שזירות במצבים קוונטיים

לב קביטני

שזירות במצבים קוונטיים

חיבור על עבודת גמר

לשם מילוי חלקי של הדרישות לקבלת התואר
מגיסטר למדעים בפיסיקה

לב קביטני

הוגש לסנט הטכניון - מכון טכנולוגי לישראל

סיון תשע״א　　חיפה　　יוני 2011

עבודת הגמר נעשתה בהנחיית פרופ׳ יוסף אברון בפקולטה לפיסיקה.

תוכן עניינים:

תקציר 1
רשימת סמלים וקיצורים 2
פרק 1 – מבוא ורקע ספרותי 3
1.1 – מטריצת צפיפות 3
1.2 – הגדרת מצב שזור 3
1.3 – קריטריון פרס לפריקות במצבים קוונטיים 3
1.4 – הפעלת קריטריון פרס על מצב בפירוק שמידט 4
1.5 – מצבי Bound Entanglement 6
1.5.1 – הגדרת מצב Bound-Entangled באמצעות UPB 6
1.5.2 – דוגמא ליצירת מצבי Bound Entanglement באמצעות UPB 7
פרק 2 – מטרת העבודה 9
פרק 3 – תיאור העבודה 10
פרק 4 – שיטות העבודה 13
4.1 – הגדרת נתונים התחלתיים 13
4.2 – תיאור הלולאה הראשית בתוכנית 13
4.3 – פלט התוכנית 14
פרק 5 – תוצאות העבודה 15
5.1 – מצב שזור 15
5.2 – מצב פריק 16
5.3 – מצב פריק-טהור 17
5.4 – מצב מעורב תלוי פרמטר 19
פרק 6 – דיון והסקת מסקנות 21
נספח א' – קוד התוכנית בשפת MATLAB 22
תוכנית "separability" 22
פונקציית עזר "ro_p_comp" 25
נספח ב' – פתרון בעיית תכנות ריבועית ב-MATLAB 26
רשימת מקורות 27

רשימת איורים:

1	דוגמא ליצירת UPB באמצעות וקטורי פירמידה מחומשת	7
2	תיאור סכמטי של חיפוש קירוב פריק טוב יותר למצב נתון	10

תקציר:

שזירות הינה תופעה במכניקת הקוונטים שבה מצבים קוונטיים של שתי מערכות (או יותר) מתוארים תמיד אחד ביחס לשני, גם אם המערכות מרוחקות פיזית זו מזו. כך למשל, ברגע בו נערכת מדידה על אחת המערכות, היא משתקפת באופן מיידי גם במערכת השניה.

קורלציה כזו לא ניתן לתאר במונחי הסתברות של הפיסיקה הקלאסית אלא רק כפונקציית גל המתארת את המצב הקוונטי המשותף של כל המערכות.

השזירות משחקת תפקיד מפתח ביישומים רבים של חישוב קוונטי ואינפורמציה קוונטית, כגון טלפורטציה וקריפטוגרפיה.

במסגרת עבודה זו נבחנו מספר שיטות קיימות לאיתור השזירות ולהגדרת המצבים השזורים.

בתחילת החיבור מוצגת הגדרה של מטריצת צפיפות, תכונותיה, והגדרה של מצב שזור. מיד לאחר מכן מוצג קריטריון פרס לפריקות (ספרביליות – ההיפך משזירות). מדובר בקריטריון אלגברי פשוט, אותו ניתן להפעיל על מטריצה המתארת מצב קוונטי המורכב משתי תת-מערכות (למשל, שני קיוביטים). הקריטריון מספק תנאי הכרחי לפריקות, ועבור מצבים ממימד נמוך (מצב של שני קיוביטים, או מצב של קיוביט וקיוטריט) הקריטריון הוא גם תנאי מספיק. בהמשך, מופיעה הוכחה שעבור מצב טהור, קריטריון פרס לפריקות שקול לקריטריון שמידט לפריקות. ההוכחה הינה באמצעות הפעלת קריטריון פרס על מצב בפירוק שמידט.

בהמשך החיבור מוצגים מצבי Bound-Entanglement - אלה מצבים שזורים של מספר מערכות ממימד גבוה (למשל, שני קיוטריטים ומעלה), שעל אף היותם שזורים, מטריצות הצפיפות שלהם כן מקיימות את קריטריון פרס לפריקות. אחת הדרכים לבניית מצבים כאלה הינה באמצעות UPB (Unextendible Product Basis). זהו בסיס אורתוגונאלי המורכב ממצבי מכפלה טנזורית של מספר תת-מערכות קוונטיות. בסיס זה אינו שלם, ותת-המרחב המשלים שלו אינו מכיל אף מצב של מכפלה טנזורית. המצב המעורב מקסימאלית מעל תת המרחב המשלים של כל UPB הוא מצב Bound Entangled.

החלק המרכזי של העבודה עוסק בשיטה נומרית לחקירה של פריקות ושזירות. השיטה מתבססת על העובדה שכל קבוצה של מצבים פריקים הינה קמורה, ועושה שימוש בתהליך איטרטיבי למציאת המצב הפריק הקרוב ביותר למטריצת צפיפות נתונה, ללא הגבלה של מימדי המטריצה.

מהות העבודה הינה מימוש אלגוריתם זה בתוכנת MATLAB והרצת התוכנית על מטריצות שונות המייצגות מצבים של שני קיוביטים. התוכנית מקבלת מטריצה בגודל 4x4 וקובעת האם מטריצה זו פריקה או שזורה. אם המטריצה פריקה, התוכנית מחזירה בפלט את פירוקה למצבים פריקים-טהורים, המהווים את נקודות הקיצון של תחום המטריצות הפריקות. אם המטריצה שזורה, התוכנית מחזירה בפלט את המטריצה הפריקה הקרובה אליה ביותר ואת המרחק ביניהן.

גם אופן המימוש של האלגוריתם וגם הדוגמאות להרצות השונות שבוצעו (עבור מצב שזור, מצב פריק, מצב פריק-טהור, ומצב תלוי פרמטר) מתוארים בפרקים המתאימים של העבודה.

בסיכום העבודה מוצגות המסקנות מכל ההרצות שבוצעו, לרבות השוואת הפלטים של תוכנית המחשב אל מול התוצאות המתקבלות מבדיקה של אותן מטריצות באמצעות קריטריון פרס (שכזכור, עבור מצבים של שני קיוביטים קובע חד-חד ערכית אם המצב פריק או שזור).

לסיכום, התוכנית המממשת את האלגוריתם מסווגת נכון את המטריצות המתארות שני קיוביטים בהיבט שזירות/פריקות. כיוון שהאלגוריתם אוניברסלי לכל מימד של מטריצה נתונה, ניתן להתאים את התוכנית (על ידי עדכון פשוט של הקוד) כך שתתמוך גם בחקירת מצבים ממימדים גבוהים יותר (למשל מצב של שני קיוטריטים), היכן שקריטריון פרס הינו רק תנאי הכרחי (אך לא מספיק) לפריקות.

רשימת סמלים וקיצורים:

PT	Partial Transform
UPB	Unextendible Product Basis
ρ	מטריצת צפיפות

פרק 1 – מבוא ורקע ספרותי:

1.1 – מטריצת צפיפות:

מטריצת צפיפות ρ המתארת מערכת קוונטית (למשל, קיוביט) הינה בעלת התכונות הבאות:

1) אי-שלילית:

$$(1)\quad \forall|\phi\rangle,\quad \langle\phi|\rho|\phi\rangle \geq 0$$

(מכאן נובע שהמטריצה ρ גם הרמיטית: $\rho^\dagger = \rho$)

2) מנורמלת:

$$(2)\quad Tr(\rho) = 1$$

מטריצות המקיימות תכונות אלו יוצרות סט קמור וניתנות לכתיבה כ:

$$(3)\quad \rho = \sum_k p_k |\psi_k\rangle\langle\psi_k|$$

כאשר $|\psi_k\rangle$ הינם וקטורי יחידה במרחב הילברט והמקדמים הינם הסתברויות ($\sum_k p_k = 1, \quad p_k \geq 0$). כלומר, p_k הינו ההסתברות למצוא את המערכת במצב טהור $|\psi_k\rangle$.

המצבים הטהורים מהצורה $|\psi_k\rangle\langle\psi_k|$ הינם נקודות הקיצון (קודקודים) של הסט הקמור של מטריצות הצפיפות, ועל-ידי קומבינציה קמורה של מצבים טהורים אלו ניתן לבנות מטריצת צפיפות של כל מצב מעורב.

1.2 – הגדרת מצב שזור:

מצב המורכב מ-2 תת מערכות הינו פריק אם ניתן לכתוב את מטריצת הצפיפות שלו כסכום של מכפלות טנזוריות [1]:

$$(4)\quad \rho^{AB} = \sum_l w_l \rho_l^A \otimes \rho_l^B$$

כאשר ρ^A, ρ^B הן מטריצות הצפיפות של תת-המערכות A ו-B (בהתאמה), והמשקלים החיוביים w_l מקיימים:

$$(5)\quad \sum_l w_l = 1$$

המצב מוגדר כשזור כשלא ניתן להשיג עבורו פירוק (4) כזה.

1.3 – קריטריון פרס לפריקות במצבים קוונטיים:

אשר פרס נתן קריטריון אלגברי פשוט [2] שמהווה תנאי הכרחי לקיום ההצגה של סכום מכפלות טנזוריות (4), ומכאן, לפריקות.

נסתכל על אלמנטי מטריצת הצפיפות ρ^{AB}:

$$(6)\quad \rho^{AB}_{m\mu,n\upsilon} = \sum_l w_l \left(\rho_l^A\right)_{mn} \cdot \left(\rho_l^B\right)_{\mu\upsilon}$$

כאן m,n הם האינדקסים של מערכת A ו- μ,υ הם האינדקסים של מערכת B.

כעת נגדיר מטריצה חדשה σ, אותה נקבל ע"י ביצוע שחלוף חלקי (PT – Partial Transpose) למטריצת ρ^{AB}:

$$(7)\quad \sigma_{m\mu,n\upsilon} \equiv \rho^{AB}_{n\mu,m\upsilon} = \sum_l w_l \left(\rho_l^A\right)_{nm} \cdot \left(\rho_l^B\right)_{\mu\upsilon}$$

כלומר, ביצענו שחלוף רק למטריצות הצפיפות של מערכת A :

$$\rho^{AB} \xrightarrow{PT} \sigma = \sum_l w_l \left(\rho_l^A\right)^T \otimes \rho_l^B \qquad (8)$$

הטרנספורמציה מ- ρ^{AB} ל- σ אינה יוניטרית אך עדיין המטריצה σ היא הרמיטית.

כעת נשים לב כי המטריצות ρ_l^A הינן מטריצות צפיפות ולכן הן אי-שליליות ובעלות $tr(\rho_l^A)=1$.
מטריצות השחלוף שלהן $\left(\rho_l^A\right)^T$ גם הן מטריצות אי-שליליות עם $trace=1$, ולכן גם הן מטריצות צפיפות חוקיות. לכן המטריצה σ (המורכבת ממטריצות צפיפות) הינה בהכרח מטריצת צפיפות חוקית עם ערכים עצמיים אי-שליליים.

קיבלנו, אם כן, שמטריצת הצפיפות של מצב פריק ρ^{AB} תמיד נשארת חיובית גם לאחר ביצוע שחלוף חלקי.

איך מיישמים את הקריטריון?
למשל, נתונה מטריצת צפיפות של מערכת המורכבת משני קיוביטים :

$$\rho^{AB} = \begin{pmatrix} \rho_{11} & \rho_{12} & \rho_{13} & \rho_{14} \\ \rho_{21} & \rho_{22} & \rho_{23} & \rho_{24} \\ \rho_{31} & \rho_{32} & \rho_{33} & \rho_{34} \\ \rho_{41} & \rho_{42} & \rho_{43} & \rho_{44} \end{pmatrix} \triangleq \begin{pmatrix} (A)_{2x2} & (B)_{2x2} \\ (C)_{2x2} & (D)_{2x2} \end{pmatrix} \qquad (9)$$

נמצא את σ ע"י ביצוע שחלוף חלקי :

$$\rho^{AB} \xrightarrow{PT} \rho^{PT} \equiv \sigma = \begin{pmatrix} (A)_{2x2} & (C)_{2x2} \\ (B)_{2x2} & (D)_{2x2} \end{pmatrix} = \begin{pmatrix} \rho_{11} & \rho_{12} & \rho_{31} & \rho_{32} \\ \rho_{21} & \rho_{22} & \rho_{41} & \rho_{42} \\ \rho_{13} & \rho_{14} & \rho_{33} & \rho_{34} \\ \rho_{23} & \rho_{24} & \rho_{43} & \rho_{44} \end{pmatrix} \qquad (10)$$

כעת, יש לבדוק את הערכים העצמיים של σ :

1) אם σ היא מטריצה שלילית (כלומר, לפחות ערך עצמי אחד שלה שלילי) אזי המטריצה ρ^{AB} מייצגת בהכרח מצב שזור (אינו פריק). משפט זה נכון ללא תלות במימדים של תת-המערכות A ו-B.
2) אם σ היא מטריצה אי-שלילית (כלומר, כל הערכים העצמיים שלה אי-שליליים), אזי המטריצה ρ^{AB} מייצגת מצב פריק שניתן לכתיבה כ-(4). משפט זה נכון רק עבור מערכת המורכבת משני קיוביטים או מערכת המורכבת מקיוביט וקיוטריט.

לסיכום, קריטריון פרס הינו תנאי הכרחי לפריקות אך לא תמיד תנאי מספיק.

1.4 – הפעלת קריטריון פרס על מצב בפירוק שמידט :

נתון מצב טהור $\left|\psi^{AB}\right\rangle$ המתאר 2 תת-מערכות A ו-B :

$$\left|\psi^{AB}\right\rangle = \sum_{i,k} a_{ik} \left|i_A\right\rangle \left|k_B\right\rangle \qquad (11)$$

בהכרח קיים בסיס אורתונורמאלי $\left|\varphi_j\right\rangle$ עבור מערכת A, ובסיס אורתונורמאלי $\left|\chi_j\right\rangle$ עבור מערכת B, כך ש- $\left|\psi^{AB}\right\rangle$ ניתן לכתיבה כפירוק שמידט :

$$|\psi\rangle = \sum_j \lambda_j |\varphi_j\rangle |\chi_j\rangle \quad (12)$$

כאשר המקדמים λ_j נקראים "מקדמי שמידט". מקדמים אלו ממשיים, חיוביים ומקיימים:

$$\sum_j \lambda_j^2 = 1 \quad (13)$$

מספר המקדמים שאינם אפס נקרא "מספר שמידט".

כידוע, קריטריון שמידט קובע שמצב טהור של מערכת המורכבת משתי תת-מערכות הינו פריק רק אם מספר שמידט שלו שווה ל-1. כלומר, בפירוק שמידט של המצב ישנו רק איבר אחד עם מקדם יחיד: $\lambda_j = 1$

אנו נרצה להוכיח שעבור מצב טהור קריטריון פרס לפריקות שקול לקריטריון שמידט לפריקות. כלומר נרצה להוכיח את השקילות:

$$\forall|\phi\rangle, \quad \langle\phi|\rho^{PT}|\phi\rangle \geq 0 \quad \Leftrightarrow \quad \lambda_1 = 1, \quad \lambda_{j\neq 1} = 0 \quad (14)$$

<u>הוכחה:</u>
נסתכל על מצב כללי בפירוק שמידט (12).

מטריצת הצפיפות המתארת מצב זה הינה:

$$\rho = |\psi\rangle\langle\psi| = \sum_{jk} \lambda_j \lambda_k \left(|\varphi_j\rangle\langle\varphi_k|\right) \otimes \left(|\chi_j\rangle\langle\chi_k|\right) \quad (15)$$

נבצע למצב זה שחלוף חלקי ונקבל:

$$\sigma = \rho^{PT} = \sum_{jk} \lambda_j \lambda_k \left(|\varphi_k\rangle\langle\varphi_j|\right) \otimes \left(|\chi_j\rangle\langle\chi_k|\right) \quad (16)$$

כעת נבדוק באילו תנאים אופרטור זה חיובי, כלומר מתי מתקיים (1):

$$\forall\phi \quad \langle\phi|\rho^{PT}|\phi\rangle \geq 0 \quad (17)$$

לבדיקה נבחר מצב $|\phi\rangle$ כללי ביותר בבסיס של מצבי שמידט של $|\psi\rangle$:

$$|\phi\rangle = \sum_{i,\ell} \mu_{i\ell} |\varphi_i\rangle |\chi_\ell\rangle \quad (18)$$

מכאן ש:

$$\begin{aligned} (19) \quad \langle\phi|\rho^{PT}|\phi\rangle &= \sum_{i,i',\ell,\ell',k,j} \lambda_j \lambda_k \mu_{i\ell} \mu_{i'\ell'} \langle\varphi_{i'}|\langle\chi_{\ell'}|\left(|\varphi_k\rangle\langle\varphi_j|\right) \otimes \left(|\chi_j\rangle\langle\chi_k|\right)|\varphi_i\rangle|\chi_\ell\rangle = \\ &= \sum_{i,i',\ell,\ell',k,j} \lambda_j \lambda_k \mu_{i\ell} \mu_{i'\ell'} \cdot \underbrace{\langle\varphi_{i'}|\varphi_k\rangle}_{\delta_{i'k}} \underbrace{\langle\varphi_j|\varphi_i\rangle}_{\delta_{ji}} \otimes \underbrace{\langle\chi_{\ell'}|\chi_j\rangle}_{\delta_{\ell'j}} \underbrace{\langle\chi_k|\chi_\ell\rangle}_{\delta_{k\ell}} = \\ &= \sum_{k,j} \lambda_j \lambda_k \mu_{jk} \mu_{kj} = \langle\phi|\rho^{PT}|\phi\rangle \end{aligned}$$

כעת נבדוק עבור אילו ערכי λ_j, λ_k מתקיים:

$$\sum_{k,j} \lambda_j \lambda_k \mu_{jk} \mu_{kj} \geq 0 \quad (20)$$

עבור λ_j יחיד (כלומר, $\lambda_1 = 1, \quad \lambda_{j\neq 1} = 0$) קריטריון פרס מתקיים מיד:

$$\langle\phi|\rho^{PT}|\phi\rangle = \sum_{k,j} \lambda_j \lambda_k \mu_{jk} \mu_{kj} = \lambda_1^2 \cdot \mu_{11}^2 \geq 0 \quad (21)$$

אם יש יותר מ-λ_j אחד (כלומר, $|\psi\rangle$ בפירוק שמידט הוא סכום של שני איברים או יותר) נראה שתמיד קיימת דוגמא נגדית שתביא לתוצאה שלילית:

$$\langle\phi|\rho^{PT}|\phi\rangle = \sum_{k,j} \lambda_j \lambda_k \mu_{jk} \mu_{kj} < 0 \quad (22)$$

למשל, הדוגמא הבאה:

$$(23)\quad \mu_{jk}=\begin{cases}\neq 0, & j,k=a,b\\ \neq 0, & j,k=b,a\\ =0, & else\end{cases}$$

במקרה זה, מכל הסכום $\sum_{k,j}\lambda_j\lambda_k\mu_{jk}\mu_{kj}$ יישארו שני איברים בלבד:

$$(24)\quad \sum_{k,j}\lambda_j\lambda_k\mu_{jk}\mu_{kj}=\lambda_a\lambda_b\mu_{ab}\mu_{ba}+\lambda_b\lambda_a\mu_{ba}\mu_{ab}=2\lambda_a\lambda_b\mu_{ab}\mu_{ba}$$

נבחר למשל, $\mu_{ab}=\lambda_a\lambda_b$ (מכאן ש: $\mu_{ba}=-\sqrt{1-\lambda_a^2\lambda_b^2}$ כי $|\phi\rangle$ מנורמל), נציב ב-(24):

$$(25)\quad \sum_{k,j}\lambda_j\lambda_k\mu_{jk}\mu_{kj}=2\lambda_a\lambda_b\mu_{ab}\mu_{ba}=-2\underbrace{(\lambda_a\lambda_b)^2}_{\geq 0}\cdot\underbrace{\sqrt{1-\lambda_a^2\lambda_b^2}}_{\geq 0}\leq 0$$

קיבלנו שהמטריצה ρ^{PT} הינה חיובית רק במקרה שהמצב בפירוק שמידט מכיל איבר יחיד ($\lambda=1$). כלומר, רק אם מספר שמידט שווה ל-1, רק אז מתקיים קריטריון פרס לפריקות:

$$(26)\quad \forall\phi\quad \langle\phi|\rho^{PT}|\phi\rangle\geq 0$$

וזה בדיוק מה שנדרש היה להוכיח (14). □

1.5 – מצבי Bound Entanglement:

מצבי Bound Entanglement הינם מצבים שזורים מעורבים שלא ניתן לזקק מתוכם שזירות טהורה. בפרט, מטריצות הצפיפות של מצבים אלה מקיימות את קריטריון פרס לפריקות (מתקבלת ρ^{PT} אי-שלילית), על אף שבפועל המצבים הינם שזורים.

1.5.1 – הגדרת מצב Bound-Entangled באמצעות UPB [3]:

נסתכל על מערכת קוונטית המורכבת מ-2 תת-מערכות: $\mathrm{H}=\mathrm{H}_1\otimes\mathrm{H}_2$. כל תת-מערכת הינה בעלת מימד d_i $(i=1,2)$.

S – הינו בסיס אורתוגונאלי לא שלם (מבחינת מימד) של מצבי מכפלה טנזורית, הפורסים תת מרחב H_S של H.

S מוגדר כ- UPB (Unextendible Product Basis) אם תת המרחב המשלים שלו $(\mathrm{H}-\mathrm{H}_S)$ אינו מכיל אף מצב של מכפלה טנזורית.

באמצעות ה-UPB ניתן לבנות מצבי Bound Entanglement: דהיינו, אם נתון בסיס UPB בעל n איברים: $\{|\psi_j\rangle:\quad j=1,....,n\}$

אזי המצב המעורב מקסימאלית מעל תת המרחב המשלים של UPB זה (במרחב הילבט ממימד כולל D) הוא מצב Bound Entangled:

$$(27)\quad \bar{\rho}=\frac{1}{D-n}\left(\mathrm{I}-\sum_{j=1}^{n}|\psi_j\rangle\langle\psi_j|\right)$$

<u>הוכחה:</u>

1) האופרטור $\bar{\rho}$ הוא ההטל לתוך תת-המרחב המשלים של ה-UPB (נוצר ע"י הפרש בין הטלה על כל המרחב ה-D מימדי (I_{DXD}) לבין ההטלה על המרחב ה-n מימדי של UPB נתון). לפי ההגדרה של UPB, תת המרחב המשלים שלו אינו מכיל מצבי מכפלה טנזורית, לכן $\bar{\rho}$ בהכרח מתאר מצב שזור.
2) נשים לב כי $\bar{\rho}$, מעצם הגדרתו, מקיים את קריטריון פרס לפריקות: מטריצת היחידה הינה אינווריאנטית תחת שחלוף חלקי (PT), ומצבי המכפלה הטנזורית המרכיבים את ה-

UPB הופכים, לאחר ביצוע שחלוף חלקי, לסט אורתוגונאלי אחר של מצבי מכפלה טנזורית. כך שבסך הכל, $\bar{\rho}^{PT}$ הינה גם מטריצת צפיפות חוקית, ולכן הינה חיובית.

לכן $\bar{\rho}$ הינו מצב שזור שכן מקיים את קריטריון פרס, כלומר, זהו מצב Bound Entangled. □

1.5.2 – דוגמא ליצירת מצבי Bound Entanglement באמצעות UPB:

נתבונן בפירמידה הבאה [3]:

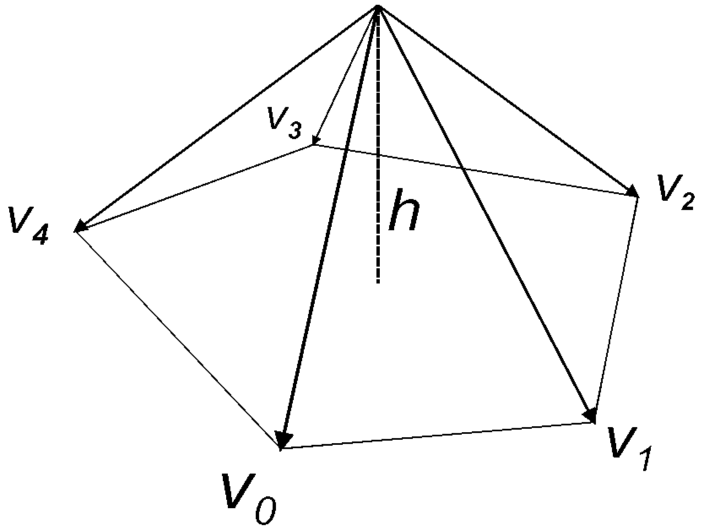

Figure 1 – A pentagonal pyramid in 3D space. The height (h) of the pyramid is chosen such that nonadjacent vectors are orthogonal. Using the vectors (V_1,..,V_5) one can combine 5 states in 3x3 Hilbert space that form a UPB. The entire 4-dimensional subspace complementary to the UPB contains no product-state, so that the uniform mixed state over that subspace is a bound entangled state.

איור 1 – פירמידה מחומשת במרחב תלת-מימדי שבעזרת וקטורי המקצועות שלה ניתן לייצר 5 מצבים במרחב הילברט 3x3, כך שיהוו בסיס אורתוגונאלי שאינו ניתן להרחבה (UPB). תת-המרחב המשלים ל-UPB זה (תת-מרחב 4 מימדי) אינו מכיל אף מכפלה טנזורית, ולכן המצב המעורב מקסימאלית מעל תת-מרחב זה הינו Bound Entangled. הגובה (h) של הפירמידה הוא כזה שכל שני וקטורים שאינם סמוכים – מאונכים זה לזה.

$\vec{v}_0,\vec{v}_1,..,\vec{v}_4$ הם 5 וקטורים במרחב ממשי תלת מימדי המהווים מקצועות של פירמידה מחומשת. גובה הפירמידה (h) נבחר כך שכל שני וקטורים שאינם סמוכים – אורתוגונליים זה לזה. הוקטורים הם:

$$j=0,..,4 \quad \vec{v}_j = N\left(\cos\frac{2\pi j}{5},\quad \sin\frac{2\pi j}{5},\quad h\right) \qquad (28)$$

כאשר: $N = 2\big/\sqrt{5+\sqrt{5}}$ ו- $h=\frac{1}{2}\sqrt{1+\sqrt{5}}$

כעת ניצור מוקטורים אלה 5 מצבי 3x3 (כלומר, הרכבה של שני קיוטריטים – אחד מתת-מערכת A והשני מתת-מערכת B):

$$\left|\psi_j\right\rangle = \left|\vec{v}_j\right\rangle_A \otimes \left|\vec{v}_{2j \bmod 5}\right\rangle_B, \quad j=0,..,4 \qquad (29)$$

נרשום אותם בצורה מפורשת:

$$\begin{cases} \left|\psi_0\right\rangle = \left|\vec{v}_0\right\rangle \otimes \left|\vec{v}_0\right\rangle \\ \left|\psi_1\right\rangle = \left|\vec{v}_1\right\rangle \otimes \left|\vec{v}_2\right\rangle \\ \left|\psi_2\right\rangle = \left|\vec{v}_2\right\rangle \otimes \left|\vec{v}_4\right\rangle \\ \left|\psi_3\right\rangle = \left|\vec{v}_3\right\rangle \otimes \left|\vec{v}_1\right\rangle \\ \left|\psi_4\right\rangle = \left|\vec{v}_4\right\rangle \otimes \left|\vec{v}_3\right\rangle \end{cases} \qquad (30)$$

נשים לב לתכונות הבאות של המצבים:

1) הם כולם אורתוגונליים זה לזה: המצבים שהפרש האינדקסים ביניהם הוא 2 mod 5 אורתוגונליים בגלל אורתוגונליות של המצבים בתת-מערכת A ($\psi_0 \perp \psi_2, \psi_1 \perp \psi_3, \psi_2 \perp \psi_4, \psi_3 \perp \psi_0, \psi_4 \perp \psi_1$). והמצבים שהפרש האינדקסים ביניהם הוא 1 mod 5 אורתוגונליים בגלל אורתוגונליות של המצבים בתת-מערכת B ($\psi_0 \perp \psi_1, \psi_1 \perp \psi_2, \psi_2 \perp \psi_3, \psi_3 \perp \psi_4, \psi_4 \perp \psi_0$).
2) כדי שמצב חדש כלשהו יהיה אורתוגונלי לכל חמשת המצבים הקיימים, עליו להיות אורתוגונאלי לפחות ל-3 מצבים של תת-מערכת A או לפחות ל-3 מצבים של תת-מערכת B. זה, כמובן, בלתי אפשרי כי כל סט של 3 וקטורים $\vec{v}_j$ פורס את המרחב השלם התלת מימדי, ולכן מונע קיום של וקטור נוסף שיהיה אורתוגונאלי אליהם.

לסיכום, בסיס המכפלה הטנזורית שיצרנו $\{|\psi_0\rangle,..,|\psi_4\rangle\}$ הינו אורתוגונאלי, אינו שלם (מכיל רק 5 מצבים בעוד שהמרחב הכולל הוא 9 מימדי), ואינו ניתן להרחבה (לא ניתן למצוא מצבי מכפלה טנזורית נוספים שיהיו אורתוגונאליים ל<u>כל</u> חמשת המצבים הקיימים במרחב הילברט נתון). לכן בסיס זה הינו UPB, ותת המרחב המשלים שלו (מרחב 4 מימדי) אינו מכיל אף מצב של מכפלה טנזורית.

כעת ניתן לבנות מצבי bound-entanglement מעל תת-המרחב המשלים:

$$\bar{\rho} = \frac{1}{D-n}\left(\mathrm{I} - \sum_{j=0}^{4} |\psi_j\rangle\langle\psi_j|\right) \qquad (31)$$

במקרה שלנו, D=9, n=5, I הינה מטריצת יחידה 9x9, ו- $|\psi_j\rangle$ הינם מתוך משוואה (30).

חשוב לציין שיצירת מצבי Bound Entanglement באמצעות UPB הינו רק אופן אחד ליצירת מצבים כאלו. כלומר, קיימים מצבי Bound Entanglement נוספים שלא ניתן ליצור אותם באמצעות UPB אלא באופנים אחרים [4].

<u>פרק 2 – מטרת העבודה:</u>

כאמור, עבור מצב של שתי תת-מערכות ממימד גבוה, קריטריון פרס אינו מהווה תנאי הכרחי בקביעת פריקות, ומצב שזור (במימדים כאלה) יכול או לקיים או שלא לקיים את הקריטריון. כך למשל ראינו את מצבי Bound Entanglement שעל אף היותם שזורים, מקיימים את קריטריון פרס לפריקות (מתקבלת ρ^{PT} אי-שלילית).

מטרת העבודה הינה פיתוח תוכנת מחשב בשפת MATLAB המממשת אלגוריתם נומרי לחקירת שזירות ופריקות במצבים קוונטיים [5]. להבדיל מקריטריון פרס, אלגוריתם זה אוניברסלי למצבים מכל מימד.

פרק 3 – תיאור העבודה:

להלן, הצגת האלגוריתם הנומרי לחקירת שזירות ופריקות במצבים קוונטיים [5].

האלגוריתם מתבסס על העובדה שכל סט של מצבים פריקים הוא קמור, וכל מצב (פריק) במרחב זה ניתן לכתיבה כצירוף לינארי של המצבים הפריקים-הטהורים שבקודקודי המרחב, לפי הנוסחה (3) שבפרק הראשון.

האלגוריתם הינו תהליך איטרטיבי למציאת המצב הפריק הקרוב ביותר למטריצת צפיפות נתונה:

נתונה מטריצת צפיפות ρ המתארת מצב של 2 תת-מערכות (למשל, שני קיוביטים). מטרתנו לחקור מצב זה ולברר האם הוא פריק או שזור.

נסמן ב- ρ_s את המטריצה המייצגת את המצב הפריק הקרוב ביותר למצב ρ באיטרציה נתונה.
מכאן ש- ρ_s נמצאת בתוך הסט הקמור S של מטריצות הצפיפות הפריקות (ראה איור 2).

בשלב הראשון נקבע, למשל, את ρ_s להיות המצב המעורב מקסימאלית:

$$(1) \quad \rho_s = I/n$$

נסמן את ההפרש בין ρ ל- ρ_s:

$$(2) \quad \sigma = \rho - \rho_s$$

בכדי לשפר את הקירוב ρ_s, נחפש מצב פריק טהור ρ_p, כזה שמביא למקסימום את המכפלה הסקלרית הבאה:

$$(3) \quad s = Tr\left[(\rho_p - \rho_s)\sigma\right]$$

או לחילופין את הביטוי הבא:

$$(4) \quad s' = Tr(\rho_p \sigma)$$

המצב ρ_p, מעצם היותו מצב פריק טהור, הינו אחד הקודקודים שעל השפה של התחום הקמור של המצבים הפריקים.

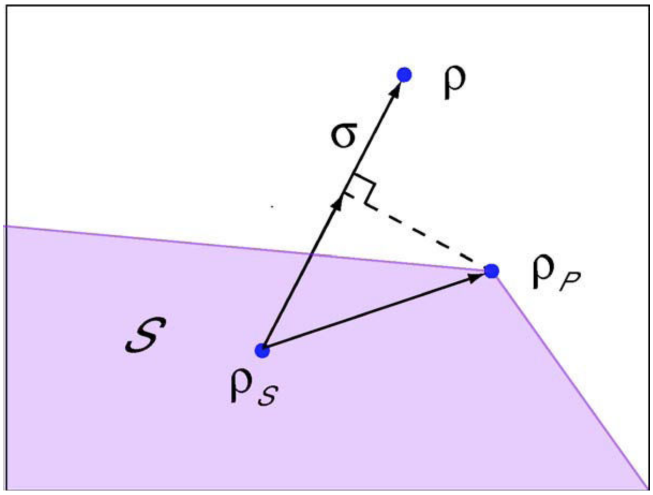

Figure 2 – Schematic illustration of searching for a better separable approximation to a given state ρ. The shaded area S represents the convex set of separable density matrices. ρ_s is the current separable approximation to ρ (at a given iteration step) and ρ_p is a pure product state that maximizes the scalar product $Tr\left[(\rho_p - \rho_s)(\rho - \rho_s\right]$, and is used to improve the separable approximation (ρ_s) at the next iteration step.

איור 2 – תיאור סכמטי של חיפוש קירוב פריק טוב יותר למצב נתון ρ. S הוא התחום הקמור של מטריצות הצפיפות הפריקות. ρ_s הוא הקירוב הפריק ל- ρ באיטרציה הנוכחית, ו- ρ_p (על קודקוד התחום S) הינו מצב פריק טהור. האלגוריתם מחפש ρ_p שיביא למקסימום את המכפלה הסקלרית: $Tr\left[(\rho_p - \rho_s)(\rho - \rho_s\right]$, ויתרום לקירוב טוב יותר (ρ_s) באיטרציה הבאה.

ככזה, ρ_p הינו מהצורה:

$$\rho_p = \left(|\phi\rangle\langle\phi|\right)\otimes\left(|\chi\rangle\langle\chi|\right) \quad (5)$$

ואלמנטי המטריצה שלו הינם:

$$\langle ij|\rho_p|kl\rangle = \phi_i\chi_j\phi_k^*\chi_l^* \quad (6)$$

כאשר $\sum_i |\phi_i|^2 = \sum_j |\chi_j|^2 = 1$

כעת, ניתן לפתח את הביטוי ל-s':

$$s' = \max_{\rho_p} Tr\left[\rho_p\sigma\right] = \max_{|\phi\rangle,|\chi\rangle}\left(\left(\langle\phi|\otimes\langle\chi|\right)\sigma\left(|\phi\rangle\otimes|\chi\rangle\right)\right) =$$

$$= \begin{cases} \max_{|\phi\rangle}\langle\phi|A_\chi|\phi\rangle, & A_\chi = \max_{|\chi\rangle}\langle\chi|\sigma|\chi\rangle \\ \max_{|\chi\rangle}\langle\chi|B_\phi|\chi\rangle, & B_\phi = \max_{|\phi\rangle}\langle\phi|\sigma|\phi\rangle \end{cases} \quad (7)$$

כלומר, הפכנו את הביטוי לבעיה כפולה של ערכים עצמיים (ערכו המקסימאלי של s' זהה לערכו המקסימלי של הערך העצמי μ):

$$\sum_k A_{ik}\phi_k = \mu\phi_i, \quad \sum_l B_{jl}\chi_l = \mu\chi_j \quad (8)$$

כאשר:

$$A_{ik} = \sum_{j,l}\chi_j^*\sigma_{ij:kl}\chi_l \quad B_{jl} = \sum_{i,k}\phi_i^*\sigma_{ij:kl}\phi_k \quad (9)$$

את הערך המקסימאלי של μ נמצא בתהליך איטרטיבי:

1) נתחיל מכך שנגריל ווקטור יחידה $|\chi\rangle$ כלשהו:

$$|\chi\rangle = \sum_j \chi_j|j\rangle_B \in \mathrm{H}_B \quad (10)$$

2) נחשב מתוך $|\chi\rangle$ את המטריצה ההרמיטית A.

3) כעת נחשב את ווקטור היחידה $|\phi\rangle$, שהוא וקטור עצמי של מטריצה A בעל הערך העצמי הגדול ביותר:

$$|\phi\rangle = \sum_i \phi_i|i\rangle_A \in \mathrm{H}_A \quad (11)$$

4) בעזרת $|\phi\rangle$ נחשב את המטריצה ההרמיטית B.

5) כעת נחשב וקטור יחידה חדש $|\chi\rangle$, שהוא הווקטור העצמי של מטריצה B בעל הערך העצמי הגדול ביותר.

6) נחזור באופן איטרטיבי על השלבים 2-5 עד להתכנסות של μ (s') לערכו המקסימאלי.

במהלך האיטרציות נקבל סדרה לא-יורדת של ערכי s' שמתכנסת לערך המקסימאלי. המקסימום במקרה זה הוא לפחות מקסימום לוקאלי, התלוי בבחירה האקראית של נקודת ההתחלה ($|\chi\rangle$).

כעת ניתן להרכיב את המטריצה ρ_p לפי נוסחה (5) ולחשב את ערכו של s לפי נוסחה (3).

אם מתקבל $s > 0$, משמע קיים מצב פריק ρ_s' (על הקו שבין ρ_s ל-ρ_p) הקרוב ל-ρ יותר מאשר ρ_s:

$$\rho_s' = (1-\lambda)\rho_s + \lambda\rho_p \quad (12)$$

כאשר $0 < \lambda \le 1$.

בפועל, אין טעם להגביל את עצמנו בחיפוש ρ_s' רק על הקו שבין ρ_s ל- ρ_p. נכון יותר יהיה לצרף את ρ_p לסדרת המצבים הפריקים ρ_{pk} שמצאנו כבר בשלבים הקודמים, ואז להביא למינימום את הביטוי הבא:

$$F = Tr\left(\rho - \sum_k \lambda_k \rho_{pk}\right)^2 \qquad (13)$$

ביטוי זה הינו פונקציה ריבועית של וקטור המקדמים λ ($\lambda_k \geq 0$) שמקיים: $\sum_k \lambda_k = 1$.

תוצאת האופטימיזציה הינה סדרת מקדמים λ_k המביאה את F לערכו המינימאלי.

כעת ניתן לבנות את ה- ρ_s' (ρ_s של השלב הבא):

$$\rho_s' = \sum_k \lambda_k \rho_{pk} \qquad (14)$$

כאשר מתוך כלל מצבי ρ_{pk} שיש לנו, נשמיט את אלה שמקדמי ה- λ_k שלהם מתאפסים (כלומר את אלה שאינם תורמים למינימיזציה של F).

חשוב להבהיר שכאן מסתיים רק שלב אחד בחיפוש אחר המצב הפריק הקרוב ביותר ל- ρ. התהליך חוזר על עצמו מספר פעמים, כאשר בכל שלב משתמשים ב- ρ_s האופטימלי שמצאנו בשלב הקודם.

תהליך איטרטיבי זה נמשך כל עוד המכפלה הסקלרית במשוואה (3) יוצאת חיובית ($s > 0$). קבלת תוצאה שלילית ($s \leq 0$) באחת האיטרציות, משמעותה שלא ניתן כבר למצוא מצב פריק טהור ρ_p שיתרום לקירוב טוב יותר (ρ_s) למטריצה הנתונה ρ. מכאן, שהקירוב שנמצא באיטרציה הקודמת, הוא **בהחלט** המצב הפריק הקרוב ביותר ל- ρ.

פרק 4 – שיטות העבודה:

להלן התיאור של התוכנית "separability" המיישמת את האלגוריתם שתואר בפרק הקודם. קוד התוכנית (בתוספת פונקציית עזר "ro_p_comp") מופיע בנספח א'.

4.1 – הגדרת נתונים התחלתיים:

1) קליטת מטריצה ρ המייצגת מצב של שתי תת-מערכות (למשל, שני קיוביטים, או קיוביט וקיוטריט, או שני קיוטריטים, וכו') מהמשתמש. זו תהיה המטריצה הנתונה שתיבדק במהלך התוכנית ובהמשך ייקבע אם היא מייצגת מצב פריק או שזור.
2) בדיקת חוקיות הקלט: נבדק אם המטריצה ρ חיובית והרמיטית. אם אינה, התוכנית מחזירה הודעת שגיאה למשתמש. כמו כן, התוכנית מוודאת ש-$Tr(\rho)=1$, ואם לא – מתבצע נרמול.
3) קביעת מטריצה ρ_s (המייצגת את המצב הפריק הקרוב ביותר למטריצה ρ באיטרציה נתונה) עבור האיטרציה הראשונה – נקבעת שרירותית להיות מטריצת יחידה במימדים זהים למימדי ρ (בסוף כל איטרציה נחשב ρ_s חדש הקרוב עוד יותר ל- ρ).
4) קביעת המשתנה iter_num המייצג את מספר האיטרציות המקסימאלי שתבוצענה בחיפוש אחר הקירוב הפריק ל- ρ.
5) הקצאת מערך של מטריצות פריקות טהורות – ro_p_storage (המספר המקסימאלי של מטריצות ρ_{pk} במערך זה יהיה – iter_num). תחילה, כל המטריצות במערך מוגדרות שרירותית כמטריצות אפס.

4.2 – תיאור הלולאה הראשית בתוכנית:

לאחר הגדרת הנתונים ההתחלתיים נפתחת לולאה שתחזור על עצמה עד iter_num פעמים והמהווה את לב התוכנית:

1) הגדרת מטריצה σ לפי משוואה (2) של הפרק הקודם.
2) הגרלת וקטור $\langle\chi|$ אקראי. למשל, עבור ρ שמייצג מצב של שני קיוביטים, $\langle\chi|$ הינו וקטור קומפלקסי דו-מימדי מהצורה:

$$\varphi=\begin{pmatrix}\cos(2\pi X)\cdot e^{i\cdot 2\pi Y}\\ \sin(2\pi X)\end{pmatrix} \quad (1)$$

(כאשר X ו-Y מוגרלים באופן בלתי תלוי וערכם נע בין 0 ל-1)

3) **ביצוע לולאה פנימית:** מתבצעת פנייה מחזורית לפונקציית עזר "ro_p_comp" שמקבלת וקטור $\langle\chi|$, מטריצה σ, ואת מימדי הבעיה. הפונקציה מחשבת את המטריצה A, מחשבת בעזרתה את הוקטור $\langle\phi|$, בונה את המטריצה B, ומחשבת מתוכה וקטור $\langle\chi|$ חדש. את הפניה הראשונה לפונקציה "ro_p_comp" מבצעים עם וקטור $\langle\chi|$ שהגרלנו, ואת שאר הפניות מבצעים עם וקטור $\langle\chi|$ שהתקבל בפלט של הפניה הקודמת לפונקציה זו. זאת עד להתכנסות לערך המקסימאלי של הערך העצמי μ. בסיום ההתכנסות, וביציאה מהלולאה, מתקבלים ערכי וקטורים $\langle\chi|$ ו-$\langle\phi|$ סופיים.
4) בניית מטריצת ρ_p, לפי משוואה (5) של הפרק הקודם, והוספתה למאגר המטריצות הפריקות הטהורות שהקצאנו בתחילת התוכנית - ro_p_storage.
5) חישוב המכפלה הסקלרית s לפי משוואה (3) של הפרק הקודם. כזכור, ה-ρ_p שמצאנו אמור להביא מכפלה סקלרית זו לערכה המקסימאלי.

6) בדיקה באם $s>0$. במידה ו- $s\le 0$, מתרחשת יציאה יזומה מהלולאה הראשית וקפיצה להמשך התוכנית.

7) **פתרון בעיית תכנות ריבועית (quadratic programming problem) למציאת קירוב טוב יותר ל- ρ :**

א. יש לנו מאגר של מטריצות ρ_p – ro_p_storage, המכיל את כל המטריצות ρ_{pk} שמצאנו באיטרציות שבוצעו עד כה (לרבות האיטרציה הנוכחית).

ב. עלינו למצוא את וקטור המקדמים λ שמביא למינימום את הביטוי :

$$F = Tr\left(\rho - \sum_k \lambda_k \rho_{pk}\right)^2 \qquad (2)$$

ג. זוהי בעיית תכנות ריבועית (quadratic programming problem) : בהינתן מספר סופי של מצבים פריקים טהורים ρ_{pk}, יש למצוא מתוכם קומבינציה קמורה הקרובה ביותר למצב נתון ρ. תוצאת האופטימיזציה הינה סדרת מקדמים λ_k המביאה את F לערכו המינימאלי.

ד. לפתרון הבעיה נשתמש בפונקציה פנימית של תוכנת MATLAB שנקראית "quadprog". תיאור הפונקציה ואופן השימוש בה מפורטים בנספח ב'.

ה. כעת, מתוך וקטור המקדמים λ שקיבלנו לאחר האופטימיזציה, נשמיט את אותם מקדמי λ_k ששואפים לאפס (כלומר, אלה שאינם תורמים למינימיזציה של F). באופן דומה, מתוך מאגר המטריצות ro_p_storage נשמיט את מטריצות ה- ρ_{pk} המתאימות, ונחשב את $\rho_s{}'$ – הקירוב החדש למטריצה ρ – על פי נוסחה (14) של הפרק הקודם.

8) חזרה לתחילת הלולאה הראשית וביצוע האיטרציה הבאה – חזרה על הסעיפים 1-7 עם ה- ρ_s החדש.

כאמור, ישנן שתי אפשרויות ליציאה מהלולאה הראשית : או קבלת $s\le 0$ באחת מהאיטרציות, או סיום הביצוע של כל האיטרציות שתוכננו (סך הכל – iter_num איטרציות).

4.3 – פלט התוכנית :

לאחר היציאה מהלולאה הראשית, מגיע שלב הפלט של התוכנית. תחילה מחושב המרחק בין המטריצה ρ הנתונה לבין הקירוב הפריק הכי טוב שלה (ה- ρ_s האחרון שחושב) :

$$distance = Tr\left(\rho - \rho_s\right)^2 \qquad (3)$$

אם המרחק שואף לאפס, משמע ש- $\rho = \rho_s$. מכאן שהמטריצה ρ הינה פריקה, והתוכנית מחזירה את פירוקה למצבים פריקים-טהורים (מוצג מספר המצבים הפריקים-טהורים שפורשים את ρ, וקטור המקדמים).

אם המרחק שונה מאפס, משמע שהמטריצה ρ מייצגת מצב שזור והתוכנית מחזירה את המטריצה הפריקה הקרובה אליה ביותר (ρ_s) ואת המרחק ביניהן.

בשני המקרים ניתן להציג בפלט, במידת הצורך, גם את המטריצות הפריקות הטהורות, מהן מורכבת ρ_s.

פרק 5 – תוצאות העבודה:

התוכנית המתוארת בפרק הקודם הורצה על מטריצות ρ בגודל 4X4 המייצגות מצב של שני קיוביטים. זאת, כדי לבדוק נכונות התוצאות מול קריטריון פרס שבמימדים אלה קובע חד-חד ערכית אם המטריצה היא שזורה או פריקה. כמו כן, עבור כל מטריצה ρ בוצעו כ-20 הרצות לווידוא עקביות התוצאות.

להלן דוגמאות להרצות שבוצעו עבור המצבים השונים.

5.1 – מצב שזור:

עבור המצב השזור הבא:

$$\rho = \frac{1}{2}\begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 1 & -1 & 0 \\ 0 & -1 & 1 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}$$

פלט התוכנית שהתקבל:

```
Insert a density matrix ro of size 4x4:
1/2*[0 0 0 0; 0 1 -1 0; 0 -1 1 0; 0 0 0 0]

The matrix ro is entangled. The closest separable state is:

ro_s_final =

   0.1661             -0.0008 - 0.0004i  -0.0009 - 0.0000i  -0.0002 - 0.0004i
  -0.0008 + 0.0004i   0.3338             -0.1662 + 0.0003i   0.0001 - 0.0004i
  -0.0009 + 0.0000i  -0.1662 - 0.0003i   0.3337              0.0001 + 0.0002i
  -0.0002 + 0.0004i   0.0001 + 0.0004i   0.0001 - 0.0002i   0.1664

This separable state is a mixing of 6 pure product states, with the weights

lambda_final =

    0.1735
    0.1701
    0.1904
    0.1344
    0.1762
    0.1554

The distance between ro and the closest separable state is:

min_merhak =

    0.3334
```

כלומר, התוכנית הצליחה לסווג נכון את המטריצה כמצב שזור.

כעת נוסיף הפרעה קטנה למטריצה זו:

$$\rho \to \frac{\rho + \varepsilon M^{\dagger} M}{Tr(\rho + \varepsilon M^{\dagger} M)}$$

כאשר M הינה מטריצה קומפלקסית אקראית (המטריצה $M^{\dagger}M$ בהכרח חיובית), ו- $\varepsilon = 0.01$

Insert a density matrix ro of size 4x4:
[0.0256 0.0074-0.0063i 0.0032+0.0129i 0.0224-0.0010i;
0.0074+0.0063i 0.5281 -0.4986+0.0144i -0.0010+0.0051i;
0.0032-0.0129i -0.4986-0.0144i 0.5155 -0.0012-0.0119i;
0.0224+0.0010i -0.0010-0.0051i -0.0012+0.0119i 0.0279]

The matrix ro is entangled. The closest separable state is:

ro_s_final =

0.1665	0.0041 - 0.0060i	0.0019 + 0.0103i	0.0203 - 0.0004i
0.0041 + 0.0060i	0.3370	-0.1686 + 0.0041i	-0.0037 + 0.0052i
0.0019 - 0.0103i	-0.1686 - 0.0041i	0.3258	-0.0026 - 0.0113i
0.0203 + 0.0004i	-0.0037 - 0.0052i	-0.0026 + 0.0113i	0.1708

This separable state is a mixing of 6 pure product states, with the weights

lambda_final =

0.2071
0.1745
0.1790
0.1622
0.1804
0.0968

The distance between ro and the closest separable state is:

min_merhak =

0.2469

כצפוי, גם לאחר הוספת ההפרעה המצב נותר שזור אבל המרחק בינו לבין המצב הפריק הקרוב ביותר הצטמצם.

5.2 – מצב פריק:

עבור המצב הפריק הבא:

$$\rho = \begin{pmatrix} 0.7 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0.3 \end{pmatrix}$$

פלט התוכנית שהתקבל:

Insert a density matrix ro of size 4x4:
[0.7 0 0 0; 0 0 0 0; 0 0 0 0; 0 0 0 0.3]

Ro represents a separable state that consists of 2 pure product-states, with the weights

lambda_final =

```
    0.7000
    0.3000
```

גם הפעם, התוכנית הצליחה לסווג נכון את המטריצה כמצב פריק.

כעת נוסיף הפרעה קטנה למטריצה זו :

Insert a density matrix ro of size 4x4:

```
 [0.7202 0.0196+0.0070i 0.0105-0.0037i 0.0066-0.0000i;
  0.0196-0.0070i 0.0274 0.0083-0.0126i 0.0024-0.0087i;
  0.0105+0.0037i 0.0083+0.0126i 0.0275 0.0109+0.0052i;
  0.0066+0.0000i 0.0024+0.0087i 0.0109-0.0052i 0.3240]
```

Ro represents a separable state that consists of 8 pure product-states, with the weights

lambda_final =

```
    0.6429
    0.2824
    0.0112
    0.0115
    0.0184
    0.0182
    0.0078
    0.0076
```

כפי שניתן לראות, גם לאחר הוספת ההפרעה המצב נותר פריק, אך כעת נדרשים הרבה יותר מצבים פריקים-טהורים (8 במקום 2) בשביל לפרוש אותו.

<u>5.3 – מצב פריק-טהור :</u>

כעת נריץ את התוכנית עבור מצב פריק-טהור :

$$\rho = \frac{1}{4}\begin{pmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \end{pmatrix}$$

הפלט שהתקבל :

Insert a density matrix ro of size 4x4:
1/4*ones(4,4)

Ro represents a separable state that consists of 1 pure product-states, with the weights

lambda_final =

```
     1
```

כצפוי, קיבלנו שזהו מצב פריק טהור (איבר יחיד בווקטור λ).

כעת נוסיף הפרעה קטנה למטריצה זו :

Insert a density matrix ro of size 4x4:
[0.2855 0.2776+0.0019i 0.2826-0.0013i 0.2814+0.0020i;
0.2776-0.0019i 0.2814 0.2797-0.0025i 0.2786-0.0013i;
0.2826+0.0013i 0.2797+0.0025i 0.2844 0.2827+0.0035i;
0.2814-0.0020i 0.2786+0.0013i 0.2827-0.0035i 0.2868]

Ro represents a separable state that consists of 6 pure product-states, with the weights

lambda_final =

0.9875
0.0036
0.0042
0.0022
0.0016
0.0009

כלומר, לאחר הוספת ההפרעה המצב נותר פריק, אך לא פריק-טהור כמו קודם, אלא מעורב (נפרש כעת ע"י 6 מצבים פריקים-טהורים). מכאן, שההפרעה "הזיזה" את המצב שהיה על גבול התחום של המטריצות הפריקות **לתוך** תחום זה.

כעת נוסיף הפרעה קטנה אחרת למצב המקורי :

Insert a density matrix ro of size 4x4:
[0.2767 0.2661-0.0091i 0.2765-0.0056i 0.2752-0.0038i;
0.2661+0.0091i 0.2671 0.2710+0.0056i 0.2694+0.0075i;
0.2765+0.0056i 0.2710-0.0056i 0.2827 0.2828-0.0001i;
0.2752+0.0038i 0.2694-0.0075i 0.2828+0.0001i 0.2894]

The matrix ro is entangled. The closest separable state is:

ro_s_final =

0.2463	0.2382 - 0.0063i	0.2469 - 0.0023i	0.2449 - 0.0039i
0.2382 + 0.0063i	0.2405	0.2422 + 0.0046i	0.2409 + 0.0039i
0.2469 + 0.0023i	0.2422 - 0.0046i	0.2566	0.2515 - 0.0021i
0.2449 + 0.0039i	0.2409 - 0.0039i	0.2515 + 0.0021i	0.2566

This separable state is a mixing of 6 pure product states, with the weights

lambda_final =

0.9713
0.0042
0.0021
0.0068
0.0075
0.0082

The distance between ro and the closest separable state is:

min_merhak =

8.5584e-005

הפעם, ההפרעה "הזיזה" את המצב (שהיה על גבול התחום של המטריצות הפריקות) **אל מחוץ** לתחום זה, ולכן המצב הפך לשזור.

5.4 – מצב מעורב תלוי פרמטר:

ניקח מצב שהוא צירוף לינארי של מצב פריק ומצב שזור:

$$\rho = q\underbrace{\begin{pmatrix}1&0&0&0\\0&1&0&0\\0&0&1&0\\0&0&0&1\end{pmatrix}}_{\text{separable}} + (1-q)\underbrace{\begin{pmatrix}1&0&0&1\\0&0&0&0\\0&0&0&0\\1&0&0&1\end{pmatrix}}_{\text{entangled}} \underset{\text{after normalization}}{\Longrightarrow} \rho = \frac{1}{1+2q}\begin{pmatrix}1&0&0&1-q\\0&q&0&0\\0&0&q&0\\1-q&0&0&1\end{pmatrix}$$

לאחר הפעלת קריטריון פרס, מתקבל שהמצב ρ פריק ($\rho^{PT} \geq 0$) עבור $q \geq \frac{1}{2}$.

נריץ את האלגוריתם עבור $q = 0.49$:

Insert a density matrix ro of size 4x4:
(1/(2+2*0.49))*[1 0 0 1-0.49; 0 0.49 0 0; 0 0 0.49 0; 1-0.49 0 0 1];

The matrix ro is entangled. The closest separable state is:

ro_s_final =

```
   0.3333             0.0015 - 0.0004i  -0.0016 - 0.0002i   0.1666 + 0.0007i
   0.0015 + 0.0004i   0.1662            -0.0001 + 0.0000i   0.0002 - 0.0006i
  -0.0016 + 0.0002i  -0.0001 - 0.0000i   0.1673            -0.0002 - 0.0005i
   0.1666 - 0.0007i   0.0002 + 0.0006i  -0.0002 + 0.0005i   0.3332
```

This separable state is a mixing of 7 pure product states, with the weights

lambda_final =

```
    0.1696
    0.1700
    0.1904
    0.1969
    0.2124
    0.0446
    0.0162
```

The distance between ro and the closest separable state is:

min_merhak =

7.5686e-005

כעת נריץ את האלגוריתם עבור $q = 0.51$:

Insert a density matrix ro of size 4x4:
(1/(2+2*0.51))*[1 0 0 1-0.51; 0 0.51 0 0; 0 0 0.51 0; 1-0.51 0 0 1];

Ro represents a separable state that consists of 11 pure product-states, with the weights

lambda_final =

    0.0413
    0.1287
    0.1680
    0.0072
    0.0016
    0.1872
    0.1932
    0.1994
    0.0619
    0.0069
    0.0046

כצפוי, קיבלנו שהמצב שזור עבור $q < \frac{1}{2}$ ופריק עבור $q \geq \frac{1}{2}$.

פרק 6 – דיון והסקת מסקנות:

להלן המסקנות המרכזיות לאחר ביצוע ההרצות שתוארו בפרק הקודם:

1) כשמריצים את האלגוריתם מספר פעמים עבור אותה מטריצה נתונה, התוצאות המתקבלות שונות לעיתים. הדבר בולט בעיקר במקרה של מטריצות שזורות, עבורן הפלט *distance* אינו זהה בכל הרצה.

הסיבה לשונות התוצאות הינה ההגרלה האקראית של וקטורים המבוצעת בעת חיפוש מט' פריקה טהורה ρ_p המביאה למקסימום את הביטוי $s'=Tr(\rho_p\sigma)$. כזכור, המקסימום המתקבל בסיום האיטרציות הינו לוקאלי (כי תלוי בבחירה האקראית של הוקטור הראשון) ולכן משפיע גם על התוצאה הסופית – על קביעת המטריצה הפריקה הקרובה ביותר למטריצה הנתונה (ρ) (ρ_s), ועל המרחק ביניהן.

כדי לוודא קבלת תוצאות גלובאליות ככל הניתן, בוצע שינוי בתוכנה הראשית, כך שמעתה כל האלגוריתם שתואר לעיל מורץ 10 פעמים והתוצאות המוחזרות בפלט שייכות לאותה הרצה שבה התקבל המרחק (*distance*) ה<u>מינימאלי</u> בין ρ ל- ρ_s. (דוגמאות ההרצה המוצגות בפרק הקודם הינן של התוכנית **לאחר ביצוע שינוי זה**).

2) עבור מטריצות בגודל 4X4 הלולאה המרכזית חוזרת על עצמה מקסימום כ-30 פעמים (עד לקבלת $s \le 0$ באחת האיטרציות). לכן, הפרמטר iter_num נקבע לערך של 100, כדי להבטיח התכנסות למטריצה הפריקה הקרובה ביותר (כלומר, קבלת $s \le 0$) **לפני** שריצת הלולאה מסתיימת.
להרצת התוכנית עבור מטריצות גדולות יותר, ככל הנראה יהיה צורך להגדיל את הפרמטר iter_num מעבר לערך זה.

3) התוצאות של כל ההרצות נבדקו גם באמצעות קריטריון פרס. באף מקרה לא התגלו סתירות בין 2 השיטות.

4) כש- ρ מתאר מצב פריק, הוא נפרש על-ידי מקסימום 11 מצבים פריקים-טהורים.
כש- ρ מתאר מצב שזור, המצב הפריק הקרוב אליו ביותר (ρ_s) נפרש ע"י מקסימום 8 מצבים פריקים-טהורים.

5) ריצת התוכנית נמשכת מקסימום 3 שניות.

לסיכום, התוכנית המממשת את האלגוריתם מספקת תוצאות נכונות עבור מצבים המתארים שני קיוביטים (מטריצות של 4x4).
חשוב לציין שהאלגוריתם אוניברסלי לכל מימד של מטריצה נתונה, והתועלת האמיתית שלו היא דווקא בחקירת מצבים ממימד גבוה (למשל מצב של קיוטריט+קיוטריט, ומעלה), כאשר קריטריון פרס כבר אינו מהווה תנאי מספיק לפריקות, ומופיעה תופעת ה- Bound Entanglement.
ניתן להתאים את קוד התוכנית, בקלות יחסית, כך שתתמוך גם במצבים ממימדים גבוהים אלו.

נספח א' – קוד התוכנית בשפת MATLAB:

תוכנית "separability":

```
clear all;
close all;
clc;

%Initial data definition:

ro=input('Insert a density matrix ro of size 4x4:\n');

[eivec,eival]=eig(ro);
if min(diag(eival))<0
   disp('Your matrix is an illegal density matrix - ro must be
        positive!');
   break
end

if (ro~=ro')
   disp('Your matrix is an illegal density matrix - ro must be
         hermitian!');
   break
end

ro=ro/(trace(ro));

n=size(ro,1);
iter_num=100; %number of iterations in the main loop

ro_p_final=[];
lambda=[];
lambda_final=[];
ro_s_final=[];
min_merhak=100;

for y=1:10  %the algorithm runs 10 times for a given matrix ro

    ro_s=eye(n)./n;
    ro_p_storage=zeros([n n iter_num]);

    for x=1:iter_num  %the main loop of the algorithm
        sig=ro-ro_s;

        %generating of random vector "kai" to start the iterative
        %search for pure product state
        X=rand;
        Y=rand;
        kai=[cos(2*pi*X)*exp(2*pi*i*Y); sin(2*pi*X)];

        %looking for a pure product state ro_p that maximizes the
        %scalar product s=trace((ro_p-ro_s)*sigma)
        for t=1:20
            [kai_new,phi_new,miu1,miu2]=ro_p_comp(kai,sig,n);
            kai=kai_new;
            if abs(miu2-miu1)<abs(miu2)/1000
                break;
            end;
```

```
        end;

        ro_p_storage(:,:,x)=kron(phi_new*phi_new',kai_new*kai_new');
        s=trace((ro_p_storage(:,:,x)-ro_s)*sig);
        if s<=0
            break;
        end;

        %Solving quadratic programming problem for best-weighted
        %sum of pure product states
        H=zeros(x,x);
        for a=1:x
            for b=1:x
                H(a,b)=2*abs(trace(ro_p_storage(:,:,a)*
                       ro_p_storage(:,:,b)));
            end;
        end;

        f=zeros(x,1);
        for a=1:x
            f(a,1)=-2*abs(trace(ro*ro_p_storage(:,:,a)));
        end;

        A=[];
        b=[];
        Aeq=ones(1,x);
        beq=ones(1,1);
        lb=zeros(x,1);
        ub=ones(x,1);

        lambda=quadprog(H,f,A,b,Aeq,beq,lb,ub);
        lambda_ind=find(lambda>0.0001);
        lambda=lambda(lambda_ind);
        ro_p_new=ro_p_storage(:,:,lambda_ind);

        %Estimation of new ro_s
        ro_s=zeros(n,n);
        for a=1:size(lambda_ind)
            ro_s=ro_s+lambda(a)*ro_p_new(:,:,a);
        end;

        if abs(trace((ro-ro_s)^2))<=0.000001
            break
        end;

    end; %end of the main loop

    merhak=abs(trace((ro-ro_s)^2));

    if (merhak<min_merhak)&&(sum(lambda~=0))
        lambda_final=lambda;
        ro_p_final=ro_p_new;
        ro_s_final=ro_s;
        min_merhak=merhak;
    end;

end; %end of the algorithm repetitions

if min_merhak>0.00001
```

```
    disp('The matrix ro is entangled. The closest separable
           state is:');
    ro_s_final
    disp(['This separable state is a mixing of'
          ,num2str(length(lambda_final)), ' pure product states,
          with the weights']);
    lambda_final
    disp('The distance between ro and the closest separable
          state is:');
    min_merhak
else
    disp(['Ro represents a separable state that consists of '
          ,num2str(length(lambda_final)), ' pure product-states,
          with the weights']);
    lambda_final
end
```

פונקציית עזר "ro_p_comp":

```
function [kai,phi,miu1,miu2]=ro_p_comp(kai,sig,n)

% Generating of Matrix A

A_her=zeros(sqrt(n),sqrt(n));

for i=1:size(A_her,2)
    for k=1:size(A_her,2)
        A_her(i,k)=0;
        for j=1:sqrt(n)
            for l=1:sqrt(n)
                temp_prod=(kai(j,1))'*sig((sqrt(n)*(i-1)+j),
                          (sqrt(n)*(k-1)+l))*kai(l,1);
                A_her(i,k)=A_her(i,k)+temp_prod;
            end;
        end;
    end;
end;

% Phi vector computation (eigenvector of A_her)
[V,eig_vals]=eig(A_her);
[miu1,ind]=max(diag(eig_vals));
phi=V(:,ind);

% Generating of Matrix B
B_her=zeros(sqrt(n),sqrt(n));

for j=1:sqrt(n)
    for l=1:sqrt(n)
        B_her(j,l)=0;
        for i=1:size(B_her,2)
            for k=1:size(B_her,2)
                temp_prod=(phi(i,1))'*sig((sqrt(n)*(i-1)+j),
                          (sqrt(n)*(k-1)+l))*phi(k,1);
                B_her(j,l)=B_her(j,l)+temp_prod;
            end;
        end;
    end;
end;

% New Kai vector computation  (eigenvector of B_her)
[V,eig_vals]=eig(B_her);
[miu2,ind]=max(diag(eig_vals));
kai=V(:,ind);
```

נספח ב' – פתרון בעיית תכנות ריבועית ב-MATLAB:

1) עלינו למצוא וקטור מקדמים λ שמביא למינימום את הביטוי:

$$F = Tr\left(\rho - \sum_k \lambda_k \rho_{pk}\right)^2 \qquad (1)$$

2) זוהי בעיית תכנות ריבועית, ולטובת פתרונה נשתמש בפונקציה פנימית של MATLAB שנקראית "quadprog".

3) הפונקציה מקבלת את הפרמטרים: $H, f, A, b, Aeq, beq, lb, ub$, ועושה מינימיזציה לביטוי:

$$\frac{1}{2}\lambda^T H \lambda + f^T \lambda \qquad (2)$$

תחת האילוצים הבאים:

$$A \cdot \lambda < b \qquad (3)$$

$$Aeq \cdot \lambda = beq \qquad (4)$$

$$lb < \lambda < ub \qquad (5)$$

הפונקציה מחזירה את וקטור העמודה λ שמביא את הביטוי (1) למינימום.

4) תחילה נרצה להביא את הביטוי (1) לתצורה מתאימה (2), בכדי שניתן יהיה להשתמש בפונקציית "quadprog":

$$F = Tr\left(\rho - \sum_k \lambda_k \rho_{pk}\right)^2 = Tr\left(\rho^2\right) - \\ -\sum_k \lambda_k \cdot Tr\left(\rho_{pk} \cdot \rho + \rho \cdot \rho_{pk}\right) + \sum_{k,l} \lambda_k \cdot Tr\left(\rho_{pk}\rho_{pl}\right) \cdot \lambda_l \qquad (6)$$

כיוון שהגורם הראשון – $Tr\left(\rho^2\right)$ – הינו קבוע (אינו תלוי ב-λ), ניתן להתעלם ממנו ולהביא למינימום רק את הביטוי:

$$F' = \sum_{k,l} \lambda_k \cdot Tr\left(\rho_{pk}\rho_{pl}\right) \cdot \lambda_l - \sum_k 2 \cdot Tr\left(\rho_{pk} \cdot \rho\right) \lambda_k \qquad (7)$$

5) כעת, ביטוי זה (7) זהה בתצורתו לביטוי (2), כאשר H הינה מטריצה מהצורה:

$$H_{k,l} = 2 \cdot Tr\left(\rho_{pk} \cdot \rho_{pl}\right) \qquad (8)$$

f הינו וקטור מהצורה:

$$f_k = -2 \cdot Tr\left(\rho_{pk} \cdot \rho\right) \qquad (9)$$

<u>והאילוצים שלנו:</u>

א. כל אחד מהמקדמים λ_k הוא בין 0 ל-1, לכן lb יהיה וקטור של אפסים ואילו ub יהיה וקטור של אחדות.

ב. האילוץ היחיד לגבי סכום המקדמים λ_k הינו: $\sum_k \lambda_k = 1$ ולכן נקבע את Aeq להיות וקטור שורה של אחדות, ו-$beq = 1$.

ג. האילוץ $A \cdot \lambda < b$ - לא קיים אצלנו, לכן נשאיר את המטריצה A ואת וקטור b ריקים.

6) כעת ניתן להפעיל את הפונקציה "quadprog", תוך הזנת כלל הפרמטרים הללו, בכדי לקבל בפלט את וקטור המקדמים λ שמביא למינימום את הביטוי (1).

רשימת מקורות:

[1] Michael A. Nielsen, Isaac L. Chuang. *Quantum Computation and Quantum Information*. Cambridge University Press (2000)

[2] Asher Peres, Phys. Rev. Lett. 77 (8) (1996) 1413–1415

[3] Charles H. Bennett, David P. DiVincenzo, Tal Mor, Peter W. Shor, John A. Smolin, and Barbara M. Terhal; Phys. Rev. Lett. 82, 5385–5388 (1999)

[4] D. Bruss and A. Peres, Phys. Rev. A 61, 030301(R) (2000)

[5] Jon Magne Leinaas, Jan Myrheim, Eirik Ovrum; Phys. Rev. A 74, 012313 (2006)

The main part of the work deals with a general numerical method for examing entanglement and separability. This method exploits the fact that each set of separable states is convex, and is based on an iterative scheme to find the closest separable state (and the distance to it) for any chosen density matrix (test state), with no limitations of this matrix's dimensions. There are two separate mathematical sub-problems that have to be solved numerically in this scheme. The first problem is to find iteratively the pure product state maximizing a scalar product. The second problem is the so-called quadratic programming problem: given a finite number of pure product states, to find the convex combination of these which is closest to the given state.

The nature of this work is the implementation of the algorithm in a MATLAB program and its execution on matrices representing different two-qubit states. The program takes a 4x4 matrix and determines whether this matrix is entangled or separable. If the matrix is separable, the program outputs the pure separable states of which the original matrix is composed (these are the extremal points of the convex set of separable density matrices). If the matrix is entangled, the program outputs the closest separable matrix and the Hilbert-Schmidt distance between them.

Further chapters of this work describe the implementation of the iterative algorithm and the examples of different runs made (for an entangled state, a separable state, a pure separable state, and for a parameter dependent state).

As part of this work's summary, conclusions are made regarding the different runs of the program, including comparisons of the outputs of the computer program against results of the execution of the Peres Criterion on these states (in these dimensions separable states can be fully identified by use of the Peres criterion).

In conclusion, the program implementing the algorithm classifies correctly matrices that represent two-qubit states for entanglement and separability. As a universal algorithm for a given matrix of any dimension, one can adjust the program (by a simple update of the code) to support the studying of the composite states of higher dimensions (e.g. two-qutrit states), where Peres criterion is only a necessary condition (but not sufficient) for separability.

Abstract:

Entanglement is a phenomenon in quantum mechanics in which quantum states of two (or more) systems are always described in relation one to the other, even if the systems are physically remote from each other. For example, when a measurement takes place on one of the systems, it is immediately reflected in the other.

Such a correlation cannot be described in terms of classical physics probability, but rather by a wave function that describes the joint quantum state of all the systems.

The entanglement plays a key role in many applications of quantum computation and quantum information, such as teleportation and cryptography.

In this work we examined several existing methods for identifying entanglement and for defining entangled states.

This work begins with a definition of a density matrix, of its characteristics (Hermiticity, positivity, normalization), and a definition of an entangled state. Immediately thereafter the Peres Criterion for Separability (the opposite of Entanglement) is defined. This is a simple algebraic criterion, which can be executed on a matrix that describes a quantum state consisting of two subsystems (e.g., two qubits). The criterion states that every separable density matrix is mapped into a positive semi-definite matrix by partial transposition. The criterion provides a necessary condition for separability, and for the special case of 2X2 system as well as for a 2X3 system (i.e., a state of two qubits, or a state of qubit and qutrit), the Peres criterion is both necessary and sufficient for a density matrix to be separable.

Later, a proof is shown that for a pure state, the Peres criterion for separability is equivalent to the Schmidt criterion for separability. The proof is achieved by executing the Peres criterion on a Schmidt decomposed state.

In higher dimensions the Peres criterion is not sufficient to identify the separable states. In other words, there exist entangled states that remain positive after a partial transposition.

Bound-Entanglement states are then shown – these are entangled states of several higher dimensional systems (e.g., two qutrits and above). The density matrices of these states, though entangled, remain positive under the partial transposition condition of Peres. One way to construct a wide variety of such states is by using UPB (Unextendible Product Basis). An Unextendible Product Basis for a multipartite system is an incomplete orthogonal product basis whose complementary subspace contains no product state. The uniform mixed state over the subspace complementary to any UPB is a Bound Entangled state.

List of Figures:

1	An example of forming UPB by pentagonal pyramid vectors	7
2	Schematic illustration of searching for a better separable approximation to a given state	10

Contents:

Abstract 1
List of symbols and abbreviations 2
Chapter 1 – Introduction 3
1.1 – Density matrix 3
1.2 – Entangled state definition 3
1.3 – Peres criterion for separability in quantum states 3
1.4 – Executing the Peres criterion on a Schmidt decomposed state 4
1.5 – Bound entangled states 6
1.5.1 – Definition of Bound-Entangled state using UPB 6
1.5.2 – An example of constructing a Bound-Entangled state using UPB ...7
Chapter 2 – Purpose of the final paper 9
Chapter 3 – Description 10
Chapter 4 – Methods 13
4.1 – Initial data definition 13
4.2 – Description of the main loop of the program 13
4.3 – Programs output 14
Chapter 5 – Results 15
5.1 – Entangled state 15
5.2 – Separable state 16
5.3 – Pure separable state 17
5.4 – Parameter dependant mixed state 19
Chapter 6 – Discussion and conclusions 21
Appendix A – Code of the MATLAB program 22
Main program - "Separability" 22
Auxiliary function - "ro_p_comp" 25
Appendix B – Quadratic programming problem solution in MATLAB 26
References 27

The final paper was done under the supervision of Prof. Joseph E. Avron in the Department of Physics.

Entanglement in Quantum States

Final Paper

Submitted in Partial Fulfillment of the Requirements for the Degree
of Master of Science in Physics

Lev Kvietny

Submitted to the Senate of the Technion –
The Israel Institute of Technology

Sivan, 5771 Haifa June 2011

Entanglement in Quantum States

Lev Kvietny