**תרגיל בית 5 – מודלים לא לינאריים בחקר ביצועים**

**מגישים :**

**אלעד בוכריס – 206202426**

**משה דידי – 311395834**

שאלה 1 :

*נתונה פונקציית המטרה :*

*נראה כי האיבר הכללי של הסכום גזיר ברציפות ולכן גזירה ברציפות כסכום של פונקציות גזירות ברציפות.*

*נשים לב כי ולכן הפונקציה מוגדרת.*

*נסמן ניתן לראות כי היא פונקציה לינארית ולכן גזירה ברציפות.*

*כעת, גזיר ברציפות כפונקציה אלמנטרית ולכן גזיר ברציפות.*

*כתוצאה מכך*

*גזירה ברציפות.*

*נגדיר:*

*נמצא קבוע ליפשיץ לפי הגדרה :*

*נגדיר*

*הוכחה:*

*ניתן לראות כי הנגזרת השניה תמיד שלילית, לכן, זוהי נקודת מקסימום גלובלית.*

*מתקבל כי*

*כעת, מטענה (5) מתקבל כי ליפשיציאנית עם קבוע .*

*ולכן :*

*כעת, נשים לב כי*

*כעת, לפי טענה (4) מתקיים :*

*ולכן :*

1. *השורה ה-i במטריצה A.*
2. *אי שוויון קושי שוורץ לנורמות.*

*מש"ל*

***שאלה 2 :***

*כאשר :*

*נתון אי השוויון הבא :*

*כאשר מתקיים :*

*סיכום ביניים :*

*מש"ל*

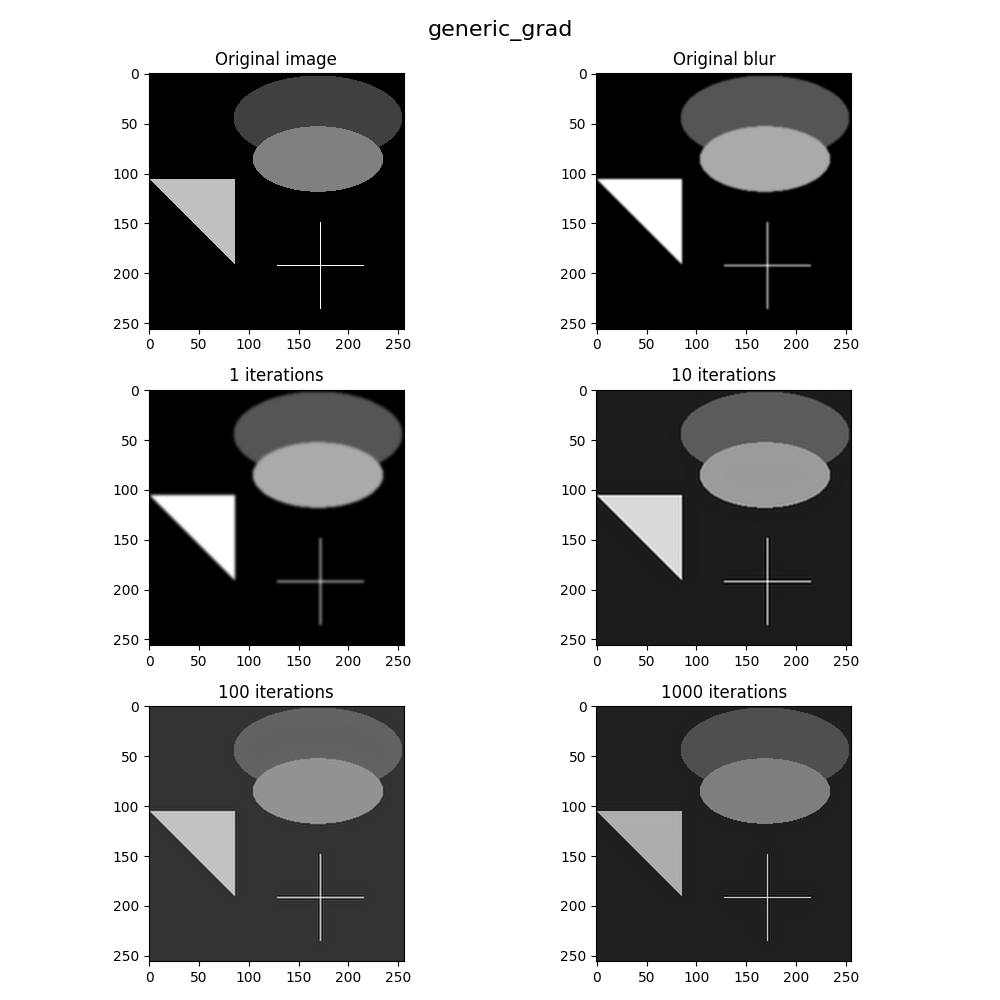
1. *מכיוון ש- נקבל כי*
2. *הצבת r.*
3. *אי השוויון הנתון.*

***שאלה 3 :***

1. *קוד:*

dimen = 256  
A, b, original\_x = blur(dimen, 5, 1)  
  
def f(x):  
 return np.square(np.linalg.norm(A.dot(x) - b))  
  
def gf(x):  
 return 2 \* A.T.dot(A.dot(x) - b)  
  
def const\_step(s):  
 return lambda \_f, xk, gk: s  
  
def exact\_quad(A):  
 return lambda \_f, xk, gk: 0.5 \* np.square(np.linalg.norm(gk)) / np.square(np.linalg.norm(A.dot(gk)))  
  
def generic\_grad(f, gf, lsearch, x0, num\_of\_iterations):  
 xk = x0  
 xk\_1 = xk - lsearch(f, xk, gf(xk)) \* gf(xk)  
 fs, gs, ts = [f(xk)], [np.linalg.norm(gf(xk))], [time.time()]  
 for i in range(num\_of\_iterations):  
 xk = xk\_1  
 xk\_1 = xk - lsearch(f, xk, gf(xk)) \* gf(xk)  
 fs.append(f(xk))  
 gs.append(np.linalg.norm(gf(xk)))  
 ts.append(time.time())  
 return xk, fs, gs, ts  
  
def q3\_a():  
 plt.figure(figsize=(10, 10)).suptitle('generic\_grad', fontsize=16)  
  
 ax = plt.subplot(3, 2, 1)  
 plt.imshow(original\_x.reshape(dimen, dimen), cmap='gray')  
 ax.set\_title(f'Original image')  
  
 ax = plt.subplot(3, 2, 2)  
 plt.imshow(b.reshape(dimen, dimen), cmap='gray')  
 ax.set\_title(f'Original blur')  
  
 x0 = np.zeros(A.shape[0]).reshape(dimen \* dimen, 1)  
  
 for i, j in enumerate([1, 10, 100, 1000]):  
 ax = plt.subplot(3, 2, i + 1 + 2)  
 xk, fs, gs, ts = generic\_grad(f, gf, exact\_quad(A), x0, j)  
 plt.imshow(xk.reshape(dimen, dimen), cmap='gray')  
 ax.set\_title(f'{j} iterations')  
  
 plt.show()

*גרף:*

**

*לפי ליפשיציות הגרדיאנט מתקיים :*

*כאשר הוא הע"ע הגדול ביותר של .*

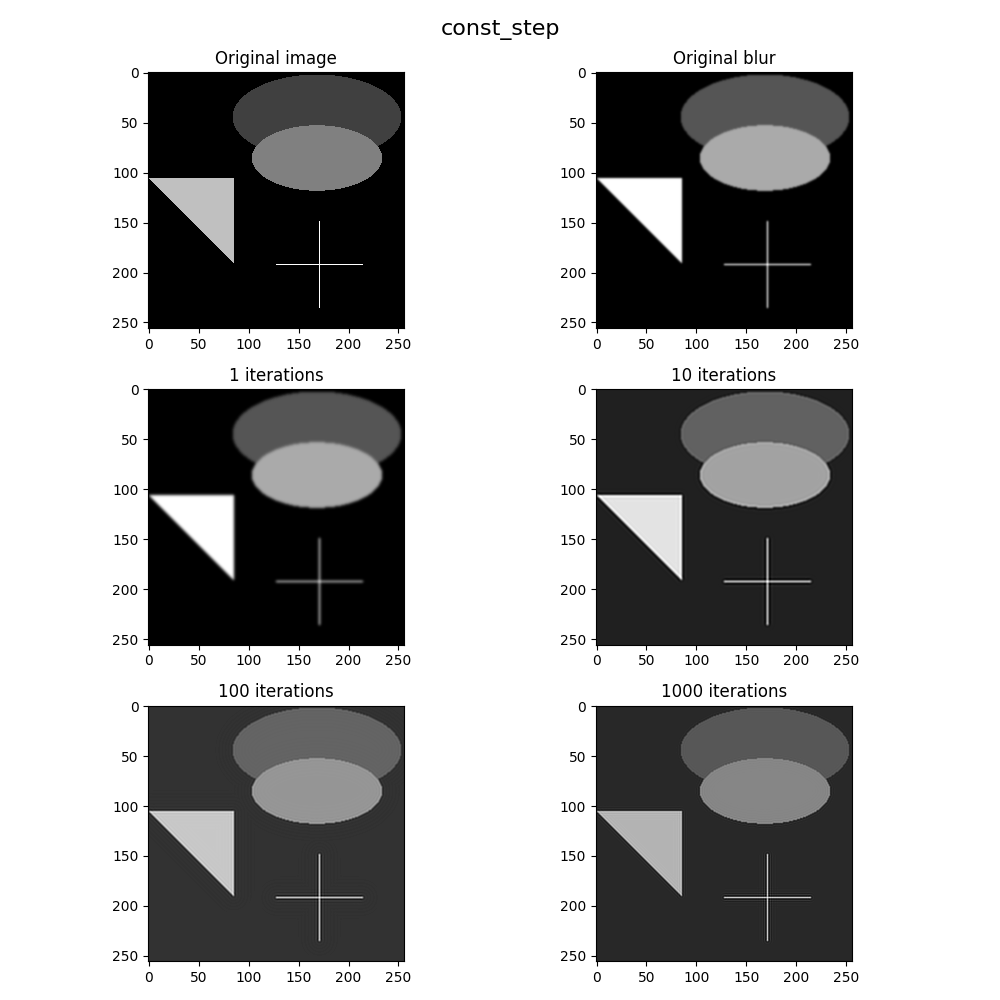
*נציב ונקבל :*

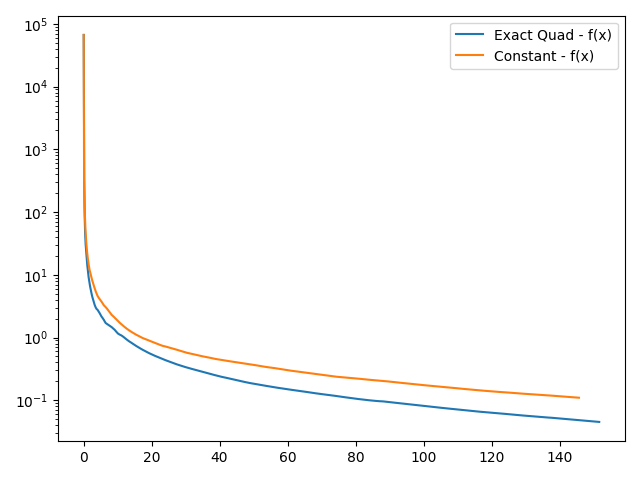
*נבחר ו- כך ש- עבור , מכיוון ש סימטרית אז ניתן לבצע את הבחירה הזו וניקח את מנורמל () לכן נקבל :*

*לכן, זהו קבוע ליפשיץ ההדוק ביותר.*

*קוד:*

def q3\_b():  
 plt.figure(figsize=(10, 10)).suptitle('const\_step', fontsize=16)  
 ax = plt.subplot(3, 2, 1)  
 plt.imshow(original\_x.reshape(dimen, dimen), cmap='gray')  
 ax.set\_title(f'Original image')  
  
 ax = plt.subplot(3, 2, 2)  
 plt.imshow(b.reshape(dimen, dimen), cmap='gray')  
 ax.set\_title(f'Original blur')  
  
 x0 = np.zeros(A.shape[0]).reshape(dimen \* dimen, 1)  
 s = 1 / (2 \* eigs(A.T.dot(A), k=1, return\_eigenvectors=False)[0].real)  
  
 for i, j in enumerate([1, 10, 100, 1000]):  
 ax = plt.subplot(3, 2, i + 1 + 2)  
 xk, fs, gs, ts = generic\_grad(f, gf, const\_step(s), x0, j)  
 plt.imshow(xk.reshape(dimen, dimen), cmap='gray')  
 ax.set\_title(f'{j} iterations')  
  
 plt.show()

*גרף:*

*בגרף הבא ציר הx הוא סך הזמן וציר הy הוא ערכי הפונקציה.*

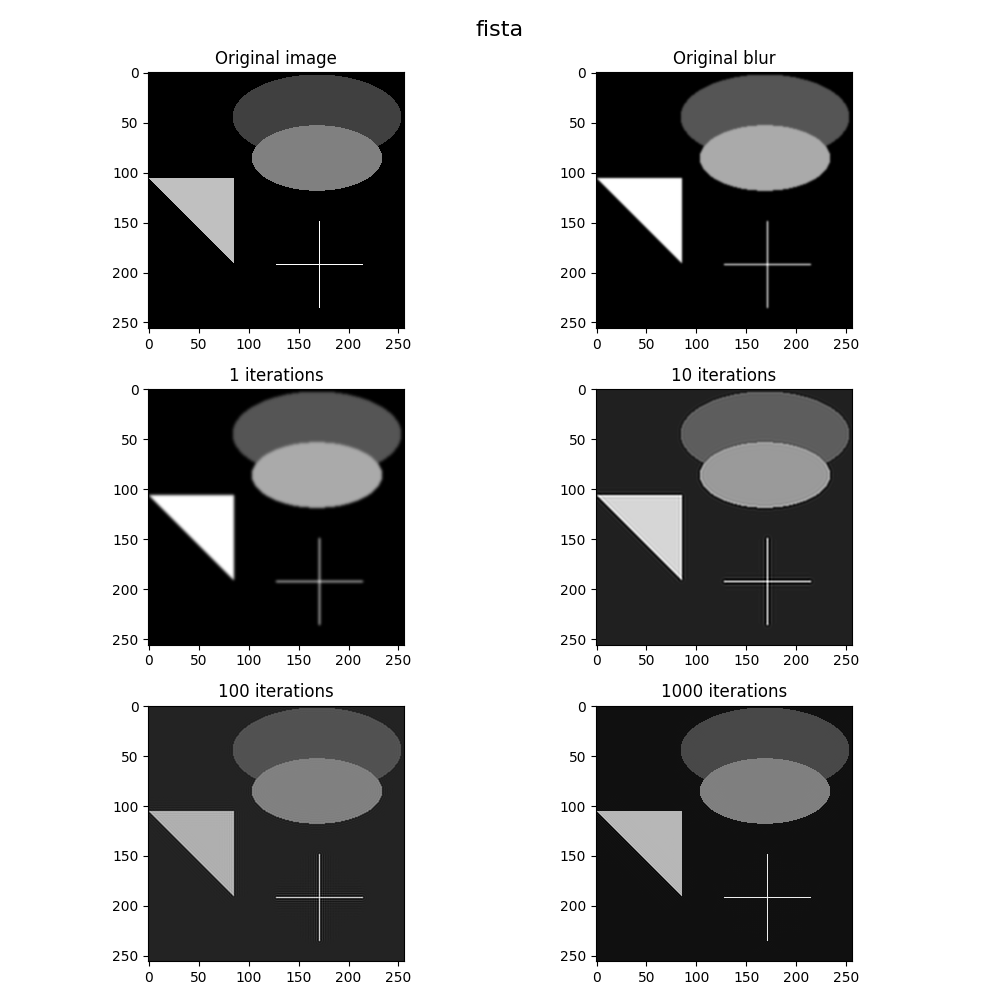
*ניתן לראות כי שיטת הגראדינט עם צעד קבוע מהירה יותר ומסיימת את 1000 האיטרציות מהר יותר אך שיטת הגראדינט עם חיפוש קווי מדויק מגיעה לערך פונקציית מטרה טוב יותר ולכן תתן תוצאה איכותית יותר.*

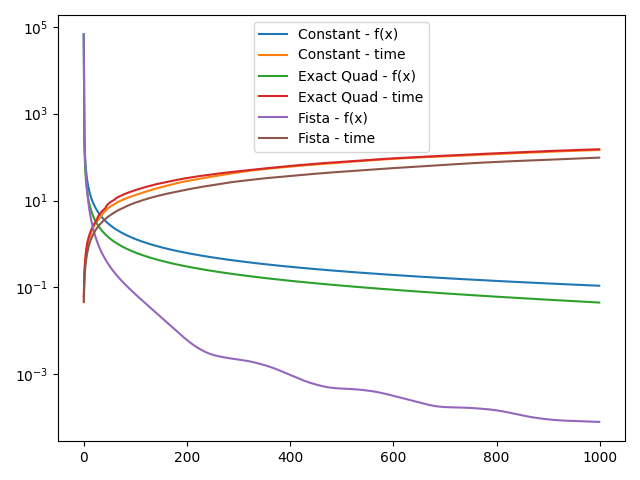
1. *קוד:*

def fista(f, gf, L, xk, num\_of\_iterations):  
 yk, tk = xk, 1  
 fs, gs, ts = [f(xk)], [np.linalg.norm(gf(xk))], [time.time()]  
 for i in range(num\_of\_iterations):  
 xk\_1 = xk  
 xk = yk - (1 / L) \* gf(yk)  
 tk\_1 = tk  
 tk = (1 + np.sqrt(1 + 4 \* np.square(tk))) / 2  
 yk = xk + ((tk\_1 - 1) / tk) \* (xk - xk\_1)  
 fs.append(f(xk))  
 gs.append(np.linalg.norm(gf(xk)))  
 ts.append(time.time())  
 return xk, fs, gs, ts

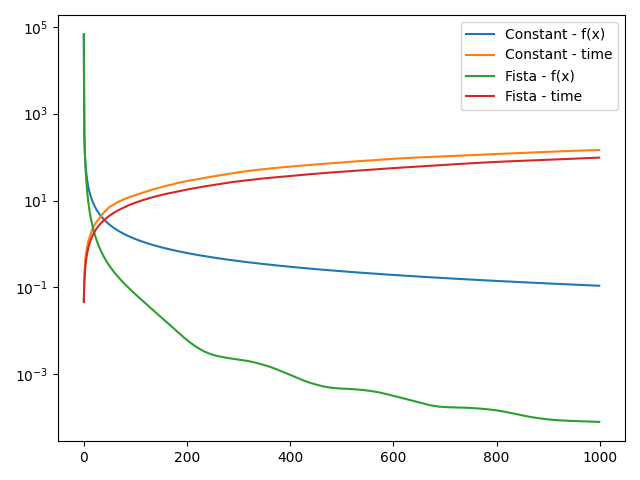
1. *קוד:*

def q3\_d():  
 plt.figure(figsize=(10, 10)).suptitle('fista', fontsize=16)  
 ax = plt.subplot(3, 2, 1)  
 plt.imshow(original\_x.reshape(dimen, dimen), cmap='gray')  
 ax.set\_title(f'Original image')  
  
 ax = plt.subplot(3, 2, 2)  
 plt.imshow(b.reshape(dimen, dimen), cmap='gray')  
 ax.set\_title(f'Original blur')  
  
 x0 = np.zeros(A.shape[0]).reshape(dimen \* dimen, 1)  
 L = 2 \* eigs(A.T.dot(A), k=1, return\_eigenvectors=False)[0].real  
  
 for i, j in enumerate([1, 10, 100, 1000]):  
 ax = plt.subplot(3, 2, i + 1 + 2)  
xk, fs, gs, ts = fista(f, gf, L, x0, j)  
 plt.imshow(xk.reshape(dimen, dimen), cmap='gray')  
 ax.set\_title(f'{j} iterations')  
  
 plt.show()

*גרף:*

**

*ניתן לראות כי משלושת השיטות שיטת הגראדינט עם צעד fista מתכנסת לערך הטוב ביותר לאחר 1000 איטרציות ולכן תיתן את התוצאה האיכותית ביותר.*

**

*ניתן לראות כי סך הזמן שלקח לשיטת הגראדינט עם צעד fista עבור אלף איטרציות קטן יותר מהזמן שלקח עם צעד קבוע, כמו כן צעד fista משפרת את ערך פונקציית המטרה משמעותית מהר יותר מצעד קבוע ולכן זקוקה לפחות איטרציות ע"מ להגיע לערך פונקציית מטרה נמוך יותר.*

***שאלה 4 :***

1. *מייצג את מיקום החיישן ה-i. לכן : מייצג את מרחק בין המיקום של העצם לחיישן ה-i (אוקלידי).*

*בנוסף, מייצג את המרחק שהחיישן ה-i מציג שבינו לבין העצם.  
לכן, על מנת להעריך את מיקום העצם, ניתן להסתכל על חיישן i והביטוי הבא : ונרצה למזער ביטוי זה עבור כל החיישנים על מנת להגיע קרוב ככל הניתן למיקום האמיתי של העצם.  
למזער ביטוי זה שקול למזעור ומכיוון שאנו רוצים שדבר זה יקרה לכל i נמזער את הסכום על כל החיישנים, כלומר :*

2. *לפי שיטת הגרדיאנט מתקיים:*

*נבחר ונקבל :*

***שאלה 5 :***

*תהי ו- . בנוסף, . נסמן : וב- ו- את הפתרונות של :*

*יש להוכיח כי :*

*בנוסף*

*כעת נשים לב כי :*

*לכן ניתן לכתוב את הדבר הבא :*

*מש"ל*

*משמעות אי השוויון – ניתן להסתכל עליו הצורה הבאה :*

*- הפתרון ה"מורעש" של .*

*- הערך האמיתי "מורעש" לעומת הערך האמיתי .*

*לכן, ככל שמספר ההתניה גדול יותר , היחס את הנורמות של הפתרון המורעש לפתרון האמיתי יכול להיות גדול יותר.*

*כלומר, מספר ההתניה נותן לנו חסם על היחס בין הפתרון המורעש לאמיתי.*

*ב.   
נשים לב כי אנו לוקחים את אותה תבנית פונקציונלית למזעור :*

*ומתחילים מאותה נקודת התחלה.  
ראינו בהרצאה כי ככל שמספר ההתניה גדול יותר, התכנסות שיטת הגרדיאנט איטית יותר.*

*נחשב ונקבל כי :*

*לכן, נצפה כי הבעיה המוגדרת על ידי G תתכנס מהר יותר.*