

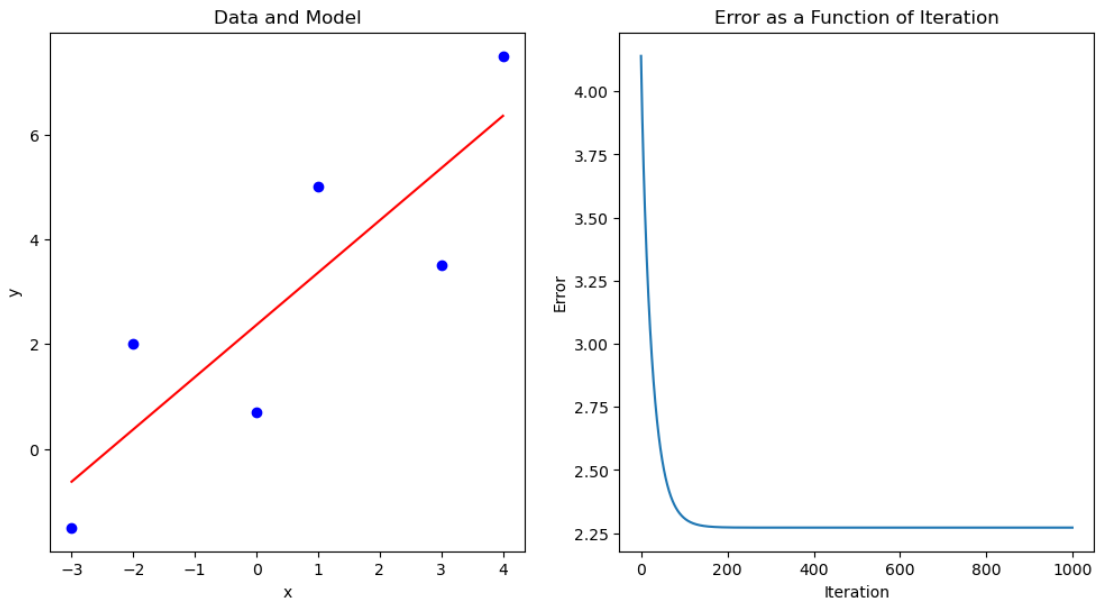
פרויקט סופי

שאלה 1

א.

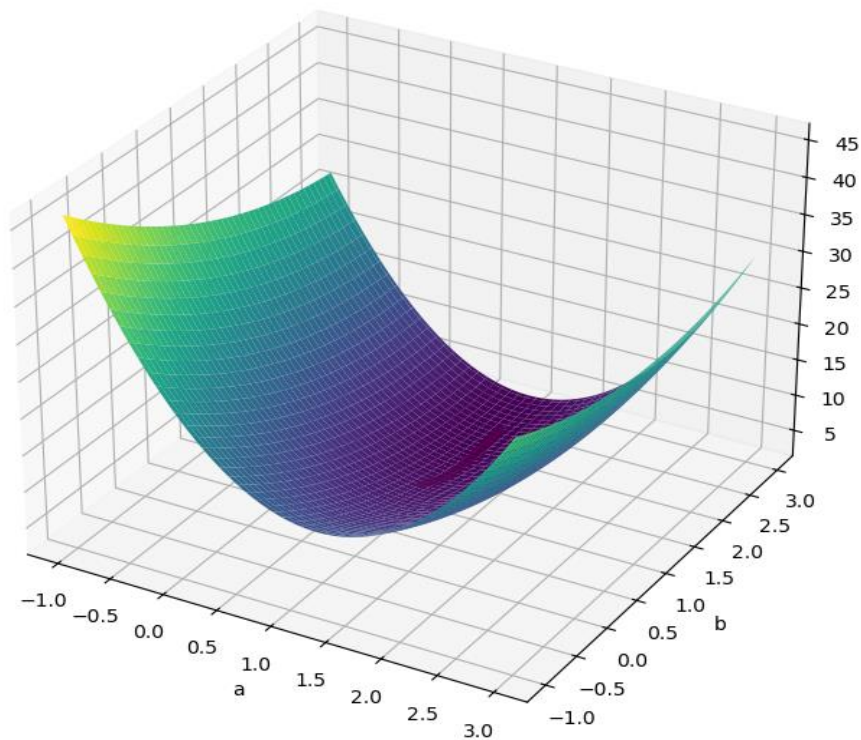
Final parameters: $a=0.997333333849981$, $b=2.367999994270291$

2. ו-3.

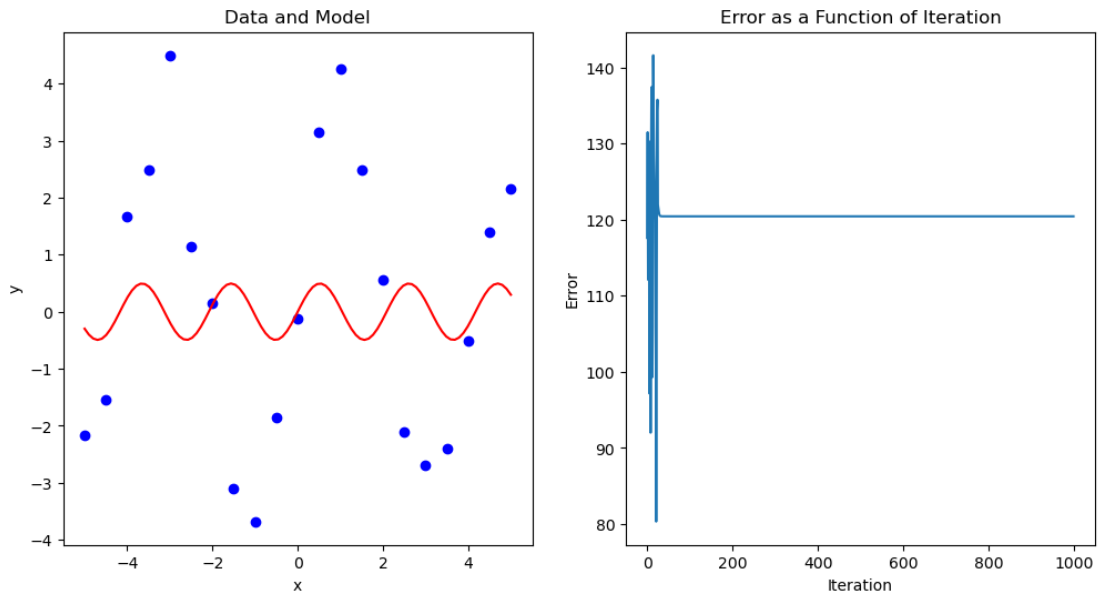


4.

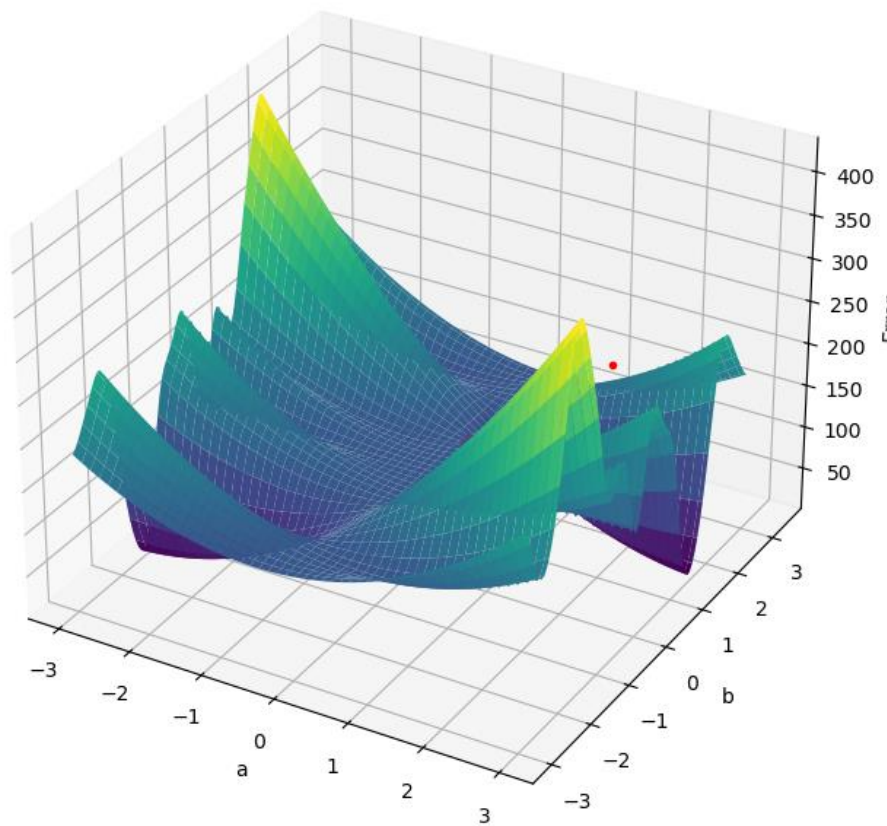
Error Surface Over Parameters



ג.



Error Surface Over Parameters



מגישים:
שחר אשר – 209305408
הדר ליאל הרוש - 211721568

קצב למידה: 0.01

```
# Learning rate  
lr = 0.01 # You might need to adjust the learning rate
```

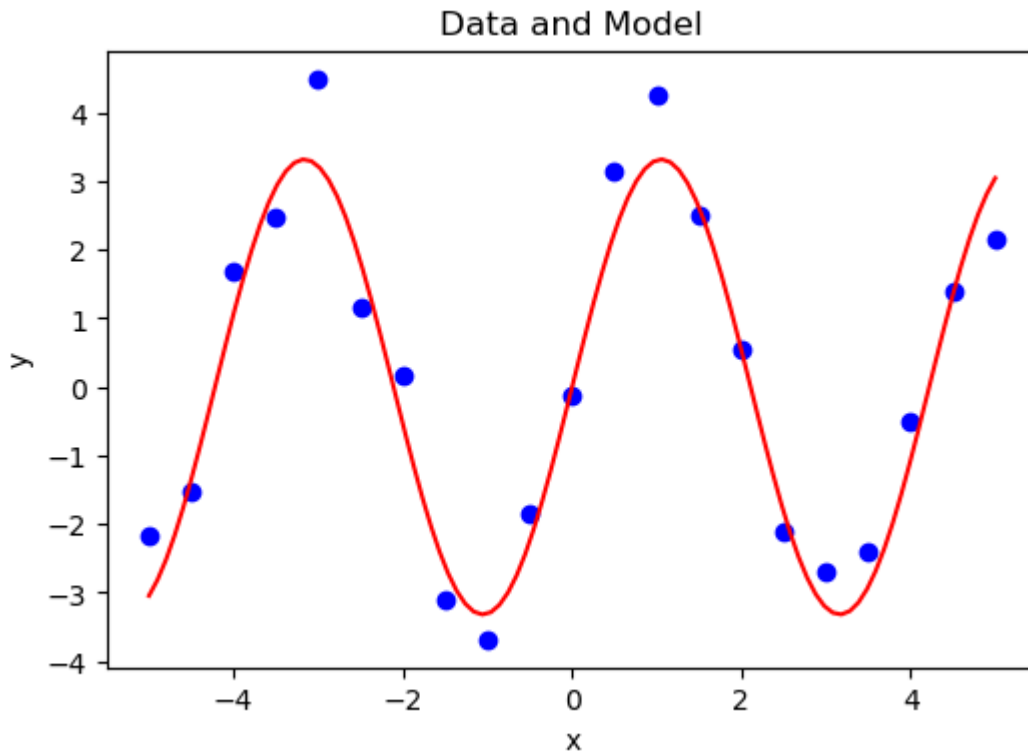
איטרציות: 1000

```
# Number of iterations  
iterations = 1000
```

הפרמטרים שמצאנו הם:

Final parameters: $a=0.4946555998532448$, $b=3.0123813348953465$

.T



פרמטרים סופיים:

Final parameters: $a=3.318145436025057$, $b=1.488977981749276$

נראה שהמודל מתאים באופן טוב מאוד לנתונים. הקו האדום (המודל) עובר דרך רוב הנקודות הכחולות (הנתונים). זה מראה שהפונקציה הסינוסואידלית מתאימה היטב למבנה של הנתונים. הפרמטרים הסופיים שהפונקציה `curve_fit` מצאה הם:

- $a = 3.318145436025057$

- $b = 1.488977981749276$

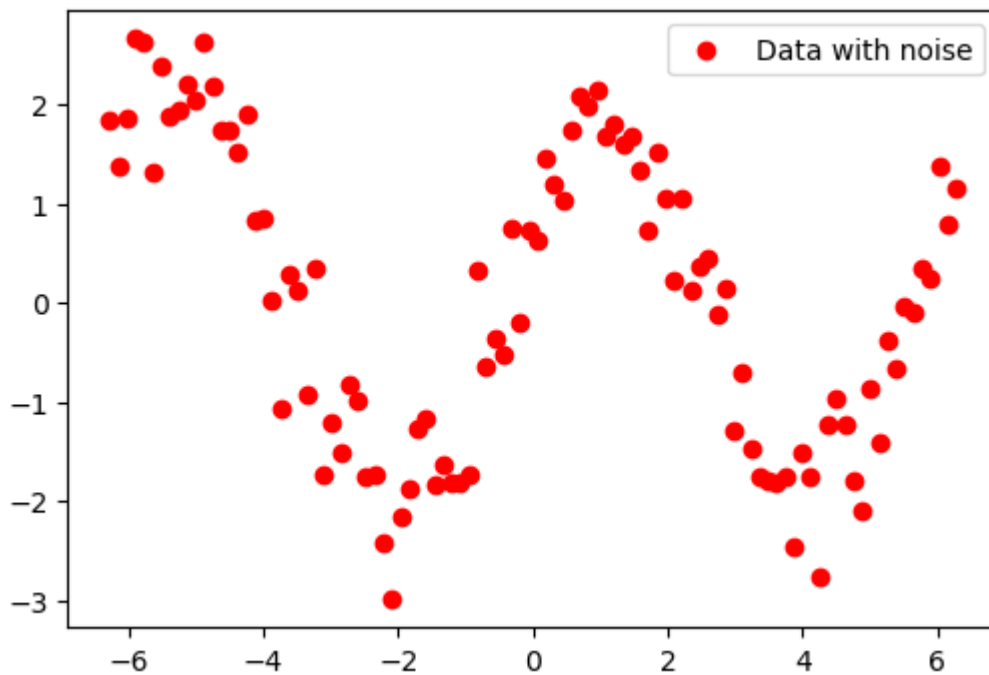
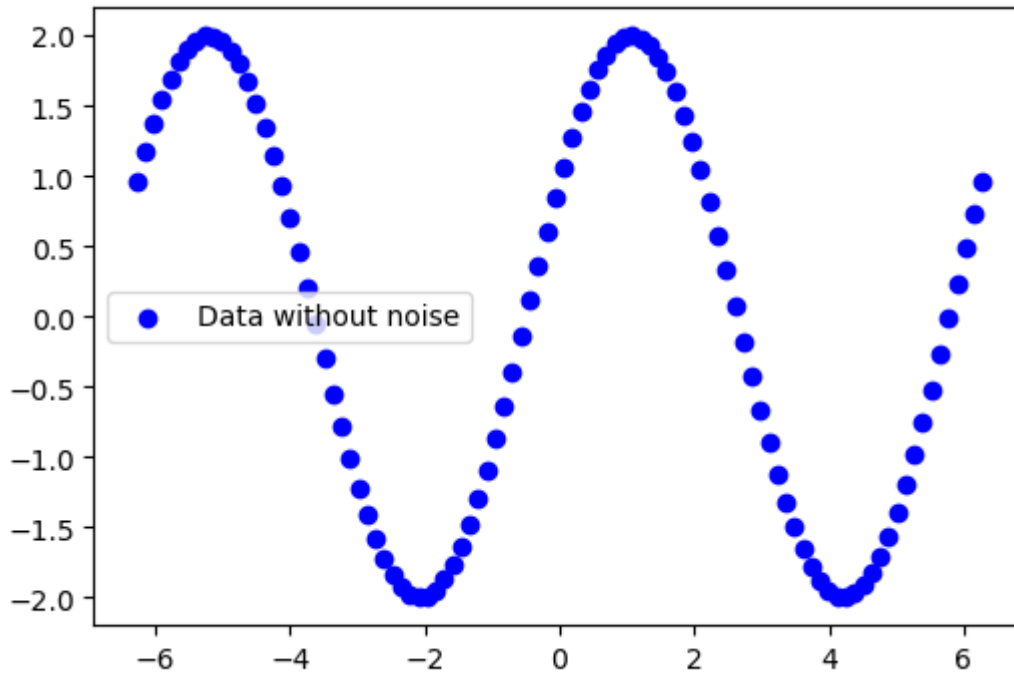
הפרמטר a הוא הערך שבו הפונקציה הסינוסואידלית מתחילה (הגובה שלה), והפרמטר b הוא התדירות של הפונקציה. אם נשווה את הערכים האלה לערכים האמיתיים שהשתמשת בהם כדי ליצור את הנתונים, נראה שהפונקציה `curve_fit` מצאה פרמטרים שדי קרובים לערכים האמיתיים. זה מראה שהפונקציה `curve_fit` יכולה להיות כלי יעיל למציאת הפרמטרים של מודל שמתאים לנתונים.

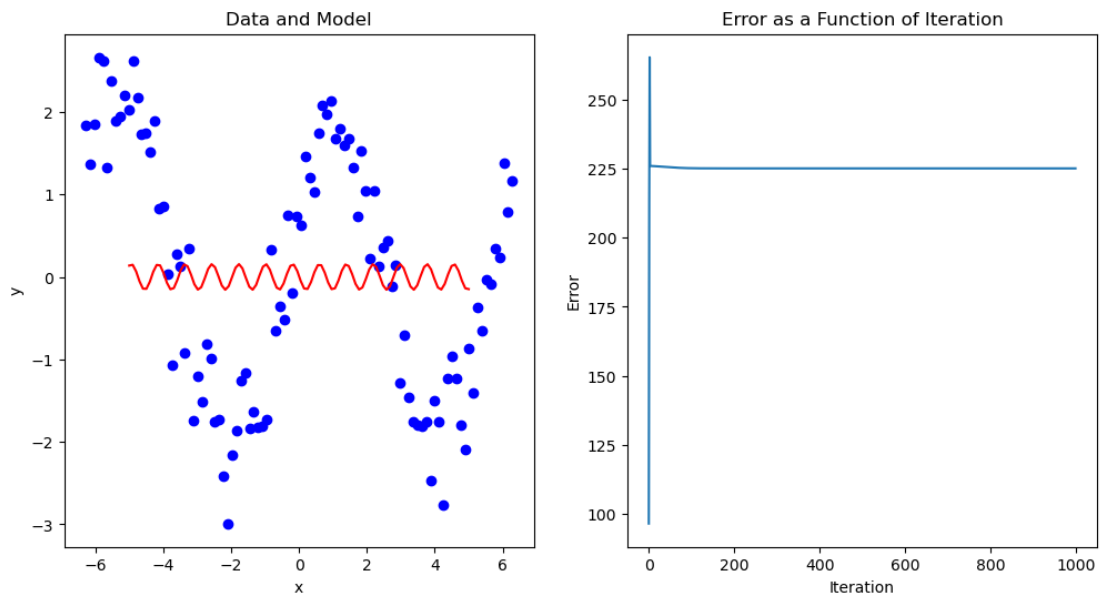
ה. המודל שבחרו הוא: $f(x) = a \cdot \sin(b \cdot x + c)$.

a, b ו- c הם הפרמטרים של המודל.

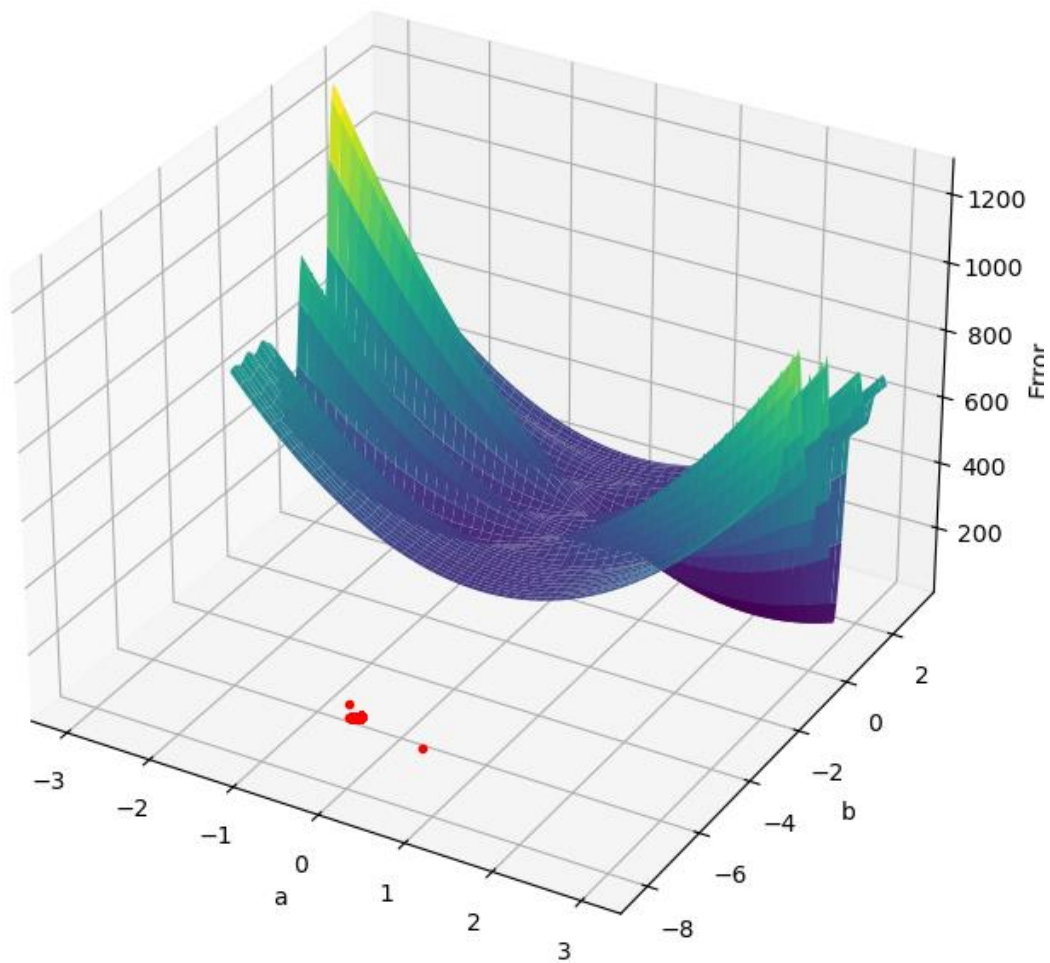
הפרמטרים ההתחלתיים שבחרנו הם: $a_val = 1, b_val = 1, c_val = 1$

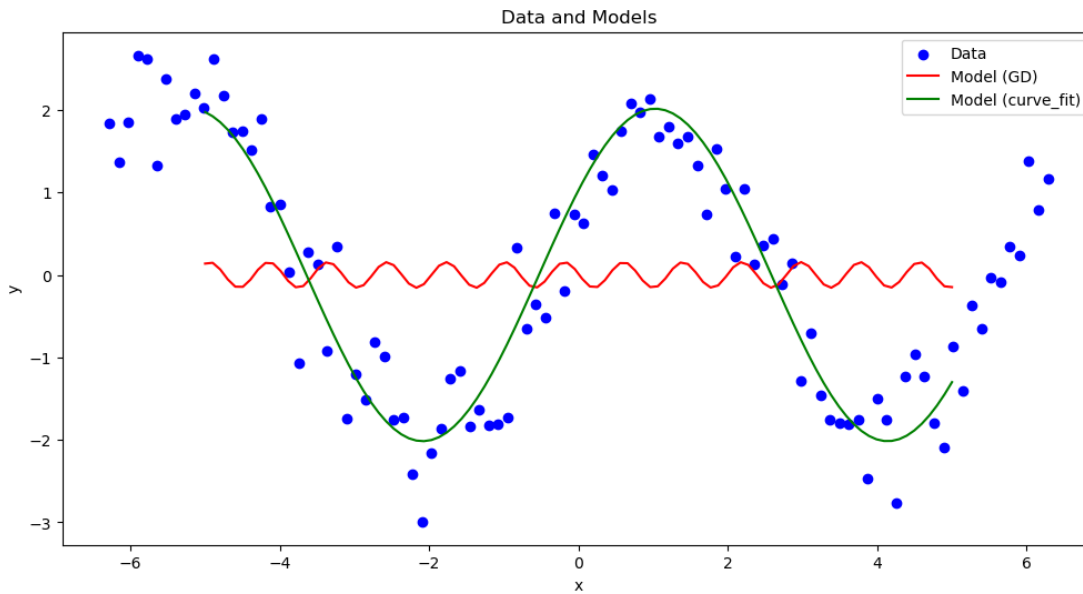
```
# Initial parameters  
a_val, b_val, c_val = 1, 1, 1
```





Error Surface Over Parameters





פרמטרים שנמצאו על ידי Gradient descent:

Final parameters: $a=0.15534552265847998$, $b=-7.934037154730188$,
 $c=0.0804089890396507$

פרמטרים שנמצאו על ידי curve_fit:

Parameters found by curve_fit: $a=2.0156616351282866$,
 $b=1.0093224363174194$, $c=0.5359304117967073$

הזמן שלוקח ל- Gradient descent:

Time elapsed for gradient descent: 2.051109552383423 seconds

הזמן שלוקח ל- curve_fit:

Time elapsed for curve_fit: 0.0010044574737548828 seconds

ההבדלים בין שתי השיטות, Gradient descent (GD) ו-curve_fit :

- **זמן הריצה:**

- Gradient descent: לקח כ-2.05 שניות למצוא את הפרמטרים האופטימליים.
- curve_fit: שיטה זו הייתה מהירה באופן משמעותי, ולקחה רק כ-0.001 שניות.

- **הפרמטרים הסופיים:**

- Gradient descent: הפרמטרים הסופיים היו $a=0.1553$, $b=-7.9340$, $c=0.0804$.
- curve_fit: הפרמטרים היו שונים באופן משמעותי עם הערכים $a=2.0157$, $b=1.0093$, $c=0.5359$.

- **איכות ההתאמה (כפי שנראית מהגרף):**

- המודל של Gradient descent: הקו האדום אינו מתאים לנקודות הנתונים באופן מדויק; נראה שהמודל מתחת למידה הנדרשת.
- המודל של curve_fit: הקו הירוק מתאים לנקודות הנתונים באופן מדויק יותר, ומתפס את המגמה הבסיסית באופן יעיל.

ההבדלים נובעים מהתכונות הייחודיות של כל שיטה:

- **Gradient descent:** זהו אלגוריתם אופטימיזציה איטרטיבי שמשמש לעיתים קרובות למציאת המינימום של פונקציה; במקרה זה, ייתכן שהוא לא התכנס לערכים האופטימליים במהלך האיטרציות או שנתקע במינימום מקומי, מה שהוביל להערכות פרמטר שאינן אופטימליות.
- **curve_fit:** בדרך כלל משתמשת בריבועים פחותים לא ליניאריים להתאמת פונקציה לנתונים, שיכולה להיות יעילה ומדויקת יותר אך יכולה גם להתלוות על הערכות פרמטר ראשוניות.

1. $(p_1 \text{ and not } p_2) \text{ or } (\text{not } p_3 \text{ and not } p_4)$

2. $(p_1 \text{ and } p_5) \text{ or } (p_6 \text{ and not } p_9)$

3. $(p_2 \text{ and } p_8) \text{ or } (p_7 \text{ and not } p_3)$