24 יוני 2024

הנדון: הוצאת נתונים ממקלטי GNSS אזרחיים

## כללי

* 1. במהלך השנים האחרונות ביצענו שימוש במקלטי GNSS אזרחיים ברחבי העולם בעיקר לטובת ניתוח שגיאות יונוספירה לאורך השנים שעברו.
  2. במהלך המחקר פותח באוניברסיטת אריאל כלי המבוסס python להורדת נתונים מהאינטרנט, עיבוד שלהם וסידור ב-data frames שקל לעבוד איתם.
  3. במסמך זה נפרט אודות אותו כלי והשימוש בו.

## מקורות מידע

* 1. התחנות מהן ניתן לשאוב את המידע מפורטות ב - <https://network.igs.org>, ארבעת האותיות הראשונות ב-site name זהו שם התחנה, ניתן לראות גם את מיקומה בעולם, סוג המקלט, מתי הופעלה לאחרונה וכו'...
  2. המקורות מהם הקוד מוריד את הנתונים הם:
     1. <https://cddis.nasa.gov/archive/gnss/products>.
     2. <https://urs.earthdata.nasa.gov/archive/gnss/products>.
  3. הנתונים מאוגדים באתר של Nasa בשם [CDDIS](https://cddis.nasa.gov/) בו ניתן למצוא מגוון מידעים, בפרט מידע רב על תחנות GNSS ברחבי העולם.
  4. בשביל להוציא את המידע נדרש להירשם לאתר (בחינם ובקלות), בהמשך נראה איך ניתן להשתמש בשם המשתמש והסיסמה להורדה אוטומטית.
  5. להורדה ידנית של קבצים ניתן ללחוץ על: CDDIS -> Data and Products -> GNSS ומשם ניתן לבחור את המוצר הרצוי ולקבל הסבר מפורט עליו ועל הפורמט שלו.

## סוגי קבצים

* 1. **קובץ Rinex Observable** (סיומת .o) – קובץ בפורמט Rinex המכיל את מדידות הPR מכל לוויין ועוד מעט תיקונים בתחנה מסוימת. הקובץ יכול להיות שעתי, ויכול להיות יומי. הסבר מפורט יותר לגבי הפורמט ניתן למצוא באינטרנט, הסבר על איך השמות של הקבצים עובדים ניתן למצוא באתר CDDIS.
  2. **קובץ Rinex Navigation** (סיומת .n) – קובץ של תחנה אחת בו מוצגות כל קליטות הEphemeris (מיקומי הלוויינים בשמיים וההודעות שלהם למשתמש בקצה) לאורך היום. הקליטות האלה מתקבלות לכל לוויין שנקלט בתחנה באותו הזמן. **\*\*קובץ זה אינו בשימוש בסימולציית ה-python עקב שימוש בהרחבה נוחה יותר שלו לכלל הלוויינים בשמיים\*\***
  3. נשים לב כי שילוב של שני הקבצים הללו יכול לאפשר לנו פתרון ניווט מלא. לקבצי ה-Rinex גרסאות שונות, בכל גרסה הפורמט שונה מהותית.
  4. **קובץ SP3** (סיומת .SP3) – קובץ הרחבה לקבצי ה-Rinex Navigation, בקובץ זה מוצגים המסלולים וההודעות של כלל הלוויינים בשמיים במהלך היום ללא תלות בתחנה הקולטת (כלומר רואים את המיקומים של כולם בשמיים בכל רגע נתון).
  5. **קובץ Ionex** (סיומת .INX/.YYi כאשר YY זוהי השנה הרלוונטית) – קובץ בפורמט מיוחד בו מוצגות מפות ה-VTEC בשמיים לאורך היום. בעזרתו ניתן להבין את מצב היונוספרה ולנסות גם לתקן שגיאות.
  6. לקובץ זה מגוון גרסאות שונות מבחינת איכות המידע המועבר בו, ישנן מפות תחזית שמועלות אפילו לפני שמגיע אותו היום, מפות אלה באיכות ובדיוק גרוע, ישנן מפות שמועלות ישר אחרי, בדיוק ובאיכות סבירים, וישנן מפות המועלות ימים אחרי בעלי איכות ודיוק גבוהים מאוד. ניתן לזהות את האיכויות השונות על ידי ההתחלות השונות בשם הקובץ כדוגמת c1p, c2p, igs וכו'...
  7. **קובץ CLK** (סיומת .clk) – קובץ המכיל תיקוני שעון בכדי להגיע למיקום המושלם של המקלט באותו הזמן, משמש בעיקר למציאת הרפרנס אליו אנו רוצים להשוות את פתרון הניווט שקיבלנו. הקובץ גם כאן הוא יומי. מכיוון שהמקלט כמעט ולא זז (תזוזה בעיקר של לוחות טקטוניים ופעילויות נוספות באדמה) **gLAB (המוסבר מטה) יודע להשתמש גם במיקום המשוער של עצמו הנמצא בתוך קבצי ה-RINEX ומתעדכן פעם בכמה זמן גם בלי קבצי ה-CLK, דבר זה אינו ישפיע על הניתוח**.

## הורדת gLAB

* 1. gLAB הינו המקלט בו אנו משתמשים למציאת פתרון הניווט לאורך היום בעזרת הקבצים המוסברים לעיל. למקלט זה ישנו GUI בו ניתן להטעין את הקבצים ולשחק עם ההגדרות, וכן ניתן ממש להפעיל את אותו המקלט ולקנפג לו הגדרות דרך הקוד עצמו.
  2. [בקישור](https://gage.upc.edu/en/learning-materials/software-tools/glab-tool-suite-links/glab-download) מוצגות ההורדות של המקלט למערכות ההפעלה השונות, אנו ממליצים להוריד את הגרסה הכי חדשה של המקלט.
  3. נשים לב כי הקוד שפותח תומך בגרסת Windows בלבד (ועם קצת התאמצות ניתן למצוא גם מוחבא בתוך הקוד את הגרסה שתומכת גם ב-Linux). התמיכה מתבטאת בסוג הקריאה ל-gLAB.
  4. לאחר בחירת ההורדה של הגרסה הכי חדשה ל-Windows יש להתקין את הקובץ, לאחר מכן בשביל בדיקה שההתקנה צלחה יש לחפש במחשב את gLAB\_GUIולראות אם רואים את המסך הבא:



* 1. נדרש בנוסף להכניס את ה-path של ה-gLAB למשתני הסביבה בעזרת המדריך [בקישור](https://www.youtube.com/watch?v=gb9e3m98avk&ab_channel=easytechtutorials) הבא. ככל הנראה שהמיקום של ה-gLAB יהיה ב-C:\gLAB\win (ניתן לחפש את המיקום בעזרת חיפוש של הקובץ gLAB\_GUI ומיקומו).

## קוד להורדת הקבצים

### מציאת הקוד ותפעולו

* 1. כל הקוד וגם הוראות אלה נמצאים ב: <https://github.com/boro3040/iono_code>.
  2. בשביל שהקוד יעבוד נדרש להפעיל את pycharm או כל IDE אחר במצב מנהל, מדריך להגדרת הפעלה תמידית במצב מנהל ניתן למצוא [בקישור](https://www.youtube.com/watch?v=Sisk7O8O4Q8&ab_channel=DigiDank) הבא.

### תפעול הקוד

* 1. הקוד הוא קוד פייתון המשתמש בהרבה ספריות, נדרש לבדוק כי כל הספריות מותקנות על המחשב.
  2. הקוד מורכב מ-5 חלקים (נפרט עליהם בהמשך):
     1. **download.py** – הורדה של הקבצים מאתרי האינטרנט השונים.
     2. **gLab\_executer.py** –הרצה של המקלט והוצאת פלט של כל השגיאות בשלושת הצירים במהלך היום.
     3. **gLab\_to\_netCDF.py** – העברה של התוצאות בפורמט של gLAB לפורמט של Xarry (אותו האפליקציה יודעת לקרוא).
     4. **NIC to df.ipynb** – להמיר את ה-Xarry לאקסל.
     5. **application.py** – הרצה של האפליקציה איתה ניתן לנתח את הנתונים באופן כללי (לא מאוד מומלץ, פיתוח לבעיה מאוד ספציפית של החברה מאוניברסיטת אריאל).

#### download.py

* 1. בקובץ זה מורידים את הקבצים מאתרי האינטרנט השונים. לחלק מהאתרים נדרש שם משתמש וסיסמה, כרגע יש שם משתמש וסיסמה דיפולטיביים בקובץ ".netrc" ולא נדרש לגעת בהם. אם פג תוקפו של המשתמש או שיש בו בעיות ניתן להירשם לאתרים מחדש בחינם ולהזין את שם המשתמש והסיסמה מחדש. לאתרים בהם לא נדרשת סיסמה כתוב: " login anonymous password anonymous".
  2. הקוד משתמש ב-3 פרמטרים שביחד קובעים לנו את מתאר ההרצה:
     1. **סוכנויות (agencies)** – כלל המקומות מהם אנו מנסים להפיק את המידע, אין צורך לגעת אם לא יודעים מה עושים.
     2. **שמות התחנות (stations)** – רשימה של כלל התחנות להורדה, קישור בין שם התחנה למיקומה ניתן למצוא באתר שהזכרנו קודם.
     3. **רשימת התאריכים (dates\_sets)** – רשימה של רשימות כאשר בכל רשימה נמצאים התאריכים אותם אנו רוצים להריץ בפורמט datetime.
  3. **כאשר ישנם קבצי pickle בתיקייה (.pk) נתונים אלה ישאבו אוטומטית מאותם הקבצים**. אם אין כאלה ניתן להגדיר את פרמטרי ההרצה בתוך הקוד עצמו.
  4. הקוד משתמש בפונקציות הורדה הנמצאות ב-utilities, איתן הוא מוריד את הקבצים, קבצים שהוא לא מצליח להוריד עוברים לקבצי log המתווספים לתיקייה כאשר יש תקלה בהורדת הקובץ.
  5. בסך הכל קוד זה מוריד קבצים מכל הסוגים המוזכרים בפסקאות הקודמות, הם יורדו לתיקיות: ION, SP3, CLK, RNX בהתאם לקובץ המורץ.
  6. אם רוצים להבין מאילו אתרים ההורדה נלקחת יש להתחקות אחרי הפונקציות בתוך הקוד, בתחילת כל פונקצית הורדה קיים הקישור ממנו נלקח המידע.
  7. נדגיש כי זהו השלב הקריטי ביותר בדרך, כלומר אם אחד מהקבצים (למעט CLK) של אותם התאריכים לא ירד לא נוכל להמשיך לשלבים הבאים באותו התאריך והתחנה הספציפית, לפעמים יכולות להיות תקלות בחיבור, ולפעמים הקובץ לא נמצא. אנו ממליצים לבקר באתר CDDIS ידנית ולנסות למצוא שם את הקובץ החסר בעצמנו.

#### gLab\_executer.py

* 1. קוד זה מריץ את המקלט על הקבצים שנוצרו ומוציא מהם פלט שגיאות בשלושת הצירים. התחנות והתאריכים שיתקבלו יהיה לפי קבצי ה-pickle שהוגדרו קודם, כלומר אם רוצים להריץ תאריכים אחרים כרגע יש לשנות את קבצי ה-pickle, מצד שני אם יש המון קבצים בתיקיות אבל בקובץ ה-pickle יש הפניה אך ורק לתחנות ותאריכים ספציפיים נקבל רק אותם.
  2. השורות הקריטיות בקוד זה הן השורות האחרונות בהן מוגדר ה-command, נפתח ה-subprocess והרצת מקלט ה-gLAB.
  3. נשים לב כי כל הקונפיגורציה והגדרות ההרצה של המקלט נמצאים בקובץ: "gLAB/tmp\_config.cfg", ניתן להיעזר במדריכים במקרה בו רוצים לשנות את ההגדרות של המקלט (זה כמו שינוי ההגדרות בגוי עצמו רק בעזרת הקוד).
  4. נשים לב גם כן כי באמצע הקוד יש את ההגדרה של המשתנה "date", הוא לוקח אך ורק רשימה אחת מבין הרשימות הנמצאות בקובץ ה-pickle של התאריכים (ניתן לשנות לפי הצורך).
  5. לאחר הרצת מקלט ה-gLAB נוצרת תיקיית OUT בה יש תתי תיקיות לפי שמות התחנות ולאחר מכן השנים השונות, הקבצים שיצאו מהמקלט יהיו בתתי התיקיות.

#### gLab\_to\_netCDF.py

* 1. הקוד ממיר את הפלט שיצא מהמקלט לפורמט Xarry נוח לעבודה. נשים לב כי גם כאן הקבצים עליהם יתבצע העיבוד הם אך ורק הקבצים המוזכרים בקבצי ה-pickle בתיקייה.
  2. לאחר הרצת קוד זה תפתח תיקייה חדשה בשם netCDF בה קיימים כלל קבצי ה-Xarry שקיבלנו עד כה.
  3. נשים לב כי בשורה 168 קיים ההשמה של שם הקובץ, ניתן קצת לשחק שם עם הדברים בשביל להביא שם יותר מתאים (אם זה לא שורה זו חפשו .nc בחיפוש ומצאו את השורה בה נמצאת ההשמה של השם).

#### NIC to df.ipynb

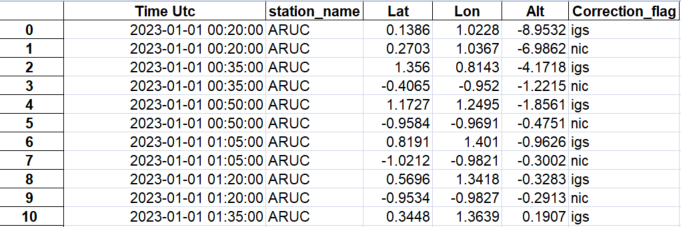
* 1. הקוד נמצא במחברת jupyter, בקוד זה מומר ה-Xarry לאקסל עם שדות אינטואיטיביים. יש לשים לב כי מחברת זו אינה עובדת יותר עם ה-pickle, יש לבחור בעצמנו איזה קובץ netCDF אנו רוצים להמיר לאקסל בעצמנו.

#### application.py

* 1. קוד פייתון הפותח אפליקציה לניתוח הנתונים, האפליקציה פותחה באוניברסיטת אריאל.
  2. בכדי לפתוח אותה יש להריץ את הקוד, ניתן לבחור קובץ netCDF כרצוננו ולראות סטטיסטיקות שעתיות שלו כמגמה לאורך כל השנה, קורלציות ועוד. לא ארחיב על האפליקציה כי ככל הנראה השימוש בה הוא פחות רלוונטי לצרכינו.

## הפורמט המתקבל

* 1. שם קובץ הפלט המסודר היוצא לאחר כל השלבים יכול להיקרא באיזה שם שנבחר. הפלט יוצא כקובץ אקסל עם 6 עמודות כמתואר בתמונה הבאה, נתאר מה משמעות כל שדה.



* 1. כל שורה מייצגת נתונים שונים של הרצה, השדה הראשון, זמן לפי UTC הוא זמן ותאריך המדידה לפי זמן UTC בפורמט datetime של פייתון. כפי שניתן לראות ההפרש בין מדידות שונות הוא בדיפולט 5 דקות, המדידות בפועל בקבצים השונים נמצאים בהפרשים של חצי דקה, ולכן ככל הנראה שניתן לשנות זאת בקוד אם נדרש.
  2. שם התחנה זה הייצוג בעל ארבעת האותיות של התחנה עליה עשינו את המדידות.
  3. השדות Lat, Lon, Alt מייצגים את ההפרש בין המיקום האמיתי של המקלט לבין המדידה עצמה בשלושת הצירים הטבעיים. **עקב כך אם אנו רוצים להבין לאן המקלט יחטיא יש לקחת את הערך השלילי של כל שדה** (מה שבאמת מעניין אותנו).
  4. Correction\_flag יכול לקבל 3 ערכים:
     1. nic – no ionospheric correction, ללא תיקון יונוספירה כלל.
     2. ckm – תיקון קלובושר בלבד של שגיאות היונוספירה ביחס למקדמים של אותו השבוע שעוברים דרך הודעות הלוויין.
     3. igs – שימוש במפות היונוספירה שנטענו לקיזוז השגיאות.