

המחלקה להנדסת תוכנה

חישוב פרקולציה מוליכות בעזרת רנורמליזציה Calculating percolation of conductivity using renormalization

מאת:

זינה אבו צווי 203591672

פואד נאסר אדין 201211570

תאריך:

אישור:

מנחה אקדמי: ד"ר יהודה חסין

תאריך:

אישור:

אחראי תעשייתי: ד"ר יהודה חסין

תאריך:

אישור:

רכז הפרויקטים: ד"ר שפנייר אסף

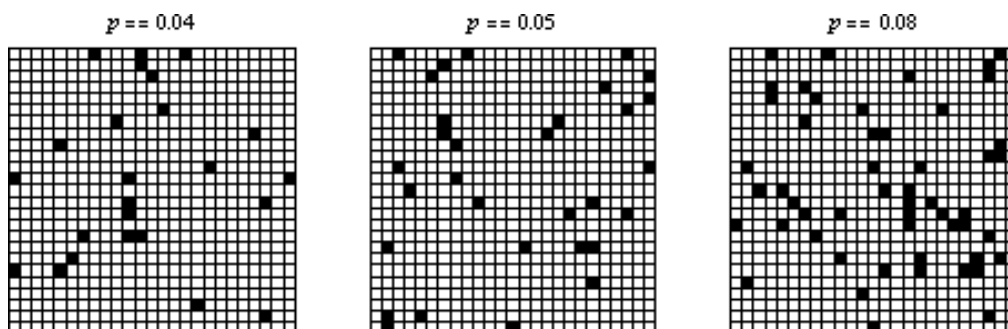
June 2018

סיוון ה'תשע"ח

תקציר

תיאוריית הפרקולציה מתארת את הפיזיקה והמתמטיקה של חיבורים ארוכי טווח במערכות אקראיות. היא תופעה המתארת חלחול של נוזל דרך חומר, והיא חלה על מגוון רחב של מערכות פיסיקליות, כולל מוליכות, נקבוביות ופולימרים. אך ניתן להשתמש בה גם לניתוח מערכות שונות כגון שריפות יער, מגיפות מחלות, ורשתות חברתיות.

במסגרת הפרויקט, שנעשה בעבודת צוות, פותחה תוכנה בשפת Java שמחשבת בצורה מדויקת עבור גריד דו ממדי בגודל 2×2 , 3×3 , ו- 4×4 את ערכי המוליכויות החשמליות האפשריים וההסתברות לקבלת כל ערך כזה. ועבור כל גריד דו ממדי בגודל אחר, מחשבת את הערכים הנ"ל בקירוב בעזרת הנורמליזציה.





הצהרה

העבודה נעשתה בהנחיית ד"ר יהודה חסין במחלקה להנדסת תוכנה,
בשיתוף פעולה עם ד"ר דורון אזולאי וד"ר איתי פורמן,
עזריאלי – המכללה האקדמית להנדסה ירושלים.
החיבור מציג את עבודתנו האישית ומהווה חלק מהדרישות לקבלת תואר ראשון
בהנדסה.

תודות

אנו רוצים להודות לד"ר יהודה חסין על האמון, המקצועיות, התמיכה ויחסו האישי שניתנה לנו במהלך הפרויקט.

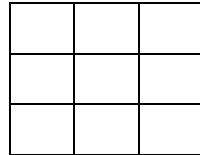
אנו גם רוצים להודות לד"ר דורון אזולאי וד"ר איתי פורמן שלאורך הפרויקט שיתפו איתנו את הידע שלהם בפיזיקה שגרים ועזר לנו מאוד להמשיך ולהתקדם בפרויקט.

ואנו מודים לשלושתם על הזמן היקר ששיתפו איתנו, והסבלנות לאורך כל הדרך.

תוכן עניינים

6	מילון מונחים, סימונים וקיצורים	
7	1. תיאור מסגרת הפרויקט	
8	2. תיאור הבעיה	
2.1	הקדמה	8
2.2	דרישות ואפיון הבעיה	11
2.3	הבעיה מבחינת הנדסת תוכנה	12
3.	3. תיאור הפתרון	12
3.1	תיאור המערכת	12
3.2	פירוט המערכת	13
3.3	תהליכים ונתוני המערכת	14
3.4	תיאור הכלים המשמשים לפתרון	15
4.	4. תוכנית בדיקות	15
5.	5. השוואה לספרות	16
6.	6. סיכום ומסקנות	16
7.	7. ספרות	17
8.	8. מערכות ניהול הפרויקט	17
9.	9. נספחים	18

מילון מונחים, סימונים וקיצורים



- גריד דו ממדי (דוגמא של גודל 3X3):

- Graphical User Interface – GUI : ממשק משתמש לתוכנה המבוסס על עיצוב גרפי של המסך המוצג למשתמש.

- JAVA: שפת תכנות מונחה עצמים אחת התוכנות הנפוצים בעולם.

- הסתברות קיומה של קשת בגריד: מספר בין 0 ל- 1.

- מוליכות: מוליכות חשמלית, מידת היכולת של זרם חשמלי לעבור בגוף מסויים.
מוליכות = 1 לחלק להתנגדות.

- וקטור ערכי מוליכות: טבלה של ערכי מוליכות שונים וההסתברות לקבלת כל ערך.

1. תיאור מסגרת הפרויקט

במשך זמן רב, מדענים חקרו את הבעיה של פרקולציה ומצאו עבודה פתרון. לאחרונה התחילו הפיזיקאים במכללה להתעסק עם החישובים של בעית הפרקולציה עבור חומרים חדשים, וגילו שהפתרון הקיים של הבעיה אינו מספיק מדויק. מה שגרם להם לעשות את החישובים הלא פשוטים והדורשים זמן רב, ידנית.

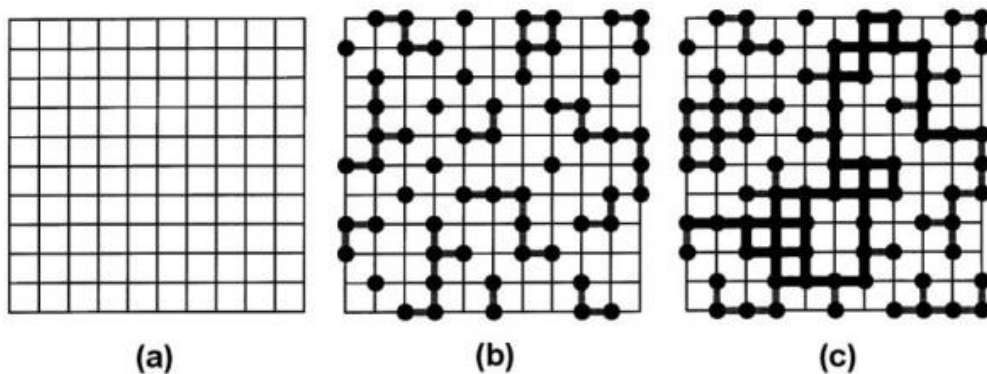
בקיץ שעבר פנו אלינו שני פיזיקאים מהמכללה וביקשו כלי תוכנה שיחשב פרקולציה מדויקת עבור מקרים שונים.

הוסכם איתם שנפתח להם תוכנה שעומדת בדרישות שלהם וממלא את צרכיהם. אשר התוכנה תקל עליהם ותחשב את החישובים שהיו דורשים מהם חודשים של עבודה ידנית, שלא לדבר על כל מיני פרמטרים שמשתנים ומצריכים ביצוע החישובים מחדש, בזמן קצר.

התוכנה מחשבת עבור גריד חשמלי דו ממדי עם מספר וסדר כלשהם של נגדים (כך שכל קשת בגריד היא בעצם נגד), את ערכי המוליכויות השונים וההסתברות לקבלת כל ערך כזה. כאשר עבור גרידים בגודל 2×2 , 3×3 , ו- 4×4 החישובים נעשו בצורה מדויקת. ו עבור גרידים יותר גדולים החישובים נעשו בצורה מקורבת ע"י שימוש ברנורמליזציה, שהיא סוג של רקורסיה המשתמשת בגרידים קטנים המהווים אבן בניין כדי לבצע חישובים על גרידים יותר גדולים.

2. תיאור הבעיה

2.1 הקדמה

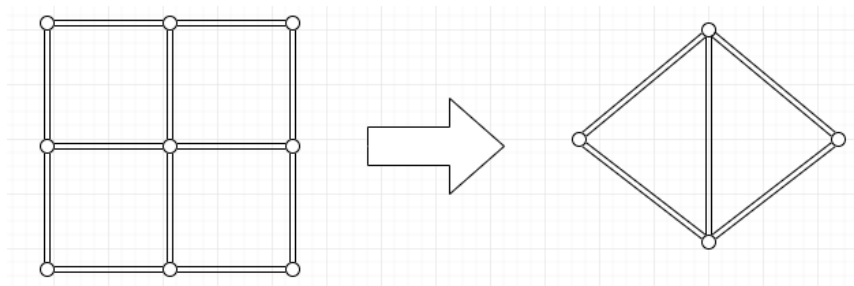


- האזור לעיל מציג גריד בגודל 10×10 בשלושה מצבים אקראיים שונים.
- מצב (a) : אין שום קשת שקיימת על הגריד ולכן ערך המוליכות שווה ל-0.
- מצב (b) : יש מספר של קשתות שקיימות על הגריד אך אין מסלול של קשתות שמגיע מצד אחד לשני ולכן המוליכות שווה ל-0.
- מצב (c) : יש מספר של קשתות שקיימות בגריד וקיים מסלול של קשתות שמגיע מצד לשני ולכן ערך המוליכת שונה מ-0.

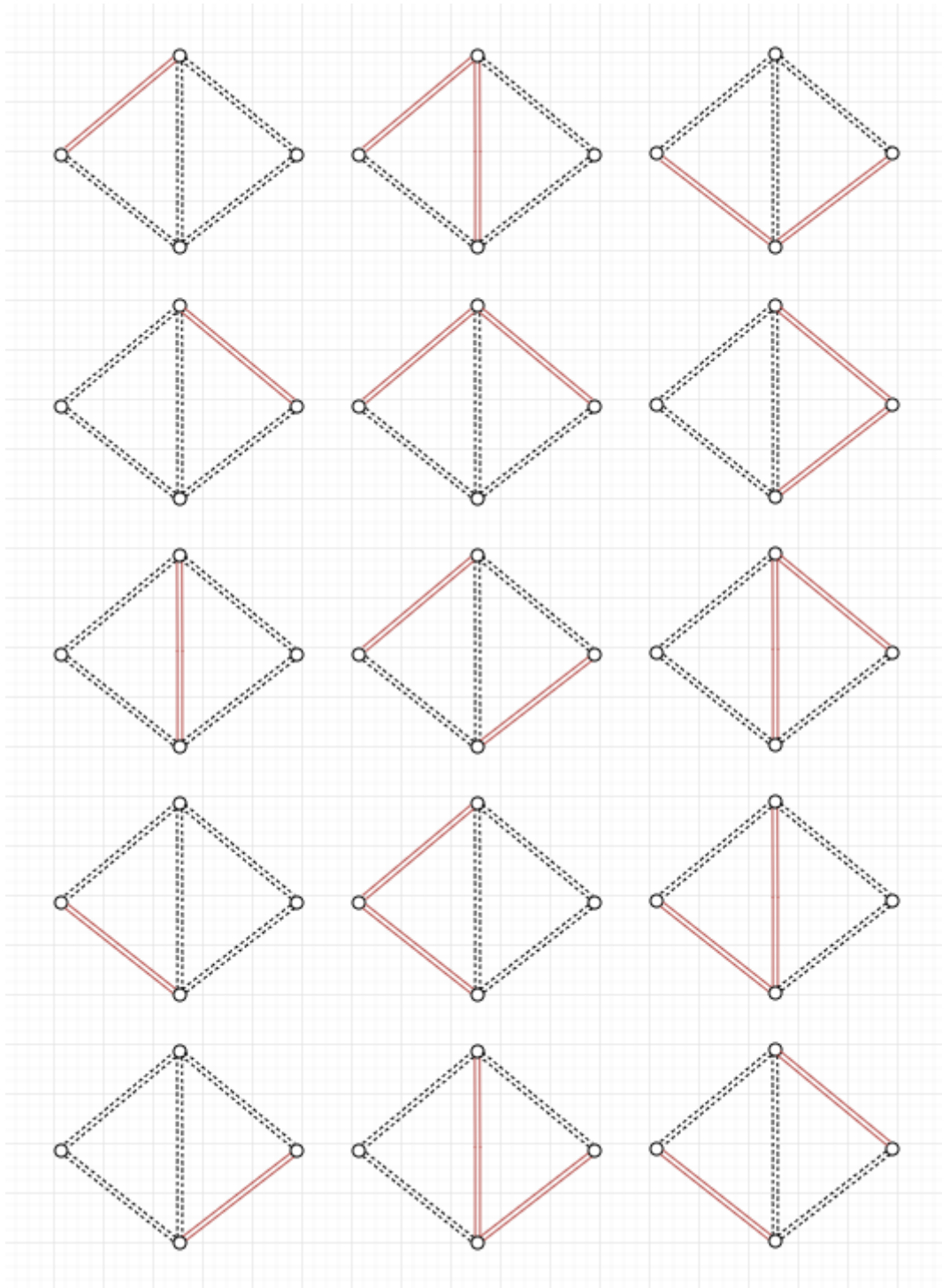
במסגרת התוכנה שפיתחנו, אנו מחשבים עבור כל גריד כזה את כל האפשרויות השונות של מספר וסדר הקשתות הקיימות, את ערך המוליכות שהוא בעצם שווה ל-ערך ההתנגדות/1, ואת ההסתברות לקבלת הערך הזה.

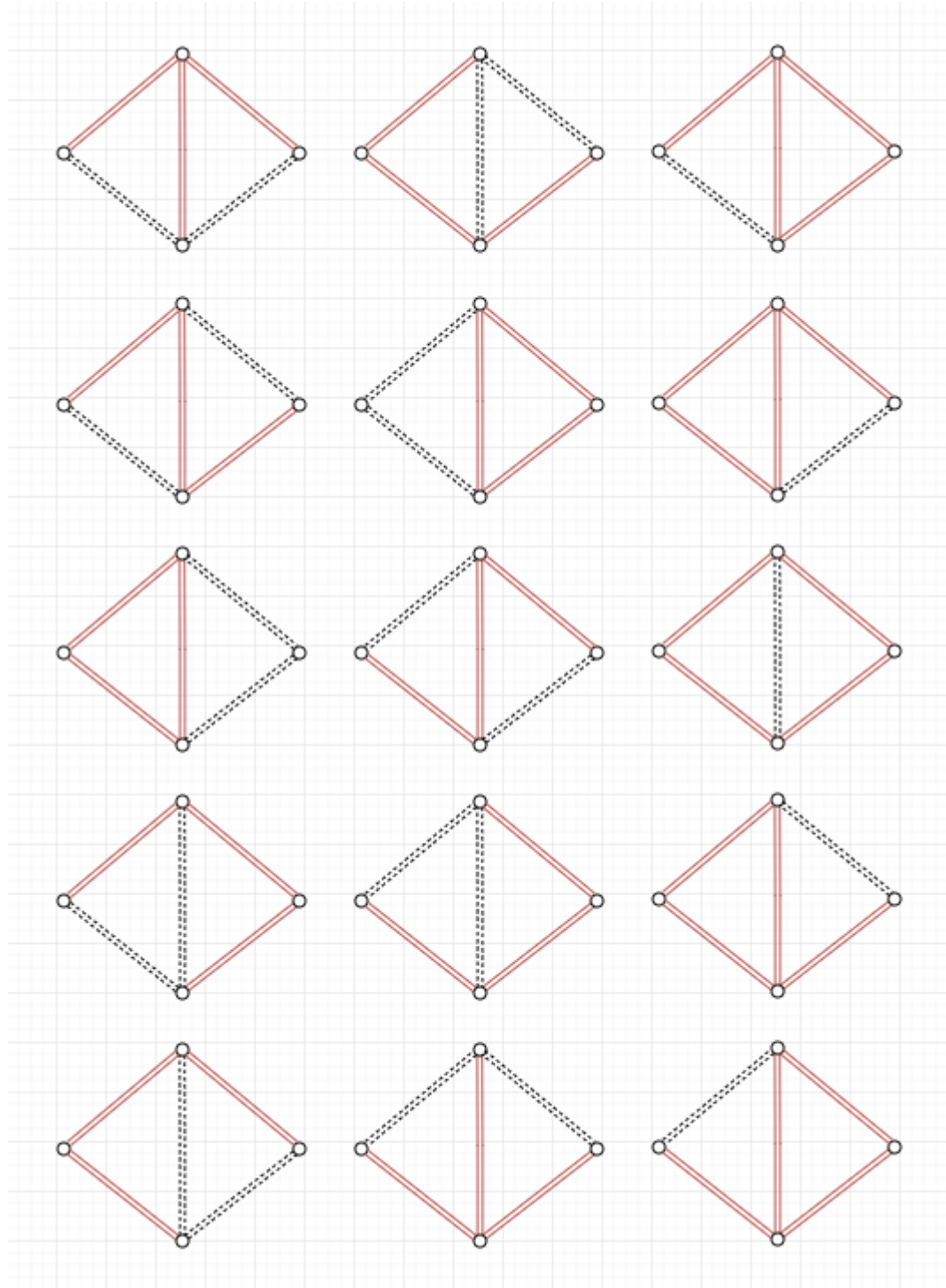
לדוגמא, עבור גריד בגודל 2×2 . קיימות 2^5 אפשרויות של תתי גרידים עם מספר וסדר שונים של קשתות קיימות.

על הגריד עצמו אפשר להסתכל באופן הבא :

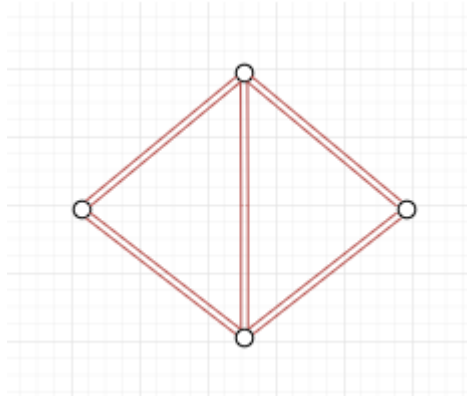


וכל האפשרויות של התתי גרידים הן (קשת מסומנת באדום היא קשת קיימת):





ובמקרה שכל הקשתו קיימות, נקבל את התת גריד הבא:



למשל עבור התת גריד הזה הפתרון של ערך המוליכות יהיה הפתרון של מערכת המשוואות הבאה:

כאשר כל עמוד וכל שורה מייצגים קודקוד מסויים, במקרה שקיימת קשת בין שני קודקודים 0 ו-1. נמלא את המטריצה במקומות המתאימים שהם (1, 0) ו- (0, 1) עם הערך 1. ובאלכסון נכניס את מספר השכנים לכל קודקוד, אז (0, 0) נמלא עם 2 ו- (1, 1) עם 3 לפי האיור הנ"ל.

וכך הלאה עד שנקבל את המטריצה הבאה:

$$\begin{pmatrix} 2 & -1 & -1 & 0 \\ -1 & 3 & -1 & -1 \\ -1 & -1 & 3 & -1 \\ 0 & -1 & -1 & 2 \end{pmatrix} \begin{pmatrix} v_1 \\ v_2 \\ v_3 \\ v_4 \end{pmatrix} = \begin{pmatrix} 1 \\ 0 \\ 0 \\ -1 \end{pmatrix}$$

פותרים את מערכת המשוואות הלינארית שמתקבלת ומוצאים את ערך ההתנגדות המתאים. ומשם נוציא את ערך המוליכות.

בקשר לערך ההסתברות המתאים, מכיוון שקיימות 5 קשתות ובהנחה ש-P זהו ערך ההסתברות של קיומה של קשת אחת, מקבלים P^5 .

2.2 דרישות ואפיון הבעיה

בעיית הפרקולציה מורכבת משני שלבים:

- שלב א': חישוב מדויק של בעיית הפרקולציה עבור רשתות קטנות בגודל 2x2, 3x3 ו- 4x4.

בהינתן P -הסתברות קיומה של קשת(נגד) בגריד. נעבור על כל התתי

גרידים שאפשר לקבל ונחשב את וקטור ערכי המוליכות הכוללים עבור כל תת גריד ואת הסיכוי של קבלת כל ערך.

- שלב ב': רנורמליזציה.

בשלב הזה לוקחים את הגריד הקטן שביצענו עליו את החישובים בשלב א', משכפילים אותו מספר פעמים ומקבלים גריד הרבה יותר גדול, ומחשבים את וקטור ערכי המוליכות הכוללים עבורו.

בתוכנה שפיתחנו, סיכמנו עם הפיזיקאים שערך R ההתחלתי יהיה 1. המשתמש יכניס את גודל הרשת שעבורה הוא מעוניין בחישובים, ואת P - הסתברות קיומה של קשת ברשת. לאחר הרצת התוכנה המשתמש יקבל וקטור ערכי המוליכות הכוללים. - איזה פרמטרים נדרוש מהמשתמש להכניס כדי שנאפשר לו הרצת ניסויים שהוא יחשוב עליהם.

2.3 הבעיה מבחינת הנדסת תוכנה

- יצירת תוכנה ידידותית למשתמש, בעלת ממשק נוח ונכון בהתאם לצרכיו ובהתאם לבעיה שהתוכנה באה לפתור.
- קביעת סביבת פיתוח מתאימה לפרויקט מסוג זה ולימוד נכון של השימוש בשפה המתאימה לה.
- שילוב של חישובים פיזיקליים לא פשוטים עם גרפים.
- ביצוע חישובים מתמטיים, פיזיקליים וחיפושים ביעילות גבוהה. לדוגמא, עבור גריד בגודל 2×2 - הגריד הכי קטן שאפשר לקבל - קיימים 32 תתי גרידים שעבור כל אחד מהם צריך לבצע את החיפושים ואת החישובים הפיזיקליים והמתמטיים.

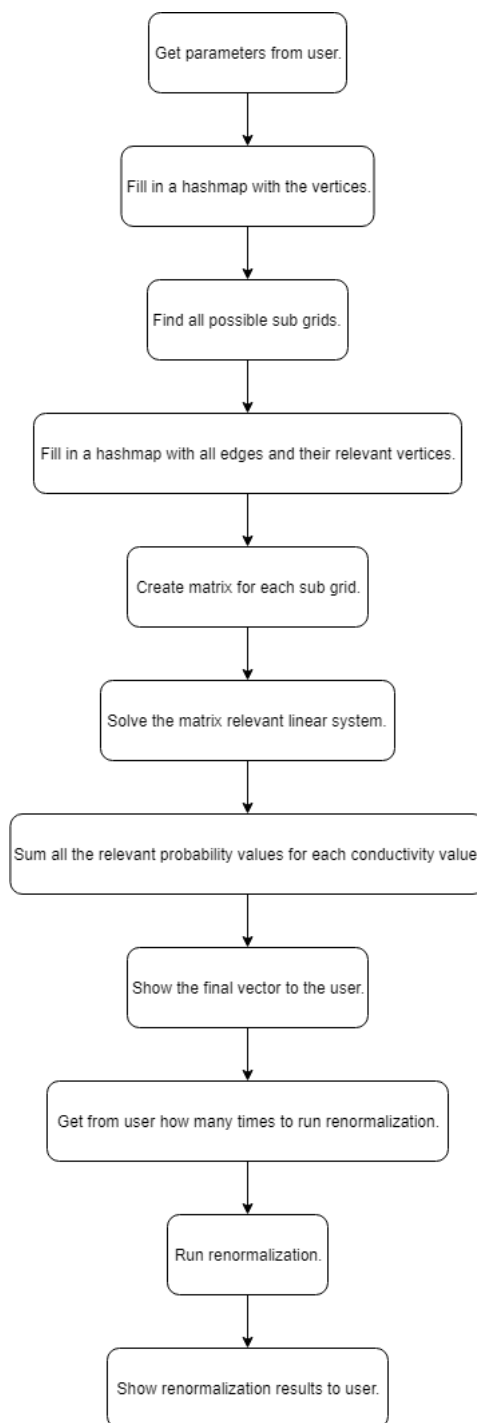
3. תיאור הפתרון

3.1 תיאור המערכת

המערכת שלנו באה לענות על הצורך הקיים אצל הפיזיקאים למערכת שתוכל לעזור להם בחישובי פרקולציה, כך שיוכלו להגיע לתשובות שהם מחפשים בצורה הכי מהירה שיכולה להיות במקום לעשות את הכל בצורה פרימיטיבית וידנית. המערכת שלנו היא תוכנת מחשב בשפת JAVA שעושה את החישובים של שלב א' ושלב ב' של הפרקולציה והרנורמליזציה. בסוף החישובים, תופיע למשתמש טבלה של וקטור ערכי המוליכות הכוללים עם הסיכוי לקבלת כל ערך.

התוכנה פותחה בצורה טובה וגינרית כך שיהיה אפשר בעתיד להשתמש בה לכל מיני חישובים שונים. לדוגמא, יהיה אפשר להשתמש בתוכנית לפתרון עוד בעיות של חישובי זרימה, לאו דווקא מוליכות פיזיקלית.

3.2 פירוט המערכת



1. התוכנה מקבלת מהמשתמש את גודל הגריד שעבורו הוא מעוניין לחשב את הפתרון של פרקולצייה, ואת ההסתברות לקיומה של קשת בגריד.
2. התוכנה מכניסה לתוך HashMap את המספרים מ-0 עד למספר הקודקודים של הגריד-1. כי היא תשתמש ב- HashMap הזה בהמשך למציאת התתי גרידים.
3. התוכנה מחשבת את כל תתי הגרידים של הגריד הנ"ל בעזרת פונקציית `subGrids(int edgesNum, HashMap<Integer, Integer> edges)` שמתמשת ב- HashMap מהשלב הנ"ל.
4. התוכנה מכניסה את הקודקודים והקשתות הרלוונטיים לתוך HashMap בעזרת פונקציית `buildHashMap(int size)`.
(לדוגמא, אם קשת מספר 1 קושרת בין הקודקודים 0 ו-1, נכניס את הערך 1 כמפתח ב- HashMap ואת 0 ו-1 כערכים של המפתח הזה)
5. התוכנה עוברת על כל תת גריד ובונה מטריצה מתאימה באופן הבא :
- במטריצה מספר השורות והעמודות שווה למספר הקודקודים בתת גריד.
- עבור כל זוג של אינדקסים, אם קיימת קשת ביניהם בתת גריד נכניס בתאים במטריצה 1- אחרת נכניס 0.
- באלכסון, עבור כל אינדקס, נכניס את מספר השכנים שלו בתת גריד ואז פותרת את מערכת המשוואות הרלוונטיות בעזרת פונקציית `createMatrix(int n, ArrayList<String> list, int minEdges, double p, double g)` שמתמשת בספריית `apache commons` לפתרון מערכות משוואות לינאריות.
6. התוכנה עוברת על כל הערכים של המוליכות שחישבה ומתחילה לסכום את כל הערכים השונים של ההסתברויות שקיבלה עבור כל ערך של מוליכות בעזרת פונקציית `calculateVals()`.
7. התוכנה מקבלת מהמשתמש מספר איטרציות של רנורמליזציה שהוא מעוניין לבצע. מבצעת רנורמליזציה על התוצאות שחישבה בשלב הקודם בעזרת פונקציית `renormalization(double p, int times)`.

3.3 תהליכים ונתוני המערכת

למערכת יש סוג אחד של משתמשים ; שהם הפיזיקאים המעוניינים בשימוש בתאוריית הפרקולצייה כדי למצוא ערכי מוליכות אפשריים עבור רשת חשמלית . התוכנה מקבלת את גודל הרשת של הבעיה, את- P הסתברות קיומה של קשת ברשת. ואז היא מחשבת את החישובים הנחוצים.

3.4 תיאור הכלים המשמשים לפתרון

- סביבת הפיתוח לתוכנת המחשב: Eclipse.
- בחרנו ב- Eclipse לפיתוח התוכנה משיקולי נוחות, וכיוון שהידע שלנו בה הינו עשיר ומגוון. וגם אפשרות בניית הממשק למשתמש הינה מגוונת ונוחה.
- שפות תכנות: Java.
- בחרנו בשפת Java מהנסיון שיש לנו בשפה הזאת, כמוכן ביצענו פרוייקטים קודמים באותה השפה במסגרת הלימודים והיה לנו מאוד נוח לעבוד איתה.
- מחשב נייד/נייד להפעלת התוכנה.
- יומן עבודה: Google calendar.
- מאגר קוד: Github.
- מערכת הפעלה: Windows.

4. תוכנית בדיקות

- חישוב ידני של כל תתי הרשתות של הרשת בגודל 2×2 , וחישוב תוצאות המוליכות וההסתברות עבור כל תת רשת. והשוואה בין התוצאות שקיבלנו מהחישובים שנעשו ידנית לבין מה שהתוכנה מחזירה.
- בדיקת שהתוכנה מחשבת את כל התתי רשתות עבור רשת נתונה ללא חוסר.
- השוואת ד"ר דורון אזולאי לתוצאות שהתוכנה מחזירה עבור הרשתות בגודל 2×2 ו- 3×3 , מול התוצאות שהוא קיבל בחישובים שעשה ידנית.
- בדיקת שעבור כל ערך אפשרי של מוליכות, וקטור המוליכות הסופי סוכם את כל ההסתברויות השונות שהתקבלו מחישובים על תתי רשתות שונים.
- בדיקת שכל תוצאות החישוב מיוצגות בחלון ה-GUI.
- בדיקת התאמה בין ערכי המוליכות וערכי ההסתברות המיוצגות ב-GUI.
- בדיקת חוקיות הפרמטרים שהמשתמש מכניס.

5. השוואה לספרות

מצאנו קוד פתוח שעושה ויזואליזציה לבעיית הפרקולציה. התוכנה מראה למשתמש את כל האופציות של זרימה עבור גריד דו ממדי. אך לא מצאנו אף תוכנה שפותרת את בעיית הפרקולציה הפיזיקאלית מספרית.

6. סיכום ומסקנות

התוכנה שפותחה באה לענות על צורך אצל הפיזיקאים במכללה, וייתכן שישנה דרישה לתוכנה כזו בשוק. התוכנה פותחה בצורה כך שתוכל לחשב את הפתרון לבעיית הפרקולציה בצורה הכי טובה, מדויקת, ומהירה שיש כדי לחסוך לפיזיקאים את כל ההשקעה הקשה והזמן הארוך שהם שמים בפתרון הבעיה.

בתחילת העבודה על הפרויקט נפגשנו עם ד"ר דורון אזולאי וד"ר איתי פורמן כדי להבין מהי בדיוק בעיית הפרקולציה ומהי רנורמליזציה. והמשכנו להיפגש איתם לאורך כל הדרך מכיוון שגם אם קראנו על הבעיה ועל הפתרון שלה התקשנו להבין.

בהתחלה פיתחנו קוד שהתייחס לגרידים כאובייקטים של גרפים (כתבנו מחלקה של גרף). מצאנו את תתי הגרידים הרלוונטיים והרצנו על כל אחד כזה אלגוריתם BFS כדי למצוא לאיזה מהם אפשר ממש להגיע מצד לשני. ואז חישבנו לאלה את המוליכות ואת ההסתברות לקבלת הערך הזה. אבל לא המשכנו עם הפתרון הזה מכיוון שנתקענו בשלב שבו היינו צריכים לעשות את החישובים המתמטיים על האובייקטים.

אחר כך פיתחנו קוד שפותר בצורה מדויקת את בעיית הפרקולציה של גריד בגודל 2×2 . אך הקוד הזה לא מצא חן בעיניי המנחה מכיוון שהוא לא היה מספיק גינרי והיה קשה לפתח אותו עוד כדי שיטפל בחישובים מדויקים עבור גרידים בגודל 3×3 ו- 4×4 .

כמעט כל הסמסטר הראשון עבדנו על הרעיונות הנ"ל.

בסמסטר השני עלה לנו הרעיון שיישמו לפתרון הבעיה. שהוא בגדול מתבסס על תרגום התתי גרידים למטריצות ומשתמש ב- Apache commons לפתרון המשוואות הלינאריות הרלוונטיות.

התוכנה לוקחת בערך 10 שעות כדי למצוא פתרון מדויק של בעית הפרקולציה עבור גריד בגודל 4×4 . מכיוון שיש בגריד הזה 25 צלעות ולכן 2^{25} תתי גרידים אפשריים. מה שאומר שלחשב במדויק את הפתרון עבור גרידים בגודל 5×5 ומעלה זה כמעט בלתי אפשרי ולכן משתמשים ברנורמליזציה. מכיוון שהפתרון של גריד בגודל 2×2 מחושב מהר, הסכמנו שהגריד הזה יהווה גריד יסודי שישמש אותנו לרנורמליזציה.

7. ספרות

Apache commons: <https://commons.apache.org/>

SWT(The Standard Widget Toolkit): <https://www.eclipse.org/swt/>

Percolation theory: Bernasconi physrevb.18.2185 1978

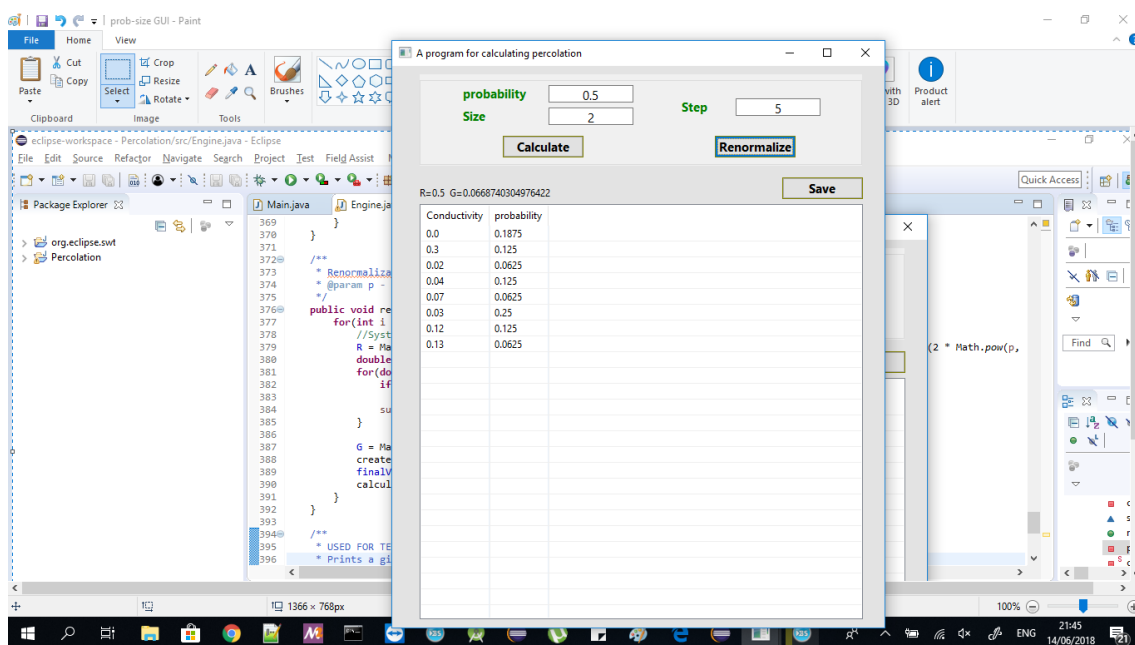
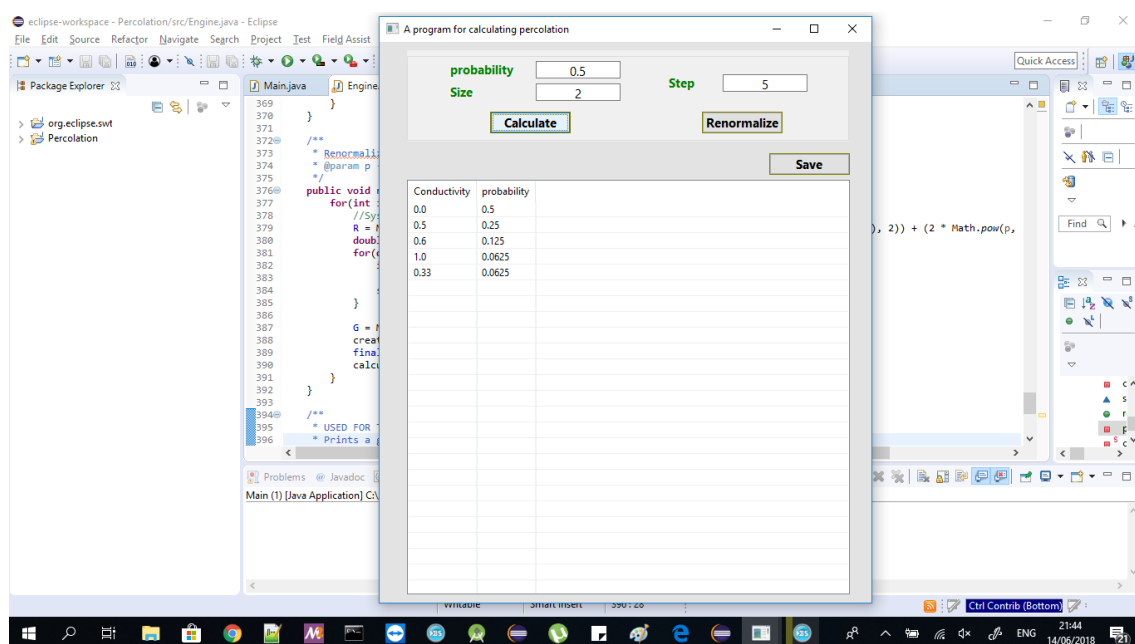
8. מערכות ניהול הפרויקט

מיקום	מערכת
https://github.com/ZeinaAbuSwai/Calculating-Percolation-Program	מאגר קוד
https://calendar.google.com/calendar/embed?src=27bpfsem1mrdg1l29dfge3l0e4%40group.calendar.google.com&ctz=Asia%2FJerusalem	יומן
https://biteable.com/watch/percolation-1917736/	סרטון סופי

9. נספחים

● מסכים

[illegible]





```
result - Notepad
File Edit Format View Help
p=0.5 d=2
0.0 0.1875
0.3 0.125
0.02 0.0625
0.04 0.125
0.07 0.0625
0.03 0.25
0.12 0.125
0.13 0.0625
R=0.5 G=0.0668740304976422
```

• **תכנון הפרויקט**

פגישת הצעת פרויקט עם ד"ר יהודה חסין	08.11.2017
פגישת עם ד"ר יהודה חסין ובירור דרישות	22.11.2017
הגשת דו"ח הצעה	26.11.2017
חישוב מדויק של שלב א' של סריג 2X2	30.12.2017
הגשת דו"ח אלפא וסרטון	01.02.2018
החלטה על שינוי הקוד	15.02.2018
חישוב מדויק של שלב 2X2 בעזרת קוד חדש	10.04.2018
חישוב מדויק של שלב ב' של סריג 2X2	20.04.2018
דו"ח גירסת בטא	25.04.2018
קבלת החלטה לשנות את הקוד וכתובת קוד חדש	01.05.2018
הגשת תוכנה סופית, סרטון, ודו"ח סופי	14.06.2018

• **טבלת סיכונים**

#	הסיכון	חומרה	מענה אפשרי
1	התעקבות בסיום הפרויקט	גבוה	חלוקה ברורה של התפקידים, ושמירה של תכנון פרויקט מעודכן על מנת לפתור את כל המכשולים הצפויים בהקדם האפשרי
2	חוסר ידע בשפת התכנות	בינוני	שימוש בלמידה מכוונת
3	חוסר ידע בפיזיקה	בינוני	פגישות עם הפיסיקאים במכללה
4	אי מציאת API לחישוב מוליכות	בינוני	כתיבת קטעי קוד שעושים את החישובים
5	כתיבת תוכנה גרועה מבחינת יעילות	גבוה	חיפוש אלגוריתמים יעילים קיימים שיכולים לעזור בחשיבה על קוד יותר יעיל
6	לקוח לא מרוצה	בינוני	פגישות עוקבות
7	איבוד הקוד	גבוה	גיבוי הקוד בענן

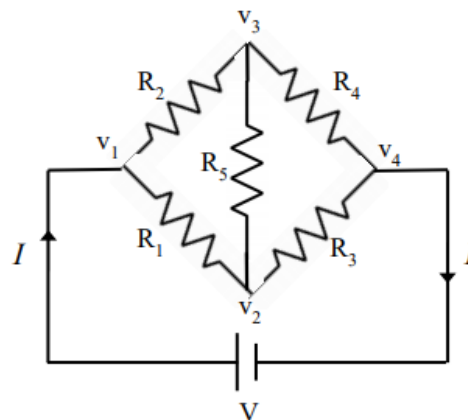


• טבלת דרישות

#	תיאור
1	מבנה קל לשימוש
2	קבלת פרמטרים P-I size מהמשתמש
3	חישוב מדויק עבור גרידים 4X4, 3X3, 2X2
4	ביצוע החישובים במהירות
5	הצגת וקטור החישובים הסופי בצורת טבלה של ערכי מוליכות שהתקבלו ואת הסיכוי לקבלת כל ערך
6	תיכון טוב של התוכנה, כך שבעתיד יהיה אפשר להשתמש בתוכנה לעוד חישובים שונים ולא רק לחישובי מוליכויות

Analysis of a resistors network

ננתח את המעגל הבא:



נסמן: $g=1/R$

$g_{ij} = g_{ji}$ – המוליכות בין שתי הנקודות (ij) ומתקיים

מחוקי קירכהוף:

$$g_{12}(v_1-v_2)+g_{13}(v_1-v_3)=I$$

$$g_{21}(v_2-v_1)+g_{23}(v_2-v_3)+g_{24}(v_2-v_4)=0$$

$$g_{31}(v_3-v_1)+g_{32}(v_3-v_2)+g_{34}(v_3-v_4)=0$$

$$g_{42}(v_4-v_2)+g_{43}(v_4-v_3)=-I$$

$$R_{total}=1/G_{total}=V/I=(v_1-v_4)/I \quad \text{וכן:}$$



$$\begin{pmatrix} g_{12} + g_{13} & -g_{12} & -g_{13} & 0 \\ -g_{21} & g_{21} + g_{23} + g_{24} & -g_{23} & -g_{24} \\ -g_{31} & -g_{32} & g_{31} + g_{32} + g_{34} & -g_{34} \\ 0 & -g_{42} & -g_{43} & g_{42} + g_{43} \end{pmatrix} \cdot \begin{pmatrix} v_1 \\ v_2 \\ v_3 \\ v_4 \end{pmatrix} = \begin{pmatrix} I \\ 0 \\ 0 \\ -I \end{pmatrix}$$

In the simple case where $g(i,j) = 1/R$

In your simple case:

$$\begin{pmatrix} 2/R & -1/R & -1/R & 0 \\ -1/R & 3/R & -1/R & -1/R \\ -1/R & -1/R & 3/R & -1/R \\ 0 & -1/R & -1/R & 2/R \end{pmatrix} \cdot \begin{pmatrix} v_1 \\ v_2 \\ v_3 \\ v_4 \end{pmatrix} = \begin{pmatrix} I \\ 0 \\ 0 \\ -I \end{pmatrix}$$

Abstract

Percolation theory describes the physics and mathematics of long-term connections in random systems. It is a phenomenon that describes the permeability of liquid through matter, and it applies to a wide variety of physical systems, including conductivity, porosity and polymers.

But can also be used to analyze systems such as forest fires, disease epidemics, and social networks.

In this project that has been done by a teamwork, a standalone Java software was developed to calculate for a network of any size with any number and order of resistors, the possible electrical conductivity values and the probability of obtaining any such value.



Software Engineering Department

Calculating percolation of conductivity using renormalization

By:

Zeina Abu-Swai 203591672

Fuad Naser Aldin 201211570

Academic supervisor: Dr. Yehuda Hassin

Industrial supervisor: Dr. Yehuda Hassin

Projects coordinator: Dr. Assaf Spanier

Approval:

Approval:

Approval:

Date:

Date:

Date:

June 2018

סיוון ה'תשע"ח