**חלק 1:**

1. מצא מטריצה כך שהבעיות שקולות:

**פתרון:**

נגדיר כלומר האיבר ה-  הוא הדגימה ה - בחזקת .

נשים לב כי: 

לכן: והבעיות שקולות עבור המטריצה הזאת.

1. נראה כעת כי  קמורה, ונמצא פתרון סגור.

**פתרון:**



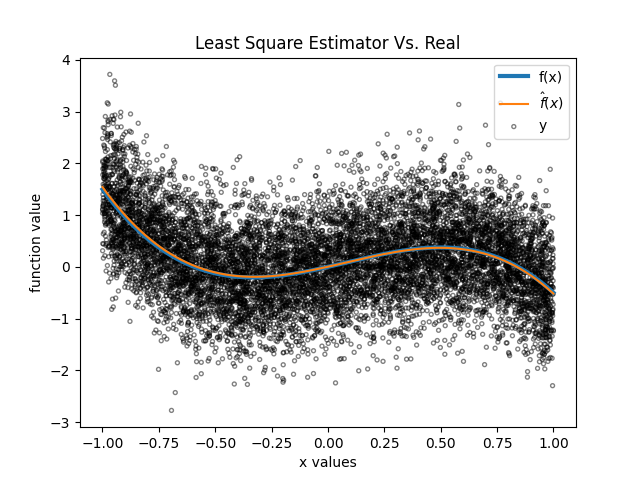
והפתרון הסגור:



נשים לב כי הפתרון קיים אם הפיכה.

1. המקדמים שהפונקציה שלנו החזירה הם: Graphical user interface

   Description automatically generated
2. **Least Square Estimator Vs. Real:**

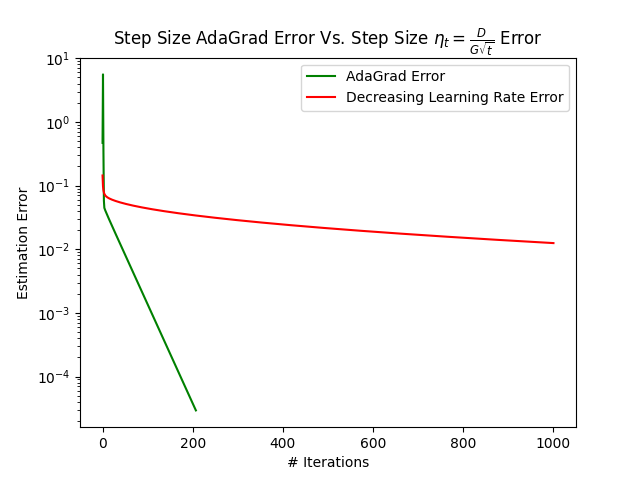


ניתן לראות מהגרף, כי הדגימות הרועשות מופלגות בצורה גאוסית סביב הפונקציה המקורית, כלומר הסבירות לדגימה רועשת ממש (רחוקה מאוד מהפונקציה) קטנה (יש מעט דגימות שכאלה).

בנוסף ניתן לראות שלמרות שהדגימות שלנו רועשות אכן שערוך שמצאנו משערך היטב את הפונקציה.

**חלק 2 Projected Gradient Descent-**

1. מהחלק הקודם של השאלה ראינו כי:  
   .
2. השוואה בין ה גודל הצעד כאשר הוא AdaGrad או :

**

*הרצנו את אלגוריתם Projected gradient descent המבוקש עם פרמטר סף של* .

כפי שניתן לראות מהגרף, עבור גודל צעד דועך האלגוריתם עדיין לא התכנס לטווח השגיאה הרצוי כי ראינו שלצורך כך דרושות  איטרציות, כלומר סדר גודל של 1,000,000 איטרציות. לעומת זאת, כאשר גודל הצעד הוא AdaGrad הביצועים הם הרבה יותר טובים, אנו משיגים שגיאה קטנה מהסף בהרבה פחות איטרציות. לפי מה שראינו בתרגול 6, AdaGrad יכול להשיג ביצועים הרבה יותר טובים כאשר הנורמה של הגרדיאנטים קטנה משמעותית מ-G.

1. *צריך להוכיח כי קבוע החלקות של פונקציית המטרה*  הוא: .  
   **הוכחה:** נראה כי הפונקציה היא בעלת גרדיאנט ליפשצי עם קבוע :



נסמן ב- , כמו כן  כי זאת מטריצה PSD ובפרט,

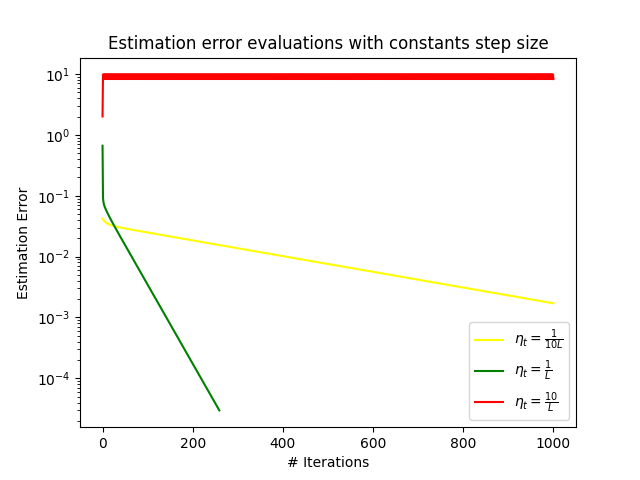


זאת מכיוון ש – הוא ערך עצמי של  ולכן עם אותו וקטור עצמי הוא גם ערך עצמי של  והוא גם מקסימלי עבור מטריצה זו אחרת זו סתירה שהוא מקסימלי עבור המטריצה .

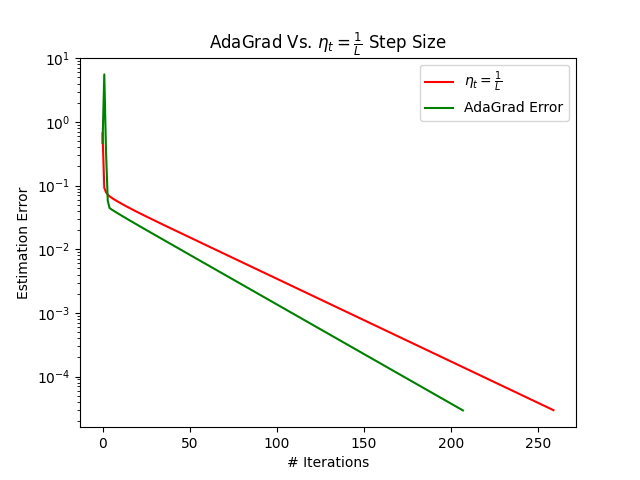
לכן: .

ומכיוון שהפונקציה קמורה וגזירה ברציפות ובעל גרדיאנט לפישצי עם קבוע אזי היא  חלקה.

1. **Constant Step Size:**



*מכיוון שאנו מניחים כי דרגת המטריצה* היא הרי הדטרמיננטה שלה שונה מאפס בוודאות ומכיוון ש , אזי ההיסאן של פונקציית המטרה בעל דטרמיננטה שונה מאפס בוודאות ולכן בהכרח כלומר פונקציית המטרה היא strongly convex וחלקה אזי כפי ראינו בכיתה נוכל לבחור בגודל צעד קבוע השווה ל- , וקצב ההתכנסות הוא מעריכי במספר איטרציות. בנוסף, ניתן לראות בגרף שגודל צעד קטן מידי מוביל להתכנסות איטית יותר וגודל צעד גדול לא מוביל להתכנסות.

1. **

*כפי שניתן לראות מהגרף שני גדלי הצעד משיגים קצב התכנסות אקספוננציאלי זאת מכיוון שהגרף הוא לינארי בסקלה לוגריתמית. עבור גודל הצעד הקבוע ראינו והוכחנו זאת בתרגול. לעומת זאת, בתרגול הוכחנו כי עבור המקרה הגרוע ביותר קצב ההתכנסות של גודל צעד AdaGrad הוא* . אמנם, מה שמייחד את *AdaGrad שהוא אדפטיבי והוא גורם לגודל הצעד להתאים את עצמו לתהליך עצמו על ידי שימוש בגרדיאנטים. לכן, הוא גם משיג קצב מעריכי.*

**חלק 3 Projected Stochastic Gradient Descent-**

1. Chart, histogram

   Description automatically generatedניתן להבין מהגרף כי אלגוריתם ה-SGD הוא יותר רועש מהאלגוריתם ה-GD, זאת מכיוון שהגרדיאנט נמדד רק על פה כמה מדידות ולא על פי כולן לכן לא בהכרח שבכל איטרציה אנחנו נתקדם לעבר המינימום של הפונקציה הכוללת. כמו שעוד ניתן לשים ככל שאנחנו דוגמים יותר דגימות יחד, כלומר bath יותר גדול התהליך פחות רועש כי כיוון הגרדיאנט יותר נכון.  
   נשים לב, כי ככל שאנחנו דוגמים יותר אז אנחנו מתכנסים יותר מהר והשגיאה קטנה יותר מהר ביחס למספר איטרציות.
2. Chart, line chart

   Description automatically generated

בגרף הזה אנחנו רואים בבירור את היתרון של SGD ביחס לGD. נכון שביחס לכמות איטרציות batch גדול יותר עשוי להתכנס מהר יותר. אמנם, אם נסתכל ביחס לזמן אנחנו רואים כי ככל ש-batch גדול כמו כל הדגימות אזי השגיאה יורדת בקצב יותר איטי וזאת מכיוון שכל איטרציה עבור batch גדול יותר לוקחת יותר זמן.  
וזה החיסרון של batch הזמן בפועל של ההתכנסות יותר איטי כי כל איטרציה לוקחת יותר זמן, יותר כבדה חישובית, אמנם ככל הנראה ההתכנסות תהיה יותר מדויקת ופחות רועשת.