

מבוא לאנליזה נומרית

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. (ניתן לבחור בכל שפת תכנות)

שאלה מספר 1

גרסת התוכנה המקורית בטיל Patriot ייצגה את פרק הזמן של עשירית השנייה ע"י מספר בינארי של 23 ספרות אחרי הנקודה הבינארית וחיתוך. (להלן $0.\tilde{1}$).

המערכת עקבה אחר מטרות אפשריות. על מנת למדוד את המרחק שעברו המטרות בין שתי נקודות בזמן (להלן t_1, t_2) המערכת חישה את מכפלת הזמן שחלף במהירות המטרה (נסמנה V) (כשהפרשי הזמן מחושבים מתוך שעון המערכת המונה ביחידות של עשיריות שנייה (שם לב, המונה הינו מונה בשלמים. כל פעימת מונה מייצגת עשירית שנייה שעברה).

n_1, n_2 – מייצגים את מספר פעימות המונה כפי שנספרו ע"י המערכת מאז איתחולה האחרון. נסמן:

$$\begin{aligned}\tilde{t}_1 &\leftarrow 0.\tilde{1} \times n_1 \\ \tilde{t}_2 &\leftarrow 0.\tilde{1} \times n_2 \\ \Delta\tilde{t} &\leftarrow \tilde{t}_2 - \tilde{t}_1\end{aligned}$$

א. מהם 0.1 ו $0.\tilde{1}$ בבסיס 2?

ב. מה השגיאה המוחלטת ומהי השגיאה היחסית ב $0.\tilde{1}$?

ג. לכמה ספרות בינאריות משמעותיות מקרב $0.\tilde{1}$ את 0.1 ?

ד. נניח כי n_1 נקרא 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 $0.\tilde{1}$.

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!)

שאלה מספר 2

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

חשב מהו המספר הגדול ממש מ-0 הקטן ביותר הניתן לייצוג במקרים הבאים:

א. בייצוג עם 32 ביט (single)

a. בייצוג נורמלי ($1. fraction \times \dots$)

b. בייצוג תת-נורמלי ($0. fraction \times \dots$)

ב. בייצוג עם 64 ביט (double)

a. בייצוג נורמלי

b. בייצוג תת-נורמלי

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

שאלה מספר 3

הוכח את הטענות הבאות (הראו דרך מלאה, גם אם יש צורך לחזור על חלקים דומים מההרצאה/תרגול):

א. $\Delta(\tilde{x} - \tilde{y}) \leq \Delta\tilde{x} + \Delta\tilde{y}$

ב. $\delta\left(\frac{\tilde{x}^2}{\tilde{y}^2}\right) \lesssim 2(\delta\tilde{x} + \delta\tilde{y})$

שאלה מספר 4

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 $\Delta X = \Delta Y = 2$, מהו תחום הערכים של α אשר יבטיח שגיאה יחסית ב- Z הקטנה מ-5%?

ג. ניתן לחשב את Z בעזרת שיטה הזוהי אלגברית לזו שבראש השאלה: $Z = \frac{e^{\alpha X}}{e^{\alpha Y}}$. האם השיטה

החדשה עדיפה מבחינת השגיאה היחסית של Z ? נמקו.

שאלה מספר 5

כתוב תוכנית המדמה חיבור של מספרים במחשב בעל מערכת נקודה צפה של שלוש ספרות עשרוניות משמעותיות וחיתוך (האקספוננט אינו מוגבל, הינכם רק נדרשים לשמור אחרי כל פעולה את 3 הספרות העשרוניות המשמעותיות ולאפס את השאר. ניתן להניח שכל המספרים חיוביים).

השתמש בשגרה שכתבת למימוש "צובר" (נסמנו בקוד acc קיצור של Accumulator) - תוכנית המחברת לצובר מספר קבוע נתון, C , n פעמים.

כמובן שזה פשוט $acc + n \cdot C$, אבל אנחנו מעוניינים לבצע פעולות חיבור נפרדות:

$$n \cdot \begin{cases} acc \leftarrow acc + C \\ \vdots \\ acc \leftarrow acc + C \end{cases}$$

הנח כי בהתחלה $C = 0.004$ והצובר מאותחל ל- $acc = 0.001$.

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

(א) מהי השגיאה המוחלטת המתקבלת אחרי $n=70$ איטרציות? מהי השגיאה המוחלטת

המתקבלת אחרי $n=7000$ איטרציות?

(השגיאה נמדדת אל מול הערך המתמטי האמיתי $n \cdot C$)

מדוע שגיאת הצובר גדולה יותר ככל שעובר הזמן?

(ב) מהו ההפרש בין השגיאה אחרי $n=70$ איטרציות לבין השגיאה אחרי $n=72$ איטרציות?

מהו ההפרש בין השגיאה אחרי $n=8000$ איטרציות לבין השגיאה אחרי $n=8002$ איטרציות?

מדוע שני ההפרשים הנ"ל כה שונים זה מזה?

צרף את התוכנית שכתבת לדף ההגשה.

לנוחיותכם מצורף קוד פייתון אשר מחזיר מספר מבוקש של ספרות משמעותיות עבור קלט שלם. בנוסף הפונקציה מחזירה את מספר הספרות שאינם משמעותיות בקלט. שימו לב כי יש דברים רבות לעשות את הסימולציה. אין חובה להשתמש בפונקציה הנתונה ואף אין חובה לכתוב בפייתון.

```
import math

# [i] python has a simple typing mechanism.
#      ': <type>' signifies the type of the parameter (integer in the
#      following)
def most_significant(num: int, digits_to_keep: int):
    """ return the digits_to_keep most significant digits of num
    :return (the kept digits, the number of non-significant (zeroed-
    out) digits)
    """
    # [i] 'assert' is a special keyword in python.
    #      it verifies if the condition is True
    assert digits_to_keep > 0, 'digits_to_keep should be positive'

    num_digits = math.floor(math.log10(num)) + 1
    non_significant = max(0, num_digits - digits_to_keep)

    # [i] '//' is division without remainder in python 3+
    return (num // 10 ** non_significant), non_significant
```