מבוא לאנליזה נומרית – na211
Assignment 1

הדרכה: אם לא מצוין אחרת, יש להציג ולהסביר את צעדי החישוב שביצעתם.
אם התבקשתם לכתוב קוד, יש לצרף תדפיס של הקוד והפלט של התכנית. בנוסף עליכם לצרף את הקוד בקובץ נפרד כולל הוראות מדויקות להרצתו, כך שיפיק את הפלט הנדרש.

<u>שאלה מספר 1</u>

גרסת התוכנה המקורית בטיל Patriot ייצגה את פרק הזמן של עשירית השנייה ע"י מספר בינארי של 23 ספרות אחרי הנקודה הבינארית וחיתוך. (להלן $\widetilde{0.1}$) .

המערכת עקבה אחר מטרות אפשריות. על מנת למדוד את המרחק שעברו המטרות בין שתי נקודות בזמן (להלן t1,t2) המערכת חישבה את מכפלת הזמן שחלף במהירות המטרה (נסמנה V) כשהפרשי הזמן מחושבים מתוך שעון המערכת המונה ביחידות של עשיריות שנייה (שם לב, המונה הינו מונה ב<u>שלמים</u>. כל פעימת מונה מייצגת עשירית שנייה שעברה).

n2,n1 – מייצגים את מספר פעימות המונה כפי שנספרו ע"י המערכת מאז איתחולה האחרון. נסמן:

$$\tilde{t}_1 \leftarrow \widetilde{0.1} \times n_1$$
$$\tilde{t}_2 \leftarrow \widetilde{0.1} \times n_2$$
$$\Delta\tilde{t} \leftarrow \tilde{t}_2 - \tilde{t}_1$$

א. מהם 0.1 ו $\widetilde{0.1}$בבסיס 2?

ב. מה השגיאה המוחלטת ומהי השגיאה היחסית ב $\widetilde{0.1}$?

ג. לכמה ספרות בינאריות משמעותיות מקרב $\widetilde{0.1}$את 0.1 ?

ד. נניח כי n_1נקרא משעון המערכת 8 שעות לאחר איתחולה, ו- n_2 נקרא 2 שניות מאוחר יותר. מהן השגיאות המוחלטות והיחסיות ב $\Delta\tilde{t}$ במקרה זה?

ה. חזור על סעיף ד', עבור 100 שעות פעילות.

ו. נניח שתוכנת המעקב עודכנה בחישובי זמנים מדויק יותר, אך עקב טעות בוצע השיפור באופן חלקי בלבד. חזור על ד' תחת ההנחה שכעת $\tilde{t}_2$ מחושב על בסיס 0.1 אך $\tilde{t}_1$ מחושב עדיין על בסיס $\widetilde{0.1}$.

ז. חזור על סעיף ו' עבור 100 שעות פעילות. (זהו המצב שגרם לכשל טיל הפטריוט בערב הסעודית בחודש פברואר 1991 והסתיים במותם של 28 נחתים!)

<u>שאלה מספר 2</u>

בייצוג מספרים בינאריים במחשב על פי סטנדרט IEEE 754, ענה על השאלות הבאות:

חשב מהו המספר הגדול ממש מ-0 *<u>הקטן ביותר</u>* הניתן לייצוג במקרים הבאים:

א. בייצוג עם 32 ביט (single)
   a. בייצוג נורמלי $(1.fraction \times \ldots)$
   b. בייצוג תת-נורמלי $(0.fraction \times \ldots)$

ב. בייצוג עם 64 ביט (double)
   a. בייצוג נורמלי
   b. בייצוג תת-נורמלי

* ראו גם https://en.wikipedia.org/wiki/Normal_number_(computing)

<u>שאלה מספר 3</u>

יהי K מספר שלם חיובי. ניתן להראות כי לכל מספר ממשי a, לכל K מספרים ממשיים $(l_j)_{j=1}^{K}$, ולכל מספר שלם $j \in \{1,2,\dots,K\}$ מתקיימת הזהות הבאה:

$$\frac{\exp(l_j)}{\sum_{j'=1}^{K}\exp(l_{j'})} = \frac{\exp(l_j - a)}{\sum_{j'=1}^{K}\exp(l_{j'} - a)}$$

(כאשר $\exp(x)$ מסמל את e^x). לדוגמה, עבור $(l_1, l_2, l_3) = (-0.1, 5, -10)$, ו-$a = 0.5$, קל לבדוק כי 3 הזהויות הבאות נכונות (סמכו עלינו, אל תבזבזו זמן לבדוק זאת בעצמכם):

$$(1)\quad \frac{\exp(-0.1)}{\exp(-0.1)+\exp(5)+\exp(-10)} = \frac{\exp(-0.1-0.5)}{\exp(-0.1-0.5)+\exp(5-0.5)+\exp(-10-0.5)};$$

$$(2)\quad \frac{\exp(5)}{\exp(-0.1)+\exp(5)+\exp(-10)} = \frac{\exp(5-0.5)}{\exp(-0.1-0.5)+\exp(5-0.5)+\exp(-10-0.5)};$$

$$(3)\quad \frac{\exp(-10)}{\exp(-0.1)+\exp(5)+\exp(-10)} = \frac{\exp(-10-0.5)}{\exp(-0.1-0.5)+\exp(5-0.5)+\exp(-10-0.5)}.$$

כעת, בשימושים רבים במדעי המחשב אנו נתקלים בצורך בחישוב ערכים מהצורה

$$\frac{\exp(l_j)}{\sum_{j'=1}^{K}\exp(l_{j'})}$$

עם זאת, חישוב זה נוטה ליצור בעיות נומריות מסוימות הקשורות למגבלות הנובעות מייצוג המספרים במחשב באמצעות כמערכת נקודה צפה. הגישה הסטנדרטית להתמודדות עם בעיות אלו היא לחשב:

$$\frac{\exp(l_j - a)}{\sum_{j'=1}^{K}\exp(l_{j'} - a)}$$

כאשר:

$$a = \max(l_1, \dots, l_K)$$

באילו מן הבעיות הבאות עשויה שיטה זו לעזור? נמק בקצרה.

i. איבוד משמעות

ii. רוב המספרים הממשיים אינם ניתנים לייצוג באמצעות מערכת ייצוג בינארית

iii. סכימה של מספרים קטנים ביחד עם מספרים גדולים עלולה להביא לשגיאות מהותיות.

iv. Underflow*

* underflow מתאר מצב שבו תוצאה של חישוב מובילה למספר קטן בערך מוחלט (קרוב ל-0) אשר לא ניתן לייצוג במערכת.

ראו: https://en.wikipedia.org/wiki/Arithmetic_underflow

<u>שאלה מספר 4</u>

נתונות שתי מדידות X ו-Y בעלות שגיאות מדידה מוחלטות ΔX ו-ΔY, בהתאמה. ממדידות אלו מחשבים מספר שלישי, Z, לפי הנוסחה $Z = e^{\alpha(X-Y)}$, כאשר α – קבוע כלשהו.

א. מצאו נוסחאות מתאימות לשגיאה המוחלטת ולשגיאה היחסית בחישוב Z.

ב. אם נתון ΔX=ΔY=2, מהו תחום הערכים של α אשר יבטיח שגיאה יחסית ב-Z הקטנה מ-5%?

ג. ניתן לחשב את Z בעזרת שיטה הזהה אלגברית לזו שבראש השאלה: $Z = \frac{e^{\alpha X}}{e^{\alpha Y}}$. האם השיטה החדשה עדיפה מבחינת השגיאה היחסית של Z? נמקו.

<u>שאלה מספר 5</u>

כתוב תוכנית המדמה (מסמלצת) חיבור של מספרים במחשב בעל מערכת נקודה צפה של שלוש ספרות עשרוניות משמעותיות וחיתוך (אינך מוגבל לסטנדרט כלשהו) .

השתמש בשגרה שכתבת למימוש "צובר" (Accumulator) - תוכנית המחברת קבוע נתון C=0.001 מספר נתון של פעמים- n, קרי , לאחר- n איטרציות תוצאת הצובר הינה: $\sum_{i=1}^{n} c$

. הנח כי הצובר מאותחל לאפס.

הזן לתוכניתך את הערכים הבאים עבור- n וענה על השאלות:

(א) מהי השגיאה המתקבלת אחרי n=80 איטרציות? מהי השגיאה המתקבלת אחרי n=8000 איטרציות?

מדוע שגיאת הצובר גדולה יותר ככל שעובר הזמן?

(ב) מהו ה<u>הפרש</u> בין השגיאה אחרי n=80 איטרציות לבין השגיאה אחרי n=82 איטרציות?

מהו <u>ההפרש</u> בין השגיאה אחרי n=8000 איטרציות לבין השגיאה אחרי n=8002 איטרציות?

מדוע שני ההפרשים הנ"ל כה שונים זה מזה? (ישנו שוני בסדר גודל של 10^{15})

צרף את התוכנית שכתבת לדף ההגשה.

<u>הדרכה (מומלץ אך לא מחייב):</u> כתוב תחילה שגרה בשם adder המחברת בין שני מספרים a ו- b . לשם נוחות ולצורך התרגיל נשתמש ב- mantissa כשבר ולא כשלם, כלומר נייצג את המספר בצורה: $x = mantissa * 10^{\exp oenent}$ (ראה דוגמא עבור 0.001 בקוד המצורף). כמו כן, לשם נוחות, אין צורך לנרמל את ה- exponent וניתן להניח כי הוא אינו מוגבל לטווח

כלשהו. כיוון שאנו עובדים במקרה זה רק עם מספרים חיוביים אין צורך להקצות סיבית לסימן. נתון בדף האחרון מבנה התוכנית בMATLAB לסטודנטים שמחליטים לכתוב ב MATLAB, אפשר להשתמש בכל שפה.

```
function[man,exp]= adder(man_a,exp_a,man_b,exp_b)

    %%%%%  The program assumes that a>=b;  %%%%


      %%%%% …  %%%%

    man=floor(man*1000)/1000;

end

function[err]= tar7_ass1(n)


        step_man=0.1;   % mantissa of 0.001

        step_exp=-2;   % exponent of 0.001


        res_man=0.1;

        res_exp=-2;


        for k=2:n

[res_man,res_exp]=adder(res_man,res_exp,step_man,step_exp);

        end


        real_res=  % ….

        approximate_res=  % ….

        err=abs(real_res - approximate_res);

end
```