**תרגיל בית 6 – מודלים לא לינאריים בחקר ביצועים**

**מגישים :**

**אלעד בוכריס – 206202426**

**משה דידי – 311395834**

שאלה 1 :

*נתונה הפונקציה:*

*כאשר הפונקציה שאנו רוצים למזער מוגדרת להיות המרכז האנליטי של .*

1. *נוכיח כי עבור x השייכים לפוליגון והפונקציה מוגדרת בהם.*

*נגדיר:*

*ונגדיר :*

*ניתן לראות כי*

*כאשר :*

*לכן:*

*נחשב את :*

*ניתן לראות כי*

1. *מטריצה אלכסונית שכל איבר באלכסון חיובי.*

*לכן, אנו יודעים כי :*

1. *מכיוון שייתכן ו- אז הוא לא מוגדר חיובית.*
2. *נניח כי*

*נשים לב כי*

*ולכן הפונקציה יורדת לכל x בתחום, כלומר, אין לה מינימום גלובלי או לוקלי ולכן אין פתרון מיטבי לבעיה.*

1. *נתונים:*

*ראשית, נמצא את התחום שמקיים את האינטריור :*

*ולכן :*

*נכתוב את*

*לכן נקבל כי בנקודה סטציונרית מתקיים :*

*ניתן לראות כי ישר זה שייך ל-interior of P.*

*נכתוב את:*

*נסמן:*

*ראשית, עבור מתקבלים ע"ע שליליים ולכן המטריצה לא יכולה להיות מוגדרת אי שלילית.*

*נחלק למקרים :*

*על מנת שגם ה-trace יהיו חיובים, צריך להתקיים ולכן התחום ריק.*

*במצב זה ולכן ה-trace שלילי וקיימים ע"ע שליליים, כלומר המטריצה לא יכולה להיות מוגדרת אי שלילית.*

*עבור כל .*

*בנוסף, הראנו בסעיף הקודם כי ההסיאן מוגדר אי שלילית לכל x בתחום, ולכן, כל נקודה סטציונרית היא נקודת מינימום גלובלי של הפונקציה. לכן, קיים פתרון מיטבי.*

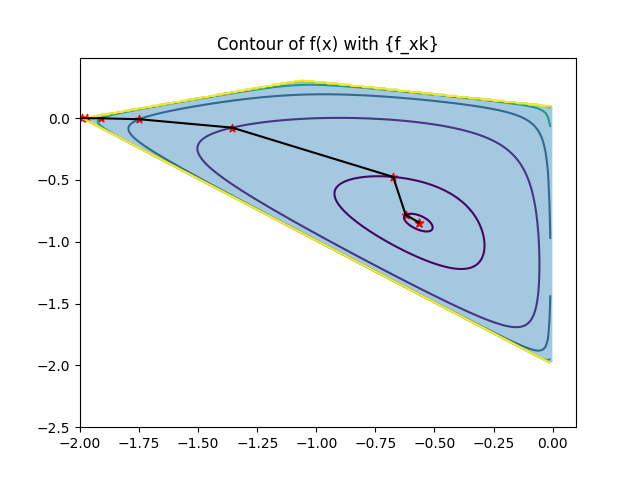
1. *analytic\_center:*

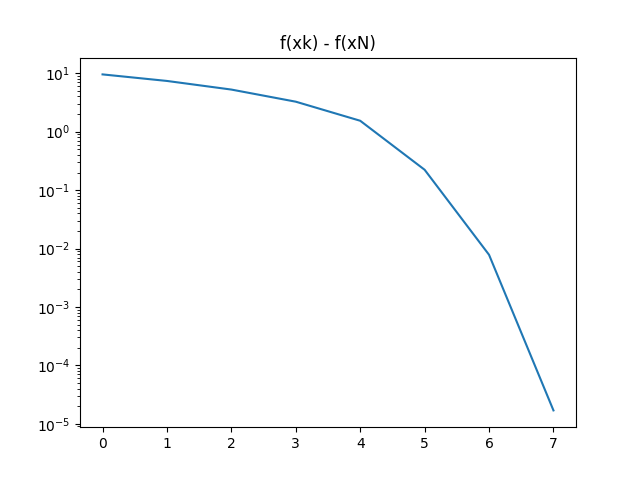
def armijo\_newton(f, df, ddf, alpha, beta, s, xk, eps):  
 xs, fs = [], []  
 while np.linalg.norm(df(xk)) > eps:  
 f\_xk, df\_xk = f(xk), df(xk)  
 xs.append(xk.T)  
 fs.append(f\_xk)  
 dk = np.linalg.solve(ddf(xk), -df\_xk)  
 tk = s  
 while f(xk + tk \* dk) >= f\_xk + alpha \* tk \* df\_xk.T.dot(dk):  
 tk = beta \* tk  
 xk = xk + tk \* dk  
 if len(xs) == 0:  
 xs, fs = [xk.T], [f(xk)]  
 return xs, fs  
  
  
def analytic\_center(A, b, x0):  
 if not all([(A[i].T.dot(x0) <= b[i]) for i in range(len(b))]):  
 raise Exception('x0 is not in interior(P)')  
 xs, fs = armijo\_newton(f(A, b), df(A, b), ddf(A, b),  
 alpha=1 / 4,  
 beta=1 / 2,  
 s=2,  
 xk=x0,  
 eps=1 / np.power(10, 6))  
 return np.array(xs), np.array(fs)

1. *test\_analytic\_center:*

def test\_analytic\_center():  
 A = np.array([[2, 10],  
 [1, 0],  
 [-1, 3],  
 [-1, -1]])  
 b = np.array([1, 0, 2, 2])  
 x0 = np.array([-1.99, 0])  
 xs, fs = analytic\_center(A, b, x0)  
  
 X, Y = np.meshgrid(np.arange(-2, 0, 0.01), np.arange(-2.5, 0.5, 0.01))  
 Z = X.copy()  
  
 f\_xk = f(A, b)  
 for i in range(len(X)):  
 for j in range(len(X[0])):  
 Z[i, j] = f\_xk((X[i, j], Y[i, j]))  
  
 Z[(np.isnan(Z))] = 10 \*\* 10  
 fig, ax = plt.subplots(1, 1)  
 ax.contour(X, Y, Z, levels=fs[::-1], extend='both')  
 patches = []  
 polygon = Polygon([[-2, 0],  
 [-1.0604, 0.3072],  
 [-0.001, 0.1],  
 [-0.001, -1.98]], True)  
  
 patches.append(polygon)  
  
 p = PatchCollection(patches, alpha=0.4)  
 ax.add\_collection(p)  
 ax.scatter(xs[:, 0], xs[:, 1], marker='\*', color='red')  
 ax.plot(xs[:, 0], xs[:, 1], color='black')  
 plt.title("Contour of f(x) with {f\_xk}")  
 plt.show()  
  
 plt.figure()  
 plt.title("f(xk) - f(xN)")  
 plt.semilogy(np.arange(len(fs) - 1), np.array(fs[0:len(fs) - 1]) - fs[-1], label="f(xk) - f(xN)")  
 plt.show()

*תרשימים:*

**

**

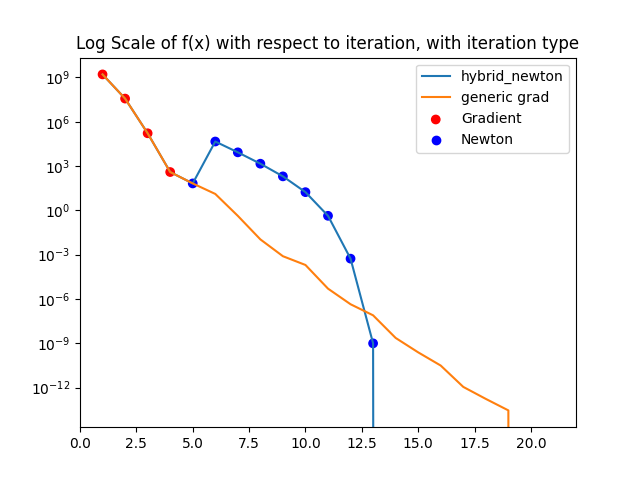
*שאלה 2 :*

1. *hybrid\_newton:*

def hybrid\_newton(f, gf, hf, lsearch, xk, eps):  
 f\_xk, df\_xk, hf\_xk = f(xk), gf(xk), hf(xk)  
 fs, gs, ts, newton = [f\_xk], [np.linalg.norm(df\_xk)], [0], []  
 while np.linalg.norm(gf(xk)) > eps:  
 start\_time = time.time()  
 try:  
 np.linalg.cholesky(hf(xk))  
 dk = np.linalg.solve(hf\_xk, -df\_xk)  
 tk = lsearch(xk, df\_xk, [dk, 'newton'])  
 newton.append(1)  
 except LinAlgError:  
 dk = -df\_xk  
 tk = lsearch(xk, df\_xk, [dk, 'grad'])  
 newton.append(0)  
 xk = xk + tk \* dk  
 f\_xk, df\_xk, hf\_xk = f(xk), gf(xk), hf(xk)  
 fs.append(f\_xk)  
 gs.append(np.linalg.norm(df\_xk))  
 ts.append(time.time() - start\_time + ts[-1])  
 newton.append(1)  
 return xk, fs, gs, ts[1:], newton

1. *hybrid\_back:*

def hybrid\_back(f, alpha, beta, s):  
 def lsearch(xk, gk, direction):  
 if direction[1] == 'newton':  
 return 1  
 else:  
 dk = direction[0]  
 tk = s  
 while f(xk + tk \* dk) >= f(xk) + alpha \* tk \* gk.T.dot(dk):  
 tk \*= beta  
 return tk  
  
 return lsearch

**

1. *תרשים:*

*האלגוריתם ההיברידי אינו אלגוריתם ירידה, כפי שניתן לראות באיטרציה החמישית עקב שיטת ניוטון ערך הפונקציה עולה.*

*ניתן להוסיף בדיקה שרק במידה ובאמצעות שיטת ניוטון נקבל ערך פונקציה נמוך יותר נבחר בה, אחרת נשתמש בשיטת הגרדיאנט שאנו יודעים כי היא מונוטונית יורדת.*

***שאלה 3 :***

*נתונה בעיית פרמה וובר*

*נגדיר:*

1. *נשים לב כי עבור מתקיים כי y קבוע.*

*ניתן לראות כי זו פונקציה ריבועית עם מטריצה מובילה .*

*הראנו כי*

*ולכן נקבל כי :*

1. *ניתן לראות כי*

*לכן :*

*צריך להוכיח כי*

*נראה זאת על ידי כך שנוכיח כי האיבר הכללי של הסכום השמאלי גדול או שווה לאיבר הכללי של הסכום הימני, כלומר :*

*אי השוויון האחרון נכון תמיד ולכן*

*כעת, ניתן להסיק כי :*

***שאלה 4 :***

*עבור m פונקציות גזירות ברציפות ו- m קבועים .*

*נתונה הבעיה הבאה :*

*נגדיר כך :*

*נסמן את מטריצת היעקוביאן של F ב- .*

1. *נשים לב כי*

*מכיוון שמטריצת הנגזרות החלקיות של F מוגדרת להיות מטריצת יעקובי.*

1. *ניתן לראות כי עבור שיטת גאוס ניוטון מתקיים :*

*נגדיר:*

*ונקבל :*

1. *נניח כי מדרגת עמודות מלאה.*

*על ידי מזעור הפונקציה נקבל כי :*

*שיטת הגראדינט המדורגת נראית כך :*

*נגדיר :*

*ונקבל את שיטת הגרדיאנט.*

*נשים לב כי נתון ש- מדרגת עמודות מלאה, לכן, אין לה ע"ע אפס.  
לאור זאת*

*מש"ל.*