנים מ-0
ולכן ע"פ משפט עבור מטריצה סימטרית שכל הע"ע שליליים אז:
לכל $x\in\mathbb{R}^n$ $x^T\cdot f' x<0$ ובפרט $f(\mu+\alpha)-f(\mu)<0$

$\Leftarrow$ f יורדת עבור $\mu>0$.

ד. יהי $\alpha\in\mathbb{R}$. אנו מעוניינים להפעיל את שיטת החצייה בשביל למצוא $\mu>0$ המקיים את המשוואה $f(\mu)=\alpha$. מיצאו תנאי על α עבורו בהכרח קיים פתרון למשוואה זו בתחום $\mu>0$. הוכיחו את תשובתכם.

בנוסף, כאשר קיים פתרון למשוואה, הגדירו לולאה המוצאת את קצוות הקטע ההתחלתי ממנו ניתן לאתחל את השיטה.

הכוונה: חלקו את תשובתם לפי סימן הערך העצמי הקטן ביותר של Q.

מכיוון ש-f רציפה ויורדת מספיק למצוא c כך ש-$f(c)>\alpha$ ו-d כך ש-$f(d)<\alpha$

נראה ש-f רציפה. ניתן לרשום $f(\mu)=\sum_{i=1}^n \frac{1}{\lambda(Q)+\mu}\cdot(a^T u_i)^2$ מכיוון שהפונקציה מוגדרת ע"י הרכבה של פונקציות אלמנטריות ומספרים בפרט מתקיים כי f רציפה.

$\Leftarrow$ ניתן להפעיל את שיטת החצייה על f.

אלגוריתם שימצא את קצוות הקטע ההתחלתי: נאתחל $0<c<d$

```
while f(c)<α or f(d)>α
    if f(c)>α: c=c/3
    if f(d)<α: d=3·d
return [c,d]
```

ה. העריכו את כמות האיטרציות הדרושות עד התכנסות השיטה בהסתמך על הקטע שמצאתם בסעיף הקודם.

$$|d^n - c^n| = \frac{1}{2^n}|d^0 - c^0|$$

כאשר תנאי העצירה הוא $|d^n - c^n| < \varepsilon$

$$\Downarrow$$

$$\frac{1}{2^n}|d^0 - c^0| < \varepsilon \Rightarrow \left(\frac{1}{2}\right)^k < \frac{\varepsilon}{|d^0 - c^0|}$$

$$\Downarrow$$

$$k \geq \left\lceil \log_2\left(\frac{d-c}{\varepsilon}\right)\right\rceil$$

ש3.

$$p(x) = -3.55x^3 + 1.1x^2 + 0.765x - 0.74$$

$$x_0 = 0.5554$$

א

נשים לב ש-$p(x)$ היא סכום של פונ' רציפות ולכן רציפה בעצמה.

עבור $x=1$: $0 > p(x)$ $(-3.55+1.1+0.765-0.74<0)$

ועבור $x=-1$: $0 < p(x)$ $(3.55+1.1-0.765-0.74>0)$

לכן ממשפט ערך הביניים ק"מת נקודה $\underline{c \in (-1,1)}$

עבורה : $p(c) = 0$.

∎

ב

עדכון בכל איטרציה k בשיטת ניוטון :

$$x^{k+1} = x^k - \frac{g(x^k)}{g'(x^k)}$$

נתחיל ב- $x_0 = 0.5554$

$$p(x) = -3.55x^3 + 1.1x^2 + 0.765x - 0.74$$

$$p'(x) = -10.65x^2 + 2.2x + 0.765$$

1. $x^1 = x_0 - \frac{p(x_0)}{p'(x_0)} = 0.5554 - \frac{(-0.584)}{(-1.2983)} = 0.1056$

2. $x^2 = x_1 - \frac{p(x_1)}{p'(x_1)} = 0.1056 - \frac{(-0.6511)}{0.8786} = 0.8467$

3. $x^3 = x_2 - \frac{p(x_2)}{p'(x_2)} = 0.8467 - \frac{(-1.4585)}{-5.0073} = 0.5554$

נשים לב שהסדרה מחזור על עצמה כי קיבלנו $x_3 = x_0$

ולכן $\{x^n\}_{n=1}^{\infty}$ תיראה בצורה הבאה:

$x_0 = 0.5554,\ x_1 = 0.1056,\ x_2 = 0.8467,\ x_3 = 0.5554,\ x_4 = 0.1056, \dots$

נסתכל על שני תת-סדרות הבאות:
$$\begin{cases} a_n = \{x_i\},\ i \bmod 3 = 0 \\ b_n = \{x_i\},\ i \bmod 3 = 1 \end{cases}$$

לכן: a_n, b_n - סדרה של קבועים ולכן כל אחת מהן מתכנסת

לאיבר שהיא מכילה. כלומר,
$$a_n \xrightarrow[n\to\infty]{} 0.5554$$
$$b_n \xrightarrow[n\to\infty]{} 0.1056$$

ולכן קיימים שני תת-סדרות שמתכנסות לשני גבולים שונים, לכן

לא קיים הגבול $\lim_{n\to\infty} \{x_i\}_{i=0}^{\infty}$

ג

$$p'(x) = -10.65x^2 + 2.2x + 0.765 = 0 \Rightarrow$$

$$\Rightarrow x_{1,2} = \frac{-2.2 \pm 6.1179}{-21.3} \quad \begin{cases} x_1 = -0.1839 \\ x_2 = 0.3905 \end{cases}$$

נשים לב להגיון של האלגוריתם. ככל שהגרדיאנט גדול יותר

"גודל העדכון" כלומר $x_{k+1}-x_k$ יהיה קטן יותר.

נניח שבאיזה שהוא שלב הגענו ל- x_k שהיא נקודה סטציונרית.

אז לא נוכל לבצע צעד עדכון משום ש- $p'(x_k)=0$ ולכן

$\frac{p(x_k)}{p'(x_k)}$ לא מוגדר.

ב השיטה ההיברידית מתמודדת עם המצב המתואר בסעיף

הקודם בצורה של האלגוריתם שראינו בתרגול. אם

$p'(x_k)=0$, אז $x_{k+1}=\infty$ והגענו "כשל".

הסברה המוצגת ע"י האלגוריתם אכן מתכנסת כי במצב הכי גרוע,

נשתמש כל פעם בשיטת החציה שמובטחת להתכנס כי התחלנו

את האלגוריתם בקטע $(-1,1)$.

code:

```
def main():
    k = 50
    eps = 1e-5

    p = lambda x: -3.55 * x ** 3 + 1.1 * x ** 2 + 0.765 * x - 0.74
    dp = lambda x: -10.65 * x ** 2 + 2.2 * x + 0.765

    x, fv = generic_hybrid(p, p, dp, -1, 1, 0, eps, k)
    print(f'Root: {x}')
```

output:

```
Root: -0.6081366095016632
```

(ק.4

$\forall 1<k<n \quad \phi_k(t) = f(x_1,\dots,x_{k-1},t,x_{k+1},\dots,x_n) = \alpha\|x-s\|_2^2 + \sum_{i=2}^{n}|x_i-x_{i-1}|$

$= \alpha\cdot\left(\sum_{i=1}^{k-1}(x_i-s_i)^2 + (t-s_k)^2 + \sum_{i=k+1}^{n}(x_i-s_i)^2\right) + \sum_{i=2}^{n}|x_i-x_{i-1}| =$

$= \alpha(t-s_k)^2 + |t-x_{k-1}| + |t-x_{k+1}| + \underbrace{\alpha\left[\sum_{i=1}^{k-1}(x_i-s_i)^2 + \sum_{i=1+k}^{n}(x_i-s_i)^2\right) + \sum_{i=2}^{k-1}|x_i-s_{i-1}| + \sum_{i=k+2}^{n}|x_i-x_{i-1}|}_{*}$

⇐ נקבל את הביטוי הנדרש עבור

$c=x_{k+1},\ b=x_{k-1},\ a=s_k$

$\mu=\alpha,\ C=1,\ D=*$

עבור $k=1$:

$\phi_k(t) = f(t,x_2,\dots,x_n) = \alpha\cdot\left(\sum_{i=1}^{n}(x_i-s_i\right) + \sum_{i=2}^{n}|x_i-x_{i-1}|$

$= \alpha\left((t-s_1)^2 + \sum_{i=2}^{n}(x_i-s_i)^2\right) + \sum_{i=2}^{n}|x_i-x_{i-1}|$

$= \frac{1}{2}\left[2\alpha(t-s_1)^2 + 2\cdot|t-x_2|\right] + \alpha\sum_{i=2}^{n}(x_i-s_i)^2 + \sum_{i=3}^{n}|x_i-x_{i-1}|$

⇐ נקבל קבועים עבור $\phi_k(t)$: $D=\alpha\sum_{i=2}^{n}(x_i-s_i)^2 + \sum_{i=3}^{n}|x_i-x_{i-1}|$

$c=x_2,\ b=x_2,\ a=s_1,\ \mu=2\alpha,\ C=\frac{1}{2}$

$$\min_{\mathbf{x}\in\mathbb{R}^n}\left\{f(\mathbf{x})\equiv\alpha\|\mathbf{x}-\mathbf{s}\|_2^2+\sum_{i=2}^{n}|\mathbf{x}_i-\mathbf{x}_{i-1}|\right\}$$

באופן דומה עבור $k=n$:

$$\phi_n(t)=f(x_1,\dots,x_{n-1},t)=\alpha\sum_{i=1}^{n}(x_i-s_i)^2+\sum_{i=2}^{n}|x_i-x_{i-1}|$$

$$\underset{\text{נציב } t}{=}\alpha(t-s_n)^2+|t-x_{n-1}|+\alpha\sum_{i=1}^{n-1}(x_i-s_i)^2+\sum_{i=2}^{n-1}|x_i-x_{i-1}|$$

$$=\frac{1}{2}\left[2\alpha(t-s_n)^2+\alpha\cdot|t-x_{n-1}|\right]+\underbrace{\alpha\sum_{i=1}^{n-1}(x_i-s_i)^2+\sum_{i=2}^{n-1}|x_i-x_{i-1}|}_{(*)}$$

$\Leftarrow$ נקבל קבועים שמ"מים את המשוואה לעיל:

$D=(*)$, $C=x_{n-1}$, $b=x_{n-1}$, $a=s_n$, $\mu=2\alpha$, $c=\frac{1}{2}$

ב. נתונה הפונקציה $\varphi:\mathbb{R}\to\mathbb{R}$ המקיימת

$$.\varphi(t)=\mu(t-a)^2+|t-b|+|t-c|$$

ונגדיר $l\equiv\min\{a,b,c\}-1$ ו־$u\equiv\max\{a,b,c\}+1$. הוכיחו כי אם ל־φ קיים מינימום גלובלי $t^*\in\mathbb{R}$, אז $.t^*\in[l,u]$

נניח ש-φ קיים מינימום גלובלי $t^*\in\mathbb{R}$.

אם $\mu<0$ אזי $\mu t^2-2\mu at+a^2+|t-b|+|t-c|\xrightarrow{t\to\infty}-\infty$ $\Leftarrow$ אין מינימום גלובלי והטענה נכונה באופן ריק

עבור $\mu>0$:

$$\varphi'(t)=2\mu t-2\mu a+\frac{t-b}{|t-b|}+\frac{t-c}{|t-c|}$$

$$\varphi'(l)=2\mu(\min(a,b,c)-1-a)+\frac{\min\{a,b,c\}-1-b}{|\text{"}\quad\text{"}|}+\frac{\min\{a,b,c\}-1-c}{|\text{"}\quad\text{"}|}$$

ניתן לראות שהביטוי שלילי ובאופן דומה $\varphi'(u)>0$

לכן קיימת $t^*\in[l,u]$ שבה סטציונרי כך ש $\varphi'(t^*)=0$

ו- $\varphi'(t)<0$ עבור $\forall t\in[-\infty,\ell]$ ולכן תמיד יורדת ואין מינימום גלובלי שם

ובאופן דומה $\forall t\in[u,\infty]: \varphi'(t)>0$ ובפרט אין מינימום שם.

$\Leftarrow \exists t^*\in[\ell,u]$ שהוא מינימום גלובלי.

4ג) מכיון ש-f unimodular והוכחנו מקודם כי נק' מינ' בקטע $[\ell,u]$, f אוניפורמית $\Leftarrow$ f רציפה ולכן מתקיים התנאי להתכנסות ה-Gold שמוצאת מינימום.

```python
def phi(t, mu, a, b, c):
    return mu * (t - a) ** 2 + abs(t-b) + abs(t-c)


4 usages  new *
def gs_denoise_step(mu, a, b, c):
    if mu < 0:
        return
    return (generic_gs_ex4(phi, max(a, b, c) + 1, min(a, b, c) - 1, mu, a, b, c,  eps: 10e-10))[0]
```

```python
def generic_gs_ex4(f, l, u, mu, a, b, c, eps):
    tau = (3 - 5 ** 0.5) / 2
    fv = [f(u, mu, a, b, c)]
    iter = 0
    x2 = l + (u - l) * tau
    x3 = l + (u - l) * (1 - tau)

    while abs(u - l) > eps:
        if f(x2, mu, a, b, c) < f(x3, mu, a, b, c):
            u = x3
            x3 = x2
            x2 = l + (u - l) * tau
        else:
            l = x2
            x2 = x3
            x3 = l + (u - l) * (1 - tau)
        fv.append(f(u, mu, a, b, c))
        iter += 1
    return l, fv
```

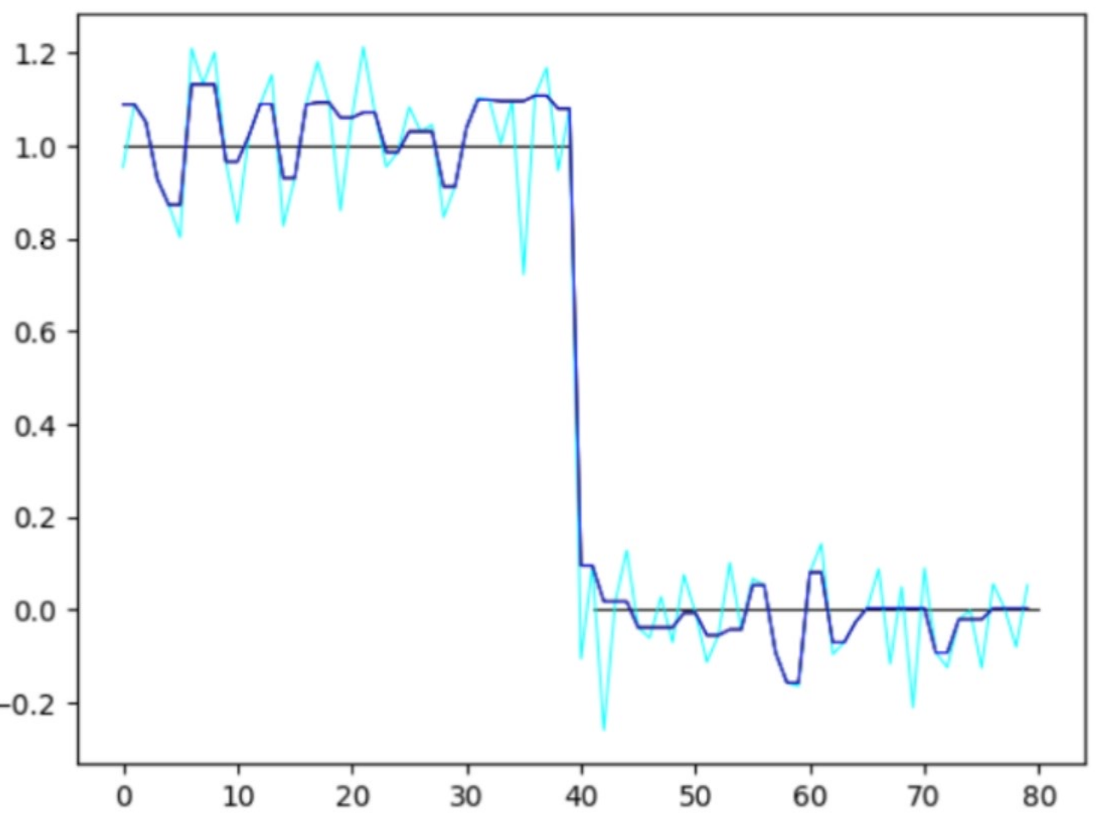
ב)

```python
def gs_denoise(s, alpha, N):
    x = copy.deepcopy(s)
    for i in range(N+1):
        for k in range(len(s)):
            if k == 0:
                x[k] = gs_denoise_step(2*alpha, s[0], x[1], x[1])
            elif k == len(s) - 1:
                x[k] = gs_denoise_step(2*alpha, s[k], x[k - 1], x[k - 1])
            else:
                x[k] = gs_denoise_step(alpha, s[k], x[k - 1], x[k + 1])
    return x
```

```python
# plotting the real discrete signal
real_s_1 = [1.]*40
real_s_0 = [0.]*40

plt.plot( *args: range(40), real_s_1, 'black', linewidth=0.7)
plt.plot( *args: range(41, 81), real_s_0, 'black', linewidth=0.7)


# solving the problem
s = np.array([[1.]*40 + [0.]*40]).T + 0.1*np.random.randn(80, 1) # noised signal
x1 = gs_denoise(s,  alpha: 0.5,  N: 10)
x2 = gs_denoise(s,  alpha: 0.5,  N: 20)
x3 = gs_denoise(s,  alpha: 0.5,  N: 30)

plt.plot( *args: range(80), s, 'cyan', linewidth=0.7)
plt.plot( *args: range(80), x1, 'red', linewidth=0.7)
plt.plot( *args: range(80), x2, 'green', linewidth=0.7)
plt.plot( *args: range(80), x3, 'blue', linewidth=0.7)

plt.show()
```

(ו)

פלט של הגרף:



עבור 10,20,30 ניתן לראות שאין הרבה הבדל.
ניתן לראות ש-x משפר יותר טוב מ-s.
יש משפר יחסית טוב מכיוון שהפרעה לא עולה על 0.2.