**סדנת עיבוד נתונים**

בסדנה זו תלמדו לעבד את נתוני המעבדה ב**עזרת שפת Python**. אנחנו נשתמש בתוכנה Spyder לכתיבה והרצה של הקוד. המטרה היא ליצור מיומנות בסיסית של ניתוח נתונים והצגתם בגרפים כדי שתוכלו לעבד את הנתונים של המעבדה בקלות. בנוסף, הקוד שתכתבו בסדנה יישמר אצלכם ותוכלו לחזור ולהיעזר בו בעתיד.

**היקף הסדנה** הוא **שני מפגשים של 3 שעות** בסופן יהיו ברשותכם שני קבצי קוד שתיצרו בשמות **tutorial.py**  ו-**data\_analysis.py** , אלו יכילו את כל הפקודות שלמדתם ותוכלו לעשות בהם שימוש במהלך המעבדה.

במפגש הראשון נגיע עד וכולל ניסוי מיפוי פוטנציאל.

# תוכן הסדנה

[חלק א' – התקנת Anaconda ו-Spyder 1](#_Toc95753580)

[חלק ב' – סביבת עבודה ב-Spyder 1](#_Toc95753581)

[חלק ג' – שימוש בסיסי ב-Python 1](#_Toc95753582)

[חלק ד – עיבוד נתוני המעבדה 2](#_Toc95753583)

[מיפוי פוטנציאל חשמלי 2](#_Toc95753584)

[קבל לוחות 3](#_Toc95753585)

[חוק אוהם 6](#_Toc95753586)

[השראות 7](#_Toc95753587)

# חלק א' – התקנת Anaconda ו-Spyder

**בצעו** התקנה של **Anaconda**:

<https://docs.anaconda.com/anaconda/install/windows/>

**התקנה זו מתקינה גם Spyder**.

# חלק ב' – סביבת עבודה ב-Spyder

הערכת זמן: כ- 15 דק'

1. **פתחו** את Spyder.
2. **צרו קובץ חדש** **tutorial.py** ושמרו אותו בתיקיה שתוקדש לסדנה.
3. **עיברו** עם העכבר מעל האייקונים של פקודות ההרצה

Graphical user interface, application, Teams

Description automatically generated

ו**קראו** את קיצורי המקשים של הרצת קוד ב-Spyder.

1. הסתכלו על הממשק של Spyder **וזהו** את החלקים הבאים:
   1. סרגל פקודות ותפריטים, תיקיית ההרצה.
   2. חלון הקובץ – אזור כתיבת הקוד.
   3. **ה-Console**, שימו לב שישנם **שני Tabs** לחלון זה.
   4. **חלון עזרה**, שימו לב שישנם **4 tabs: Help, Variable Explorer, Plots, Files**
2. **שנו** את תיקיית ההרצה לתיקיית הסדנה – כדי שהקוד יוכל למצוא את הקבצים שנמצאים בה.

# תפריט הלמידה המוצע באתר learnpython חלק ג' – שימוש בסיסי ב-Python

הערכת זמן: כשעה וחצי

**היכנסו** לאתר, כאן נלמד את השיעורים המוגדרים *כלימודי הבסיס וניתוח נתונים* - <https://www.learnpython.org/en/Welcome>

אנחנו נבצע את השיעורים עד וכולל Pandas (ראו רשימה). אנחנו נקדיש לכך כ**שעה וחצי.** במידה ויש צורך תוכלו להמשיך.

דגשים:

1. במקום להריץ באתר – **העתיקו** את הקוד לקובץ ב-Spyder ו**הריצו** את הפקודות שם. באופן זה תשמרו את כל מה שלמדתם.
2. אם אתם מבינים את דוגמת הקוד, אתם לא חייבים להריץ אותה. תעתיקו ותריצו רק מה שאתם רוצים לוודא שאתם מבינים.
3. **התמקדו** בלהבין את הדוגמאות (ופחות בהבנת הטקסט).
4. **בצעו** את התרגיל בסוף כל שיעור באתר (ב-Spyder).

\*לאחר הסדנה, אתם מוזמנים לבצע את שאר השיעורים אם אתם רוצים.

**שמרו** את הקוד שכתבתם בתיקיה.

**קחו** לכם הפסקה קטנה של רבע שעה.

הלינק אצלכם, הקוד אצלכם, תוכלו להמשיך ולהתקדם בכל זמן שתרצו, **הכל זמין עבורכם**.

# חלק ד – עיבוד נתוני המעבדה

באופן כללי, הפעולות הנדרשות לעיבוד הנתונים הן:

1. כתיבת וקטור נתונים (למדתם בשיעור Numpy Arrays)
2. חישוב עקום תיאורטי
3. טעינת נתונים מקובץ
4. הצגת נתונים בגרף
5. ביצוע חישוב על וקטורי נתונים (כתיבת פונקציה)
6. ביצוע רגרסיה לינארית
7. ביצוע fit (לעקום לא לינארי)
8. ביצוע אינטגרציה לווקטור נתונים

בשאר הסדנה תלמדו ותבצעו את כל הפעולות האלה, על נתונים מניסויי המעבדה.

כדי לעבד את הנתונים, נשתמש בmodules הבאים:

import numpy as np # math functions

import scipy # scientific functions

import scipy.stats # contains linregress (for linear regression)

import matplotlib.pyplot as plt # for plotting figures and setting their properties

import pandas as pd # handling data structures (loaded from files)

from scipy.optimize import curve\_fit as cfit # non-linear curve fitting

1. **פתחו** קובץ קוד חדש בשם **data\_analysis.py**
2. **העתיקו** את השורות האלה לתחילת הקובץ של הקוד **והריצו** אותן.

נתחיל לעבור יחד על ניסוי המעבדה באמצעות רקע קצר על מנת שתוכלו לנתח את הנתונים

## תרגיל - מיפוי פוטנציאל חשמלי

**רקע תאורטי**

בניסוי מיפוי פוטנציאל חשמלי אתם תמדדו את הפוטנציאל החשמלי על דף מוליך שמחוברות אליו אלקטרודות. בתרגיל הזה אנחנו נשרטט את הפוטנציאל החשמלי התיאורטי של שתי אלקטרודות ונחשב במפורש את שינוי הפוטנציאל מאלקטרודה אחת לשנייה.

נניח שישנן שתי אלקטרודות (חיובית ושלילית), בנקודות ו-. הפוטנציאל מסביב לאלקטרודה בודדת הוא:

*כאשר מתקיים (מהגדרה זו) . הפוטנציאל משתי אלקטרודות הוא סכום הפוטנציאלים מכל אחת מהן. נבחר את להיות ראשית הצירים, איפה ש- ונקבל*

***המשימה*** *- נשרטט את הפוטנציאל.*

1. ***פתחו*** *section חדש בקוד (#%%) בשם potential*
2. ***העתיקו*** *את השורות הבאות:*

C = 1

a = 1

L = 3

N = 100

coord = np.linspace(-L, L , N)

coord\_x, coord\_y = np.meshgrid(coord, coord)

1. ***הריצו*** *את הקוד ו****הסתכלו*** *ב-Variable Explorer על משתני הקואורדינטות.*
2. ***כתבו*** *פונקציה שמחשבת את הפוטנציאל: def potential(x, y, a, C). (****היעזרו*** *ב: np.log(), np.sqrt(), np.power())*
3. ***חשבו*** *את הפוטנציאל בקואורדינטות הנתונות* ***וציירו*** *אותו באמצעות השורות הבאות:*

V\_xy = potential(coord\_x, coord\_y, a, C)

plt.figure()

plt.pcolormesh(coord\_x, coord\_y, V\_xy)

plt.colorbar()

1. *כעת נוסיף קווים שווי-פוטנציאל, בהם הפוטנציאל הוא בעל ערך קבוע:*

plt.contour(coord\_x, coord\_y, V\_xy, np.sort([-1 , 0 , 1]), cmap='hot')

***הוסיפו*** *ערכים קבועים לציור, כך שיהיו בסך הכול 9.*

*כעת נצייר את הפוטנציאל שבין שתי האלקטרודות (על ציר X):*

1. ***הכינו*** *וקטור של x,* ***חשבו*** *עבורו את V\_x וציירו את העקום בעזרת plt.plot(x,V\_x,'.')*

***מה למדנו?***

*סימלצנו פוטנציאל חשמלי (יצרנו עקום תיאורטי) והצגנו אותו בשלושה סוגי גרפים: (plot, pcolormesh, contour).*

## תרגיל - קבל לוחות

**רקע תאורטי**

קיבול הוא היחס בין המטען החשמלי למתח של גוף מוליך:

הקיבול של קבל לוחות עם לוחות עגולים בקוטר D ומרווח d בין הלוחות הוא:

*כאשר הוא המקדם הדיאלקטרי של ריק. כשנותנים לקבל C להיפרק דרך נגדים בעלי התנגדות כוללת של , המתח שעל הקבל דועך בצורה אקסופננציאלית:*

*כש- הוא קבוע הדעיכה. אם נבצע ln לשני האגפים נקבל קשר לינארי:*

*הזרם שזורם בקבל זורם גם בנגד R ולכן שווה למתח הנגד חלקי ההתנגדות:*

*מכיוון שהמטען הוא אינטגרל של הזרם, מקבלים:*

**המשימה:** עיבוד נתוני הניסוי

1. **פתחו** section חדש בקוד בשם capacitor
2. בהינתן **חשבו** את הקיבול C\_theoretical

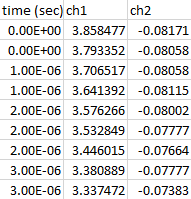
eps0 = 8.854e-12 # F/m

D = 18e-2 # m

d = 0.5e-3 # m

1. בהינתן , **חשבו** את tau\_theoretical

הקובץ capacitor.csv מכיל מדידות של הפריקה:



המתח על הנגד R הוא ch2 והמתח על הקבל הוא ch1-ch2.

1. **טענו** את הנתונים שבקובץ באופן הבא:

C\_data = pd.read\_csv('capacitor.csv')

**הסתכלו** ב-Variables Explorer איזה מידע מופיע ב-C\_data.

1. **שנו** את השם של עמודת הזמן ל- "t" ואת השם של “ch2” ל-"V\_R":

C\_data = C\_data.rename(columns = {"time (sec)":"t", "ch2":"V\_R"})

1. **הוסיפו** עמודה של ל-C\_data:

C\_data["V\_C"] = C\_data["ch1"] - C\_data["V\_R"]

1. **ציירו** את מתח הקבל כתלות בזמן

(**רמז:** למדנו לצייר גרף בתרגיל מיפוי פוטנציאל חשמלי – אתם יכולים להעתיק את הפקודות משם).

1. **בצעו** fit לגרף בעזרת הקוד הבא, המכיל כתיבה של פונקציה המתארת את המודל אליו אנחנו מבצעים את ההתאמה וקריאה לפונקציה curve\_fit (אותה כינינו cfit בפקודת ה-import בתחילת הקוד):

def V\_decay(t,a,b):

return a\*np.exp(-t\*b)

fit2 = cfit(V\_decay,C\_data['t'], C\_data["V\_C"])

1. **הוסיפו** לגרף הנתונים את עקום ה-fit באופן הבא:

plt.plot(C\_data['t'], V\_decay(C\_data['t'],fit2[0][0],fit2[0][1]))

1. על גרף נפרד, **ציירו** את לוג המתח כתלות בזמן. אמור להתקבל קשר לינארי. **והוסיפו** grid: plt.grid(“both”)
2. לפי הגרף, **בחרו** מקטע זמנים בו מתקיים הקשר הלינארי, **צרו** וקטור של אינדקסים של מקטע זה, **וציירו** את המקטע שבחרתם:

inds = (C\_data['t'] > t1) & (C\_data['t'] < t2)

plt.plot(C\_data['t'][inds], np.log(C\_data["V\_C"])[inds],'.')

כעת נבצע רגרסיה לינארית כדי לחלץ את V\_0 ואת tau.

1. **בצעו** רגרסיה לינארית:

reg2 = scipy.stats.linregress(C\_data['t'][inds], np.log(C\_data["V\_C"][inds]))

print(reg2)

1. **חשבו** מתוך reg2.slope ו-reg2.intercept את V\_0\_reg ואת tau\_reg
2. **בצעו** אינטגרציה למתח על הנגד ושמרו את התוצאה ב-C\_data[“int\_V\_R”]. (השתמשו ב-scipy.integrate.cumtrapz(V\_R, x = t, initial = 0))
3. **שרטטו** את מתח הקבל כתלות באינטגרל של מתח הנגד. אמור להתקבל קשר לינארי. אם הוא לינארי רק במקטע – **הגדירו** את האינדקסים של מקטע זה ב-inds2.
4. **בצעו** רגרסיה לינארית reg3 (אתם מוזמנים להעתיק את הקוד ממקודם) **וחשבו** את הקיבול C\_meas.
5. **הוסיפו** לגרף את עקום הרגרסיה, מקרא, grid וכותרות לצירים:

plt.plot(C\_data["int\_V\_R"],C\_data["int\_V\_R"]\*reg3.slope + reg3.intercept)

plt.xlabel("integral of V\_R")

plt.ylabel("V\_C")

plt.legend(["data","regression"])

plt.grid("both")

התרשמו ממידת ההתאמה של הערכים התיאורטיים לערכים שחישבתם מהנתונים. הם צריכים להיות לפחות מאותו סדר גודל.

**מה למדנו?**

למדנו כיצד לטעון נתונים מקובץ, לבצע עליהם חישובים, להוסיף עמודות נתונים, לחתוך מתוך הנתונים את המקטע הרצוי, לבצע אינטגרציה, לבצע fit ורגרסיה לינארית ולעצב את הגרפים.

(אם אתם לא חושבים שזה מה שלקחתם מהתרגיל –אנחנו מציעים בחום, קראו שוב את הקוד שיצרתם - הכול שם)

שימו לב! כעת אתם יודעים לבצע את כל הפעולות הנדרשות לעיבוד הנתונים – ראו את [הרשימה מעלה.](#_חלק_ד_–)

**התרגילים הבאים הם תרגול של פעולות אלו.**

## תרגיל - חוק אוהם

**רקע תאורטי**

כאשר זורם זרם בנגד, ההספק המתבזבז עליו הוא

*והאנרגיה הזאת הופכת לאנרגית חום:*

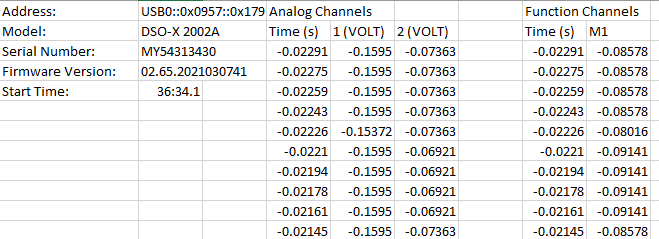
*טמפרטורת הנגד עולה, לפי קיבול החום שלו:*

*וכאשר הוא מתחמם, ההתנגדות של הנגד משתנה באופן לינארי:*

*כלומר, בסך הכול:*

*וזהו הקשר הלינארי אותו מוצאים בניסוי.*

*הקובץ ohm.csv מכיל את המדידות הבאות:*

**

כאשר המתח על הנגד המתחמם הוא המתח בערוץ 1 פחות המתח בערוץ 2, והזרם הוא המתח בערוץ 2 חלקי התנגדות R1 של 5.48 אוהם.

1. **צרו** section חדש בשם Ohm
2. **כתבו** פונקציה שמחשבת זרם: def I\_R(V2, R1)
3. **כתבו** פונקציה שמחשבת את המתח שעל הנגד: def V\_R(V1, V2)
4. **כתבו** פונקציה שמחשבת את התנגדות הנגד: def R\_t(V\_R, I\_R)
5. **כתבו** פונקציה שמחשבת את ההספק שמתבזבז בנגד: def P\_t(V\_R, I\_R)
6. **כתבו** פונקציה שמחשבת את האנרגיה שמחממת את הנגד: def Energy(P\_t, t)
7. **טענו** את הנתונים מהקובץ: R\_data=pd.read\_csv(“ohm.csv”, header=1)
8. **שנו** את שם העמודה של הזמן ל-t ואת שמות הערוצים ל-V1, V2.
9. **השתמשו** בפונקציות שכתבתם וחשבו את התנגדות הנגד כתלות בזמן ואת האנרגיה כתלות בזמן.
10. **שרטטו** גרף של ההתנגדות כתלות באנרגיה.
11. **בצעו** רגרסיה לינארית (לקטע הלינארי).
12. **חשבו** את ואת .

## תרגיל - השראות

בניסוי זה אתם תעבדו על סט של קבצי נתונים – **וזו חשיבות התרגיל!**

**רקע תאורטי**

כאשר יש שינוי בשטף המגנטי שבתוך סליל, מתפתח עליו מתח שנקרא כא"מ (כוח אלקטרו מניע) לפי הנוסחה הבאה:

במערכת הניסוי, מפילים מגנט דרך זוג סלילים ומודדים את המתחים שמתפתחים עליהם. מכיוון שהמתח תלוי במהירות המגנט, צריך לאפיין את התנועה שלו – כלומר, את המהירות ההתחלתית ואת התאוצה שלו.

לשם כך, נמדדו חמישה סטים של מתחים [Trace 0.csv, …., Trace 4.csv], כאשר הסליל העליון (ref) הוחזק במיקום קבוע והסליל התחתון (signal) הורחק ממנו ב-

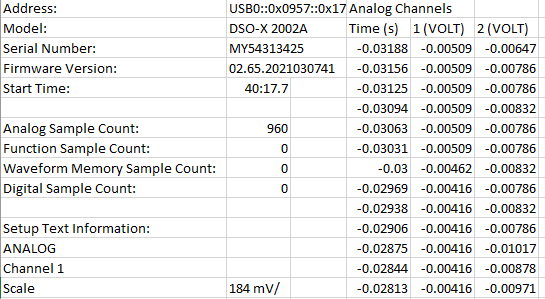
*סנטימטרים.*

*נגדיר את זמן המעבר בכל סליל בתור הזמן בו השטף הוא מקסימאלי. אם נקבע את להיות זמן המעבר בסליל הראשון, אז הזמן בו עובר המגנט את הסליל השני מקיים:*

*וניתן לרשום:*

1. ***פתחו*** *section חדש בשם inductance*

*קבצי המדידה נראים כך:*

**

*כאשר ערוץ 1 הוא המתח בסליל הראשון, הקבוע, וערוץ 2 הוא המתח בסליל הזז.*

1. ***צרו*** *את L בתור numpy array (np.array([…]))*
2. ***טענו*** *את המדידות באמצעות לולאת for:*

*שימו לב לשם הקבצים.*

Ind\_data = []

for n in range(0,5,1):

df = pd.read\_csv('Trace %d.csv'%(n,),header = 1)

Ind\_data.append(df)

*משלב זה, אתם מוזמנים לערוך את לולאת ה-for.*

1. ***שנו*** *את השמות של וקטורי הנתונים: זמן ל-“t”, ערוץ 1 ל-“ref” וערוץ 2 ל-"signal"*
2. ***כתבו*** *פונקציה לחישוב השטף המגנטי מתוך המתח (בערך מוחלט).*
3. ***כתבו*** *פונקציה שמחזירה את הזמן בו השטף מקסימאלי (np.argmax()).*
4. *לכל מדידה,* ***חשבו*** *את זמני המעבר בשני הסלילים ושמרו את הפרש הזמנים בווקטור נתונים t\_coil.*
5. ***שרטטו*** *גרף לינארי מתוך L ו-t\_coil (לפי הנוסחה).*
6. *נניח שגיאה של 0.2 ס"מ ב-L ושגיאה של 10 מילישניות בזמן t\_coil.* ***קראו*** *את התיעוד של הפונקציה matplotlib.pyplot.errorbar()* ***וצרו*** *גרף שמציג שגיאות אלו (יש לחשב שגיאות נגררות).*
7. ***חשבו*** *את המהירות ההתחלתית ואת התאוצה בעזרת רגרסיה לינארית.*
8. ***הוסיפו*** *את עקום הרגרסיה לגרף עם השגיאות.*