

CurviFit – Curve Fitting and Regression Analysis

עבודה מסכמת בקורס "יישומי מחשב למדעים", שקד טובל 313360661

הקדמה

כימאים אנליטיים וכימאים פיזיקליים, יחד עם מדענים מתחומים אחרים, נדרשים פעמים רבות לניתוח של נתונים שנאספו ממדידות בניסוי כלשהו. פעמים רבות, שלב אחד בניתוח הנתונים הוא התאמת הנתונים למשוואת פונקציה. התאמה של נתונים לעקומה נדרשת פעמים רבות כאשר המדידות שלנו אמורות לציית למשוואה מתמטית כלשהי, ואנו רוצים לוודא זאת, או שאנו רוצים לחלץ גדלים מסוימים הנוגעים למערכת ונוכל לעשות זאת על ידי מציאת הפרמטרים של המשוואה המתמטית הזו.

סביבת העבודה LabWindows/CVI מכילה פונקציות ספריה להתאמת נתונים לעקומה, עבור מספר פונקציות התאמה כמו ישר, פולינום, אקספוננט ועוד. עם זאת, פונקציות הספריה המובנות מאפשרות התחשבות בשגיאות המדידה בציר Y בלבד, ומתעלמות מהשגיאות בציר X. עם זאת, השגיאות בציר X יכולות להשפיע על הפרמטר χ^2 בעזרתו מתבצעת ההתאמה (פירוט על כך בהמשך). ההשפעה של שגיאות אלו לא בהכרח גדולה, אך רציתי לכתוב תוכנית המאפשרת להתחשב בהן.

השימוש בפונקציות ההתאמה המובנות הכרחי כיוון שהאלגוריתם בו עשיתי שימוש עבור ההתאמה דורש ניחושים ראשוניים עבור הפרמטרים של הפונקציה (פירוט בהמשך), ולצורך כך השתמשתי בפרמטרים המתקבלים מהפונקציות המובנות. בנוסף, בחרתי להציג למשתמש את פרמטרי ההתאמה של הפונקציות המובנות ושל הפונקציה שלי, וכך ניתן להשוות ביניהם וגם לבדוק את ההשפעה של השגיאות בציר X על ידי השוואת פרמטרי טיב ההתאמה.

התוכנית שכתבתי, CurviFit, כוללת את התכונות הבאות:

- ייבוא נתונים בשתי דרכים: מקובץ על ידי פתיחתו דרך ה-GUI או על ידי הדבקתם ישירות מה-Clipboard.
- שרטוט לגרף של המדידות כולל צלבי שגיאה.
- שינוי טווח המדידות המשמש להתאמה מתוך המדידות שיובאו.
- התאמה ראשונית של הנתונים על ידי פונקציות הספריה של CVI.
- שרטוט לגרף של פונקציית ההתאמה הראשונית.
- תמיכה בהתאמה לפונקציות: ישר, פולינום עד דרגה 10, אקספוננט, גאוסיאן ולוגריתם (טבעי ובסיס 10).
- ביצוע התאמה על ידי אלגוריתם שאני כתבתי.
- חישוב שגיאות עבור פרמטרי ההתאמה שהתקבלו ותלות הדדית ביניהם (covariance).
- חישוב פרמטרי טיב התאמה לשתי ההתאמות: χ^2 , p-value עבורו, χ^2 מצומצם (לדרגת חופש).
- שרטוט לגרף של פונקציית ההתאמה המתקבלת.
- שרטוט של גרף שארים של ההתאמה.
- התאמה אישית של כותרות הגרפים והצירים על ידי המשתמש.
- ייצוא פרמטרי ההתאמה לקובץ txt.
- חלון עזרה המסביר את השימוש בתוכנית.

החלון של הגרפים כולל את התכונות הבאות:

- שינוי טווח הצירים.
- הוספת עד 5 תוויות מידע לנקודות על הגרף ומחיקתן.
- בחירת הגרפים שמוצגים בו.
- ייצוא הגרף כתמונה ושמירתו במחשב.

ממשק המשתמש הגרפי

להלן דוגמת הרצה עבור ההתאמה לפולינום מדרגה 4.

החלון הראשי לאחר בחירת סוג ההתאמה (פולינום מדרגה 4), פתיחת קובץ המדידות ובחירה של טווח ערכי X להתאמה:

החלון של פרמטרי ההתאמה המופיע לאחר לחיצה על "Fit!" בחלון הראשי:

Fit Parameters

Polynomial fit
 $y = a_0 + a_1 \cdot x + a_2 \cdot x^2 + \dots$

Iteration no. 331
 Initial parameters' values:
 $a_0 = 1.999932 \pm 0.080395$
 $a_1 = 4.999782 \pm 0.080395$
 $a_2 = -2.141253 \pm 0.080395$
 $a_3 = -4.197777 \pm 0.080395$
 $a_4 = 1.155373 \pm 0.080395$
 $\chi^2 = 38.504559$
 $\chi^2_{red} = 9.626140$
 $p_prob = 0.000000$

Fitted parameters' values:
 $a_0 = 2.005670 \pm 0.154798$
 $a_1 = 5.023101 \pm 0.269557$
 $a_2 = -2.155117 \pm 0.228559$

Save

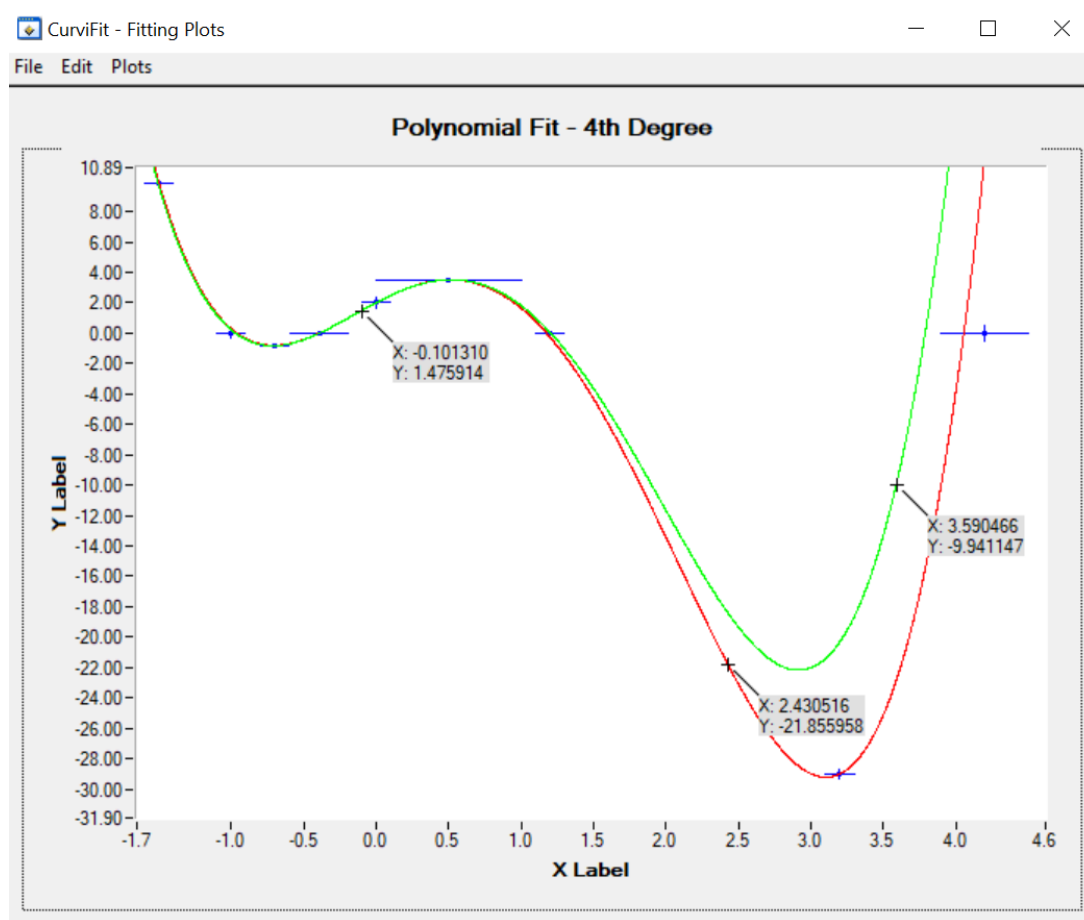
Fit Parameters

$a_1 = 5.023101 \pm 0.269557$
 $a_2 = -2.155117 \pm 0.228559$
 $a_3 = -4.347948 \pm 0.358266$
 $a_4 = 1.120106 \pm 0.063481$
 $cov(a_0, a_1) = -0.033648$
 $cov(a_0, a_2) = -0.027968$
 $cov(a_0, a_3) = 0.064374$
 $cov(a_0, a_4) = -0.015744$
 $cov(a_1, a_2) = -0.101191$
 $cov(a_1, a_3) = 0.199168$
 $cov(a_1, a_4) = -0.048562$
 $cov(a_2, a_3) = -0.129501$
 $cov(a_2, a_4) = 0.032017$
 $cov(a_3, a_4) = -0.030901$
 $\chi^2 = 0.352840$
 $ndf = 4$
 $\chi^2_{red} = 0.088210$
 $p_prob = 0.986153$

Save

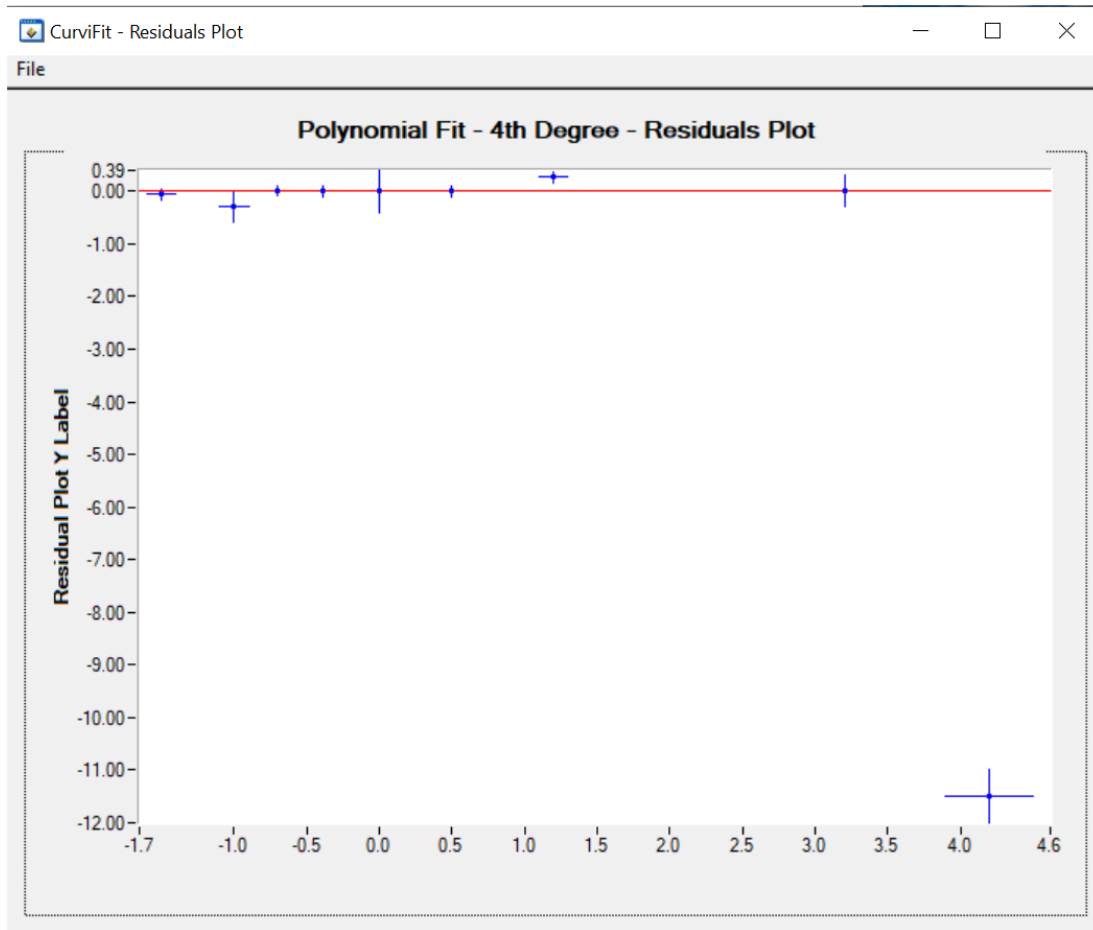
ניתן לראות שערך χ^2_{red} המצומצם (שהתקבל עבור אלגוריתם ההתאמה שלי טוב יותר מאשר זה של ההתאמה הראשונית (של פונקציית הספרייה של CVI)), על פי ערך ה-p-value (p_prob).

החלון של גרפי ההתאמה לאחר שרטוט של המדידות (כחול) גרף ההתאמה הראשונית (ירוק) וגרף ההתאמה (אדום), על ידי לחיצה על הכפתורים המתאימים בחלון הראשי (ניתן גם להסתיר/להציג כל אחד מהגרפים בעזרת התפריט "Plots" לאחר פתיחת חלון הגרפים בפעם הראשונה ע"י לחיצה על אחד הכפתורים בחלון הראשי) ולאחר הוספה של 3 תוויות מידע (על ידי לחיצה על הגרף; ניתן למחוק אותן על ידי עכבר ימני או דרך התפריט "Edit"):



גם כאן ניתן לראות שההתאמה של האלגוריתם שלי (אדום) טובה יותר מההתאמה הראשונית (ירוק).

החלון של גרף השאריות, המופיע לאחר לחיצה על "Plot Residuals Plot":



חלון העזרה המופיע על ידי לחיצה על "Help" בחלון הראשי:

CurviFit - Help

CurviFit - Curve Fitting and Regression Analysis

v1.0 - March 2021

Created by Shaked Tuval as a final project for the course "Computer Application for Science"

Fitting process

1. Choose fit type. Supported fit functions:
 - Linear: $y = a_0 + a_1 \cdot x$
 - Exponential: $y = a_0 \cdot \exp(a_1 \cdot x)$
 - Polynomial of degree n: $y = \sum_{i=0}^n a_i \cdot x^i$
 - Base 10 logarithm: $y = a_0 \cdot \log(a_1 \cdot x)$
 - Natural logarithm: $y = a_0 \cdot \ln(a_1 \cdot x)$
2. Import data - you can paste a table (from MS Excel for example) or open a data file (supported filetypes: .txt, .csv). Data format must be a 4-column table in that order: x, dx, y, dy.
3. You can choose a custom range of X values to fit. If selected, you must specify both minimum, and maximum values.
4. Click "Fit!" to fit data. Fitting parameters will display on a new window. These can be exported to a text file by clicking "Save".
5. Specify titles for the graph and axes. These can be empty.
6. Click one of the plotting buttons to open the graph window and plot the appropriate plot. All plots except the residual plot are plotted in a single graph; residual plot is plotted in a separate graph in another window.

האלגוריתמים בהם נעשה שימוש

להלן תיאור של הפונקציות העיקריות שכתבתי והסבר שלהן. שם הפונקציות מופיע בסוגריים.

1. פתיחת קובץ ו-Parsing של המדידות (OpenDataFile)

בחירת הקובץ על ידי המשתמש ופתיחתו נעשות על ידי פונקציות ספריה של CVI (OpenFile ו-FileSelectPopupEx), (בהתאמה). לאחר מכן, נעשה שימוש בפונקציית הספריה ScanFile על מנת לבצע פורמטיזציה של המידע מהקובץ, שצריך להכיל טבלה בעלת 4 טורים (x, dx, y, dy), אל תוך מערכים מתאימים המכילים את המדידות להתאמה. במקרה שאחת השורות בקובץ לא מכילה 4 איברים, מופיע פופ-אפ המעדכן את המשתמש על כך והפונקציה מחזירה שגיאה.

2. הדבקה מ-Clipboard ו-Parsing של המדידות (PasteData)

הדבקת טבלה מה-Clipboard נעשית על ידי פונקציית הספריה ClipboardGetText. במקרה שה-Clipboard לא מכיל טקסט, מופיע פופ-אפ והפונקציה מחזירה שגיאה. לאחר מכן נעשה שימוש בפונקציה Scan על מנת לבצע פורמטיזציה של המידע אל המערכים המתאימים, בדומה לפורמטיזציה מקובץ, וגם במקרה זה אם אחת השורות לא מכילה 4 איברים מופיע פופ-אפ וחוזרת שגיאה.

3. שינוי טווח המדידות להתאמה (ChangeDataRange)

הקריאה לפונקציה הזו נעשית על ידי הפונקציה Fit, לפני ביצוע ההתאמה הראשונית. במקרה שהמשתמש סימן בחלון הראשי את האופציה לשנות את טווח הנקודות להתאמה, הפונקציה לוקחת מהחלון את הנתונים שהמשתמש הזין עבור X מינימלי ומקסימלי. במקרה שהמשתמש לא סימן את האופציה לשנות את הטווח, הפונקציה מוצאת את הערך המינימלי והמקסימלי במערך בעזרת פונקציית הספריה MaxMin1D.

לאחר מכן, הפונקציה מעתיקה את הנקודות שבין ערכי X המינימלי והמקסימלי (שהמשתמש הזין או שנמצאו על ידי MaxMin1D) מכל המערכים x, dx, y, dy אל מערכים חדשים שישמשו להתאמה. השימוש בשני מערכים נועד לאפשר את שינוי טווח הנקודות להתאמה על ידי המשתמש – בצורה כזו יש רק סט אחד של מערכים המשמש להתאמה ונחסך הצורך להשתמש במערכים שונים במקרים שונים ומתקבל קוד פשוט יותר (על חשבון זיכרון כמובן).

4. התאמה ראשונית והדפסת תוצאות ההתאמה (Fit)

זו פונקציית ה-Callback העיקרית המבצעת מספר פעולות. ראשית, היא מבצעת התאמה ראשונית של הנתונים על ידי פונקציית הספריה המתאימה לפונקציה שהמשתמש בחר (GaussFit, ExpFitEx, PolyFitWithWeight, LinearFitEx, LogFit). לאחר מכן Fit קוראת לפונקציה GeneralFit, המבצעת את ההתאמה השנייה, ומספקת לה כארגומנטים את הפרמטרים שהתקבלו מההתאמה הראשונית.

לאחר שהפונקציה GeneralFit מחזירה את פרמטרי ההתאמה, Fit מחשבת את פרמטרי טיב ההתאמה של ההתאמה הראשונית (עבור ההתאמה השנייה הם מחושבים על ידי GeneralFit), ומדפיסה לחלון יעודי את כל הפרמטרים שהתקבלו.

5. התאמה לפונקציה (GeneralFit)

השיטה הנפוצה להתאמת נתונים לפונקציה היא שיטת הריבועים המינימליים (Least Squares) המניחה שהפונקציה המתאימה ביותר למדידות היא זו בה הערכים שלה הם הקרובים ביותר למדידות. הערך המייצג את המרחק בין הפונקציה למדידות הוא χ^2 (סכום המרחקים של כל המדידות, מחולקים בשגיאת המדידה, בריבוע).

ההתאמה הטובה ביותר אם כך היא זו לה χ^2 מינימלי. עבור התאמה לקו ישר או לפולינום, ניתן למצוא את פרמטרי ההתאמה בצורה אנליטית על ידי מזעור של χ^2 , ועם התחשבות בשגיאות בציר Y בלבד. לא קיים פתרון אנליטי עבור פונקציות מורכבות יותר, או כאשר יש שגיאות בשני הצירים, לכן ניתן להשתמש בפתרון נומרי.

האלגוריתם הנומרי בו השתמשתי נקרא Levenberg–Marquardt, והוא מבוצע על ידי הפונקציה GeneralFit בעזרת קריאה למספר פונקציות משנה, כולן נמצאות בקובץ המקור המשני generalfit.c.

להלן תיאור של האלגוריתם:

- GeneralFit מקבלת כארגומנטים את פונקציית ההתאמה, נתוני המדידות וניחושים ראשוניים עבור הפרמטרים.
- GeneralFit מחשבת את χ^2 עבור ההתאמה עם הפרמטרים הראשוניים בעזרת הפונקציה CalcChi2 (המחשבת אותו בעזרת השגיאות בשני הצירים).
- GeneralFit מחשבת את גודל ההזזה של הפרמטרים.
- GeneralFit קוראת ל-GradStep בלולאה עד שמושג ערך χ^2 מינימלי (המחושב על ידי CalcChi2):
 - א. GradStep מקבלת כארגומנטים את פונקציית ההתאמה, נתוני המדידות, הפרמטרים הנוכחיים (שמוזזים בכל איטרציה) ואת גודל ההזזה עבור הפרמטרים (גם משתנה כל איטרציה).
 1. GradStep קוראת ל-CalcGrad, המקבלת את אותם הארגומנטים, ומחשבת את הגרדיאנט בנקודה במרחב הפרמטרי (עבור הפרמטרים הנוכחיים), אותו היא מחזירה ל-GradStep.
 2. GradStep מזיזה את הפרמטרים הראשוניים בלולאה בכיוון המינימום (לפי הגרדיאנט) ומקטינה את גודל ההזזה עבור הפרמטרים עד שערך χ^2 (מחושב על ידי CalcChi2) מתחיל לרדת (בכל איטרציה הפרמטרים הראשוניים הם אלה שמוזזים).
 3. א. GradStep מזיזה את הפרמטרים בלולאה עד להגעה למינימום של χ^2 (בכל איטרציה מוזזים הפרמטרים החדשים בגודל הזזה קבוע, שנקבע בלולאה הקודמת).
 4. א. GradStep מחזירה ל-GeneralFit את הפרמטרים החדשים ואת המרחק הכולל שהפרמטרים הוזזו בו.
 - ב. GeneralFit משנה את גודל ההזזה עבור הפרמטרים למרחק הכולל שהפרמטרים הוזזו בו על ידי GradStep, והלולאה חוזרת על עצמה עד שמושג ערך χ^2 מינימלי.
- GeneralFit קוראת ל-Errors, המקבלת כארגומנטים את פונקציית ההתאמה, נתוני המדידות ואת הפרמטרים החדשים שהתקבלו. Errors מחזירה ל-GeneralFit את השגיאות של כל אחד מהפרמטרים ואת התלות ההדדית (covariance) ביניהם.
- GeneralFit מחשבת את χ^2 של הפרמטרים החדשים, את מספר דרגות החופש, את χ^2 המצומצם, ואת p-value. היא מחזירה את כל אלו (ואת הפרמטרים החדשים, שגיאותיהם והתלות ההדדית ביניהם, ואת מספר האיטרציות שנעשו).

6. ייצוא של הגרף כתמונה (SaveGraph)

בחירת מיקום הקובץ לשמירה על ידי המשתמש נעשית על ידי פונקציית ספריה של CVI (FileSelectPopupEx). לאחר מכן, נעשה שימוש בפונקציית הספריה SaveCtrlDisplayToFile על מנת לשמור את הגרף לקובץ.

7. בחירת הגרפים המוצגים דרך התפריט (PlotMenuClick)

כיוון שניתן לייצא את הגרף כתמונה, רציתי שתהיה גם אופציה לבחור את השרטוטים המוצגים עליו כדי שלמשתמש תהיה יותר שליטה על הגרף שהוא מייצא. כדי לעשות זאת הוספתי תפריט Plots בחלון של הגרף, ממנו ניתן לבחור איזה שרטוטים מוצגים על ידי לחיצה על שם הגרף המתאים, כאשר השרטוטים המוצגים מסומנים על ידי וי. לחיצה על כל אחד מהם קוראת לפונקציה PlotMenuClick.

אם השרטוט אותו המשתמש בחר כבר מוצג, הפונקציה מסתירה אותו (אך משאירה אותו בזכרון). אם השרטוט אינו קיים, הפונקציה קוראת לפונקציה המתאימה על מנת לשרטט אותו (אותן הפונקציות המשמשות לשרטוט בלחיצה על הכפתורים שבחלון הראשי – PlotMenuClick מדמה לחיצה על הכפתור המתאים). אם השרטוט מוסתר (קיים בזכרון אך אינו מוצג), הפונקציה מציגה אותו.

לחיצה על הכפתורים בחלון הראשי תמיד מציגה את השרטוט המתאים, גם אם הוא הוסתר על ידי המשתמש דרך התפריט, וגם מסמנת אותו ב-וי בתפריט, כך שיש סנכרון בין לחיצה על הכפתורים בחלון הראשי לבין התפריט.

8. הוספת ומחיקת תוויות מידע בגרף (GraphClick)

בגרף מוגדרות מראש 5 annotations ו-cursor אחד, כולם מוסתרים.

לחיצה שמאלית על הגרף קוראת לפונקציה עם `EVENT_LEFT_CLICK`, המכיל ב-`eventdata` שלו את מיקום העכבר ביחס לגרף. פונקציית הספרייה `GetGraphCoordsFromPoint` ממירה את מיקום העכבר לקואורדינטות X ו-Y של הגרף. מיקום ה-cursor בגרף נקבע על פי קואורדינטות אלו, והוא מוגדר כברירת מחדל להיצמד לשרטוט הקרוב אליו ביותר (ניתן לשנות זאת דרך התפריט Graph בחלון הגרף), כך שהוספה של תווית תמיד נצמדת לנקודות של שרטוט כלשהו (דבר זה אינו אפשרי עם annotations בלבד, ולכן ה-cursor נדרש – והוא תמיד מוסתר כיוון שזה השימוש היחיד שלו). כעת הפונקציה לוקחת את הקואורדינטות החדשות של ה-cursor, ומציגה annotation עבורו.

בגרף מוגדרות מראש 5 annotations, ובכל לחיצה שמאלית על הגרף האינדקס של ה-annotation מקודם ב-1, וכך בכל פעם מוצגת annotation אחרת. כשכל החמש מוצגות, לחיצה נוספת תחזיר את האינדקס ל-1 ובלחיצה הבאה ה-annotation הראשונה תשנה את הפרמטרים שלה על פי הלחיצה החדשה וכן הלאה. בצורה זו יש מספר מוגבל של annotations כך שלא ניתן להגיע למצב בו הגרף עמוס מדי, וניתן למחוק אותן אחת אחת.

לחיצה ימנית על הגרף קוראת לפונקציה עם `EVENT_RIGHT_CLICK`. במקרה זה הפונקציה מורידה את האינדקס ב-1 ומסתירה את ה-annotation עם האינדקס המתאים, כך שה-annotation האחרונה שהוספה היא זו שמוסתרת, לפי סדר הופכי לסדר ההוספה.

הפונקציה `ClearAnnotations` לה ניתן לקרוא דרך התפריט `Edit` בחלון הגרף עוברת בלולאה על כל ה-annotations ומסתירה אותן.

בעיות שהתעוררו ופתרון

1. הדבקת מידע מה-Clipboard

לאחר שכתבתי את הקוד לפורמטיזציה של מידע מה-Clipboard (בפונקציה `PasteData`), הפונקציה החזירה לי רק את הערכים שהיו בשורה הראשונה בקובץ ממנו העתקתי את המידע, והמשיכה להעתיק אותם לאורך המערכים (כלומר מערך X לדוגמה הכיל את אותה נקודה המון פעמים וכך גם שאר המערכים).

לאחר ניסיונות רבים של שינוי ה-`format string` ששימש אותי בפונקציה `Scan` המשמשת לפורמטיזציה הזו הבנתי שהיא ככל הנראה שונה מהפונקציה `ScanFile` המשמשת לפורמטיזציה של מידע מקובץ בכך שהיא לא מקדמת את הפוינטר של ה-`input string` לסוף השורה בסיום כל איטרציה כמו שהפונקציה `ScanFile` ולכן היא רצה על אותה שורה שוב ושוב. כדי לפתור בעיה זו השתמשתי במספר הביטים שהפונקציה `Scan` סרקה (זהו הערך שהפונקציה מחזירה) על מנת לקדם את הפוינטר באופן ידני.

2. קריאה לפונקציה משתנה

מספר פונקציות בתוכנית נדרשות לבצע הערכה של ערכי Y עבור פונקציית ההתאמה שהתקבלה (למשל על מנת לשרטט את פונקציית ההתאמה). כיוון שיש מספר פונקציות אפשריות, ניתן היה להשתמש ב-`switch` למשל על פי סוג ההתאמה שנבחרה. כיוון שיש פונקציות רבות שנדרשות לבצע זאת, הדבר היה יוצר קוד מסורבל מאוד וארוך.

על מנת לפתור בעיה זו, כשהפונקציה `Fit` קובעת מה היא פונקציית ההתאמה שהשתמש בה, היא נותנת לפוינטר מסוים (לדוגמה פוינטר שנקרא `func`) את הכתובת של הפונקציה הרלוונטית (ישר, פולינום, אקספוננט, גאוסיאן או לגריתים). לאחר מכן הפונקציה `Fit` מעבירה את הפוינטר הזה הלאה אל כל הפונקציות שצריכות להשתמש בו על מנת

לבצע הערכה של הפונקציה. הפונקציות האלה נמצאות בקובץ `fitfunc.c`, מקבלות כארגומנט ערך x , מערך של מקדמים בהם להשתמש להערכה a ומספר מקדמים בו להשתמש na , ומחזירות את ערך Y המתאים.

בכל פעם שצריך לקרוא לאחת הפונקציות האלה להערכה שלהן, הקריאה נעשית על ידי הפוינטר בצורה הבאה:

```
func (x, a, na);
```

3. זמן ריצה ארוך מדי של התאמה פולינומית

לאחר סיום כתיבת הקוד כשניסיתי לבדוק את ההתאמות, להתאמה הפולינומית לקח זמן ארוך מאוד לרוץ (לא מדדתי, אבל לאחר כמה דקות עצרתי את התוכנית כי היא לא סיימה לרוץ). הבעיה נגרמה כיוון שהשתמשתי במקומות רבים באלגוריתם בפונקציות מובנות של CVI לביצוע פעולות על מערכים. כיוון שפעולה כמו חיבור של שני מערכים והעלאה בריבוע צריך לעשות עם שתי פונקציות שונות, במקומות רבים בקוד קראתי למספר פונקציות, שכל אחת מהם מריצה לולאה, והפונקציות האלה נקראות מספר פעמים כיוון שהן מבוצעות בכל איטרציה של האלגוריתם של הזזה של פרמטרי ההתאמה.

כדי לפתור את הבעיה עברתי שוב על הקוד והחלפתי את הפונקציות האלה לביצוע של מספר פעולות – למשל העלאה בריבוע, חיבור של מערכים וחלוקה שדרשו קריאה לשלוש פונקציות שכל אחת מהן רצה על המערך – בלולאה בודדת שמבצעת את כל הפעולות הללו. לאחר מכן הבעיה אכן נפתרה.

4. שרטוט של צלבי שגיאה וגרף שארים

ב-CVI אין אופציה מובנית לשרטוט של צלבי שגיאה, והדבר מהותי לתוכנית שלי.

על מנת לשרטט את צלבי השגיאה (בפונקציה `PlotData`) השתמשתי בלולאה שרצה על מספר הנקודות ומשרטטת קווים ישרים (בעזרת הפונקציה `PlotLine`) עבור השגיאות ב-X והשגיאות ב-Y. השרטוט של כל קו כזה נשמר במערך מתאים (עבור X ו- Y) על מנת שיהיה ניתן להסתיר ולהציג אותם מחדש בעזרת התפריט בחלון של הגרף יחד עם השרטוט של הנקודות עצמן.

בעיה נוספת הייתה כיצד להסתיר או להציג לאחר הסתרה את השרטוט של המדידות (דרך התפריט בחלון הגרף), כיוון שהוא כולל בתוכו גם את כל צלבי השגיאה. את הפתרון לכך השארתי בתוך הפונקציה `PlotData` על ידי הוספת `Event` מיוחד (CVI שומרת טווח של `Events` בתחום מ-1000 ועד 10000 עבור `Events` שהתוכנית יכול להגדיר) המשמש להסתרה/הצגה לאחר הסתרה של השרטוט של המדידות וצלבי השגיאה. הפונקציה `PlotMenuClick` קוראת לפונקציה `PlotData` עם ה-`event` הזה ועם `eventData` שאומר לה אם להציג או להסתיר את השרטוט של המדידות וצלבי השגיאה.

ניתן היה לבצע את ההצגה וההסתרה על ידי `PlotMenuClick`, אך אז במקרה שהשתמש היה מסתיר את השרטוט של המדידות דרך התפריט ולאחר מכן לוחץ על הכפתור להצגה שלו בחלון הראשי, `PlotData` לא הייתה יודעת להציג אותו מחדש (אלא אם אותו הקוד להצגה של השרטוט היה מופיע בשתי הפונקציות, ואז לא הרווחנו כלום).

5. שינוי טווח הנקודות להתאמה עבור ציר X לא בסדר עולה

בהתחלה השתמשתי בפונקציית הספרייה `Subset1D` כדי להעתיק את הנקודות מהמערכים המקוריים למערכים המשמשים להתאמה, אך הפונקציה מעתיקה את כל הערכים מאינדקס מסוים עד אינדקס מסוים, כך שהקוד התבסס על כך שערכי הנקודות ב-X הן בסדר עולה. ניתן כמובן לבקש את המשתמש להזין את הנתונים בצורה כזו שהדבר הוא כך (ניתן לסדר את הנקודות בסדר X עולה באקסל בקלות), אך העדפתי לא להסתמך על כך ולבטל את התלות הזו.

כדי לפתור זאת שכתבתי את הפונקציה `ChangeDataRange` לביצוע של לולאה על מערך הנקודות ב-X והעתקה של הנקודות המתאימות בשאר המערכים רק כאשר ערך ה-X של נקודה נמצא בין הערכים המינימלי והמקסימלי שהמשתמש הזין.

קוד המקור

כל הקבצים מוצגים מהדף הבא. שם הקובץ מופיע בראש כל דף.