Математички факултет Универзитета у Београду

Конструкције Адамарових матрица

Мастер рад

Сенад Ибраимоски

Чланови комисије:
проф. др. Миодраг Живковић - ментор
проф. др. Предраг Јаничић
проф. др. Филип Марић

Београд, Новембар 2015

University of Belgrade, School of Mathematics

Construction Methods for Hadamard Matrices

Master thesis

Senad Ibraimoski

Commission members:
prof. phD. Miodrag Zivkovic - mentor
prof. phD. Predrag Janicic
prof. phD. Filip Maric

Belgrade, November 2015

Садржај

Листа слика

Све слике су софтверски генерисане од стране аутора.

Имплементације алгоритама

Псеудокодови конструкција

Захвалница

Желео бих да се захвалим свим члановима комисије а посебно ментору проф. др. Миодрагу Живковићу на указаној помоћи, саветима и смерницама током израде овог рада. Захвалност дугујем и мојој породици на безусловној подршци током свих ових година.

Глава 1

Увод

Адамарова[1] матрица је квадратна матрица са елементима из скупа $\{-1, 1\}$, чије су сваке две различите врсте ортогоналне. Према Адамаровој хипотези Адамарова матрица реда n се може конструисати за $n = 1, 2$ и за свако n дељиво са четири. Ова хипотеза је један од најпознатијих нерешених проблема математике.
У овом раду приказују се неке од важнијих метода конструкције Адамарових матрица. Уводи се појам еквивалентности и излажу се неопходни математички појмови и тврђења. Приказују се програмске реализације наведених конструкција.

Адамарове матрице се примењују и изучавају како у науци тако и у различитим индустријским гранама. Изразито су корисне приликом процесуирања сигнала различитих природа у телекомуникацијама, кодирању, комуникационим протоколима, криптографији и статистици. Акценат овог рада је искључиво на описивању различитих конструкција Адаровиx матрица у циљу генерисања што већег броја матрица различитих редова.

[1] Jacques Salomon Hadamard (1865-1963) - Француски математичар

Глава 2

Адамарове матрице

За сваку Адамарову матрицу потребно је описати конструкцију помоћу које је она добијена. Конструкције се често добијају коришћењем знања из различитих математичких области. У овој глави наводе се основне опште ознаке, појмови и тврђења и основне особине Адамарових матрица које се користе у конструкцијама које следе.

2.1 Опште ознаке, појмови и тврђења

Означићемо са:

- $\vec{1}$ низ (врсту) који се састоји искључиво од јединица.
- $A_n[a(i,j)]$ или само A квадратну матрицу димензије n.
- A^τ транспоновану матрицу матрице A.
- J_n матрицу чији су сви елементи јединице.
- I_n јединичну матрицу димензије n.
- $\langle x, y\rangle$ операцију скаларног производа између x и y.

Дефиниција 2.1.1.

1. *За матрицу A каже се да је симетрична ако је $A = A^\tau$.*
2. *За матрицу A каже се да је косо симетрична ако је $-A = A^\tau$.*

Дефиниција 2.1.2. *Траг матрице предстаља збир елемената на главној дијагонали.*

$$\mathrm{tr}(A) = a_{11} + a_{22} + \cdots + a_{nn} = \sum_{i=1}^{n} a_{ii}$$

Дефиниција 2.1.3. *Матрица $A_n[a(i,j)]_{0\le i,j<n}$ реда n је циклична ако је:*

$$a(i,j) = a(0, j - i \bmod n)$$

.

Дефиниција 2.1.4. *Кронекеров производ две матрице $A_{n\times m}$ и $B_{p\times q}$ је матрица $C_{np\times mq}$*

$$C = \mathbf{A}\otimes\mathbf{B} = \begin{bmatrix} a_{11}\mathbf{B} & \cdots & a_{1n}\mathbf{B} \\ \vdots & \ddots & \vdots \\ a_{m1}\mathbf{B} & \cdots & a_{mn}\mathbf{B}\end{bmatrix}$$

односно,

$$C = \mathbf{A} \otimes \mathbf{B} = \begin{bmatrix} a_{11}b_{11} & a_{11}b_{12} & \cdots & a_{11}b_{1q} & \cdots & \cdots & a_{1n}b_{11} & a_{1n}b_{12} & \cdots & a_{1n}b_{1q} \\ a_{11}b_{21} & a_{11}b_{22} & \cdots & a_{11}b_{2q} & \cdots & \cdots & a_{1n}b_{21} & a_{1n}b_{22} & \cdots & a_{1n}b_{2q} \\ \vdots & \vdots & \ddots & \vdots & & & \vdots & \vdots & \ddots & \vdots \\ a_{11}b_{p1} & a_{11}b_{p2} & \cdots & a_{11}b_{pq} & \cdots & \cdots & a_{1n}b_{p1} & a_{1n}b_{p2} & \cdots & a_{1n}b_{pq} \\ \vdots & \vdots & & \vdots & \ddots & & \vdots & \vdots & & \vdots \\ \vdots & \vdots & & \vdots & & \ddots & \vdots & \vdots & & \vdots \\ a_{m1}b_{11} & a_{m1}b_{12} & \cdots & a_{m1}b_{1q} & \cdots & \cdots & a_{mn}b_{11} & a_{mn}b_{12} & \cdots & a_{mn}b_{1q} \\ a_{m1}b_{21} & a_{m1}b_{22} & \cdots & a_{m1}b_{2q} & \cdots & \cdots & a_{mn}b_{21} & a_{mn}b_{22} & \cdots & a_{mn}b_{2q} \\ \vdots & \vdots & \ddots & \vdots & & & \vdots & \vdots & \ddots & \vdots \\ a_{m1}b_{p1} & a_{m1}b_{p2} & \cdots & a_{m1}b_{pq} & \cdots & \cdots & a_{mn}b_{p1} & a_{mn}b_{p2} & \cdots & a_{mn}b_{pq} \end{bmatrix}$$

У следећој леми наводе се неке од основних особина операције Кронекеровог производа које се користе у даљем раду приликом доказивања одређених тврђења.

Лема 2.1.1. *За матрице* $A_{m\times n}, B_{r\times s}, C_{n\times p}, D_{s\times t}$ *важи:*

1. $(A \otimes B)(C \otimes D) = AC \otimes BD$.
2. $(A \otimes B)^\tau = A^\tau \otimes B^\tau$.

Доказ.

1.

$$\begin{aligned} (A \otimes B)(C \otimes D) &= \begin{bmatrix} a_{11}B & \cdots & a_{1n}B \\ \vdots & \ddots & \vdots \\ a_{m1}B & \cdots & a_{mn}B \end{bmatrix} \begin{bmatrix} c_{11}D & \cdots & c_{1p}D \\ \vdots & \ddots & \vdots \\ c_{n1}D & \cdots & c_{np}D \end{bmatrix} \\ &= \begin{bmatrix} \sum_{k=1}^n a_{1k}c_{k1}BD & \cdots & \sum_{k=1}^n a_{1k}c_{kp}BD \\ \vdots & \ddots & \vdots \\ \sum_{k=1}^n a_{mk}c_{k1}BD & \cdots & \sum_{k=1}^n a_{mk}c_{kp}BD \end{bmatrix} = AC \otimes BD. \end{aligned}$$

2.

$$(A \otimes B)^\tau = \begin{bmatrix} a_{11}B & \cdots & a_{1n}B \\ \vdots & \ddots & \vdots \\ a_{m1}B & \cdots & a_{mn}B \end{bmatrix}^\tau = \begin{bmatrix} a_{11}B^\tau & \cdots & a_{m1}B^\tau \\ \vdots & \ddots & \vdots \\ a_{1n}B^\tau & \cdots & a_{mn}B^\tau \end{bmatrix} = A^\tau \otimes B^\tau.$$

Дефиниција 2.1.5. *Ако је* $k = i_{n-1}2^{n-1} + i_{n-2}2^{n-2} + \cdots + i_1 2 + i_0, i_j \in \{0, 1\}$ *бинарни запис броја* k*, онда се броју* k *придружује вектор* $(i_0, i_1, \ldots, i_{n-1})$ *који се без опасности од забуне означава истом ознаком* k*.*

Дефиниција 2.1.6. *Коначно поље* F *је скуп од најмање два елемента, са две операције* $+$ *и* $*$*, за који важе следеће аксиоме:*

1. *Скуп* F *формира Абелову (комутативну) групу за операцију* $+$ *и неутралом групе 0.*
2. *Скуп* $F^* = F - \{0\}$ *је Абелова група за операцију* $*$ *и неутралом групе 1.*
3. *Важи дистрибутивни закон: За свако* $a, b, c \in F$ *важи* $(a+b)*c = (a*c)+(a*b)$.

Дефиниција 2.1.7. *Коначно поље F_p је коначно поље које се састоји од скупа $R_p = \{0, 1, \ldots, p-1\}$ где је p прост број, са операцијама $*_{\text{mod } p}$ и $+_{\text{mod } p}$, односно након операције сабирања и множења резултат се своди по модулу простог броја p.*

Специјално за $p = 2$, поље F_2 представља најмање коначно поље чији елементи узимају вредности из скупа $\{0, 1\}$ где су 0,1 редом адитивни и мултипликативни неутрали поља

$$x + 0 = x, x * 1 = x$$

и операцијама сабирања и множења по модулу 2.

2.2 Основни појмови и тврђења о Адамаровим матрицама

Дефиниција 2.2.1. *Адамарова матрица реда n је $n \times n$ матрица H са елементима из $\{-1, 1\}$ за коју важи:*

$$HH^{\tau} = nI_n \tag{2.1}$$

односно производ сваке две различите врсте или колоне је 0.

У следећој леми наводе се основне особине Адамарових матрица.

Лема 2.2.1. *Нека је H Адамарова матрица реда n и нека је $M = \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}$.*
Тачна су следећа тврђења:

1. *Ако је матрица H' добијена заменом неке врсте или колоне матрице H, са истом врстом или колоном помноженом са -1, тада је H' Адамарова матрица.*

2. *$-H$ матрица је такође Адамарова матрица.*

3. *H^{τ} матрица је такође Адамарова матрица.*

4. *Ако је H' Адамарова матрица реда m, Кронекеров производ $H' \otimes H$ је Адамарова матрица реда $m \cdot n$.*

5. *За свако $t \geq 1$, $(\otimes_{i=1}^{t} M) \otimes H$ је Адамарова матрица реда $2^t n$*

6. *Ако је матрица H' добијена заменом било која две врсте или колоне матрице H, матрица H' је Адамарова матрица.*

7. *Ако је H_n Адамарова матрица реда $n > 2$, n је дељиво са 4.*

Доказ.

1. *претпоставимо да се i-та врста множи са -1. Скаларни производ i-те вресте са самом собом је:*

$$\sum_{k=1}^{n} (-H(i,k))^2 = \sum_{k=1}^{n} H(i,k)^2 = n$$

док је скаларни производ i-те врсте са j-том колоном:

$$\sum_{k=1}^{n} -H(i,k)H(k,j) = -\sum_{k=1}^{n} H(i,k)H(k,j) = 0$$

Слично се показује и за колоне матрица.

2. *Непосредно се проверава применом 2.2.1.1 на све врсте матрице H.*

3. *Нека је H Адамарова матрица.* $(H^t)\cdot((H^t)^t) = (H^t \cdot H)^t = (nI_n)^t = nI_n$.

4. *Нека су $H^{'}, H$ Адамарове матрице редом реда m, n. На основу леме 2.1.1 следи:*

   $$(H^{'} \otimes H)(H' \otimes H)^{\tau} = (H^{'} \otimes H)(H^{'\tau} \otimes H^{\tau}) = H'H^{'\tau} \otimes HH^{\tau} = mI_m \otimes nI_n = mnI_{mn}.$$

5. *Пошто је M Адамарова матрица, доказ следи применом тачке 4 ове теореме коначан број пута.*

6. *Замена било које две врсте или колоне не утиче на ортогоналност, односно производ свака два различита реда или колоне остаје 0, без обзира на замене.*

7. *За сваки елемент прве врсте који је негативан, потребно је помножити колону којој тај елемент припада са -1. На тај начин добија се матрица чији су сви елементи прве врсте јединице.*
   Нека постоји i колона које почињу са $(1,1,1,\dots)$, j колона које почињу са $(1,-1,1,\dots)$, k колона које почињу са $(1,1,-1,\dots)$ и l колона које почињу са $(1,-1,-1,\dots)$.
   С обзиром на то да су врсте ортогоналне добијамо збирове:

   $$i + j + k + l = n$$

   $$i - j + k - l = 0$$

   $$i + j - k - l = 0$$

   $$i - j - k + l = 0$$

   Решавањем система једначина добија се да је $i = j = k = l$, из чега непосредно следи да је $n = 4i$, односно n је дељиво са 4.

Неке од наведених особина се користе за генерисање Адамарових матрица виших редова почев од једноставнијих.

Дефиниција 2.2.2. *За Адамарове матрице H и $H^{'}$ се каже да су еквивалентне ако се $H^{'}$ може добити коначним низом замена врста, односно колона и променом знака врста, односно колона матрице H.*

Непосредно се може верификовати да наведена дефиниција представља релацију еквиваленције.

Дефиниција 2.2.3. *Нормализована Адамарова матрица је Адамарова матрица у којој се почетна врста и колона састоје само од +1.*

Прву конструкцију Адамарових матрица објавио је Силвестер [1] 1867. године.

Теорема 2.2.1. *Нека је H_n Адамарова матрица реда n онда је*

$$H_{2n} = \begin{bmatrix} H_n & H_n \\ H_n & -H_n \end{bmatrix} \tag{2.2}$$

такође Адамарова матрица реда $2n$ [2].

[1] James Joseph Sylvester (1814-1897) - Енглески математичар

Доказ. *Тврђење је непосредна последица леме 2.2.1.4.*

Претпоставимо да је H Адамарова матрица реда n. На основу леме 2.2.1.3, H^τ је такође Адамарова матрица и за обе матрице важи из полазне дефиниције Адамарових матрица да је $HH^\tau = nI_n$.

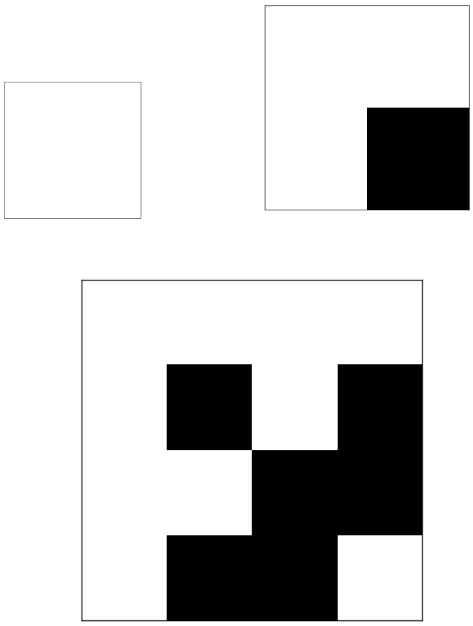
$$HH^\tau = \begin{bmatrix} H & H \\ H & -H \end{bmatrix} \begin{bmatrix} H^\tau & H^\tau \\ H^\tau & -H'^\tau \end{bmatrix} = \begin{bmatrix} nI_n + nI_n & nI_n - nI_n \\ nI_n - nI_n & nI_n + nI_n \end{bmatrix} = 2n \begin{bmatrix} I_n & 0 \\ 0 & I_n \end{bmatrix} = 2nI_{2n}. \tag{2.3}$$

Овом конструкцијом се могу добити матрице реда $2^k, k = 1, 2, \ldots, n$.

Пример 2.2.1. *Адамарове матрице реда 1,2 односно 4 су:*

$$\begin{bmatrix} 1 \end{bmatrix}, \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}, \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & -1 & 1 & -1 \\ 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 \end{bmatrix}$$

Графички се Адамарове матрице могу представити тако што негативни елементи матрице представе црним а позитивни елементи белим квадратом. Графичка представа матрица из примера 2.2.1 приказана је следећој слици:



Слика 2.1: Адамарове матрице из примера 2.2.1.

Глава 3

Конструкције

У овој глави описане су најзначајније конструкције Адамарових матрица које генеришу велики број различитих редова. За сваку конструкцију излаже се неопходна математичка теорија и примери. За већину конструкција доказана је њихова коректност док тамо где је доказ изостао наводи се референца. Свака конструкција је праћена псеудокодом и графичком репрезентацијом Адамарове матрице коју конструкција производи.

3.1 Силвестерова конструкција

Силвестерова (Sylvester) конструкција генерише фамилију матрица реда $2^t : t \geq 1$. Као што је речено, фамилија Адамарових матрица се једноставно конструише тако што се итеративно примењује Кронекеров производ.

$$H_1 = \begin{bmatrix}1\end{bmatrix}, H_2 = \begin{bmatrix}1 & 1\\ 1 & -1\end{bmatrix}, H_{2^k} = \begin{bmatrix}H_{2^{k-1}} & H_{2^{k-1}}\\ H_{2^{k-1}} & -H_{2^{k-1}}\end{bmatrix} = H_2 \otimes H_{2^{k-1}}.$$

Један од начина погодан за рачунарско генерисање Силвестерових матрица се ослања на аритметику у коначном пољу F_2.

Теорема 3.1.1. *Ако се елементи H_{2^t} индексирају природним бројевима $0 \leq i \leq 2^t - 1$ где се i, j представе у бинарној форми користећи дефиницију (2.1.5), при чему i истовремено представља и вектор од t бинарних цифара броја i, онда се може конструисати Силверстерова матрица реда 2^t користећи:*

$$H_{2^t} = [(-1)^{\langle i,j\rangle}]_{0\leq i,j\leq 2^t-1}. \tag{3.1}$$

Доказ. *Нека су u, v врсте матрице H индексиране векторима x_a, x_b добијене на основу дефиниције (2.1.6):*

$$u = ((-1)^{\langle x_1,x_a\rangle}, \cdots, (-1)^{\langle x_n,x_a\rangle}), v = ((-1)^{\langle x_1,x_b\rangle}, \cdots, (-1)^{\langle x_n,x_b\rangle}).$$

Непосредно се показује да је

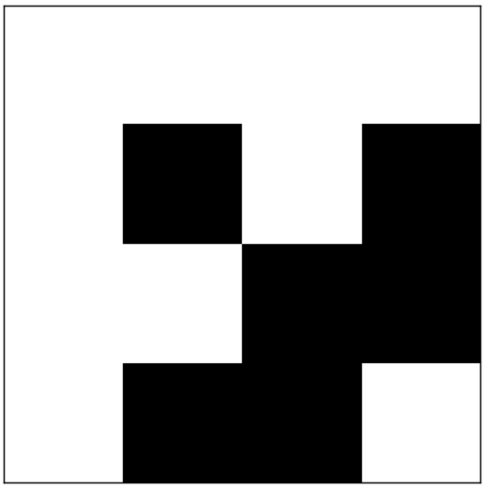
$$\langle x_a, x_b\rangle = \sum_{i=1}^{n}(-1)^{\langle x_i,x_a\rangle+\langle x_i,x_b\rangle} = \sum_{i=1}^{n}(-1)^{\langle x_i,x_a+x_b\rangle}.$$

Ако је $a = b$ следи да је скаларни проивод 0 и наведена сума је n. У супротном $x_a + x_b$ је не-нула вредности и низ $\langle x_1, x_a + x_b\rangle, \ldots, \langle x_n, x_a + x_b\rangle$ садржи подједнаки број нула и јединица на основу чега је $\langle x_a, x_b\rangle = 0$ што задовољава основну дефиницију Адамарових матрица.

Пример 3.1.1. *Адамарова матрица реда* $n = 4$

$$H_4 = \begin{bmatrix} (-1)^{\langle 00,00\rangle} & (-1)^{\langle 00,01\rangle} & (-1)^{\langle 00,10\rangle} & (-1)^{\langle 00,11\rangle} \\ (-1)^{\langle 01,00\rangle} & (-1)^{\langle 01,01\rangle} & (-1)^{\langle 01,10\rangle} & (-1)^{\langle 01,11\rangle} \\ (-1)^{\langle 10,00\rangle} & (-1)^{\langle 10,01\rangle} & (-1)^{\langle 10,10\rangle} & (-1)^{\langle 10,11\rangle} \\ (-1)^{\langle 11,00\rangle} & (-1)^{\langle 11,01\rangle} & (-1)^{\langle 11,10\rangle} & (-1)^{\langle 11,11\rangle} \end{bmatrix}$$

Графичка репрезентација претходно добијене матрице је приказана на слици 3.1.



Слика 3.1: Адамарова матрица реда 4 добијена Силвестеровом конструкцијом.

Следи алгоритам у псеудокоду који описује приказану конструкцију.

Алгоритам 1 Силвестерова конструкција

procedure SILVESTER(n)
    ▷ Улаз: n - димензија Адамарове матрице коју генеришемо.
    $H[h(i,j)] = 0$
    for $i \leftarrow 1$ to n **do**
        for $j \leftarrow 1$ to n **do**
            b_i = binary(i) ▷ Бинарна репрезентација броја i
            b_j = binary(j) ▷ Бинарна репрезентација броја j
            $H[h(i,j)] \leftarrow (-1)^{\langle b_i, b_j\rangle}$
        end for
    end for
    ▷ Излаз: H - Адамарова матрица добијена Силвестеровом конструкцијом.
    return H
end procedure

3.2 Палејева конструкција

Палеј (Paley[1]) је 1933. описао конструкцију која је поред Силвестерове конструкције једна од најзначајнијих. Конструкција се заснива на квадратним остацима и Лежан-

[1] Raymond Edward Alan Christopher Paley (1907-1933) - Енглески математичар

дровом симболу чије су дефиниције описане у наставку.

Дефиниција 3.2.1. *Нека је $a \in \boldsymbol{Z_n}$. Ако конгруенција $x^2 \equiv a \bmod n$ има решење онда се за а каже да је квадратни остатак по модулу n. Ако решење не постоји онда се каже да а није квадратни остатак по модулу n.*

Дефиниција 3.2.2. *Нека је a природан број и нека је p прост. Лежандров симбол, у ознаци $\left(\frac{a}{p}\right)$ се дефише на следећи начин:*

$$\left(\frac{a}{p}\right) = \begin{cases} 1, & \text{ако је } a \text{ квадратни остатак} \\ -1, & \text{ако } a \text{ није квадратни остатак} \\ 0, & \text{ако је } a \equiv 0 \pmod p \end{cases} \tag{3.2}$$

Следи дефиниција Јакобшталове матрице:

Дефиниција 3.2.3. *Нека је $\boldsymbol{F}$ коначно поље реда p, где је p степен простог броја чији су елемененти $i, 0 \leq i \leq p-1$ и нека је $\chi(a) = \left(\frac{a}{p}\right)$ квадратни остатак над пољем. Нека је матрица Q, Јакобшталова матрица дефинисана изразом:*

$$Q = [\chi(i,j)]_{0\leq i,j \leq p-1} = \begin{bmatrix} \chi(0-0) & \chi(0-1) & \dots & \chi(0-j_{p-1}) \\ \chi(1-0) & \chi(1-1) & \dots & \chi(1-j_{p-1}) \\ \vdots & \vdots & & \vdots \\ \chi(i_{p-1}-0) & \chi(i_{p-1}-1) & \dots & \chi(i_{p-1}-j_{p-1}) \end{bmatrix} \tag{3.3}$$

Наредна лема описује основне особине Јакобшталове матрице.

Лема 3.2.1. *Нека је $Q = [\chi(i,j)]_{0\leq i,j\leq p-1}$ Јакобшталова матрица.*

1. *Ако је $p \equiv 3 \bmod 4$, матрица Q је косо симетрична.*
2. *Ако је $p \equiv 1 \bmod 4$, матрица Q је симетрична.*
3. $QJ_{p-1} = J_{p-1}Q = 0$
4. $QQ^\tau = (p-1)I_{p-1} - J_{p-1}$
5. *За матрицу $S = \begin{bmatrix} 0 & -\vec{1} \\ \vec{1}^\tau & Q \end{bmatrix}$ важи $S^2 = sI_{s+1}$.*

Доказ наведене леме може се видети у књизи [4].

Теорема 3.2.1. *Палејева конструкција Адамарових матрица.*

- *Ако је $p \equiv 3 \bmod 4$, онда је*

$$P_p = \begin{bmatrix} 1 & -\vec{1} \\ \vec{1}^\tau & Q + I_p \end{bmatrix} \tag{3.4}$$

  Адамарова матрица реда $p+1$ позната као Палеј I матрица.

- *Ако је $p \equiv 1 \bmod 4$, онда је*

$$P_p = \begin{bmatrix} S + I_{p+1} & S - I_{p+1} \\ S - I_{p+1} & -S - I_{p+1} \end{bmatrix} \tag{3.5}$$

  Адамарова матрица реда $2(p+1)$ позната као Палеј II матрица.

Доказ. *Доказ се заснива на леми 3.2.1:*

- *Нека је* $p \equiv 3 \pmod 4$

$$HH^\tau = \begin{bmatrix} 1 & -\vec{1} \\ \vec{1}^\tau & Q + I_p \end{bmatrix} \begin{bmatrix} 1 & \vec{1} \\ -\vec{1}^\tau & Q^\tau + I_p \end{bmatrix} = \begin{bmatrix} p+1 & 0 \\ 0 & (p+1)I_p \end{bmatrix} = (p+1)I_{p+1}.$$

- *Нека је* $p \equiv 1 \pmod 4$

$$HH^\tau = \begin{bmatrix} S + I_{p+1} & S - I_{p+1} \\ S - I_{p+1} & -S - I_{p+1} \end{bmatrix} \begin{bmatrix} S + I_{p+1} & S - I_{p+1} \\ S - I_{p+1} & -S - I_{p+1} \end{bmatrix} \begin{bmatrix} (2p+2)I_{p+1} & 0 \\ 0 & (2p+2)I_{p+1} \end{bmatrix} = 2(p+1)I_{2(p+}$$

Пример 3.2.1. *Палеј I конструкција* $p = 7$

Јакобшталова матрица:

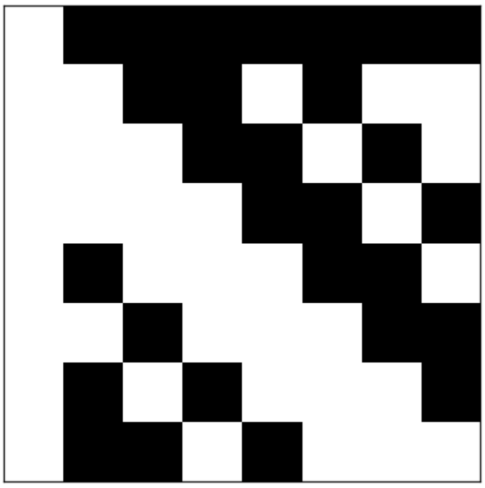
$$Q = \begin{bmatrix} \chi(0-0) & \chi(0-1) & \chi(0-2) & \chi(0-3) & \chi(0-4) & \chi(0-5) & \chi(0-6) \\ \chi(1-0) & \chi(1-1) & \chi(1-2) & \chi(1-3) & \chi(1-4) & \chi(1-5) & \chi(1-6) \\ \chi(2-0) & \chi(2-1) & \chi(2-2) & \chi(2-3) & \chi(2-4) & \chi(2-5) & \chi(2-6) \\ \chi(3-0) & \chi(3-1) & \chi(3-2) & \chi(3-3) & \chi(3-4) & \chi(3-5) & \chi(3-6) \\ \chi(4-0) & \chi(4-1) & \chi(4-2) & \chi(4-3) & \chi(4-4) & \chi(4-5) & \chi(4-6) \\ \chi(5-0) & \chi(5-1) & \chi(5-2) & \chi(5-3) & \chi(5-4) & \chi(5-5) & \chi(5-6) \\ \chi(6-0) & \chi(6-1) & \chi(6-2) & \chi(6-3) & \chi(6-4) & \chi(6-5) & \chi(6-6) \end{bmatrix} =$$

$$= \begin{bmatrix} 0 & -1 & -1 & 1 & -1 & 1 & 1 \\ 1 & 0 & -1 & -1 & 1 & -1 & 1 \\ 1 & 1 & 0 & -1 & -1 & 1 & -1 \\ -1 & 1 & 1 & 0 & -1 & -1 & 1 \\ 1 & -1 & 1 & 1 & 0 & -1 & -1 \\ -1 & 1 & -1 & 1 & 1 & 0 & -1 \\ -1 & -1 & 1 & -1 & 1 & 1 & 0 \end{bmatrix}$$

Након замене добијене матрице у (3.4) добија се Адамарова матрица реда 8.

$$H_8 = \begin{bmatrix} 1 & -1 & -1 & -1 & -1 & -1 & -1 & -1 \\ 1 & 1 & -1 & -1 & 1 & -1 & 1 & 1 \\ 1 & 1 & 1 & -1 & -1 & 1 & -1 & 1 \\ 1 & 1 & 1 & 1 & -1 & -1 & 1 & -1 \\ 1 & -1 & 1 & 1 & 1 & -1 & -1 & 1 \\ 1 & 1 & -1 & 1 & 1 & 1 & -1 & -1 \\ 1 & -1 & 1 & -1 & 1 & 1 & 1 & -1 \\ 1 & -1 & -1 & 1 & -1 & 1 & 1 & 1 \end{bmatrix}$$

Графичка репрезентација добијене матрице је приказана на слици 3.2



Слика 3.2: Адамарова матрица реда 8 добијена Палеј I конструкцијом.

Пример 3.2.2. *Paley II конструкција* $p = 5$

Јакобшталова матрица Q и матрица S:

$$Q = \begin{bmatrix} \chi(0-0) & \chi(0-1) & \chi(0-2) & \chi(0-3) & \chi(0-4) \\ \chi(1-0) & \chi(1-1) & \chi(1-2) & \chi(1-3) & \chi(1-4) \\ \chi(2-0) & \chi(2-1) & \chi(2-2) & \chi(2-3) & \chi(2-4) \\ \chi(3-0) & \chi(3-1) & \chi(3-2) & \chi(3-3) & \chi(3-4) \\ \chi(4-0) & \chi(4-1) & \chi(4-2) & \chi(4-3) & \chi(4-4) \end{bmatrix} = \begin{bmatrix} 0 & 1 & -1 & -1 & 1 \\ 1 & 0 & 1 & -1 & -1 \\ -1 & 1 & 0 & 1 & -1 \\ -1 & -1 & 1 & 0 & 1 \\ 1 & -1 & -1 & 1 & 0 \end{bmatrix}$$

$$S = \begin{bmatrix} 0 & 1 & 1 & 1 & 1 & 1 \\ 1 & 0 & 1 & -1 & -1 & 1 \\ 1 & 1 & 0 & 1 & -1 & -1 \\ 1 & -1 & 1 & 0 & 1 & -1 \\ 1 & -1 & -1 & 1 & 0 & 1 \\ 1 & 1 & -1 & -1 & 1 & 0 \end{bmatrix}$$

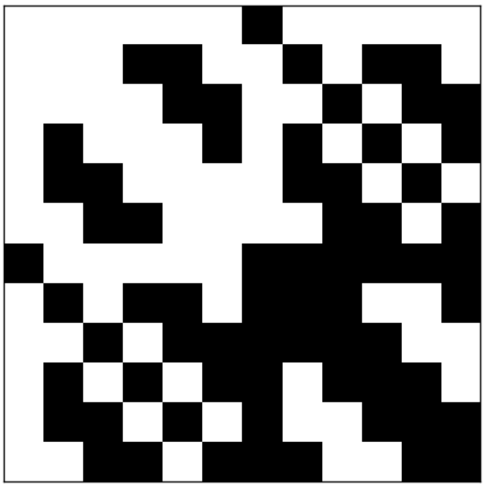
Након замене матрица Q и S у (3.5) добија се матрица реда $n = 12$:

$$H = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & -1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & -1 & -1 & 1 & 1 & -1 & 1 & -1 & -1 & 1 \\ 1 & 1 & 1 & 1 & -1 & -1 & 1 & 1 & -1 & 1 & -1 & -1 \\ 1 & -1 & 1 & 1 & 1 & -1 & 1 & -1 & 1 & -1 & 1 & -1 \\ 1 & -1 & -1 & 1 & 1 & 1 & 1 & -1 & -1 & 1 & -1 & 1 \\ 1 & 1 & -1 & -1 & 1 & 1 & 1 & 1 & -1 & -1 & 1 & -1 \\ -1 & 1 & 1 & 1 & 1 & 1 & -1 & -1 & -1 & -1 & -1 & -1 \\ 1 & -1 & 1 & -1 & -1 & 1 & -1 & -1 & -1 & 1 & 1 & -1 \\ 1 & 1 & -1 & 1 & -1 & -1 & -1 & -1 & -1 & -1 & 1 & 1 \\ 1 & -1 & 1 & -1 & 1 & -1 & -1 & 1 & -1 & -1 & -1 & 1 \\ 1 & -1 & -1 & 1 & -1 & 1 & -1 & 1 & 1 & -1 & -1 & -1 \\ 1 & 1 & -1 & -1 & 1 & -1 & -1 & -1 & 1 & 1 & -1 & -1 \end{bmatrix}$$

Графичка репрезентација добијене матрице је приказана на слици 3.3
Следећи псеудокод прегледно описује Палејеву конструкцију.

Алгоритам 2 Палејова конструкција

procedure PALEY(q)
    ▷ Са симболом **0** означавамо матрицу која се састоји само од 0.
    ▷ Улаз: q - Прост број
    $P \leftarrow \mathbf{0}$
    $Q \leftarrow \mathbf{0}$
    $S \leftarrow \begin{bmatrix} 0 & -\vec{1} \\ \vec{1}^{\tau} & Q \end{bmatrix}$
    ▷ Конструкција Јакобшталове матрице
    for $i \leftarrow 1$ to q **do**
        for $j \leftarrow 1$ to q **do**
            $Q[q(i,j)] \leftarrow \chi(i-j)$
        end for
    end for
    if $q \equiv 3 \bmod 4$ **then**
        ▷ Генерисање Палеј I матрице
        $P \leftarrow \begin{bmatrix} 1 & -\vec{1} \\ \vec{1}^{\tau} & Q + I_p \end{bmatrix}$
    else if $q \equiv 1 \bmod 4$ **then**
        ▷ Генерисање Палеј II матрице
        $P \leftarrow \begin{bmatrix} S + I_{p+1} & S - I_{p+1} \\ S - I_{p+1} & -S - I_{p+1} \end{bmatrix}$
        ▷ Излаз: P - Адамарове матрица добијена Палејовом конструкцијом.
        return P
    else
        return null
    end if
end procedure



Слика 3.3: Адамарова матрица реда 12 добијена Палеј II конструкцијом.

3.3 Вилијамсонова конструкција

Конструкција се заснива на проналажењу матрица које задовољавају одређени скуп услова. Вилијамсон (Williamson[2]) је објавио ову конструкцију 1944. године [5], међутим њена моћ постаје доступна са развојем рачунарства, паралелних и дистрибуираних алгоритама који врше потрагу за матрицама на основу следеће леме:

Лема 3.3.1. *Нека су A, B, C, D симетричне матрице чији су елементи -1 или +1 реда n за које важи:*

$$XY^\tau = YX^\tau, X, Y \in \{A, B, C, D\}. \tag{3.6}$$

и

$$AA^\tau + BB^\tau + CC^\tau + DD^\tau = 4nI_n \tag{3.7}$$

Тада је матрица

$$\begin{bmatrix} A & B & C & D \\ -B & A & -D & C \\ -C & D & A & -B \\ -D & -C & B & A \end{bmatrix} \tag{3.8}$$

Адамарова матрица реда $4n$.

Доказ. *Нека су A,B,C,D матрице задате као у претходној леми. На основу једна-*

[2]John Williamson (1901-1949) - Енглески математичар

кости 3.7 и дефиниције 2.2.1 важи:

$$HH^\tau = \begin{bmatrix} A & B & C & D \\ -B & A & -D & C \\ -C & D & A & -B \\ -D & -C & B & A \end{bmatrix} \begin{bmatrix} A & B & C & D \\ -B & A & -D & C \\ -C & D & A & -B \\ -D & -C & B & A \end{bmatrix}^\tau =$$

$$= (A^2 + B^2 + C^2 + D^2) \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} =$$

$$= (AA^\tau + BB^\tau + CC^\tau + DD^\tau) \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} = 4nI_n. \tag{3.9}$$

Важан део конструкције је проналажење матрица из претходне леме. У раду се користе већ познати мањи редови који се могу преузети из веб архиве Х. Коуковинуса [6]. Наведена архива садржи познате матрице реда $n \leq 63$ и такође су представљене у додатку Б овог рада. Поменуте матрице поседују разне корисне особине и јако су важне с обзиром на то да се могу користити и у другим конструкцијама.

Пример 3.3.1. *Адамарова матрица реда 12 добијена Вилијамсовом методом.*

У следећем примеру користе се системи матрица A,B,C,D реда $n = 3$ које задовољавају услове леме 3.3.1 и такође су део претходно наведене архиве.

$$A = \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}, B = C = D = \begin{bmatrix} 1 & -1 & -1 \\ -1 & 1 & -1 \\ -1 & -1 & 1 \end{bmatrix}$$

С обзиром на то да су матрице B, C, D једнаке и да су све симетричне $A = A^\tau, B = B^\tau, C = C^\tau, D = D^\tau$, довољно је проверити да важи:

$$AA + BB + CC + DD = AA + 3BB = 12I, AB = BA$$

Непосредно се проверава да је:

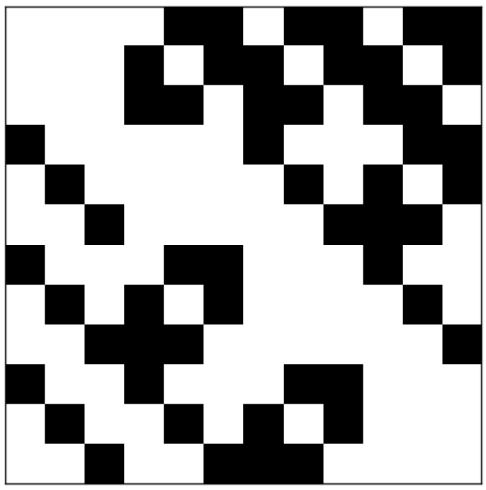
$$AB = \begin{bmatrix} -1 & -1 & -1 \\ -1 & -1 & -1 \\ -1 & -1 & -1 \end{bmatrix} = BA$$

$$AA = \begin{bmatrix} 3 & 3 & 3 \\ 3 & 3 & 3 \\ 3 & 3 & 3 \end{bmatrix}, BB = \begin{bmatrix} 3 & -1 & -1 \\ -1 & 3 & -1 \\ -1 & -1 & 3 \end{bmatrix}, AA + 3BB = \begin{bmatrix} 12 & 0 & 0 \\ 0 & 12 & 0 \\ 0 & 0 & 12 \end{bmatrix} = 12I$$

Заменом матрица A, B, C, D у (3.8) добија се Адамарова матрица H_{12}

$$H_{12} = \begin{bmatrix} 1 & 1 & 1 & 1 & -1 & -1 & 1 & -1 & -1 & 1 & -1 & -1 \\ 1 & 1 & 1 & -1 & 1 & -1 & -1 & 1 & -1 & -1 & 1 & -1 \\ 1 & 1 & 1 & -1 & -1 & 1 & -1 & -1 & 1 & -1 & -1 & 1 \\ -1 & 1 & 1 & 1 & 1 & 1 & -1 & 1 & 1 & 1 & -1 & -1 \\ 1 & -1 & 1 & 1 & 1 & 1 & 1 & -1 & 1 & -1 & 1 & -1 \\ 1 & 1 & -1 & 1 & 1 & 1 & 1 & 1 & -1 & -1 & -1 & 1 \\ -1 & 1 & 1 & 1 & -1 & -1 & 1 & 1 & 1 & -1 & 1 & 1 \\ 1 & -1 & 1 & -1 & 1 & -1 & 1 & 1 & 1 & 1 & -1 & 1 \\ 1 & 1 & -1 & -1 & -1 & 1 & 1 & 1 & 1 & 1 & 1 & -1 \\ -1 & 1 & 1 & -1 & 1 & 1 & 1 & -1 & -1 & 1 & 1 & 1 \\ 1 & -1 & 1 & 1 & -1 & 1 & -1 & 1 & -1 & 1 & 1 & 1 \\ 1 & 1 & -1 & 1 & 1 & -1 & -1 & -1 & 1 & 1 & 1 & 1 \end{bmatrix}$$

Графичка репрезентација добијене матрице је приказана на слици 3.4



Слика 3.4: Адамарова матрица реда 12 добијена Вилијамсовом конструкцијом.

Псеудокод Вилијамсове конструкције:

Алгоритам 3 Вилијамсова конструкција

procedure WILIAMSON(A,B,C,D)

  ▷ Улаз: A,B,C,D матрице које задовољавају (3.6)

  $H \leftarrow \begin{bmatrix} A & B & C & D \\ -B & A & -D & C \\ -C & D & A & -B \\ -D & -C & B & A \end{bmatrix}$

  if $AA^\tau + BB^\tau + CC^\tau + DD^\tau = 4nI_n$ **then**

    ▷ Излаз: H - Адамарове матрица добијена Вилијамсовом конструкцијом

    return H

  else

    return null

  end if

end procedure

3.4 Баумерт/Хол конструкција

Теорема 3.4.1. *Нека су A,B,C,D симетричне матрице чији су сви елементи -1 ili +1 реда n које међусобно комутирају. Баумерт/Хол (Baumert[3] /Hall[4]) Адамарова матрица реда H реда $12n$ је матрица следећег облика.*

$$H = \begin{bmatrix} A & A & A & B & -B & C & -C & -D & B & C & -D & -D \\ A & -A & B & -A & -B & -D & D & -C & -B & -D & -C & -C \\ A & -B & -A & A & -D & D & -B & B & -C & -D & C & -C \\ B & A & -A & -A & D & D & D & C & C & -B & -B & -C \\ B & -D & D & D & A & A & A & C & -C & B & -C & B \\ B & C & -D & D & A & -A & C & -A & -D & C & B & -B \\ D & -C & B & -B & A & -C & -A & A & B & C & D & -D \\ -C & -D & -C & -D & C & A & -A & -A & -D & B & -B & -B \\ D & -C & -B & -B & -B & C & C & -D & A & A & A & D \\ -D & -B & C & C & C & B & B & -D & A & -A & D & -A \\ C & -B & -C & C & D & -B & -D & -B & A & -D & -A & A \\ -C & -D & -D & C & -C & -B & B & B & D & A & -A & -A \end{bmatrix}, \tag{3.10}$$

тако да за матрицеважи: $AA^\tau + BB^\tau + CC^\tau + DD^\tau = tr(H)$.

Доказ. *Непосредно се проверава да се множењем Баумерт/Хол Адамарове матрице са њеним транспонатом добија матрица чији су сви елементи осим елемената на главној дијагонали 0.*

$$H = (3A^2 + 3B^2 + 3C^2 + 3D^2)I_n$$

С обзиром на то да су матрице A,B,C,D симетричне, слично као што је показано у доказу 3.3.1 важи:

$$3A^2 + 3B^2 + 3C^2 + 3D^2 = 3 \cdot (AA^\tau + BB^\tau + CC^\tau + DD^\tau) = 3 \cdot 4nI_n = 12nI_n.$$

Пример 3.4.1. *Адамарова матрица реда 36 добијена методом.*

[3]Leonard Daniel Baumert - Амерички математичар

[4]Marshall Hall, Jr. (1910 – 1990) - Амерички математичар

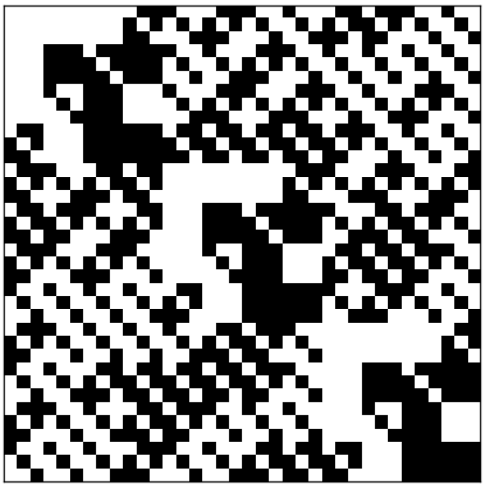
Конструкција матрице реда $n = 36$ *постиже се коришћењем матрица* A, B, C, D *из примера (3.3.1). С обзиром на то да важи* $B = C = D$ *услов комутативности је испуњен ако је* $AB = BA$*, што је такође показано у (3.3.1). Траг матрице је* $AA^\tau + BB^\tau + CC^\tau + DD^\tau = 36$*, и*

$$HH^\tau = 36I_{36}$$

Сви услови теореме су испуњени па се заменом матрица A, B, C, D *у (3.10) добија се Адамарова матрица реда* $n = 36$*.*

$$H_{36} = \left[\begin{array}{cccccccccccccccccccccccccccccccccccc}
1&1&1&1&1&1&1&1&1&1&-1&-1&-1&1&1&1&-1&-1&-1&1&1&-1&1&1&1&-1&-1&1&-1&-1&-1&1&1&-1&1&1\\
1&1&1&1&1&1&1&1&1&-1&1&-1&1&-1&1&-1&1&-1&1&-1&1&1&-1&1&-1&1&-1&-1&1&-1&1&-1&1&1&-1&1\\
1&1&1&1&1&1&1&1&1&-1&-1&1&1&1&-1&-1&-1&1&1&1&-1&1&1&-1&-1&-1&1&-1&-1&1&1&1&-1&1&1&-1\\
1&1&1&-1&-1&-1&1&-1&-1&-1&-1&-1&-1&1&1&-1&1&1&1&-1&-1&-1&1&1&-1&1&1&-1&1&1&-1&1&1&-1&1&1\\
1&1&1&-1&-1&-1&-1&1&-1&-1&-1&-1&1&-1&1&1&-1&1&-1&1&-1&1&-1&1&1&-1&1&1&-1&1&1&-1&1&1&-1&1\\
1&1&1&-1&-1&-1&-1&-1&1&-1&-1&-1&1&1&-1&1&1&-1&-1&-1&1&1&1&-1&1&1&-1&1&1&-1&1&1&-1&1&1&-1\\
1&1&1&-1&1&1&-1&-1&-1&1&1&1&-1&1&1&1&-1&-1&-1&1&1&1&-1&-1&-1&1&1&-1&1&1&1&-1&-1&-1&1&1\\
1&1&1&1&-1&1&-1&-1&-1&1&1&1&1&-1&1&-1&1&-1&1&-1&1&-1&1&-1&1&-1&1&1&-1&1&-1&1&-1&1&-1&1\\
1&1&1&1&1&-1&-1&-1&-1&1&1&1&1&1&-1&-1&-1&1&1&1&-1&-1&-1&1&1&1&-1&1&1&-1&-1&-1&1&1&1&-1\\
1&-1&-1&1&1&1&-1&-1&-1&-1&-1&-1&1&-1&-1&1&-1&-1&1&-1&-1&1&-1&-1&1&-1&-1&-1&1&1&-1&1&1&-1&1&1\\
-1&1&-1&1&1&1&-1&-1&-1&-1&-1&-1&-1&1&-1&-1&1&-1&-1&1&-1&-1&1&-1&-1&1&-1&1&-1&1&1&-1&1&1&-1&1\\
-1&-1&1&1&1&1&-1&-1&-1&-1&-1&-1&-1&-1&1&-1&-1&1&-1&-1&1&-1&-1&1&-1&-1&1&1&1&-1&1&1&-1&1&1&-1\\
1&-1&-1&-1&1&1&1&-1&-1&1&-1&-1&1&1&1&1&1&1&1&1&1&1&-1&-1&-1&1&1&1&-1&-1&-1&1&1&1&-1&-1\\
-1&1&-1&1&-1&1&-1&1&-1&-1&1&-1&1&1&1&1&1&1&1&1&1&-1&1&-1&1&-1&1&-1&1&-1&1&-1&1&-1&1&-1\\
-1&-1&1&1&1&-1&-1&-1&1&-1&-1&1&1&1&1&1&1&1&1&1&1&-1&-1&1&1&1&-1&-1&-1&1&1&1&-1&-1&-1&1\\
1&-1&-1&1&-1&-1&-1&1&1&1&-1&-1&1&1&1&-1&-1&-1&1&-1&-1&-1&-1&-1&-1&1&1&1&-1&-1&1&-1&-1&-1&1&1\\
-1&1&-1&-1&1&-1&1&-1&1&-1&1&-1&1&1&1&-1&-1&-1&-1&1&-1&-1&-1&-1&1&-1&1&-1&1&-1&-1&1&-1&1&-1&1\\
-1&-1&1&-1&-1&1&1&1&-1&-1&-1&1&1&1&1&-1&-1&-1&-1&-1&1&-1&-1&-1&1&1&-1&-1&-1&1&-1&-1&1&1&1&-1\\
1&-1&-1&-1&1&1&1&-1&-1&-1&1&1&1&1&1&-1&1&1&-1&-1&-1&1&1&1&1&-1&-1&1&-1&-1&1&-1&-1&-1&1&1\\
-1&1&-1&1&-1&1&-1&1&-1&1&-1&1&1&1&1&1&-1&1&-1&-1&-1&1&1&1&-1&1&-1&-1&1&-1&-1&1&-1&1&-1&1\\
-1&-1&1&1&1&-1&-1&-1&1&1&1&-1&1&1&1&1&1&-1&-1&-1&-1&1&1&1&-1&-1&1&-1&-1&1&-1&-1&1&1&1&-1\\
-1&1&1&-1&1&1&-1&1&1&-1&1&1&1&-1&-1&1&1&1&-1&-1&-1&-1&-1&-1&-1&1&1&1&-1&-1&-1&1&1&-1&1&1\\
1&-1&1&1&-1&1&1&-1&1&1&-1&1&-1&1&-1&1&1&1&-1&-1&-1&-1&-1&-1&1&-1&1&-1&1&-1&1&-1&1&1&-1&1\\
1&1&-1&1&1&-1&1&1&-1&1&1&-1&-1&-1&1&1&1&1&-1&-1&-1&-1&-1&-1&1&1&-1&-1&-1&1&1&1&-1&1&1&-1\\
1&-1&-1&-1&1&1&-1&1&1&-1&1&1&-1&1&1&1&-1&-1&1&-1&-1&-1&1&1&1&1&1&1&1&1&1&1&1&1&-1&-1\\
-1&1&-1&1&-1&1&1&-1&1&1&-1&1&1&-1&1&-1&1&-1&-1&1&-1&1&-1&1&1&1&1&1&1&1&1&1&1&-1&1&-1\\
-1&-1&1&1&1&-1&1&1&-1&1&1&-1&1&1&-1&-1&-1&1&-1&-1&1&1&1&-1&1&1&1&1&1&1&1&1&1&-1&-1&1\\
-1&1&1&-1&1&1&1&-1&-1&1&-1&-1&1&-1&-1&1&-1&-1&1&-1&-1&-1&1&1&1&1&1&-1&-1&-1&1&-1&-1&-1&-1&-1\\
1&-1&1&1&-1&1&-1&1&-1&-1&1&-1&-1&1&-1&-1&1&-1&-1&1&-1&1&-1&1&1&1&1&-1&-1&-1&-1&1&-1&-1&-1&-1\\
1&1&-1&1&1&-1&-1&-1&1&-1&-1&1&-1&-1&1&-1&-1&1&-1&-1&1&1&1&-1&1&1&1&-1&-1&-1&-1&-1&1&-1&-1&-1\\
1&-1&-1&-1&1&1&-1&1&1&1&-1&-1&1&-1&-1&-1&1&1&-1&1&1&-1&1&1&1&1&1&-1&1&1&-1&-1&-1&1&1&1\\
-1&1&-1&1&-1&1&1&-1&1&-1&1&-1&-1&1&-1&1&-1&1&1&-1&1&1&-1&1&1&1&1&1&-1&1&-1&-1&-1&1&1&1\\
-1&-1&1&1&1&-1&1&1&-1&-1&-1&1&-1&-1&1&1&1&-1&1&1&-1&1&1&-1&1&1&1&1&1&-1&-1&-1&-1&1&1&1\\
-1&1&1&-1&1&1&-1&1&1&1&-1&-1&-1&1&1&-1&1&1&1&-1&-1&1&-1&-1&1&-1&-1&1&1&1&-1&-1&-1&-1&-1&-1\\
1&-1&1&1&-1&1&1&-1&1&-1&1&-1&1&-1&1&1&-1&1&-1&1&-1&-1&1&-1&-1&1&-1&1&1&1&-1&-1&-1&-1&-1&-1\\
1&1&-1&1&1&-1&1&1&-1&-1&-1&1&1&1&-1&1&1&-1&-1&-1&1&-1&-1&1&-1&-1&1&1&1&1&-1&-1&-1&-1&-1&-1
\end{array}\right]$$

Графичка репрезентација добијене матрице је приказана на слици 3.5.



Слика 3.5: Адамарова матрица реда 36 добијена Баумерт/Хол конструкцијом.

Псеудокод који описује наведену конструкцију:

Алгоритам 4 Баумерт/Xol конструкција

procedure BAUMERTHALL(A,B,C,D)

▷ Улаз: A,B,C,D матрице које међусобно комутирају

$$H \leftarrow \begin{bmatrix} A & A & A & B & -B & C & -C & -D & B & C & -D & -D \\ A & -A & B & -A & -B & -D & D & -C & -B & -D & -C & -C \\ A & -B & -A & A & -D & D & -B & B & -C & -D & C & -C \\ B & A & -A & -A & D & D & D & C & C & -B & -B & -C \\ B & -D & D & D & A & A & A & C & -C & B & -C & B \\ B & C & -D & D & A & -A & C & -A & -D & C & B & -B \\ D & -C & B & -B & A & -C & -A & A & B & C & D & -D \\ -C & -D & -C & -D & C & A & -A & -A & -D & B & -B & -B \\ D & -C & -B & -B & -B & C & C & -D & A & A & A & D \\ -D & -B & C & C & C & B & B & -D & A & -A & D & -A \\ C & -B & -C & C & D & -B & -D & -B & A & -D & -A & A \\ -C & -D & -D & C & -C & -B & B & B & D & A & -A & -A \end{bmatrix}$$

▷ Провера услова леме за матрице A,B,C,D

$trace \leftarrow tr(AA^\tau + BB^\tau + CC^\tau + DD^\tau)$

$H_t \leftarrow HH^\tau$

if $H_t(h_t(i,i)) =$ trace, за свако i **then**

  return H

else

  return null

end if

end procedure

3.5 Итова конструкција

Слично као Вилијамсова, Итова[5] конструцкија се заснива на проналажењу матрица које испуњавају одређени скуп услова. Матрице добијене Итовом методом се такође у литератури срећу као матрице типа Q.

Лема 3.5.1. *Ако постоје симетричне цикличне матрице A, B, C, D реда n чији су сви елементи -1 или +1 за које важи:*

$$AB^\tau + CD^\tau = BA^\tau + DC^\tau \tag{3.11}$$

тада је матрица:

$$H = \begin{bmatrix} A & B & C & D \\ B & -A & D & -C \\ C^\tau & -D^\tau & -A^\tau & B^\tau \\ D^\tau & C^\tau & -B^\tau & -A^\tau \end{bmatrix} \tag{3.12}$$

Адамарова матрица reda $4n$.

Доказ. *Претпоставимо да је матрица H Адамарова матрица. Непосредно се проверава да је*

$$HH^\tau = (A^2 + B^2 + C^2 + D^2)I_n$$

С обзиром на то да су матрице A,B,C,D симетричне важи:

$$HH^\tau = (A^2 + B^2 + C^2 + D^2)I_n = (AA^\tau + BB^\tau + CC^\tau + DD^\tau) = 4nI_n.$$

[5]Noboru Ito - Јапански Математичар

У публикованом раду[7] Ито користи матрице добијене Paley II методом. Наводи се једноставнији пример где се користе A, B, C, D матрице из (3.4).

Пример 3.5.1. *Адамарова матрица реда 12 добијена Итовом методом.*

Нека је:

$$A = \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}, B = C = D = \begin{bmatrix} 1 & -1 & -1 \\ -1 & 1 & -1 \\ -1 & -1 & 1 \end{bmatrix}$$

С обзиром на то да су матрице A, B, C, D цикличне и симетричне услови леме се своде на:

$$AB^\tau = BA^\tau$$

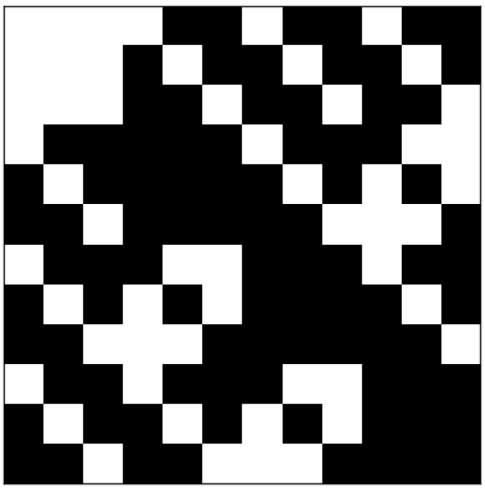
Лако се показује да:

$$AB^\tau = \begin{bmatrix} -1 & -1 & -1 \\ -1 & -1 & -1 \\ -1 & -1 & -1 \end{bmatrix} = BA^\tau$$

Заменом матрица A, B, C, D у (3.12) добија се Адамарова матрица H_{12}

$$H_{12} = \begin{bmatrix}
1 & 1 & 1 & 1 & -1 & -1 & 1 & -1 & -1 & 1 & -1 & -1 \\
1 & 1 & 1 & -1 & 1 & -1 & -1 & 1 & -1 & -1 & 1 & -1 \\
1 & 1 & 1 & -1 & -1 & 1 & -1 & -1 & 1 & -1 & -1 & 1 \\
1 & -1 & -1 & -1 & -1 & -1 & 1 & -1 & -1 & -1 & 1 & 1 \\
-1 & 1 & -1 & -1 & -1 & -1 & -1 & 1 & -1 & 1 & -1 & 1 \\
-1 & -1 & 1 & -1 & -1 & -1 & -1 & -1 & 1 & 1 & 1 & -1 \\
1 & -1 & -1 & -1 & 1 & 1 & -1 & -1 & -1 & 1 & -1 & -1 \\
-1 & 1 & -1 & 1 & -1 & 1 & -1 & -1 & -1 & -1 & 1 & -1 \\
-1 & -1 & 1 & 1 & 1 & -1 & -1 & -1 & -1 & -1 & -1 & 1 \\
1 & -1 & -1 & 1 & -1 & -1 & -1 & 1 & 1 & -1 & -1 & -1 \\
-1 & 1 & -1 & -1 & 1 & -1 & 1 & -1 & 1 & -1 & -1 & -1 \\
-1 & -1 & 1 & -1 & -1 & 1 & 1 & 1 & -1 & -1 & -1 & -1
\end{bmatrix}$$

Графичка репрезентација добијене матрице приказана на слици 3.6



Слика 3.6: Адамарова матрица реда 12 добијена Итовом конструкцијом.

Алгоритам 5 Ито конструкција

```
procedure Ito(A,B,C,D)
    ▷ Улаз: Цикличне A,B,C,D матрице
```
$$H \leftarrow \begin{bmatrix} A & B & C & D \\ B & -A & D & -C \\ C^\tau & -D^\tau & -A^\tau & B^\tau \\ D^\tau & C^\tau & -B^\tau & -A^\tau \end{bmatrix}$$
```
    if AB^τ + CD^τ = BA^τ + DC^τ then
        return H
    else
        return null
    end if
end procedure
```

3.6 Ехлихова конструкција

Ехлихова конкстукција се заснива на следећој теореми:

Теорема 3.6.1. *Нека је $A_m = [(a(i,j)]$ косо симетрична матрица димензија m где за елементе важи $a_{ii} = 0, a_{ij} = \pm 1, j \neq i$ и*

$$A_m A_m^\tau = A_m^2 = mI_m - J_m. \tag{3.13}$$

Нека је $J[j(k,l)] = +1$ матрица чији су сви елементи једнаки $+1$. Ако је G косо симетрична матрица облика тако да важи:

$$H^*_{n+1} = \begin{bmatrix} 1 & \vec{1} \\ -\vec{1}^\tau & I_n + G \end{bmatrix} \tag{3.14}$$

и нека је K косо симетрична Адамарова матрица реда $n+1$ и $(n-2) = m = q \equiv 1 \bmod 4$ где је q степен простог броја, при чему важи:

$$K = A_m \otimes G + I_m \otimes (I_n - J_n) + J_m \otimes I_n \tag{3.15}$$

тада је H Адамарова матрица реда $(n-1)^2$

$$H = \begin{bmatrix} 1 & \vec{1} \\ \vec{1}^\tau & K \end{bmatrix} \tag{3.16}$$

За доказ видети [8].
У следећем примеру матрице A, G су Јакобшталове матрице које се добијају Paley I и Paley II методома са $q = 7$ и $q = 5$ односно над F_7 и F_5 које се срећу у примерима 3.2.1 и 3.2.2.

Пример 3.6.1. *Адамарова матрица реда 36 добијена методом Ехлиха.*

$$A = \begin{bmatrix} 0 & 1 & -1 & -1 & 1 \\ 1 & 0 & 1 & -1 & -1 \\ -1 & 1 & 0 & 1 & -1 \\ -1 & -1 & 1 & 0 & 1 \\ 1 & -1 & -1 & 1 & 0 \end{bmatrix}, G = \begin{bmatrix} 0 & -1 & -1 & 1 & -1 & 1 & 1 \\ 1 & 0 & -1 & -1 & 1 & -1 & 1 \\ 1 & 1 & 0 & -1 & -1 & 1 & -1 \\ -1 & 1 & 1 & 0 & -1 & -1 & 1 \\ 1 & -1 & 1 & 1 & 0 & -1 & -1 \\ -1 & 1 & -1 & 1 & 1 & 0 & -1 \\ -1 & -1 & 1 & -1 & 1 & 1 & 0 \end{bmatrix}$$

Лако се проверава да важи

$$AA^{\tau} = A^2 = 5I_5 - J_5 = \begin{bmatrix} 4 & -1 & -1 & -1 & -1 \\ -1 & 4 & -1 & -1 & -1 \\ -1 & -1 & 4 & -1 & -1 \\ -1 & -1 & -1 & 4 & -1 \\ -1 & -1 & -1 & -1 & 4 \end{bmatrix}$$

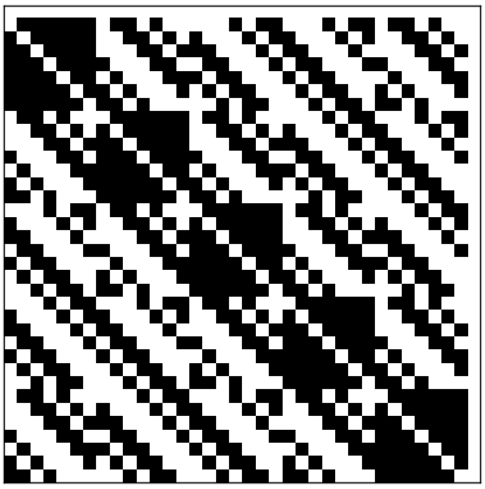
Даље је:

$$A \otimes G = \begin{bmatrix}
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & -1 & 1 & -1 & 1 & 1 & 0 & 1 & 1 & -1 & 1 & -1 & -1 & 0 & 1 & 1 & -1 & 1 & -1 & -1 & 0 & -1 & -1 & 1 & -1 & 1 & 1 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & -1 & -1 & 1 & -1 & 1 & -1 & 0 & 1 & 1 & -1 & 1 & -1 & -1 & 0 & 1 & 1 & -1 & 1 & -1 & 1 & 0 & -1 & -1 & 1 & -1 & 1 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & -1 & -1 & 1 & -1 & -1 & -1 & 0 & 1 & 1 & -1 & 1 & -1 & -1 & 0 & 1 & 1 & -1 & 1 & 1 & 1 & 0 & -1 & -1 & 1 & -1 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 1 & 1 & 0 & -1 & -1 & 1 & 1 & -1 & -1 & 0 & 1 & 1 & -1 & 1 & -1 & -1 & 0 & 1 & 1 & -1 & -1 & 1 & 1 & 0 & -1 & -1 & 1 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & -1 & 1 & 1 & 0 & -1 & -1 & -1 & 1 & -1 & -1 & 0 & 1 & 1 & -1 & 1 & -1 & -1 & 0 & 1 & 1 & 1 & -1 & 1 & 1 & 0 & -1 & -1 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 1 & -1 & 1 & 1 & 0 & -1 & 1 & -1 & 1 & -1 & -1 & 0 & 1 & 1 & -1 & 1 & -1 & -1 & 0 & 1 & -1 & 1 & -1 & 1 & 1 & 0 & -1 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & -1 & 1 & -1 & 1 & 1 & 0 & 1 & 1 & -1 & 1 & -1 & -1 & 0 & 1 & 1 & -1 & 1 & -1 & -1 & 0 & -1 & -1 & 1 & -1 & 1 & 1 & 0 \\
0 & -1 & -1 & 1 & -1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & -1 & 1 & -1 & 1 & 1 & 0 & 1 & 1 & -1 & 1 & -1 & -1 & 0 & 1 & 1 & -1 & 1 & -1 & -1 \\
1 & 0 & -1 & -1 & 1 & -1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & -1 & -1 & 1 & -1 & 1 & -1 & 0 & 1 & 1 & -1 & 1 & -1 & -1 & 0 & 1 & 1 & -1 & 1 & -1 \\
1 & 1 & 0 & -1 & -1 & 1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & -1 & -1 & 1 & -1 & -1 & -1 & 0 & 1 & 1 & -1 & 1 & -1 & -1 & 0 & 1 & 1 & -1 & 1 \\
-1 & 1 & 1 & 0 & -1 & -1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 1 & 1 & 0 & -1 & -1 & 1 & 1 & -1 & -1 & 0 & 1 & 1 & -1 & 1 & -1 & -1 & 0 & 1 & 1 & -1 \\
1 & -1 & 1 & 1 & 0 & -1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & -1 & 1 & 1 & 0 & -1 & -1 & -1 & 1 & -1 & -1 & 0 & 1 & 1 & -1 & 1 & -1 & -1 & 0 & 1 & 1 \\
-1 & 1 & -1 & 1 & 1 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 1 & -1 & 1 & 1 & 0 & -1 & 1 & -1 & 1 & -1 & -1 & 0 & 1 & 1 & -1 & 1 & -1 & -1 & 0 & 1 \\
-1 & -1 & 1 & -1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & -1 & 1 & -1 & 1 & 1 & 0 & 1 & 1 & -1 & 1 & -1 & -1 & 0 & 1 & 1 & -1 & 1 & -1 & -1 & 0 \\
0 & 1 & 1 & -1 & 1 & -1 & -1 & 0 & -1 & -1 & 1 & -1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & -1 & 1 & -1 & 1 & 1 & 0 & 1 & 1 & -1 & 1 & -1 & -1 \\
-1 & 0 & 1 & 1 & -1 & 1 & -1 & 1 & 0 & -1 & -1 & 1 & -1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & -1 & -1 & 1 & -1 & 1 & -1 & 0 & 1 & 1 & -1 & 1 & -1 \\
-1 & -1 & 0 & 1 & 1 & -1 & 1 & 1 & 1 & 0 & -1 & -1 & 1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & -1 & -1 & 1 & -1 & -1 & -1 & 0 & 1 & 1 & -1 & 1 \\
1 & -1 & -1 & 0 & 1 & 1 & -1 & -1 & 1 & 1 & 0 & -1 & -1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 1 & 1 & 0 & -1 & -1 & 1 & 1 & -1 & -1 & 0 & 1 & 1 & -1 \\
-1 & 1 & -1 & -1 & 0 & 1 & 1 & 1 & -1 & 1 & 1 & 0 & -1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & -1 & 1 & 1 & 0 & -1 & -1 & -1 & 1 & -1 & -1 & 0 & 1 & 1 \\
1 & -1 & 1 & -1 & -1 & 0 & 1 & -1 & 1 & -1 & 1 & 1 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 1 & -1 & 1 & 1 & 0 & -1 & 1 & -1 & 1 & -1 & -1 & 0 & 1 \\
1 & 1 & -1 & 1 & -1 & -1 & 0 & -1 & -1 & 1 & -1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & -1 & 1 & -1 & 1 & 1 & 0 & 1 & 1 & -1 & 1 & -1 & -1 & 0 \\
0 & 1 & 1 & -1 & 1 & -1 & -1 & 0 & 1 & 1 & -1 & 1 & -1 & -1 & 0 & -1 & -1 & 1 & -1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & -1 & 1 & -1 & 1 & 1 \\
-1 & 0 & 1 & 1 & -1 & 1 & -1 & -1 & 0 & 1 & 1 & -1 & 1 & -1 & 1 & 0 & -1 & -1 & 1 & -1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & -1 & -1 & 1 & -1 & 1 \\
-1 & -1 & 0 & 1 & 1 & -1 & 1 & -1 & -1 & 0 & 1 & 1 & -1 & 1 & 1 & 1 & 0 & -1 & -1 & 1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & -1 & -1 & 1 & -1 \\
1 & -1 & -1 & 0 & 1 & 1 & -1 & 1 & -1 & -1 & 0 & 1 & 1 & -1 & -1 & 1 & 1 & 0 & -1 & -1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 1 & 1 & 0 & -1 & -1 & 1 \\
-1 & 1 & -1 & -1 & 0 & 1 & 1 & -1 & 1 & -1 & -1 & 0 & 1 & 1 & 1 & -1 & 1 & 1 & 0 & -1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & -1 & 1 & 1 & 0 & -1 & -1 \\
1 & -1 & 1 & -1 & -1 & 0 & 1 & 1 & -1 & 1 & -1 & -1 & 0 & 1 & -1 & 1 & -1 & 1 & 1 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 1 & -1 & 1 & 1 & 0 & -1 \\
1 & 1 & -1 & 1 & -1 & -1 & 0 & 1 & 1 & -1 & 1 & -1 & -1 & 0 & -1 & -1 & 1 & -1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & -1 & 1 & -1 & 1 & 1 & 0 \\
0 & -1 & -1 & 1 & -1 & 1 & 1 & 0 & 1 & 1 & -1 & 1 & -1 & -1 & 0 & 1 & 1 & -1 & 1 & -1 & -1 & 0 & -1 & -1 & 1 & -1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
1 & 0 & -1 & -1 & 1 & -1 & 1 & -1 & 0 & 1 & 1 & -1 & 1 & -1 & -1 & 0 & 1 & 1 & -1 & 1 & -1 & 1 & 0 & -1 & -1 & 1 & -1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
1 & 1 & 0 & -1 & -1 & 1 & -1 & -1 & -1 & 0 & 1 & 1 & -1 & 1 & -1 & -1 & 0 & 1 & 1 & -1 & 1 & 1 & 1 & 0 & -1 & -1 & 1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
-1 & 1 & 1 & 0 & -1 & -1 & 1 & 1 & -1 & -1 & 0 & 1 & 1 & -1 & 1 & -1 & -1 & 0 & 1 & 1 & -1 & -1 & 1 & 1 & 0 & -1 & -1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
1 & -1 & 1 & 1 & 0 & -1 & -1 & -1 & 1 & -1 & -1 & 0 & 1 & 1 & -1 & 1 & -1 & -1 & 0 & 1 & 1 & 1 & -1 & 1 & 1 & 0 & -1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
-1 & 1 & -1 & 1 & 1 & 0 & -1 & 1 & -1 & 1 & -1 & -1 & 0 & 1 & 1 & -1 & 1 & -1 & -1 & 0 & 1 & -1 & 1 & -1 & 1 & 1 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
-1 & -1 & 1 & -1 & 1 & 1 & 0 & 1 & 1 & -1 & 1 & -1 & -1 & 0 & 1 & 1 & -1 & 1 & -1 & -1 & 0 & -1 & -1 & 1 & -1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0
\end{bmatrix}$$

$$K = \begin{bmatrix}
1 & -1 & -1 & -1 & -1 & -1 & -1 & 1 & -1 & -1 & 1 & -1 & 1 & 1 & 1 & 1 & 1 & -1 & 1 & -1 & -1 & 1 & 1 & 1 & -1 & 1 & -1 & -1 & 1 & -1 & -1 & 1 & -1 & 1 & 1 \\
-1 & 1 & -1 & -1 & -1 & -1 & -1 & 1 & 1 & -1 & -1 & 1 & -1 & 1 & -1 & 1 & 1 & 1 & -1 & 1 & -1 & -1 & 1 & 1 & 1 & -1 & 1 & -1 & 1 & 1 & -1 & -1 & 1 & -1 & 1 \\
-1 & -1 & 1 & -1 & -1 & -1 & -1 & 1 & 1 & 1 & -1 & -1 & 1 & -1 & -1 & -1 & 1 & 1 & 1 & -1 & 1 & -1 & -1 & 1 & 1 & 1 & -1 & 1 & 1 & 1 & 1 & -1 & -1 & 1 & -1 \\
-1 & -1 & -1 & 1 & -1 & -1 & -1 & -1 & 1 & 1 & 1 & -1 & -1 & 1 & 1 & -1 & -1 & 1 & 1 & 1 & -1 & 1 & -1 & -1 & 1 & 1 & 1 & -1 & -1 & 1 & 1 & 1 & -1 & -1 & 1 \\
-1 & -1 & -1 & -1 & 1 & -1 & -1 & 1 & -1 & 1 & 1 & 1 & -1 & -1 & -1 & 1 & -1 & -1 & 1 & 1 & 1 & -1 & 1 & -1 & -1 & 1 & 1 & 1 & 1 & -1 & 1 & 1 & 1 & -1 & -1 \\
-1 & -1 & -1 & -1 & -1 & 1 & -1 & -1 & 1 & -1 & 1 & 1 & 1 & -1 & 1 & -1 & 1 & -1 & -1 & 1 & 1 & 1 & -1 & 1 & -1 & -1 & 1 & 1 & -1 & 1 & -1 & 1 & 1 & 1 & -1 \\
-1 & -1 & -1 & -1 & -1 & -1 & 1 & -1 & -1 & 1 & -1 & 1 & 1 & 1 & 1 & 1 & -1 & 1 & -1 & -1 & 1 & 1 & 1 & -1 & 1 & -1 & -1 & 1 & -1 & -1 & 1 & -1 & 1 & 1 & 1 \\
1 & -1 & -1 & 1 & -1 & 1 & 1 & 1 & -1 & -1 & -1 & -1 & -1 & -1 & 1 & -1 & -1 & 1 & -1 & 1 & 1 & 1 & 1 & 1 & -1 & 1 & -1 & -1 & 1 & 1 & 1 & -1 & 1 & -1 & -1 \\
1 & 1 & -1 & -1 & 1 & -1 & 1 & -1 & 1 & -1 & -1 & -1 & -1 & -1 & 1 & 1 & -1 & -1 & 1 & -1 & 1 & -1 & 1 & 1 & 1 & -1 & 1 & -1 & -1 & 1 & 1 & 1 & -1 & 1 & -1 \\
1 & 1 & 1 & -1 & -1 & 1 & -1 & -1 & -1 & 1 & -1 & -1 & -1 & -1 & 1 & 1 & 1 & -1 & -1 & 1 & -1 & -1 & -1 & 1 & 1 & 1 & -1 & 1 & -1 & -1 & 1 & 1 & 1 & -1 & 1 \\
-1 & 1 & 1 & 1 & -1 & -1 & 1 & -1 & -1 & -1 & 1 & -1 & -1 & -1 & -1 & 1 & 1 & 1 & -1 & -1 & 1 & 1 & -1 & -1 & 1 & 1 & 1 & -1 & 1 & -1 & -1 & 1 & 1 & 1 & -1 \\
1 & -1 & 1 & 1 & 1 & -1 & -1 & -1 & -1 & -1 & -1 & 1 & -1 & -1 & 1 & -1 & 1 & 1 & 1 & -1 & -1 & -1 & 1 & -1 & -1 & 1 & 1 & 1 & -1 & 1 & -1 & -1 & 1 & 1 & 1 \\
-1 & 1 & -1 & 1 & 1 & 1 & -1 & -1 & -1 & -1 & -1 & -1 & 1 & -1 & -1 & 1 & -1 & 1 & 1 & 1 & -1 & 1 & -1 & 1 & -1 & -1 & 1 & 1 & 1 & -1 & 1 & -1 & -1 & 1 & 1 \\
-1 & -1 & 1 & -1 & 1 & 1 & 1 & -1 & -1 & -1 & -1 & -1 & -1 & 1 & -1 & -1 & 1 & -1 & 1 & 1 & 1 & 1 & 1 & -1 & 1 & -1 & -1 & 1 & 1 & 1 & -1 & 1 & -1 & -1 & 1 \\
1 & 1 & 1 & -1 & 1 & -1 & -1 & 1 & -1 & -1 & 1 & -1 & 1 & 1 & 1 & -1 & -1 & -1 & -1 & -1 & -1 & 1 & -1 & -1 & 1 & -1 & 1 & 1 & 1 & 1 & 1 & -1 & 1 & -1 & -1 \\
-1 & 1 & 1 & 1 & -1 & 1 & -1 & 1 & 1 & -1 & -1 & 1 & -1 & 1 & -1 & 1 & -1 & -1 & -1 & -1 & -1 & 1 & 1 & -1 & -1 & 1 & -1 & 1 & -1 & 1 & 1 & 1 & -1 & 1 & -1 \\
-1 & -1 & 1 & 1 & 1 & -1 & 1 & 1 & 1 & 1 & -1 & -1 & 1 & -1 & -1 & -1 & 1 & -1 & -1 & -1 & -1 & 1 & 1 & 1 & -1 & -1 & 1 & -1 & -1 & -1 & 1 & 1 & 1 & -1 & 1 \\
1 & -1 & -1 & 1 & 1 & 1 & -1 & -1 & 1 & 1 & 1 & -1 & -1 & 1 & -1 & -1 & -1 & 1 & -1 & -1 & -1 & -1 & 1 & 1 & 1 & -1 & -1 & 1 & 1 & -1 & -1 & 1 & 1 & 1 & -1 \\
-1 & 1 & -1 & -1 & 1 & 1 & 1 & 1 & -1 & 1 & 1 & 1 & -1 & -1 & -1 & -1 & -1 & -1 & 1 & -1 & -1 & 1 & -1 & 1 & 1 & 1 & -1 & -1 & -1 & 1 & -1 & -1 & 1 & 1 & 1 \\
1 & -1 & 1 & -1 & -1 & 1 & 1 & -1 & 1 & -1 & 1 & 1 & 1 & -1 & -1 & -1 & -1 & -1 & -1 & 1 & -1 & -1 & 1 & -1 & 1 & 1 & 1 & -1 & 1 & -1 & 1 & -1 & -1 & 1 & 1 \\
1 & 1 & -1 & 1 & -1 & -1 & 1 & -1 & -1 & 1 & -1 & 1 & 1 & 1 & -1 & -1 & -1 & -1 & -1 & -1 & 1 & -1 & -1 & 1 & -1 & 1 & 1 & 1 & 1 & 1 & -1 & 1 & -1 & -1 & 1 \\
1 & 1 & 1 & -1 & 1 & -1 & -1 & 1 & 1 & 1 & -1 & 1 & -1 & -1 & 1 & -1 & -1 & 1 & -1 & 1 & 1 & 1 & -1 & -1 & -1 & -1 & -1 & -1 & 1 & -1 & -1 & 1 & -1 & 1 & 1 \\
-1 & 1 & 1 & 1 & -1 & 1 & -1 & -1 & 1 & 1 & 1 & -1 & 1 & -1 & 1 & 1 & -1 & -1 & 1 & -1 & 1 & -1 & 1 & -1 & -1 & -1 & -1 & -1 & 1 & 1 & -1 & -1 & 1 & -1 & 1 \\
-1 & -1 & 1 & 1 & 1 & -1 & 1 & -1 & -1 & 1 & 1 & 1 & -1 & 1 & 1 & 1 & 1 & -1 & -1 & 1 & -1 & -1 & -1 & 1 & -1 & -1 & -1 & -1 & 1 & 1 & 1 & -1 & -1 & 1 & -1 \\
1 & -1 & -1 & 1 & 1 & 1 & -1 & 1 & -1 & -1 & 1 & 1 & 1 & -1 & -1 & 1 & 1 & 1 & -1 & -1 & 1 & -1 & -1 & -1 & 1 & -1 & -1 & -1 & -1 & 1 & 1 & 1 & -1 & -1 & 1 \\
-1 & 1 & -1 & -1 & 1 & 1 & 1 & -1 & 1 & -1 & -1 & 1 & 1 & 1 & 1 & -1 & 1 & 1 & 1 & -1 & -1 & -1 & -1 & -1 & -1 & 1 & -1 & -1 & 1 & -1 & 1 & 1 & 1 & -1 & -1 \\
1 & -1 & 1 & -1 & -1 & 1 & 1 & 1 & -1 & 1 & -1 & -1 & 1 & 1 & -1 & 1 & -1 & 1 & 1 & 1 & -1 & -1 & -1 & -1 & -1 & -1 & 1 & -1 & -1 & 1 & -1 & 1 & 1 & 1 & -1 \\
1 & 1 & -1 & 1 & -1 & -1 & 1 & 1 & 1 & -1 & 1 & -1 & -1 & 1 & -1 & -1 & 1 & -1 & 1 & 1 & 1 & -1 & -1 & -1 & -1 & -1 & -1 & 1 & -1 & -1 & 1 & -1 & 1 & 1 & 1 \\
1 & -1 & -1 & 1 & -1 & 1 & 1 & 1 & 1 & 1 & -1 & 1 & -1 & -1 & 1 & 1 & 1 & -1 & 1 & -1 & -1 & 1 & -1 & -1 & 1 & -1 & 1 & 1 & 1 & -1 & -1 & -1 & -1 & -1 & -1 \\
1 & 1 & -1 & -1 & 1 & -1 & 1 & -1 & 1 & 1 & 1 & -1 & 1 & -1 & -1 & 1 & 1 & 1 & -1 & 1 & -1 & 1 & 1 & -1 & -1 & 1 & -1 & 1 & -1 & 1 & -1 & -1 & -1 & -1 & -1 \\
1 & 1 & 1 & -1 & -1 & 1 & -1 & -1 & -1 & 1 & 1 & 1 & -1 & 1 & -1 & -1 & 1 & 1 & 1 & -1 & 1 & 1 & 1 & 1 & -1 & -1 & 1 & -1 & -1 & -1 & 1 & -1 & -1 & -1 & -1 \\
-1 & 1 & 1 & 1 & -1 & -1 & 1 & 1 & -1 & -1 & 1 & 1 & 1 & -1 & 1 & -1 & -1 & 1 & 1 & 1 & -1 & -1 & 1 & 1 & 1 & -1 & -1 & 1 & -1 & -1 & -1 & 1 & -1 & -1 & -1 \\
1 & -1 & 1 & 1 & 1 & -1 & -1 & -1 & 1 & -1 & -1 & 1 & 1 & 1 & -1 & 1 & -1 & -1 & 1 & 1 & 1 & 1 & -1 & 1 & 1 & 1 & -1 & -1 & -1 & -1 & -1 & -1 & 1 & -1 & -1 \\
-1 & 1 & -1 & 1 & 1 & 1 & -1 & 1 & -1 & 1 & -1 & -1 & 1 & 1 & 1 & -1 & 1 & -1 & -1 & 1 & 1 & -1 & 1 & -1 & 1 & 1 & 1 & -1 & -1 & -1 & -1 & -1 & -1 & 1 & -1 \\
-1 & -1 & 1 & -1 & 1 & 1 & 1 & 1 & 1 & -1 & 1 & -1 & -1 & 1 & 1 & 1 & -1 & 1 & -1 & -1 & 1 & -1 & -1 & 1 & -1 & 1 & 1 & 1 & -1 & -1 & -1 & -1 & -1 & -1 & 1
\end{bmatrix}$$

$$
H_{36} = \left[\begin{array}{rrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrr}
1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\
1 & -1 & -1 & -1 & -1 & -1 & -1 & 1 & -1 & -1 & 1 & -1 & 1 & 1 & 1 & 1 & 1 & -1 & 1 & -1 & -1 & 1 & 1 & 1 & -1 & 1 & -1 & -1 & 1 & -1 & -1 & 1 & -1 & 1 & 1 & 1 \\
-1 & 1 & -1 & -1 & -1 & -1 & -1 & 1 & 1 & -1 & -1 & 1 & -1 & 1 & -1 & 1 & 1 & 1 & -1 & 1 & -1 & -1 & 1 & 1 & 1 & -1 & 1 & -1 & 1 & 1 & -1 & -1 & 1 & -1 & 1 & 1 \\
-1 & -1 & 1 & -1 & -1 & -1 & -1 & 1 & 1 & 1 & -1 & -1 & 1 & -1 & -1 & -1 & 1 & 1 & 1 & -1 & 1 & -1 & -1 & 1 & 1 & 1 & -1 & 1 & 1 & 1 & 1 & -1 & -1 & 1 & -1 & 1 \\
-1 & -1 & -1 & 1 & -1 & -1 & -1 & -1 & 1 & 1 & 1 & -1 & -1 & 1 & 1 & -1 & -1 & 1 & 1 & 1 & -1 & 1 & -1 & -1 & 1 & 1 & 1 & -1 & -1 & 1 & 1 & 1 & -1 & -1 & 1 & 1 \\
-1 & -1 & -1 & -1 & 1 & -1 & -1 & 1 & -1 & 1 & 1 & 1 & -1 & -1 & -1 & 1 & -1 & -1 & 1 & 1 & 1 & -1 & 1 & -1 & -1 & 1 & 1 & 1 & 1 & -1 & 1 & 1 & 1 & -1 & -1 & 1 \\
-1 & -1 & -1 & -1 & -1 & 1 & -1 & -1 & 1 & -1 & 1 & 1 & 1 & -1 & 1 & -1 & 1 & -1 & -1 & 1 & 1 & 1 & -1 & 1 & -1 & -1 & 1 & 1 & -1 & 1 & -1 & 1 & 1 & 1 & -1 & 1 \\
-1 & -1 & -1 & -1 & -1 & -1 & 1 & -1 & -1 & 1 & -1 & 1 & 1 & 1 & 1 & 1 & -1 & 1 & -1 & -1 & 1 & 1 & 1 & -1 & 1 & -1 & -1 & 1 & -1 & -1 & 1 & -1 & 1 & 1 & 1 & 1 \\
1 & -1 & -1 & 1 & -1 & 1 & 1 & 1 & -1 & -1 & -1 & -1 & -1 & -1 & 1 & -1 & -1 & 1 & -1 & 1 & 1 & 1 & 1 & 1 & -1 & 1 & -1 & -1 & 1 & 1 & 1 & -1 & 1 & -1 & -1 & 1 \\
1 & 1 & -1 & -1 & 1 & -1 & 1 & -1 & 1 & -1 & -1 & -1 & -1 & -1 & 1 & 1 & -1 & -1 & 1 & -1 & 1 & -1 & 1 & 1 & 1 & -1 & 1 & -1 & -1 & 1 & 1 & 1 & -1 & 1 & -1 & 1 \\
1 & 1 & 1 & -1 & -1 & 1 & -1 & -1 & -1 & 1 & -1 & -1 & -1 & -1 & 1 & 1 & 1 & -1 & -1 & 1 & -1 & -1 & -1 & 1 & 1 & 1 & -1 & 1 & -1 & -1 & 1 & 1 & 1 & -1 & 1 & 1 \\
-1 & 1 & 1 & 1 & -1 & -1 & 1 & -1 & -1 & -1 & 1 & -1 & -1 & -1 & -1 & 1 & 1 & 1 & -1 & -1 & 1 & 1 & -1 & -1 & 1 & 1 & 1 & -1 & 1 & -1 & -1 & 1 & 1 & 1 & -1 & 1 \\
1 & -1 & 1 & 1 & 1 & -1 & -1 & -1 & -1 & -1 & -1 & 1 & -1 & -1 & 1 & -1 & 1 & 1 & 1 & -1 & -1 & -1 & 1 & -1 & -1 & 1 & 1 & 1 & -1 & 1 & -1 & -1 & 1 & 1 & 1 & 1 \\
-1 & 1 & -1 & 1 & 1 & 1 & -1 & -1 & -1 & -1 & -1 & -1 & 1 & -1 & -1 & 1 & -1 & 1 & 1 & 1 & -1 & 1 & -1 & 1 & -1 & -1 & 1 & 1 & 1 & -1 & 1 & -1 & -1 & 1 & 1 & 1 \\
-1 & -1 & 1 & -1 & 1 & 1 & 1 & -1 & -1 & -1 & -1 & -1 & -1 & 1 & -1 & -1 & 1 & -1 & 1 & 1 & 1 & 1 & 1 & -1 & 1 & -1 & -1 & 1 & 1 & 1 & -1 & 1 & -1 & -1 & 1 & 1 \\
1 & 1 & 1 & -1 & 1 & -1 & -1 & 1 & -1 & -1 & 1 & -1 & 1 & 1 & 1 & -1 & -1 & -1 & -1 & -1 & -1 & 1 & -1 & -1 & 1 & -1 & 1 & 1 & 1 & 1 & 1 & -1 & 1 & -1 & -1 & 1 \\
-1 & 1 & 1 & 1 & -1 & 1 & -1 & 1 & 1 & -1 & -1 & 1 & -1 & 1 & -1 & 1 & -1 & -1 & -1 & -1 & -1 & 1 & 1 & -1 & -1 & 1 & -1 & 1 & -1 & 1 & 1 & 1 & -1 & 1 & -1 & 1 \\
-1 & -1 & 1 & 1 & 1 & -1 & 1 & 1 & 1 & 1 & -1 & -1 & 1 & -1 & -1 & -1 & 1 & -1 & -1 & -1 & -1 & 1 & 1 & 1 & -1 & -1 & 1 & -1 & -1 & -1 & 1 & 1 & 1 & -1 & 1 & 1 \\
1 & -1 & -1 & 1 & 1 & 1 & -1 & -1 & 1 & 1 & 1 & -1 & -1 & 1 & -1 & -1 & -1 & 1 & -1 & -1 & -1 & -1 & 1 & 1 & 1 & -1 & -1 & 1 & 1 & -1 & -1 & 1 & 1 & 1 & -1 & 1 \\
-1 & 1 & -1 & -1 & 1 & 1 & 1 & 1 & -1 & 1 & 1 & 1 & -1 & -1 & -1 & -1 & -1 & -1 & 1 & -1 & -1 & 1 & -1 & 1 & 1 & 1 & -1 & -1 & -1 & 1 & -1 & -1 & 1 & 1 & 1 & 1 \\
1 & -1 & 1 & -1 & -1 & 1 & 1 & -1 & 1 & -1 & 1 & 1 & 1 & -1 & -1 & -1 & -1 & -1 & -1 & 1 & -1 & -1 & 1 & -1 & 1 & 1 & 1 & -1 & 1 & -1 & 1 & -1 & -1 & 1 & 1 & 1 \\
1 & 1 & -1 & 1 & -1 & -1 & 1 & -1 & -1 & 1 & -1 & 1 & 1 & 1 & -1 & -1 & -1 & -1 & -1 & -1 & 1 & -1 & -1 & 1 & -1 & 1 & 1 & 1 & 1 & 1 & -1 & 1 & -1 & -1 & 1 & 1 \\
1 & 1 & 1 & -1 & 1 & -1 & -1 & 1 & 1 & 1 & -1 & 1 & -1 & -1 & 1 & -1 & -1 & 1 & -1 & 1 & 1 & 1 & -1 & -1 & -1 & -1 & -1 & -1 & 1 & -1 & -1 & 1 & -1 & 1 & 1 & 1 \\
-1 & 1 & 1 & 1 & -1 & 1 & -1 & -1 & 1 & 1 & 1 & -1 & 1 & -1 & 1 & 1 & -1 & -1 & 1 & -1 & 1 & -1 & 1 & -1 & -1 & -1 & -1 & -1 & 1 & 1 & -1 & -1 & 1 & -1 & 1 & 1 \\
-1 & -1 & 1 & 1 & 1 & -1 & 1 & -1 & -1 & 1 & 1 & 1 & -1 & 1 & 1 & 1 & 1 & -1 & -1 & 1 & -1 & -1 & -1 & 1 & -1 & -1 & -1 & -1 & 1 & 1 & 1 & -1 & -1 & 1 & -1 & 1 \\
1 & -1 & -1 & 1 & 1 & 1 & -1 & 1 & -1 & -1 & 1 & 1 & 1 & -1 & -1 & 1 & 1 & 1 & -1 & -1 & 1 & -1 & -1 & -1 & 1 & -1 & -1 & -1 & -1 & 1 & 1 & 1 & -1 & -1 & 1 & 1 \\
-1 & 1 & -1 & -1 & 1 & 1 & 1 & -1 & 1 & -1 & -1 & 1 & 1 & 1 & 1 & -1 & 1 & 1 & 1 & -1 & -1 & -1 & -1 & -1 & -1 & 1 & -1 & -1 & 1 & -1 & 1 & 1 & 1 & -1 & -1 & 1 \\
1 & -1 & 1 & -1 & -1 & 1 & 1 & 1 & -1 & 1 & -1 & -1 & 1 & 1 & -1 & 1 & -1 & 1 & 1 & 1 & -1 & -1 & -1 & -1 & -1 & -1 & 1 & -1 & -1 & 1 & -1 & 1 & 1 & 1 & -1 & 1 \\
1 & 1 & -1 & 1 & -1 & -1 & 1 & 1 & 1 & -1 & 1 & -1 & -1 & 1 & -1 & -1 & 1 & -1 & 1 & 1 & 1 & -1 & -1 & -1 & -1 & -1 & -1 & 1 & -1 & -1 & 1 & -1 & 1 & 1 & 1 & 1 \\
1 & -1 & -1 & 1 & -1 & 1 & 1 & 1 & 1 & 1 & -1 & 1 & -1 & -1 & 1 & 1 & 1 & -1 & 1 & -1 & -1 & 1 & -1 & -1 & 1 & -1 & 1 & 1 & 1 & -1 & -1 & -1 & -1 & -1 & -1 & 1 \\
1 & 1 & -1 & -1 & 1 & -1 & 1 & -1 & 1 & 1 & 1 & -1 & 1 & -1 & -1 & 1 & 1 & 1 & -1 & 1 & -1 & 1 & 1 & -1 & -1 & 1 & -1 & 1 & -1 & 1 & -1 & -1 & -1 & -1 & -1 & 1 \\
1 & 1 & 1 & -1 & -1 & 1 & -1 & -1 & -1 & 1 & 1 & 1 & -1 & 1 & -1 & -1 & 1 & 1 & 1 & -1 & 1 & 1 & 1 & 1 & -1 & -1 & 1 & -1 & -1 & -1 & 1 & -1 & -1 & -1 & -1 & 1 \\
-1 & 1 & 1 & 1 & -1 & -1 & 1 & 1 & -1 & -1 & 1 & 1 & 1 & -1 & 1 & -1 & -1 & 1 & 1 & 1 & -1 & -1 & 1 & 1 & 1 & -1 & -1 & 1 & -1 & -1 & -1 & 1 & -1 & -1 & -1 & 1 \\
1 & -1 & 1 & 1 & 1 & -1 & -1 & -1 & 1 & -1 & -1 & 1 & 1 & 1 & -1 & 1 & -1 & -1 & 1 & 1 & 1 & 1 & -1 & 1 & 1 & 1 & -1 & -1 & -1 & -1 & -1 & -1 & 1 & -1 & -1 & 1 \\
-1 & 1 & -1 & 1 & 1 & 1 & -1 & 1 & -1 & 1 & -1 & -1 & 1 & 1 & 1 & -1 & 1 & -1 & -1 & 1 & 1 & -1 & 1 & -1 & 1 & 1 & 1 & -1 & -1 & -1 & -1 & -1 & -1 & 1 & -1 & 1 \\
-1 & -1 & 1 & -1 & 1 & 1 & 1 & 1 & 1 & -1 & 1 & -1 & -1 & 1 & 1 & 1 & -1 & 1 & -1 & -1 & 1 & -1 & -1 & 1 & -1 & 1 & 1 & 1 & -1 & -1 & -1 & -1 & -1 & -1 & 1 & 1
\end{array}\right]
$$

Графичка репрезентација добијене матрице је приказана на слици 3.7



Слика 3.7: Адамарова матрица реда 36 добијена Ехлиховом конструкцијом.

Псеудокод који описује наведену конструкцију:

Алгоритам 6 Ехлихова конструкција

procedure ECHLICH(A,G)
  ▷ Улаз: Матрице A, G добијене нпр. Палеј I и Палеј II методама
  if $AA^\tau = A^2$ **then**
    $K = A_m \otimes G + I_m \otimes (I_n - J_n) + J_m \otimes I_n$
    ▷ Излаз: Адамарова матрица добијена Ехлиховом конструкцијом
    return $\begin{bmatrix} 1 & \vec{1} \\ \vec{1}^\tau & K \end{bmatrix}$
  else
    return null
  end if
end procedure

3.7 Гeталс/Сајдл конструкција

Геталс-Сајдл (Goethals[6]/Seidel[7]) конструкција представља генерализацију Вилијамсове конструкције. Адамарове матрице добијене овом методом се такође називају Геталс/Сајдл матрице.

Теорема 3.7.1. *Нека су A, B, C, D цикличне матрице реда n тако да важи:*

$$AA^\tau + BB^\tau + CC^\tau + DD^\tau = 4nI_n \tag{3.17}$$

Нека је матрица R, таква да су јој једини не-нула елементи јединице на споредној дијагонали.

$$R = [r(i,j)] \begin{cases} 1, & \text{ако } i + j = n + 1 \\ 0, & \text{у супротном} \end{cases} \tag{3.18}$$

Геталс-Сајдл матрица је Адамарова матрица следећег облика:

$$H = \begin{bmatrix} A & BR & CR & DR \\ -BR & A & D^\tau R & -C^\tau \\ -CR & -D^\tau R & A & B^\tau R \\ -DR & C^\tau R & -B^\tau R & A \end{bmatrix} \tag{3.19}$$

За доказ видети [9].
С обзиром на то да су матрице A, B, C, D цикличне тј. одређене својом првом врстом довољно им је специфицирати прву врсту. Потрага за почетним врстама ових матрица је непосредно повезана са аутокорелационим функцијама.

Дефиниција 3.7.1. *За задат низ $A = (a_0, a_1, \dots, a_{n-1})$ дужине n, непериодична аутокорелациона функција N_A је задата изразом*

$$N_A(s) = \begin{cases} \sum\limits_{i=0}^{n-1-s} a_i a_{i+s}, s = 0, 1, \dots, n-1 \\ 0, \text{ако је } s \geq n. \end{cases} \tag{3.20}$$

Дефиниција 3.7.2. *Четири низа A, B, C, D чији су сви елементи -1 или +1 редом дужине $n + p, n + p, n, n$ су базни низови ако:*

$$(N_A + N_B + N_C + N_D)(s) = 0, s \geq 1. \tag{3.21}$$

[6] J. M. Goethals - Холандски математичар
[7] Johan Jacob Seidel (1919 - 2001) - Холандски математичар

Дефиниција 3.7.3. *Четири низа A, B, C, D чији су сви елементи -1 или +1 дужине n су такозвани T низови ако:*

$$(N_A + N_B + N_C + N_D)(s) = 0, s \geq 1. \tag{3.22}$$

Дефиниција 3.7.4. *Четири низа X, Y, Z, W чији су сви елементи -1 или +1 редом дужине $n, n, n, n-1$ су такозвани Туринови низови ако:*

$$(N_X + N_Y + 2N_Z + 2N_W)(s) = 0, s \geq 1. \tag{3.23}$$

Ако са 0_n означимо нула низ дужине n, коришћењем следеће теореме, која се наводи без доказа, могуће је конструисати T низове од базних, и базне од Туринових низова.

Теорема 3.7.2. *Нека су A, B, C, D базни низови реда $n+p, n+p, n, n$ тада су низови:*

$$(\frac{1}{2}(A+B), 0_n)), (\frac{1}{2}(A-B), 0_n)), (0_{n+p}, \frac{1}{2}(C+D)), (0_{n+p}, \frac{1}{2}(C-D))$$

T низови дужине $2n+p, 2n+p, 2n+p, 2n+p$.

Идеја на којој се конструкција заснива је да се генеришу четири цикличне матрице T_1, T_2, T_3, T_4 од t низова t_1, t_2, t_3, t_4, и да се матице A, B, C, D:

$$\begin{aligned}
A &= T_1 + T_2 + T_3 + T_4 \\
B &= -T_1 + T_2 + T_3 - T_4 \\
C &= -T_1 - T_2 + T_3 + T_4 \\
D &= -T_1 + T_2 - T_3 + T_4
\end{aligned}$$

замене у (3.19). На тај начин се добија матрица реда $4n$. За доказ видети [9].

Занимљива чињеница јесте да је Адамарова матрица H_{428} конструисана овом методом [1]. Туринови низови су генерисани користећи кластер од шеснаест 2.6GHz рачунара и дванаест сати рачунарског времена. Овај напор је као резултат дао до тада непознате Туринове низове реда $36, 36, 36, 35$.

Пример 3.7.1. *Адамарова матрица реда 28 добијена методом Геталс-Сајдл.*

Следећи базни низови су преузете из базе [6]. У овој архиви се могу наћи Туринови низови димензија до 36.
Нека су $a = (1, 1, -1, 1), b = (1, 1, -1, -1), c = (1, 1, 1), d = (1, -1, 1)$ базни низови редом реда $4, 4, 3, 3$, следи да је $n = 3$ и $p = 1$. Користећи Теорему (3.7.2) четири T низа реда $2n + p = 7$ су:

$$\begin{aligned}
t_1 &= (1, 1, -1, 0, 0, 0, 0) \\
t_2 &= (0, 0, 0, -1, 0, 0, 0) \\
t_3 &= (0, 0, 0, 0, 1, 0, 1) \\
t_4 &= (0, 0, 0, 0, 0, 1, 0)
\end{aligned}$$

На основу (2.1.4) формирају се чикличне матрице T_1, T_2, T_3, T_4.

$$T_1 = \begin{bmatrix} 1 & 1 & -1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & -1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & -1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & -1 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 & -1 \\ -1 & 0 & 0 & 0 & 0 & 1 & 1 \\ 1 & -1 & 0 & 0 & 0 & 0 & 1 \end{bmatrix},$$

$$T_2 = \begin{bmatrix} 0 & 0 & 0 & -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & -1 \\ -1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & -1 & 0 & 0 & 0 & 0 \end{bmatrix},$$

$$T_3 = \begin{bmatrix} 0 & 0 & 0 & 0 & 1 & 0 & 1 \\ 1 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 1 \\ 1 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 1 & 0 \end{bmatrix}, T_4 = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 \end{bmatrix}.$$

Полазећи од матрица T_1, T_2, T_3, T_4, добијају се матрице A, B, C, D:

$$A = \begin{bmatrix} 1 & 1 & -1 & -1 & 1 & 1 & 1 \\ 1 & 1 & 1 & -1 & -1 & 1 & 1 \\ 1 & 1 & 1 & 1 & -1 & -1 & 1 \\ 1 & 1 & 1 & 1 & 1 & -1 & -1 \\ -1 & 1 & 1 & 1 & 1 & 1 & -1 \\ -1 & -1 & 1 & 1 & 1 & 1 & 1 \\ 1 & -1 & -1 & 1 & 1 & 1 & 1 \end{bmatrix}, B = \begin{bmatrix} -1 & -1 & 1 & -1 & 1 & -1 & 1 \\ 1 & -1 & -1 & 1 & -1 & 1 & -1 \\ -1 & 1 & -1 & -1 & 1 & -1 & 1 \\ 1 & -1 & 1 & -1 & -1 & 1 & -1 \\ -1 & 1 & -1 & 1 & -1 & -1 & 1 \\ 1 & -1 & 1 & -1 & 1 & -1 & -1 \\ -1 & 1 & -1 & 1 & -1 & 1 & -1 \end{bmatrix},$$

$$C = \begin{bmatrix} -1 & -1 & 1 & 1 & 1 & 1 & 1 \\ 1 & -1 & -1 & 1 & 1 & 1 & 1 \\ 1 & 1 & -1 & -1 & 1 & 1 & 1 \\ 1 & 1 & 1 & -1 & -1 & 1 & 1 \\ 1 & 1 & 1 & 1 & -1 & -1 & 1 \\ 1 & 1 & 1 & 1 & 1 & -1 & -1 \\ -1 & 1 & 1 & 1 & 1 & 1 & -1 \end{bmatrix}, D = \begin{bmatrix} -1 & -1 & 1 & -1 & -1 & 1 & -1 \\ -1 & -1 & -1 & 1 & -1 & -1 & 1 \\ 1 & -1 & -1 & -1 & 1 & -1 & -1 \\ -1 & 1 & -1 & -1 & -1 & 1 & -1 \\ -1 & -1 & 1 & -1 & -1 & -1 & 1 \\ 1 & -1 & -1 & 1 & -1 & -1 & -1 \\ -1 & 1 & -1 & -1 & 1 & -1 & -1 \end{bmatrix}.$$

Непосредно се проверава да је

$$AA^\tau + BB^\tau + CC^\tau + DD^\tau = \begin{bmatrix} 28 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 28 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 28 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 28 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 28 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 28 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 28 \end{bmatrix} = 4 \cdot 7I_7$$

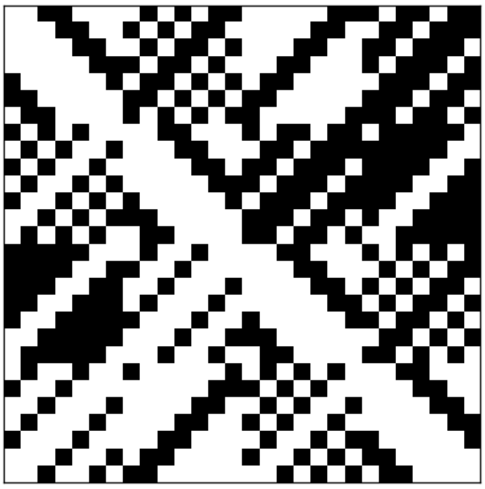
и матрица R има јединице на споредној дијагонали

$$R = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

Заменом матрица A, B, C, D, R у (3.19) добија се Адамарова матрица:

$$H_{28} = \begin{bmatrix}
1 & 1 & -1 & -1 & 1 & 1 & 1 & 1 & -1 & 1 & -1 & 1 & -1 & -1 & 1 & 1 & 1 & 1 & 1 & -1 & -1 & -1 & 1 & -1 & -1 & 1 & -1 & -1 \\
1 & 1 & 1 & -1 & -1 & 1 & 1 & -1 & 1 & -1 & 1 & -1 & -1 & 1 & 1 & 1 & 1 & 1 & -1 & -1 & 1 & 1 & -1 & -1 & 1 & -1 & -1 & -1 \\
1 & 1 & 1 & 1 & -1 & -1 & 1 & 1 & -1 & 1 & -1 & -1 & 1 & -1 & 1 & 1 & 1 & -1 & -1 & 1 & 1 & -1 & -1 & 1 & -1 & -1 & -1 & 1 \\
1 & 1 & 1 & 1 & 1 & -1 & -1 & -1 & 1 & -1 & -1 & 1 & -1 & 1 & 1 & 1 & -1 & -1 & 1 & 1 & 1 & -1 & 1 & -1 & -1 & -1 & 1 & -1 \\
-1 & 1 & 1 & 1 & 1 & 1 & -1 & 1 & -1 & -1 & 1 & -1 & 1 & -1 & 1 & -1 & -1 & 1 & 1 & 1 & 1 & 1 & -1 & -1 & -1 & 1 & -1 & -1 \\
-1 & -1 & 1 & 1 & 1 & 1 & 1 & -1 & -1 & 1 & -1 & 1 & -1 & 1 & -1 & -1 & 1 & 1 & 1 & 1 & 1 & -1 & -1 & -1 & 1 & -1 & -1 & 1 \\
1 & -1 & -1 & 1 & 1 & 1 & 1 & -1 & 1 & -1 & 1 & -1 & 1 & -1 & -1 & 1 & 1 & 1 & 1 & 1 & -1 & -1 & -1 & 1 & -1 & -1 & 1 & -1 \\
-1 & 1 & -1 & 1 & -1 & 1 & 1 & 1 & 1 & -1 & -1 & 1 & 1 & 1 & -1 & 1 & -1 & -1 & 1 & -1 & -1 & 1 & -1 & -1 & -1 & -1 & -1 & 1 \\
1 & -1 & 1 & -1 & 1 & 1 & -1 & 1 & 1 & 1 & -1 & -1 & 1 & 1 & 1 & -1 & -1 & 1 & -1 & -1 & -1 & -1 & -1 & -1 & -1 & -1 & 1 & 1 \\
-1 & 1 & -1 & 1 & 1 & -1 & 1 & 1 & 1 & 1 & 1 & -1 & -1 & 1 & -1 & -1 & 1 & -1 & -1 & -1 & 1 & -1 & -1 & -1 & -1 & 1 & 1 & -1 \\
1 & -1 & 1 & 1 & -1 & 1 & -1 & 1 & 1 & 1 & 1 & 1 & -1 & -1 & -1 & 1 & -1 & -1 & -1 & 1 & -1 & -1 & -1 & -1 & 1 & 1 & -1 & -1 \\
-1 & 1 & 1 & -1 & 1 & -1 & 1 & -1 & 1 & 1 & 1 & 1 & 1 & -1 & 1 & -1 & -1 & -1 & 1 & -1 & -1 & -1 & -1 & 1 & 1 & -1 & -1 & -1 \\
1 & 1 & -1 & 1 & -1 & 1 & -1 & -1 & -1 & 1 & 1 & 1 & 1 & 1 & -1 & -1 & -1 & 1 & -1 & -1 & 1 & -1 & 1 & 1 & -1 & -1 & -1 & -1 \\
1 & -1 & 1 & -1 & 1 & -1 & 1 & 1 & -1 & -1 & 1 & 1 & 1 & 1 & -1 & -1 & 1 & -1 & -1 & 1 & -1 & 1 & 1 & -1 & -1 & -1 & -1 & -1 \\
-1 & -1 & -1 & -1 & -1 & 1 & 1 & 1 & -1 & 1 & 1 & -1 & 1 & 1 & 1 & 1 & -1 & -1 & 1 & 1 & 1 & -1 & 1 & -1 & 1 & -1 & 1 & -1 \\
-1 & -1 & -1 & -1 & 1 & 1 & -1 & -1 & 1 & 1 & -1 & 1 & 1 & 1 & 1 & 1 & 1 & -1 & -1 & 1 & 1 & 1 & -1 & 1 & -1 & 1 & -1 & -1 \\
-1 & -1 & -1 & 1 & 1 & -1 & -1 & 1 & 1 & -1 & 1 & 1 & 1 & -1 & 1 & 1 & 1 & 1 & -1 & -1 & 1 & -1 & 1 & -1 & 1 & -1 & -1 & 1 \\
-1 & -1 & 1 & 1 & -1 & -1 & -1 & 1 & -1 & 1 & 1 & 1 & -1 & 1 & 1 & 1 & 1 & 1 & 1 & -1 & -1 & 1 & -1 & 1 & -1 & -1 & 1 & -1 \\
-1 & 1 & 1 & -1 & -1 & -1 & -1 & -1 & 1 & 1 & 1 & -1 & 1 & 1 & -1 & 1 & 1 & 1 & 1 & 1 & -1 & -1 & 1 & -1 & -1 & 1 & -1 & 1 \\
1 & 1 & -1 & -1 & -1 & -1 & -1 & 1 & 1 & 1 & -1 & 1 & 1 & -1 & -1 & -1 & 1 & 1 & 1 & 1 & 1 & 1 & -1 & -1 & 1 & -1 & 1 & -1 \\
1 & -1 & -1 & -1 & -1 & -1 & 1 & 1 & 1 & -1 & 1 & 1 & -1 & 1 & 1 & -1 & -1 & 1 & 1 & 1 & 1 & -1 & -1 & 1 & -1 & 1 & -1 & 1 \\
1 & -1 & 1 & 1 & -1 & 1 & 1 & -1 & 1 & 1 & 1 & 1 & 1 & -1 & 1 & -1 & 1 & -1 & 1 & -1 & 1 & 1 & 1 & -1 & -1 & 1 & 1 & 1 \\
-1 & 1 & 1 & -1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & -1 & -1 & -1 & 1 & -1 & 1 & -1 & 1 & 1 & 1 & 1 & 1 & -1 & -1 & 1 & 1 \\
1 & 1 & -1 & 1 & 1 & 1 & -1 & 1 & 1 & 1 & 1 & -1 & -1 & 1 & 1 & -1 & 1 & -1 & 1 & 1 & -1 & 1 & 1 & 1 & 1 & -1 & -1 & 1 \\
1 & -1 & 1 & 1 & 1 & -1 & 1 & 1 & 1 & 1 & -1 & -1 & 1 & 1 & -1 & 1 & -1 & 1 & 1 & -1 & 1 & 1 & 1 & 1 & 1 & 1 & -1 & -1 \\
-1 & 1 & 1 & 1 & -1 & 1 & 1 & 1 & 1 & -1 & -1 & 1 & 1 & 1 & 1 & -1 & 1 & 1 & -1 & 1 & -1 & -1 & 1 & 1 & 1 & 1 & 1 & -1 \\
1 & 1 & 1 & -1 & 1 & 1 & -1 & 1 & -1 & -1 & 1 & 1 & 1 & 1 & -1 & 1 & 1 & -1 & 1 & -1 & 1 & -1 & -1 & 1 & 1 & 1 & 1 & 1 \\
1 & 1 & -1 & 1 & 1 & -1 & 1 & -1 & -1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & -1 & 1 & -1 & 1 & -1 & 1 & -1 & -1 & 1 & 1 & 1 & 1
\end{bmatrix}$$

Графичка репрезентација добијене матрице је:



Слика 3.8: Адамарова матрица реда 28 добијена Геталс/Сајдл конструкцијом.

Псеудокод који описује наведену конструкцију:

Алгоритам 7 Геталс/Сајдл конструкција

procedure CIRCULAR(t)
  $A[a(i,j)] = 0$
  $n = length(t)$ ▷ Величина матрице је одређена дужином Туринових низова
  for $i \leftarrow 1$ to n **do**
    for $j \leftarrow 1$ to n **do**
      $A[a(i,j)] \leftarrow A[a(0,(j-i) \mod n)]$
    end for
  end for
  ▷ Излаз: Циркуларна матрица
  return A
end procedure

procedure GOETHALSEIDEL(t1,t2,t3,t4)
  ▷ Улаз: Туринови низови $t1$, $t2$, $t3$, $t4$
  $A \leftarrow Circular(t1)$
  $B \leftarrow Circular(t2)$
  $C \leftarrow Circular(t3)$
  $D \leftarrow Circular(t4)$
  ▷ Генерисање матрице R
  $R[r(i,j)] \leftarrow [r(i,j)] \begin{cases} 1, & \text{ако } i+j=n+1 \\ 0, & \text{у супротном} \end{cases}$
  if $AA^\tau + BB^\tau + CC^\tau + DD^\tau = 4nI_n$ **then**
    ▷ Излаз: Адамарова матрица добијена Геталс/Сајдл конструкцијом
    return $\begin{bmatrix} A & BR & CR & DR \\ -BR & A & D^\tau R & -C^\tau \\ -CR & -D^\tau R & A & B^\tau R \\ -DR & C^\tau R & -B^\tau R & A \end{bmatrix}$
  else
    return null
  end if
end procedure

3.8 Агајан/Сарукханјан конструкција

Агајан/Сарукханјан (Agaian /Sarukhanyan) конструкција представља једноставну конструкцију која се заснива на следећој теореми.

Теорема 3.8.1. *Нека су H_1 и H_2 Адамарове матрице редом реда $4h$, $4k$. Тада постоји Адамарова матрица реда $8hk$.*

Доказ. *Потребно је записати матрице H_1, H_2 на следећи начин:*

$$H_1 = \begin{bmatrix} P & Q \\ R & S \end{bmatrix} \; H_2 = \begin{bmatrix} K & L \\ M & N \end{bmatrix}$$

С обзиром на то да је $H_1 H_1^\tau = 4hI$ и $H_2 H_2^\tau = 4kI$, непосредно се показује да важи:

$$PP^\tau + QQ^\tau = RR^\tau + SS^\tau = 2hI, \; PR^\tau + QS^\tau = RP^\tau + SQ^\tau = 0$$

$$KK^\tau + LL^\tau = MM^\tau + NN^\tau = 2kl, \; KM^\tau + LN^\tau = MK^\tau + NL^\tau = 0$$

Тада је матрица H Адамарова матрица реда $8hk$ облика:

$$H = \begin{bmatrix} \frac{1}{2}(P+Q) \otimes K + \frac{1}{2}(P-Q) \otimes M & \frac{1}{2}(P+Q) \otimes L + \frac{1}{2}(P-Q) \otimes N \\ \\ \frac{1}{2}(R+S) \otimes K + \frac{1}{2}(R-S) \otimes M & \frac{1}{2}(R+S) \otimes L + \frac{1}{2}(R-S) \otimes N \end{bmatrix} \tag{3.24}$$

Што се непосредно проверава израчунавањем производа HH^τ.

Пример 3.8.1. *Конструше се Адамарова матрица реда $H_{96}, (8 \cdot 4 \cdot 3)$, коришћењем Силвестерове матрице $H_{16}(4 \cdot 4)$ и матрице $H_{12}(4 \cdot 3)$ добијене Палеј II конструкцијом.*
Силвестерова матрица:

$$H_{16} = \begin{bmatrix}
1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\
1 & -1 & 1 & -1 & 1 & -1 & 1 & -1 & 1 & -1 & 1 & -1 & 1 & -1 & 1 & -1 \\
1 & 1 & -1 & -1 & 1 & 1 & -1 & -1 & 1 & 1 & -1 & -1 & 1 & 1 & -1 & -1 \\
1 & -1 & -1 & 1 & 1 & -1 & -1 & 1 & 1 & -1 & -1 & 1 & 1 & -1 & -1 & 1 \\
1 & 1 & 1 & 1 & -1 & -1 & -1 & -1 & 1 & 1 & 1 & 1 & -1 & -1 & -1 & -1 \\
1 & -1 & 1 & -1 & -1 & 1 & -1 & 1 & 1 & -1 & 1 & -1 & -1 & 1 & -1 & 1 \\
1 & 1 & -1 & -1 & -1 & -1 & 1 & 1 & 1 & 1 & -1 & -1 & -1 & -1 & 1 & 1 \\
1 & -1 & -1 & 1 & -1 & 1 & 1 & -1 & 1 & -1 & -1 & 1 & -1 & 1 & 1 & -1 \\
1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & -1 & -1 & -1 & -1 & -1 & -1 & -1 & -1 \\
1 & -1 & 1 & -1 & 1 & -1 & 1 & -1 & -1 & 1 & -1 & 1 & -1 & 1 & -1 & 1 \\
1 & 1 & -1 & -1 & 1 & 1 & -1 & -1 & -1 & -1 & 1 & 1 & -1 & -1 & 1 & 1 \\
1 & -1 & -1 & 1 & 1 & -1 & -1 & 1 & -1 & 1 & 1 & -1 & -1 & 1 & 1 & -1 \\
1 & 1 & 1 & 1 & -1 & -1 & -1 & -1 & -1 & -1 & -1 & -1 & 1 & 1 & 1 & 1 \\
1 & -1 & 1 & -1 & -1 & 1 & -1 & 1 & -1 & 1 & -1 & 1 & 1 & -1 & 1 & -1 \\
1 & 1 & -1 & -1 & -1 & -1 & 1 & 1 & -1 & -1 & 1 & 1 & 1 & 1 & -1 & -1 \\
1 & -1 & -1 & 1 & -1 & 1 & 1 & -1 & -1 & 1 & 1 & -1 & 1 & -1 & -1 & 1
\end{bmatrix}$$

Палеј II матрица:

$$H_{12} = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & -1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & -1 & -1 & 1 & 1 & -1 & 1 & -1 & -1 & 1 \\ 1 & 1 & 1 & 1 & -1 & -1 & 1 & 1 & -1 & 1 & -1 & -1 \\ 1 & -1 & 1 & 1 & 1 & -1 & 1 & -1 & 1 & -1 & 1 & -1 \\ 1 & -1 & -1 & 1 & 1 & 1 & 1 & -1 & -1 & 1 & -1 & 1 \\ 1 & 1 & -1 & -1 & 1 & 1 & 1 & 1 & -1 & -1 & 1 & -1 \\ -1 & 1 & 1 & 1 & 1 & 1 & -1 & -1 & -1 & -1 & -1 & -1 \\ 1 & -1 & 1 & -1 & -1 & 1 & -1 & -1 & -1 & 1 & 1 & -1 \\ 1 & 1 & -1 & 1 & -1 & -1 & -1 & -1 & -1 & -1 & 1 & 1 \\ 1 & -1 & 1 & -1 & 1 & -1 & -1 & 1 & -1 & -1 & -1 & 1 \\ 1 & -1 & -1 & 1 & -1 & 1 & -1 & 1 & 1 & -1 & -1 & -1 \\ 1 & 1 & -1 & -1 & 1 & -1 & -1 & -1 & 1 & 1 & -1 & -1 \end{bmatrix}$$

Партиционисањем матрице H_{16} *на матрице* P, Q, R, S *добија се:*

$$P = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & -1 & 1 & -1 & 1 & -1 & 1 & -1 \\ 1 & 1 & -1 & -1 & 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 & 1 & -1 & -1 & 1 \\ 1 & 1 & 1 & 1 & -1 & -1 & -1 & -1 \\ 1 & -1 & 1 & -1 & -1 & 1 & -1 & 1 \\ 1 & 1 & -1 & -1 & -1 & -1 & 1 & 1 \\ 1 & -1 & -1 & 1 & -1 & 1 & 1 & -1 \end{bmatrix} \quad Q = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & -1 & 1 & -1 & 1 & -1 & 1 & -1 \\ 1 & 1 & -1 & -1 & 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 & 1 & -1 & -1 & 1 \\ 1 & 1 & 1 & 1 & -1 & -1 & -1 & -1 \\ 1 & -1 & 1 & -1 & -1 & 1 & -1 & 1 \\ 1 & 1 & -1 & -1 & -1 & -1 & 1 & 1 \\ 1 & -1 & -1 & 1 & -1 & 1 & 1 & -1 \end{bmatrix}$$

$$R = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & -1 & 1 & -1 & 1 & -1 & 1 & -1 \\ 1 & 1 & -1 & -1 & 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 & 1 & -1 & -1 & 1 \\ 1 & 1 & 1 & 1 & -1 & -1 & -1 & -1 \\ 1 & -1 & 1 & -1 & -1 & 1 & -1 & 1 \\ 1 & 1 & -1 & -1 & -1 & -1 & 1 & 1 \\ 1 & -1 & -1 & 1 & -1 & 1 & 1 & -1 \end{bmatrix} \quad S = \begin{bmatrix} -1 & -1 & -1 & -1 & -1 & -1 & -1 & -1 \\ -1 & 1 & -1 & 1 & -1 & 1 & -1 & 1 \\ -1 & -1 & 1 & 1 & -1 & -1 & 1 & 1 \\ -1 & 1 & 1 & -1 & -1 & 1 & 1 & -1 \\ -1 & -1 & -1 & -1 & 1 & 1 & 1 & 1 \\ -1 & 1 & -1 & 1 & 1 & -1 & 1 & -1 \\ -1 & -1 & 1 & 1 & 1 & 1 & -1 & -1 \\ -1 & 1 & 1 & -1 & 1 & -1 & -1 & 1 \end{bmatrix}$$

Партиционисањем матрице H_{12} *на матрице* K, L, M, N *добија се:*

$$K = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & -1 & -1 & 1 \\ 1 & 1 & 1 & 1 & -1 & -1 \\ 1 & -1 & 1 & 1 & 1 & -1 \\ 1 & -1 & -1 & 1 & 1 & 1 \\ 1 & 1 & -1 & -1 & 1 & 1 \end{bmatrix} \quad L = \begin{bmatrix} -1 & 1 & 1 & 1 & 1 & 1 \\ 1 & -1 & 1 & -1 & -1 & 1 \\ 1 & 1 & -1 & 1 & -1 & -1 \\ 1 & -1 & 1 & -1 & 1 & -1 \\ 1 & -1 & -1 & 1 & -1 & 1 \\ 1 & 1 & -1 & -1 & 1 & -1 \end{bmatrix}$$

$$M = \begin{bmatrix} -1 & 1 & 1 & 1 & 1 & 1 \\ 1 & -1 & 1 & -1 & -1 & 1 \\ 1 & 1 & -1 & 1 & -1 & -1 \\ 1 & -1 & 1 & -1 & 1 & -1 \\ 1 & -1 & -1 & 1 & -1 & 1 \\ 1 & 1 & -1 & -1 & 1 & -1 \end{bmatrix} \quad N = \begin{bmatrix} -1 & -1 & -1 & -1 & -1 & -1 \\ -1 & -1 & -1 & 1 & 1 & -1 \\ -1 & -1 & -1 & -1 & 1 & 1 \\ -1 & 1 & -1 & -1 & -1 & 1 \\ -1 & 1 & 1 & -1 & -1 & -1 \\ -1 & -1 & 1 & 1 & -1 & -1 \end{bmatrix}$$

Добијени делови су:

$$\frac{1}{2}(P+Q)\otimes K+\frac{1}{2}(P-Q)\otimes M= \left[\text{[illegible]}\right]$$

$$\frac{1}{2}(P+Q)\otimes L+\frac{1}{2}(P-Q)\otimes N= \left[\text{[illegible]}\right]$$

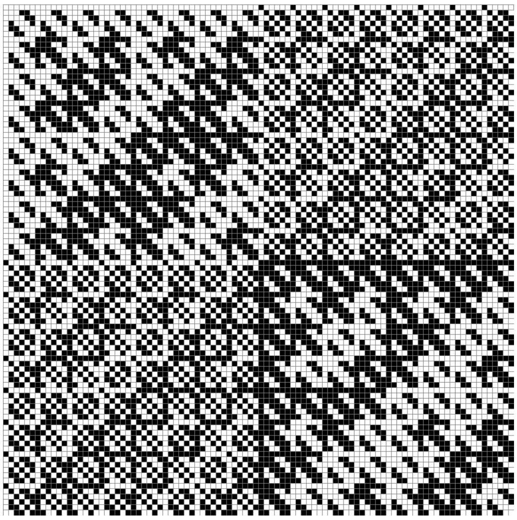
$$\frac{1}{2}(R+S)\otimes K+\frac{1}{2}(R-S)\otimes M= \left[\text{[illegible]}\right]$$

$$\frac{1}{2}(R+S)\otimes L+\frac{1}{2}(R-S)\otimes N= \left[\text{[illegible]}\right]$$

Након замене добијених партиција у (3.24), добија се Адамарова матрица реда 96.

$$[illegible]$$

На следећој слици приказана је графичка репрезентација добијене матрице.



Слика 3.9: Адамарова матрица реда 96 добијена Агајан/Сарукханјан конструкцијом.

Псеудокод конструкције.

Алгоритам 8 Агајан/Сарукханјан конструкција

procedure AGAYANSARUKHANYAN(H_1, H_2)
  ▷ Улаз: H_1, H_2 - Адамарове матрице реда редом $4h, 4k$.
  $P, Q, R, S \leftarrow partition(H_1)$ ▷ Одређују се партиције матрице H_1.
  $K, L, M, N \leftarrow partition(H_2)$ ▷ Одређују се партиције матрице H_2.
  ▷ Одређивање градивиних матрица финалне Адамарове матрице.
  $T_1 \leftarrow \frac{1}{2}(P+Q) \otimes K + \frac{1}{2}(P-Q) \otimes M$
  $T_2 \leftarrow \frac{1}{2}(P+Q) \otimes L + \frac{1}{2}(P-Q) \otimes N$
  $T_3 \leftarrow \frac{1}{2}(R+S) \otimes K + \frac{1}{2}(R-S) \otimes M$
  $T_4 \leftarrow \frac{1}{2}(R+S) \otimes L + \frac{1}{2}(R-S) \otimes N$
  $H \leftarrow \begin{bmatrix} T_1 & T_2 \\ T_3 & T_4 \end{bmatrix}$
  ▷ Излаз: Адамарова матрица реда $8hk$
  return H
end procedure

3.9 Купер/Волис конструкција

Купер/Волис (Cooper/Wallis) конструкција представља комбинацију Вилијамсонове и Геталс/Сајдл конструкције.

Теорема 3.9.1. *Нека су X, Y, Z, W матрице реда n добијене Вилијамсоновом конструкцијом, и T_1, T_2, T_3, T_4 цикличне матрице реда m добијене од четири T секвенце дужине m. Заменом матрица*

$$A = T_1 \otimes X + T_2 \otimes Y + T_3 \otimes Z + T_4 \otimes W \tag{3.25}$$

$$B = T_1 \otimes -Y + T_2 \otimes X + T_3 \otimes W + T_4 \otimes -Z \tag{3.26}$$

$$C = T_1 \otimes -Z + T_2 \otimes -W + T_3 \otimes X + T_4 \otimes Y \tag{3.27}$$

$$D = T_1 \otimes -W + T_2 \otimes Z + T_3 \otimes -Y + T_4 \otimes -X \tag{3.28}$$

у Геталс/Сајдл матрицу (3.19) тј. конкретно:

$$H = \begin{bmatrix} A & BR & CR & DR \\ -BR & A & D^{\tau}R & -C^{\tau} \\ -CR & -D^{\tau}R & A & B^{\tau}R \\ -DR & C^{\tau}R & -B^{\tau}R & A \end{bmatrix}$$

добија се Адамарова матрица реда $4nm$.

За доказ видети [10].

Пример 3.9.1. *Адамарова матрица реда 336 добијена методом Купер/Волис. Кори-*

сти се матрица реда 12 добијена Вилијамсоновом конструкцијом из примера (3.3.1)

$$H_{12} = X = Y = Z = W = \begin{bmatrix} 1 & 1 & 1 & 1 & -1 & -1 & 1 & -1 & -1 & 1 & -1 & -1 \\ 1 & 1 & 1 & -1 & 1 & -1 & -1 & 1 & -1 & -1 & 1 & -1 \\ 1 & 1 & 1 & -1 & -1 & 1 & -1 & -1 & 1 & -1 & -1 & 1 \\ -1 & 1 & 1 & 1 & 1 & 1 & -1 & 1 & 1 & 1 & -1 & -1 \\ 1 & -1 & 1 & 1 & 1 & 1 & 1 & -1 & 1 & -1 & 1 & -1 \\ 1 & 1 & -1 & 1 & 1 & 1 & 1 & 1 & -1 & -1 & -1 & 1 \\ -1 & 1 & 1 & 1 & -1 & -1 & 1 & 1 & 1 & -1 & 1 & 1 \\ 1 & -1 & 1 & -1 & 1 & -1 & 1 & 1 & 1 & 1 & -1 & 1 \\ 1 & 1 & -1 & -1 & -1 & 1 & 1 & 1 & 1 & 1 & 1 & -1 \\ -1 & 1 & 1 & -1 & 1 & 1 & 1 & -1 & -1 & 1 & 1 & 1 \\ 1 & -1 & 1 & 1 & -1 & 1 & -1 & 1 & -1 & 1 & 1 & 1 \\ 1 & 1 & -1 & 1 & 1 & -1 & -1 & -1 & 1 & 1 & 1 & 1 \end{bmatrix}$$

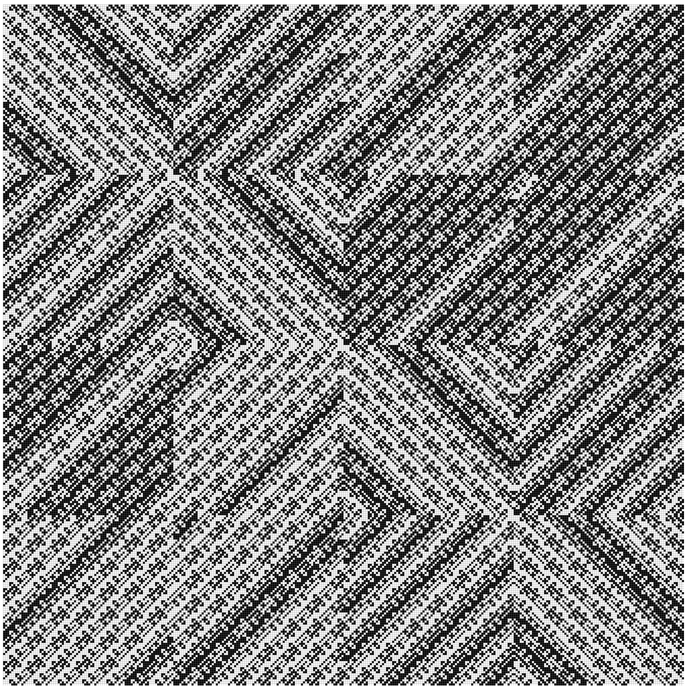
и 4 T матрице добијене из примера (3.7.1) Геталс/Сајдл конструкције:

$$T_1 = \begin{bmatrix} 1 & 1 & -1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & -1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & -1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & -1 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 & -1 \\ -1 & 0 & 0 & 0 & 0 & 1 & 1 \\ 1 & -1 & 0 & 0 & 0 & 0 & 1 \end{bmatrix},$$

$$T_2 = \begin{bmatrix} 0 & 0 & 0 & -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & -1 \\ -1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & -1 & 0 & 0 & 0 & 0 \end{bmatrix}$$

$$T_3 = \begin{bmatrix} 0 & 0 & 0 & 0 & 1 & 0 & 1 \\ 1 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 1 \\ 1 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 1 & 0 \end{bmatrix}, T_4 = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 \end{bmatrix}$$

Заменом матрица $X, Y, Z, W, T_1, T_2, T_3, T_4$ у једнакости (3.25-3.28) добијају се матрице A,B,C,D, након чега се заменом матрица A,B,C,D у Геталс/Сајдл матрицу добија Адамарова матрица реда 336. На следећој слици приказана је графичка репрезентација добијене матрице. Међурачун и коначна матрица су изостављене због превеликих димензија.



Слика 3.10: Адамарова матрица реда 336 добијена Купер/Волис конструкцијом.

Алгоритам 9 Купер/Волис конструкција

procedure CooperWallis$(X, Y, Z, W, T_1, T_2, T_3, T_4)$

$\triangleright$ Улаз: X,Y,Z,W- Адамарове матрице добијена Вилијамсоновом конструкцијом.

$\triangleright$ Улаз: $T_1 \dots T_4$ - Т матрице.

$A \leftarrow T_1 \otimes X + T_2 \otimes Y + T_3 \otimes Z + T_4 \otimes$

$B \leftarrow T_1 \otimes -Y + T_2 \otimes X + T_3 \otimes W + T_4 \otimes -Z$

$C \leftarrow T_1 \otimes -Z + T_2 \otimes -W + T_3 \otimes X + T_4 \otimes Y$

$D \leftarrow T_1 \otimes -W + T_2 \otimes Z + T_3 \otimes -Y + T_4 \otimes -X$

$n \leftarrow dim(A)$

$\triangleright$ Генерисање матрице R

$$R[r(i,j)] \leftarrow [r(i,j)] \begin{cases} 1, & \text{ако } i + j = n + 1 \\ 0, & \text{у супротном} \end{cases}$$

$$H = \begin{bmatrix} A & BR & CR & DR \\ -BR & A & D^{\tau}R & -C^{\tau} \\ -CR & -D^{\tau}R & A & B^{\tau}R \\ -DR & C^{\tau}R & -B^{\tau}R & A \end{bmatrix}$$

return H

end procedure

3.10 Јангова конструкција - Адамарове Коцке

Јангова (Yang) конструкција је везана за такозване вишедимензионе Адамарове матрице које представљају релативно ново проширење теорије Адамарових матрица и тренутно постоји веома мали број аутора који се истим бави. С обзиром на комплекност имплементација и саме теорије у раду се приказују само конструкције Адамарових коцака, односно рад са тродимензионим низовима.

Дефиниција 3.10.1. *Тродимензиони Адамаров низ се назива Адамаровом коцком ако се низ $n \times n \times n$ састоји од елемената из скупа $\{-1, 1\}$ где је сваки планаран пресек Адамарова матрица димензија $n \times n$. Планаран пресек се добија тако што се фиксира једна од димензија.*

Теорема 3.10.1. *Нека је H Адамарова матрица реда n. Тродимензиони низ*

$$C = [c_{ijk}] = [h_{ij} \cdot h_{ik} \cdot h_{jk}], i, j, k \in \{1, 2, \cdots, n\} \tag{3.29}$$

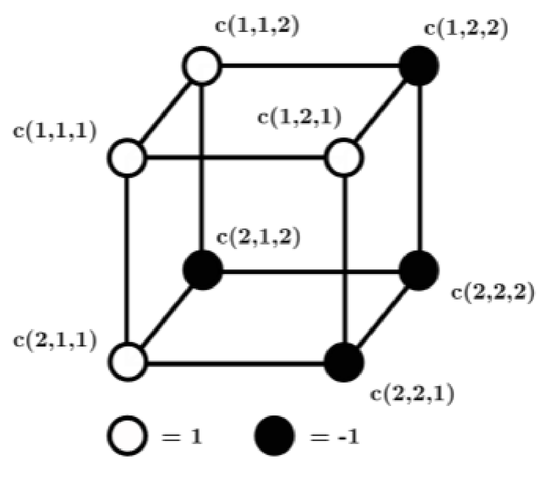
је Адамарова коцка реда n.

За доказ видети [11].

Пример 3.10.1. *Адамарова коцка реда $n = 4$ добијена конструкцијом Yang.*

Нека је $H_4 = \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}$ Адамарова матрица добијена конструкцијом Силвестер. Коришћењем претходно дефинисане теорме лако се добијају сви елементи Адамарове коцке:

$$\begin{aligned}
c_{111} &= h_{11} \cdot h_{11} \cdot h_{11} = 1 \cdot 1 \cdot 1 = 1 \\
c_{112} &= h_{11} \cdot h_{12} \cdot h_{12} = 1 \cdot 1 \cdot 1 = 1 \\
c_{121} &= h_{12} \cdot h_{11} \cdot h_{21} = 1 \cdot 1 \cdot 1 = 1 \\
c_{122} &= h_{12} \cdot h_{12} \cdot h_{22} = 1 \cdot 1 \cdot -1 = -1 \\
c_{211} &= h_{21} \cdot h_{21} \cdot h_{11} = 1 \cdot 1 \cdot 1 = 1 \\
c_{212} &= h_{21} \cdot h_{22} \cdot h_{12} = 1 \cdot -1 \cdot 1 = -1 \\
c_{221} &= h_{22} \cdot h_{21} \cdot h_{21} = -1 \cdot 1 \cdot 1 = -1 \\
c_{222} &= h_{22} \cdot h_{22} \cdot h_{22} = -1 \cdot -1 \cdot -1 = -1
\end{aligned}$$



Слика 3.11: Адамарова коцка реда 4 добијена Јанговом конструкцијом.

Псеудокод који описује приказану конструкцију:

Алгоритам 10 Јангова конструкција

```
procedure Yang(H)
    ▷ Улаз: H - Адамарова матрица добијена Силвестеровом конструкцијом.
    Y[y(i, j, k)] = 0
    n = dim(H)
    ▷ Генерација Адамарове коцке.
    for i ← 1 to n do
        for j ← 1 to n do
            for k ← 1 to n do
                Y[y(i, j, k)] ← H[h(i, j)] ∗ H[h(i, k)] ∗ H[h(j, k)]
            end for
        end for
    end for
    return Y
end procedure
```

Глава 4

Програмска реализација и резултати

За потребе овог рада, у програмском језику Пајтон (Python) написана је библиотка под називом **hm** која садржи велики број помоћних метода за генерисање Адамарових матрица и саме имплементације конструкција које су описане у раду. Имплементације алгоритама помоћу којих се конструишу различите Адамарове матрице описане су у додатку А.
Већина конструкција које су описане у раду захтевају постојање одређених улазних параметара као што су низови и матрице које задовољавају особине описане конструкцијом. Потребама овог рада послужила је база матрица Х. Коуковинуса [6].

4.1 Инсталација потребних пакета

Инсталација захтева неколико пакета у зависности од оперативног система који се користи. Подразумева се поседовање верзије Пајтон интерпретера која је већа или једнака 2.7. Препоручује се не оптерећивање локалне инсталације Пајтона инсталирањем додатних пакета од којих библиотека **hm** зависи него да се користи виртуелно Пајтон окружење.
Да бисте на оперативном систему Linux или OSX инсталирали Пајтон виртуелно окружење и додатне пакете од којих библиотека **hm** зависи потребно је извршити следеће кораке.

- Инсталирање Пајтоновог менаџера пакета: *apt-get install python-pip*
- Инсталирање системске библиотеке за креирање виртуелног Пајтон окружења: *apt-get install python-virtualenv*
- Креирање виртуелног Пајтон окружења у корисничком home директоријуму *virtualenv /home/user/hm*
- Активација виртуелног Пајтон окружења се постиже командом *source /home/user/hm/bin/activate*
- Потребно је инсталирати пакете од којих библиотека **hm** зависи. То се постиже командом *pip install -r requirements.txt* из **hm** директоријума.

4.2 Примери генерисања матрица

Библиотека није писана као апликација која производи Адамарове матрице произвољно задатог реда, него као помоћни скуп алата и конструкција које омогућавају кориснику да користи већ имплементиране конструкције и комбинује исте у циљу генерисања већег броја Адамарових матрица различитих редова. У следећим примерима свака линија означена са In представља линију написану од стране корисника односно улазне наредбе док су линије означена са Out линије са резултатима које штампа Пајтон интерпретер. Интерпретер штампа скраћену верзију матрица великих димензија и користи тачке да означи изостављени садржај.

Пример 4.2.1. *Силвестерова матрица реда 16 се може генерисати на следећи начин.*

```
In [2]: from methods.sylvester import sylvester
In [3]: s16 = sylvester(16)
In [4]: s16
Out[4]:
array([[ 1,  1,  1,  1,  1,  1,  1,  1,  1,  1,  1,  1,  1,  1,  1,  1],
       [ 1, -1,  1, -1,  1, -1,  1, -1,  1, -1,  1, -1,  1, -1,  1, -1],
       [ 1,  1, -1, -1,  1,  1, -1, -1,  1,  1, -1, -1,  1,  1, -1, -1],
       [ 1, -1, -1,  1,  1, -1, -1,  1,  1, -1, -1,  1,  1, -1, -1,  1],
       [ 1,  1,  1,  1, -1, -1, -1, -1,  1,  1,  1,  1, -1, -1, -1, -1],
       [ 1, -1,  1, -1, -1,  1, -1,  1,  1, -1,  1, -1, -1,  1, -1,  1],
       [ 1,  1, -1, -1, -1, -1,  1,  1,  1,  1, -1, -1, -1, -1,  1,  1],
       [ 1, -1, -1,  1, -1,  1,  1, -1,  1, -1, -1,  1, -1,  1,  1, -1],
       [ 1,  1,  1,  1,  1,  1,  1,  1, -1, -1, -1, -1, -1, -1, -1, -1],
       [ 1, -1,  1, -1,  1, -1,  1, -1, -1,  1, -1,  1, -1,  1, -1,  1],
       [ 1,  1, -1, -1,  1,  1, -1, -1, -1, -1,  1,  1, -1, -1,  1,  1],
       [ 1, -1, -1,  1,  1, -1, -1,  1, -1,  1,  1, -1, -1,  1,  1, -1],
       [ 1,  1,  1,  1, -1, -1, -1, -1, -1, -1, -1, -1,  1,  1,  1,  1],
       [ 1, -1,  1, -1, -1,  1, -1,  1, -1,  1, -1,  1,  1, -1,  1, -1],
       [ 1,  1, -1, -1, -1, -1,  1,  1, -1, -1,  1,  1,  1,  1, -1, -1],
       [ 1, -1, -1,  1, -1,  1,  1, -1, -1,  1,  1, -1,  1, -1, -1,  1]])
```

Пример 4.2.2. *Палеј I и Палеј II матрица реда редом 8 и 12.*

```
In [24]: from methods.paley import Paley
In [25]: p1 = Paley(7)
In [26]: p2 = Paley(5)
In [27]: # Са ._type се проверава тип матрице.
In [28]: p1._type
Out[28]: 'PaleyI'
In [29]: p2._type
Out[29]: 'PaleyII'
In [30]: p1.matrix
Out[30]:
array([[ 1, -1, -1, -1, -1, -1, -1, -1],
       [ 1,  1, -1, -1,  1, -1,  1,  1],
       [ 1,  1,  1, -1, -1,  1, -1,  1],
       [ 1,  1,  1,  1, -1, -1,  1, -1],
       [ 1, -1,  1,  1,  1, -1, -1,  1],
       [ 1,  1, -1,  1,  1,  1, -1, -1],
       [ 1, -1,  1, -1,  1,  1,  1, -1],
       [ 1, -1, -1,  1, -1,  1,  1,  1]])
In [31]: p2.matrix
Out[31]:
array([[ 1,  1,  1,  1,  1,  1, -1,  1,  1,  1,  1,  1],
       [ 1,  1,  1, -1, -1,  1,  1, -1,  1, -1, -1,  1],
       [ 1,  1,  1,  1, -1, -1,  1,  1, -1,  1, -1, -1],
       [ 1, -1,  1,  1,  1, -1,  1, -1,  1, -1,  1, -1],
       [ 1, -1, -1,  1,  1,  1,  1, -1, -1,  1, -1,  1],
       [ 1,  1, -1, -1,  1,  1,  1,  1, -1, -1,  1, -1],
       [-1,  1,  1,  1,  1,  1, -1, -1, -1, -1, -1, -1],
       [ 1, -1,  1, -1, -1,  1, -1, -1, -1,  1,  1, -1],
       [ 1,  1, -1,  1, -1, -1, -1, -1, -1, -1,  1,  1],
       [ 1, -1,  1, -1,  1, -1, -1,  1, -1, -1, -1,  1],
       [ 1, -1, -1,  1, -1,  1, -1,  1,  1, -1, -1, -1],
       [ 1,  1, -1, -1,  1, -1, -1, -1,  1,  1, -1, -1]])
```

Пример 4.2.3. *Вилијамсонова матрица реда* 76.

```
In [4]: from methods.williams import williams
In [5]: from utils import read_abcd_files_with_specified_order
# читамо матрице реда 19 да бисмо генерисали Вилијамсонову матрицу реда 76.
In [7]: A,B,C,D = read_abcd_files_with_specified_order(19, 'data/good_matrices')
In [8]: w = williams(A,B,C,D)
In [9]: w
Out[9]:
matrix([[ 1,  1, -1, ..., -1,  1,  1],
        [ 1,  1,  1, ..., -1, -1,  1],
        [-1,  1,  1, ..., -1, -1, -1],
        ...,
        [ 1,  1,  1, ...,  1,  1, -1],
        [-1,  1,  1, ...,  1,  1,  1],
        [-1, -1,  1, ..., -1,  1,  1]])

In [10]: # Ако желите да генеришете слику
In [11]: from image.image import save
In [12]: save(w, 'out/w76')
```

Пример 4.2.4. *Баумерт/Хол матрица реда* 156.

```
In [4]: from methods import baumert_hall
In [5]: from utils import read_abcd_files_with_specified_order
# читамо матрице реда 13 да бисмо генерисали Баумерт/Хол матрицу реда 156.
In [7]: A,B,C,D = read_abcd_files_with_specified_order(13, 'data/good_matrices')
In [8]: bh = baumert_hall(A,B,C,D)
In [9]: bh
Out[9]:
matrix([[ 1,  1,  1, ...,  1,  1,  1],
        [ 1,  1,  1, ..., -1,  1,  1],
        [ 1,  1,  1, ..., -1, -1,  1],
        ...,
        [ 1, -1,  1, ..., -1, -1, -1],
        [ 1,  1, -1, ..., -1, -1, -1],
        [-1,  1,  1, ..., -1, -1, -1]])
```

Пример 4.2.5. *Итова матрица реда* 68.

```
In [4]: from methods import ito
In [5]: from utils import read_abcd_files_with_specified_order
# читамо матрице реда 17 да бисмо генерисали Итову матрицу реда 68.
In [7]: A,B,C,D = read_abcd_files_with_specified_order(17, 'data/good_matrices')
In [8]: it = ito(A,B,C,D)
In [9]: it
Out[9]:
matrix([[ 1, -1, -1, ..., -1, -1, -1],
        [-1,  1, -1, ...,  1, -1, -1],
        [-1, -1,  1, ...,  1,  1, -1],
        ...,
        [-1,  1,  1, ..., -1,  1,  1],
        [-1, -1,  1, ...,  1, -1,  1],
        [-1, -1, -1, ...,  1,  1, -1]])
```

Пример 4.2.6. *Ехлихова матрица реда* 36.

```
In [2]: from methods import echlich
In [3]: from methods.paley import Paley
In [4]: p5 = Paley(5)
In [5]: j5 = p5.jacobstal
In [6]: p7 = Paley(7)
In [7]: j7 = p7.jacobstal
In [8]: ech = echlich(j5,j7)
In [9]: from utils import is_hadamard
# са is_hadamard(H) увек можете проверити да ли је добијена матрица заиста Адамарова матрица.
In [10]: is_hadamard(ech)
Out[10]: True
In [11]: ech
Out[11]:
array([[ 1,  1,  1, ...,  1,  1,  1],
       [ 1, -1, -1, ...,  1,  1,  1],
       [-1,  1, -1, ..., -1,  1,  1],
       ...,
       [ 1, -1,  1, ..., -1, -1,  1],
       [-1,  1, -1, ...,  1, -1,  1],
       [-1, -1,  1, ..., -1,  1,  1]])
```

Пример 4.2.7. *Геталс/Сајдл матрица реда* 28.

```
In [1]: from methods import goethal_seidel
In [2]: from utils import read_t_files
In [3]: t1,t2,t3,t4 = read_t_files('data/t-arrays/')
In [4]: gs = goethal_seidel(t1,t2,t3,t4)
In [5]: gs
Out[6]:
matrix([[ 1,  1, -1, ...,  1, -1, -1],
        [ 1,  1,  1, ..., -1, -1, -1],
        [ 1,  1,  1, ..., -1, -1,  1],
        [ 1,  1,  1, ..., -1,  1, -1],
        ...,
        [-1,  1,  1, ..., 1,  1, -1],
        [ 1,  1,  1, ..., 1,  1,  1],
        [ 1,  1, -1, ...,  1,  1,  1]])
```

Пример 4.2.8. *Агајан/Сарукханјан матрица реда* 40.

```
In [1]: from methods import ito, sylvester, agayan_sarukhanyan
In [2]: from utils import read_abcd_files_with_specified_order
In [3]: A,B,C,D = read_abcd_files_with_specified_order(5, 'data/good_matrices')
In [4]: it = ito(A,B,C,D)
In [5]: s = sylvester(4)
In [6]: ags = agayan_sarukhanyan(it,s)
In [7]: ags
Out[8]:
matrix([[ 1.,  1., -1., ...,  1., -1., -1.],
        [ 1., -1., -1., ..., -1., -1.,  1.],
        [-1., -1.,  1., ...,  1.,  1.,  1.],
        ...,
        [ 1., -1.,  1., ...,  1.,  1., -1.],
        [-1., -1.,  1., ...,  1., -1., -1.],
        [-1.,  1.,  1., ..., -1., -1.,  1.]])
```

Пример 4.2.9. *Купер/Волис матрица реда* 560.

```
In [1]: from methods import williams, cooper_wallis
In [2]: from utils import read_t_files, read_abcd_files_with_specified_order
In [3]: t1,t2,t3,t4 = read_t_files('data/t-arrays/')
In [4]: A,B,C,D = read_abcd_files_with_specified_order(5, 'data/good_matrices')
In [5]: X=Y=Z=W=williams(A,B,C,D)
In [7]: cw = cooper_wallis(X,Y,Z,W, t1,t2,t3,t4)
In [8]: cw
Out[9]:
matrix([[ 1, -1, -1, ...,  1,  1, -1],
        [-1,  1, -1, ...,  1, -1,  1],
        [-1, -1,  1, ..., -1,  1,  1],
        ...,
        [-1, -1,  1, ...,  1, -1, -1],
        [-1,  1, -1, ..., -1,  1, -1],
        [ 1, -1, -1, ..., -1, -1,  1]])
```

4.3 Резултати

Коришћењем библиотеке **hm** генерисано је 249 различитих матрица редова мањих од 1000. Добијени редови представљени су у табелама приказаним на страницама које следе. Из табела се јасно види да конструкције описане у овом раду не покривају све редове. Тренуно Адамарова хипотеза је и даље отворена и за сада **не постоји конструкција за редове 668, 716, 892.**

4k	Агајан/Сарукханјан	Баумерт/Хол	Купер/Волис	Ито	Палеј I	Палеј II	Силвестер	Вилијамс
4					4		4	
8	8				8		8	
12				12	12	12		12
16	16						16	
20				20	20			20
24	24				24			
28				28		28		28
32	32				32		32	
36		36		36		36		36
40	40							
44				44	44			44
48	48				48			
52				52				52
56	56							
60		60		60	60	60		60
64	64						64	
68				68	68			68
72	72				72			
76				76		76		76
80	80				80			
84		84		84	84	84		84
88	88							
92				92				92
96	96							
100				100				100
104	104				104			
108		108		108	108	108		108
112	112							
116				116				116
120	120							
124				124		124		124
128	128				128		128	
132		132		132	132			132
136	136							
140					140			
144	144							
148				148		148		148
152	152				152			
156		156						
160	160							
164				164	164			164
168	168				168			
172				172				172
176	176							
180		180		180	180	180		180
184	184							
188								
192	192				192			
196				196		196		196
200	200				200			

4k	Агајан/Сарукханјан	Баумерт/Хол	Купер/Волис	Ито	Палеј I	Палеј II	Силвестер	Вилијамс
204		204		204		204		204
208	208							
212					212			
216	216							
220				220		220		220
224	224				224			
228		228		228	228	228		228
232	232							
236								
240	240				240			
244				244				244
248	248							
252		252		252	252			252
256	256						256	
260								
264	264				264			
268								
272	272				272			
276		276				276		
280	280							
284					284			
288	288							
292								
296	296							
300		300				300		
304	304							
308					308			
312	312				312			
316						316		
320	320							
324		324						
328	328							
332					332			
336	336		336					
340								
344	344							
348		348			348	348		
352	352							
356								
360	360				360			
364						364		
368	368				368			
372		372						
376								
380					380			
384	384				384			
388						388		
392	392							
396		396				396		
400	400							

4k	Агајан/Сарукханјан	Баумерт/Хол	Купер/Волис	Ито	Палеј I	Палеј II	Силвестер	Вилијамс
404								
408	408							
412								
416	416							
420					420			
424								
428								
432	432				432			
436								
440	440				440			
444		444			444			
448	448							
452								
456	456							
460						460		
464	464				464			
468					468	468		
472								
476								
480	480				480			
484						484		
488	488				488			
492		492			492			
496	496							
500					500			
504	504				504			
508								
512	512						512	
516		516				516		
520	520							
524					524			
528	528							
532								
536								
540		540				540		
544	544							
548					548			
552	552							
556						556		
560			560					
564					564	564		
568								
572					572			
576	576							
580								
584								
588		588			588	588		
592	592							
596								
600	600				600			

4k	Агајан/Сарукханјан	Баумерт/Хол	Купер/Волис	Ито	Палеј I	Палеј II	Силвестер	Вилијамс
604								
608	608				608			
612		612						
616	616							
620					620			
624	624							
628						628		
632					632			
636						636		
640	640							
644					644			
648	648				648			
652								
656	656							
660		660			660			
664								
668								
672	672							
676						676		
680	680							
684		684			684			
688	688							
692					692			
696	696							
700						700		
704	704							
708						708		
712								
716								
720	720				720			
724								
728	728				728			
732		732						
736	736							
740					740			
744	744				744			
748						748		
752					752			
756		756						
760	760							
764								
768	768							
772								
776								
780						780		
784	784		784					
788					788			
792	792							
796						796		
800	800							

4k	Агајан/Сарукханјан	Баумерт/Хол	Купер/Волис	Ито	Палеј I	Палеј II	Силвестер	Вилијамс
804						804		
808								
812					812			
816	816							
820						820		
824					824			
828					828			
832	832							
836								
840	840				840			
844						844		
848								
852								
856								
860					860			
864	864				864			
868						868		
872								
876								
880	880							
884					884			
888	888				888			
892								
896	896							
900						900		
904								
908					908			
912	912				912			
916						916		
920	920				920			
924						924		
928	928							
932								
936	936							
940								
944								
948					948			
952	952							
956								
960	960							
964								
968	968				968			
972					972			
976	976							
980								
984	984				984			
988								
992	992				992			
996								

Глава 5

Закључак

У овом раду приказане су неке од познатијих конструкција Адамарових матрица које захтевају познавање различитих области математике. Наведене су све битне теореме које се користе у конструкцијама а неке од једноставнијих су и доказане. На конструкцијама где је доказ изостао, додате су референце ка радовима и уџбеницима које их садрже. Свака конструкција је детаљно објашњена са јасним примером. За сваку конструкцију присутан је и псеудокод конструкције као и графичка репрезентација матрице која се конструкцијом добија. У додатку А се налазе програмске имплементације приказаних конструкција у програмском језику Пајтон, док се у додатку Б могу пронаћи такозване добре матрице које се користе у примерима овог рада.

С обзиром на то да се Адамарове матрице заснивају на различитим областима математике за аутора је био прави изазов како ограничити број приказаних конструкција и задржати једноставност и јасност рада. Даљи правац овог рада би довео до конструкција комплексних Адамарових матрица и матрица развијеним над Абеловим групама.

Аутор сматра да је рад корисна референца за даље истраживање како Адамарових матрица тако и свих других градивних блокова комбинаторне теорије дизајна.

Додатак А

Имплементација

Сви алгоритми наведени у раду су имплементирани коришћењем програмског језика Пајтон. Коришћење програмског језика Пајтон омогућило је једноставност и читљивост кода самих конструкција, и у већини случајева имплементација је не много дужа од самог псеудокода конструкције. Потребама овог рада послужиле су и спољне библиотеке (numpy, scipy, matplotlib) које омогућавају једноставнији рад са матрицама и сликама. Додатне информације о самом језику Пајтон и библиотекама које се користе се могу пронаћи у [12],[13], [14], [15].

```python
"""Sylvester construction"""
import numpy
def toBinary(i):
    """
    :param i: Integer
    :return: Binary form of the provided integer
    """
    if i == 0:
        return '0'
    numeral = ''
    while i != 0:
        digit = i % 2
        numeral = str(digit) + numeral
        i = i // 2
    return numeral

def scalar_product(first, second, length):
    """
    :param first: First operand
    :param second: Second operand
    :param length: Maximum binary length
    :return: Scalar product of the first and second operand
    """
    first = toBinary(first).zfill(length)
    second = toBinary(second).zfill(length)
    digits_f = [int(d) for d in first]
    digits_s = [int(d) for d in second]
    if len(digits_f) > len(digits_s):
        digits_s.extend([0 for i in range(0, len(digits_f) - len(digits_s))])
    if len(digits_s) > len(digits_f):
        digits_f.extend([0 for i in range(0, len(digits_s) - len(digits_f))])
    and_list = list(numpy.logical_and(digits_f, digits_s))
    xor = False
    for i in range(0, len(and_list)):
        xor = xor ^ and_list[i]
    return xor

def sylvester(order):
    """
    :param order: Order to be generated
    :return: Sylvester Hadamard matrix of order :order
    """
    H = numpy.zeros(shape=(order, order), dtype=int)
```

```
    length = len(str(toBinary(order)))
    for i in range(0, order):
        for j in range(0, order):
            power = scalar_product(i, j, length)
            H[(i, j)] = (-1)**power
    return H
```

Имплементација А.1: Силвестерова конструкција

```
""" PaleyI and PaleyII construcion """
import numpy
def is_prime(number, limit):
    """
    :param number: Number to be tested
    :param limit: Max limit to check
    :return: True if number is prime else False
    """
    prime = [False] * 2 + [True] * (limit - 1)
    for n in range(int(limit ** 0.5 + 1.5)):
        if prime[n]:
            for i in range(n * n, limit + 1, n):
                prime[i] = False
    return number in [i for i, prime in enumerate(prime) if prime]

def legendre_jacobi_symbol(n, m):
    """
     :return: Legendre symbol for (n/m)
    """
    j = 1
    n %= m
    while n:
        t = 0
        while not int(n) & 1:
            n /= 2
            t += 1
        if t & 1 and m % 8 in (3, 5):
            j = -j
        if n % 4 == m % 4 == 3:
            j = -j
        n, m = m % n, n
    return j if m == 1 else 0

def ints_modulo_prime(p):
    """
    Helper class for representing x mod p
    """
    class IntegeruloPrime(object):
        def __init__(self, n):
            self.n = int(n) % IntegerModuloPrime.p
            self.field = IntegerModuloPrime
        def __sub__(self, other):
            return IntegerModuloPrime(self.n - other.n)
        def __int__(self):
            return self.n
    IntegerModuloPrime.p = p
    return IntegerModuloPrime


class Paley(object):
    def __init__(self, order=None):
        if order is None:
            self.matrix = matrix([], dtype=int)
        else:
            self.order = order
        if order % 4 == 3 and is_prime(order, order+1):
            self._type = 'PaleyI'
            self.matrix = self.construct_paley_i(order)
        elif order % 4 == 1 and is_prime(order, order+1):
            self._type = 'PaleyII'
            self.matrix = self.construct_paley_ii(order)
        else:
            raise BadPaleyOrder()
```

```
    def __str__(self):
        return str(self.matrix)

    def construct_paley_i(self, order):
        zp = ints_modulo_prime(order)
        unit = numpy.identity(order)
        mj = numpy.array([1]*order)
        mj_t = mj.transpose()
        q = numpy.zeros(shape=(order, order), dtype=int)
        s = numpy.zeros(shape=(order+1, order+1), dtype=int)
        p = numpy.zeros(shape=(order+1, order+1), dtype=int)
        for i in range(0, order):
            for j in range(0, order):
                q[(i, j)] = legendre_jacobi_symbol((zp(i) - zp(j)).n, order)
        self.jacobstal = q
        p[(0, 0)] = 1
        for i in range(1, order+1):
            p[(0, i)] = - mj[i-1]
        for i in range(1, order+1):
            p[(i, 0)] = mj_t[i-1]
        for i in range(1, order+1):
            for j in range(1, order+1):
                p[(i, j)] = q[(i-1, j-1)] + unit[(i-1, j-1)]
        return p

    def construct_paley_ii(self, order):
        zp = ints_modulo_prime(order)
        I = numpy.identity(order+1)
        mj = numpy.array([1]*order)
        mj_t = mj.transpose()
        q = numpy.zeros(shape=(order, order), dtype=int)
        s = numpy.zeros(shape=(order+1, order+1), dtype=int)
        p = numpy.zeros(shape=(2*(order+1), 2*(order+1)), dtype=int)
        for i in range(0, order):
            for j in range(0, order):
                q[(i, j)] = legendre_jacobi_symbol((zp(i) - zp(j)).n, order)
        # Jacobstal matrix
        self.jacobstal = q
        s[(0, 0)] = 0
        for i in range(1, order+1):
            s[(0, i)] = mj[i-1]
        for i in range(1, order+1):
            s[(i, 0)] = mj_t[i-1]
        for i in range(1, order+1):
            for j in range(1, order+1):
                s[(i, j)] = q[(i-1, j-1)]
        self.s = s
        tl = s + I
        tr = s - I
        bl = s - I
        br = -s - I
        for i in range(0, tl.shape[0]):
            for j in range(0, tl.shape[1]):
                p[(i, j)] = tl[(i, j)]
        for i in range(0, tr.shape[0]):
            for j in range(0, tr.shape[1]):
                p[(i, j + tr.shape[1])] = tr[(i, j)]
        for i in range(0, bl.shape[0]):
            for j in range(0, bl.shape[1]):
                p[(i+bl.shape[0], j)] = bl[(i, j)]
        for i in range(0, br.shape[0]):
            for j in range(0, br.shape[1]):
                p[(i+br.shape[0], j+br.shape[1])] = br[(i, j)]
        return p
```

Имплементација А.2: PaleyI и PaleyII конструкције

```
""" Williams construction """
import numpy
def williams(A,B,C,D):
    """
    :param A,B,C,D: Input matrices
    :return: William Hadamard matrix
```

```
    """
    n = A.shape[0]
    In = numpy.identity(n, dtype=int)
    AAT = A.dot(A.transpose())
    BBT = B.dot(B.transpose())
    CCT = C.dot(C.transpose())
    DDT = D.dot(D.transpose())
    if not numpy.array_equal(AAT+BBT+CCT+DDT, 4*n*In):
        return None
    H = numpy.bmat([[A,B,C,D],[-B,A,-D,C],[-C,D,A,-B],[-D,-C,B,A]])
    return H
```

Имплементација А.3: Вилијамсова конструкција

```
"""Baumert/Hall construction"""
import numpy
def baumert_hall(A, B, C, D):
    """
    :param A,B,C,D: Input matrices
    :return: Baumert/Hall Hadamard matrix
    """
    AAT = numpy.dot(A, A.transpose())
    BBT = numpy.dot(B, B.transpose())
    CCT = numpy.dot(C, C.transpose())
    DDT = numpy.dot(D, D.transpose())
    H = numpy.bmat([[A,A,A,B,-B,C,-C,-D,B,C,-D,-D],
                    [A,-A,B,-A,-B,-D,D,-C,-B,-D,-C,-C],
                    [A,-B,-A,A,-D,D,-B,B,-C,-D,C,-C],
                    [B,A,-A,-A,D,D,D,C,C,-B,-B,-C],
                    [B,-D,D,D,A,A,A,C,-C,B,-C,B],
                    [B,C,-D,D,A,-A,C,-A,-D,C,B,-B],
                    [D,-C,B,-B,A,-C,-A,A,B,C,D,-D],
                    [-C,-D,-C,-D,C,A,-A,-A,-D,B,-B,-B],
                    [D,-C,-B,-B,-B,C,C,-D,A,A,A,D],
                    [-D,-B,C,C,C,B,B,-D,A,-A,D,-A],
                    [C,-B,-C,C,D,-B,-D,-B,A,-D,-A,A],
                    [-C,-D,-D,C,-C,-B,B,B,D,A,-A,-A]])

    trace = numpy.trace(AAT + BBT + CCT + DDT)
    if numpy.array_equal((H.dot(H.transpose())).diagonal(),
                         [[trace]*H.shape[0]]):
        return H
    else:
        return None
```

Имплементација А.4: Баумерт/Хоl конструкција

```
""" Ito construction """
import numpy
def ito(A,B,C,D):
    """
    :param A,B,C,D: Input matrices
    :return: Ito Hadamard Matrix
    """
    n = A.shape[0]
    ABT = A.dot(B.transpose())
    CDT = C.dot(D.transpose())
    BAT = B.dot(A.transpose())
    DCT = D.dot(C.transpose())
    if not numpy.array_equal(ABT + CDT, BAT + DCT):
        return None
    H = numpy.bmat([[A,B,C,D],[B,-A,D,-C],
                    [C.transpose(),-D.transpose(),-A.transpose(),B.transpose()],
                    [D.transpose(),C.transpose(),-B.transpose(), -A.transpose()]])
    return H
```

Имплементација А.5: Итова конструкција

```python
""" Echlich construction"""
import numpy
def echlich(A, G):
    """
    :param A,G: Jacobshtal matrices
    :return: Echlich Hadamard matrix
    """
    m = A.shape[0]
    n = G.shape[0]
    Im = numpy.identity(m, dtype=int)
    In = numpy.identity(n, dtype=int)
    Jm = (numpy.zeros(shape=(m, m), dtype=int) + 1)
    Jn = (numpy.zeros(shape=(n, n), dtype=int) + 1)
    if not numpy.array_equal(A.dot(A.transpose()), A.dot(A)):
        return None
    if not numpy.array_equal(A.dot(A.transpose()),
            (numpy.subtract(numpy.multiply(m, Im), Jm))):
        return None
    temp = numpy.add(numpy.add(numpy.kron(A, G),
                                   numpy.kron(Im, numpy.subtract(In, Jn))),
                     numpy.kron(Jm, In))
    H = numpy.zeros(shape=(temp.shape[0] + 1, temp.shape[1] + 1), dtype=int)
    H[:, H.shape[1] - 1] = 1
    H[0] = 1
    for i in range(1, H.shape[0]):
        for j in range(0, H.shape[1] - 1):
            H[i, j] = temp[i - 1, j]
    return H
```

Имплементација А.6: Ехлихова конструкција

```python
""" Goethals/Seidel construction """
import numpy
def build_circular_matrix(array):
    """
    :param array: First row of the matrix
    :return: Circulant matrix
    """
    matrix = numpy.zeros(shape=(array.shape[0], array.shape[0]), dtype=int)

    for i in range(0, matrix.shape[0]):
        for j in range(0, matrix.shape[1]):
            matrix[(i,j)] = array[(j-i) % array.shape[0]]

    return matrix
def goethal_seidel(t1,t2,t3,t4):
    """
    :param t1,t2,t3,t4: Turin arrays
    :return: Goethal/Seidel Hadamard matrix
    """
    T1 = build_circular_matrix(t1)
    T2 = build_circular_matrix(t2)
    T3 = build_circular_matrix(t3)
    T4 = build_circular_matrix(t4)
    A = T1 + T2 + T3 + T4
    B = -T1 + T2 + T3 - T4
    C = -T1 - T2 + T3 + T4
    D = -T1 + T2 - T3 + T4
    R = numpy.rot90(numpy.identity(A.shape[0], dtype=int))
    AAT = A.dot(A.transpose())
    BBT = B.dot(B.transpose())
    CCT = C.dot(C.transpose())
    DDT = D.dot(D.transpose())
    if numpy.array_equal(AAT+BBT+CCT+DDT, 4*A.shape[0]* numpy.identity(A.shape[0]))
        :
        H = numpy.bmat(
            [[A, B.dot(R), C.dot(R), D.dot(R)],
            [-B.dot(R), A, (D.transpose()).dot(R), (-C.transpose().dot(R))],
            [-C.dot(R), (-D.transpose()).dot(R), A, (B.transpose().dot(R))],
            [-D.dot(R), (C.transpose()).dot(R), (-B.transpose()).dot(R), A]])
        return H
    else:
```

```
        return False
```

Имплементација А.7: Геталс/Сајдл конструкција

```
""" Agayan/Sarukhanyan construction """

import numpy
from utils import is_hadamard
def agayan_sarukhanyan(H1,H2):
    P = H1[0:H1.shape[0]/2,0:H1.shape[0]/2]
    Q = H1[0:H1.shape[0]/2,H1.shape[0]/2:H1.shape[0]]
    R = H1[H1.shape[0]/2:H1.shape[0], 0:H1.shape[0]/2]
    S = H1[H1.shape[0]/2:H1.shape[0], H1.shape[0]/2:H1.shape[0]]

    K = H2[0:H2.shape[0]/2,0:H2.shape[0]/2]
    L = H2[0:H2.shape[0]/2,H2.shape[0]/2:H2.shape[0]]
    M = H2[H2.shape[0]/2:H2.shape[0], 0:H2.shape[0]/2]
    N = H2[H2.shape[0]/2:H2.shape[0], H2.shape[0]/2:H2.shape[0]]

    f1 = numpy.kron(0.5*(P+Q), K) + numpy.kron(0.5*(P-Q),M)
    f2 = numpy.kron(0.5*(P+Q), L) + numpy.kron(0.5*(P-Q), N)
    f3 = numpy.kron(0.5*(R+S), K) + numpy.kron(0.5*(R-S), M)
    f4 = numpy.kron(0.5*(R+S), L) + numpy.kron(0.5*(R-S), N)
    H = numpy.bmat([[f1,f2],[f3,f4]])
    if is_hadamard(H, 8*H1.shape[0]/4*H2.shape[0]/4):
        return H
    else:
        return None
```

Имплементација А.8: Агајан/Сарукханјан конструкција

```
""" Cooper/Wallis construction """

import numpy
from utils import build_circular_matrix
def cooper_wallis(X,Y,Z,W, t1,t2,t3,t4):
    T1 = build_circular_matrix(t1)
    T2 = build_circular_matrix(t2)
    T3 = build_circular_matrix(t3)
    T4 = build_circular_matrix(t4)
    A = numpy.kron(T1,X) + numpy.kron(T2,Y) + numpy.kron(T3,Z) + numpy.kron(T4,W)
    B = numpy.kron(T1, -Y) + numpy.kron(T2,X) + numpy.kron(T3,W) + numpy.kron(T4,-Z
        )
    C = numpy.kron(T1, -Z) + numpy.kron(T2, -W) + numpy.kron(T3, X) + numpy.kron(T4
        ,Y)
    D = numpy.kron(T1, -W) + numpy.kron(T2, Z) + numpy.kron(T3,-Y) + numpy.kron(T4
        ,-X)
    R = numpy.rot90(numpy.identity(A.shape[0], dtype=int))
    H = numpy.bmat(
                [[A, B.dot(R), C.dot(R), D.dot(R)],
                 [-B.dot(R), A, (D.transpose()).dot(R), (-C.transpose().dot(R))],
                 [-C.dot(R), (-D.transpose()).dot(R), A, (B.transpose().dot(R))],
                 [-D.dot(R), (C.transpose()).dot(R), (-B.transpose()).dot(R), A]])
    return H
```

Имплементација А.9: Купер/Волис конструкција

Додатак B

Табела добрих матрица

Додатак садржи такозване добре матрице преузете из базе Х. Коуковинуса [6]. Матрице су цикличне, симетричне и задовољавају једнакост:

$$AA^{\tau} + BB^{\tau} + CC^{\tau} + DD^{\tau} = 4nI_n$$

Наводи се ред матрице и прве врсте матрица редом A, B, C, D. С обзиром на то да су матрице цикличне оне су и одређене првом врстом, што нам омогућава да једноставно конструишемо комплетну матрицу. Са знаком 1 означен је број 1, док је са знаком - означен број -1.

```
m=3
111
1--
1--
1--
m=5
1----
1----
11--1
1-11-
m=7
111--11
111--11
11-11-1
1-1--1-
m=9
11------1
1-1----1-
1--1--1--
1---11---
m=11
11--------1
11-11--11-1
11-1-11-1-1
1-11----11-
m=13
111-11--11-11
11-1-1111-1-1
11--1-11-1--1
1---111111---
m=15
1-11-111111-11-
1-11--1111--11-
11-11------11-1
1-1---1111---1-
m=17
1--1-11111111-1--
1-11-111--111-11-
11-1---1--1---1-1
1---111----111---
m=19
11--11-111111-11--1
1-11--1------1--11-
111---1-1--1-1---11
111---1-1--1-1---11
m=21
11-11111-1--1-11111-1
11--1-1-11--11-1-1--1
1111-1---1--1---1-111
1--1111---11---1111--
m=23
1-11-11--111111--11-11-
11---1---1-11-1---1---1
111---11-1-11-1-11---11
111-111-1------1-111-11
m=25
11-1111--1-1111-1--1111-1
1----1111--1--1--1111----
```

```
1-1-11---1--11--1---11-1-
1---1-1--11111111--1-1---
m=27
1--1---111--------111---1--
1-11-1----11----11----1-11-
1-1-111--1---11---1--111-1-
1111-1---1--1--1--1---1-111
m=29
1111-11-1---111111---1-11-111
11--1--1-111-1111-111-1--1--1
111---11--1-1----1-1--11---11
1-1---11--1-111111-1--11---1-
m=31
11-1---11111-1-11-1-11111---1-1
11111--111-1--1--1--1-111--1111
11-11--1----11----11----1--11-1
1---111-11-1-1----1-1-11-111---
m=33
111--1---1--1--------1--1---1--11
111----1-1-1--11--11--1-1-1----11
1-11--1-1---1111--1111---1-1--11-
11--1----11-1-111111-1-11----1--1
m=37
1--111-1-----1----11----1-----1-111--
11111-1-----11----11----11-----1-1111
1---11-11--1-1-11----11-1-1--11-11---
1--1-1-1-11---1--1111--1---11-1-1-1--

m=41
1-1---111-11-1--1111111111--1-11-111---1-
1-1---111-11-1--1111111111--1-11-111---1-
11----1-111-11-11---11---11-11-111-1----1
1-1111-1---1--1--111--111--1--1---1-1111-
m=43
1---11--1111-1-111-11--11-111-1-1111--11---
11-111111----1-1--11-11-11--1-1----111111-1
111-1-11--1-1-1111-1----1-1111-1-1--11-1-11
11---1111-1--1--11--------11--1--1-1111---1
m=45
111--111-11-1-11111--------11111-1-11-111--11
1--11---1--1-1-----11111111-----1-1--1---11--
11--1111-1-1-111--1--1--1--1--111-1-1-1111--1
11--1111-1-1-111--1--1--1--1--111-1-1-1111--1
m=49
1111-11-1---11-111---11-11-11---111-11---1-11-111
1111-11-1---11-111---11-11-11---111-11---1-11-111
1----1-1111--1-111-1-111--111-1-111-1--1111-1----
11111-1----11-1---1-1---11---1-1---1-11----1-1111
m=51
1---111-11-1-111--11111--11--11111--111-1-11-111---
1111---1--1-1---11-----11--11-----11---1-1--1---111
1-11-1111-1---1-1-----11-11-11-----1-1---1-1111-11-
1-11-1111-1---1-1-----11-11-11-----1-1---1-1111-11-
m=55
1-1--1-1-11--1-11111-111--1111--111-11111-1--11-1-1--1-
11-11-1-1--11-1-----1---11----11---1-----1-11--1-1-11-1
111----11-11--11----1-1-11111111-1-1----11--11-11----11
111----11-11--11----1-1-11111111-1-1----11--11-11----11
m=57
1---11-1--1111-111-11---1-111111-1---11-111-1111--1-11---
1111--1-11----1---1--111-1------1-111--1---1----11-1--111
11-1-1---11-11-1--111-----1----1-----111--1-11-11---1-1-1
11-1-1---11-11-1--111-----1----1-----111--1-11-11---1-1-1
```

```
m=61
11--1--11--1-1-1111--1-----1------1-----1--1111-1-1--11--1--1
11--1--11--1-1-1111--1-----1------1-----1--1111-1-1--11--1--1
1---1-1-1111---11--1-11-1---111111---1-11-1--11---1111-1-1---
1111-1-1----111--11-1--1-111------111-1--1-11--111----1-1-111
m=63
1111-11-1-1111-1---1---111---111111---111---1---1-1111-1-11-111
1111-11-1-1111-1---1---111---111111---111---1---1-1111-1-11-111
11-111--11-11--1--1-11-1-111--------111-1-11-1--1--11-11--111-1
1-1---11--1--11-11-1--1-1---11111111---1-1--1-11-11--1--11---1-
```

Библиографија

[1] H. Kharaghani and B. Tayfeh-Rezaie, A Hadamard matrix of order 428, J. Combinatorial Designs 13 (2005), 435–440 *Institute for studies in theoretical Physics and Mathematics (2005)*

[2] Horadam, K.J.: Hadamard Matrices and Their Applications *Phil. Mag. 34 (1867): 461-475.*

[3] J. J. Sylvester, Thoughts on inverse orthogonal matrices, simultaneous sign successions, and tesselated pavements in two or more colours, with applications to Newton's rule, ornamental tile-work, and the theory of numbers. *Princeton University Press (2006)*

[4] A.S. Hedayat, N.J.A. Sloane, John Stufken, Orthogonal Arrays: Theory and Applications *Springer Series in Statistics, Springer New York (2012)*

[5] J. Williamson, Hadamard's determinant theorem and the sum of four squares. *Duke Math. J. 11 (1944), no. 1, 65–81*

[6] Christos Koukouvinos, `http://www.math.ntua.gr/~ckoukouv/`

[7] N.Ito, Note on Hadamard matices of type Q *Studia Sci. Math. Hungar:14 (1981) 389-393.*

[8] H Echlich, Neue Hadamard-Matrizen *Arch. Math. 16(1965)*

[9] E.Spence, Skew-Hadamard matrices of order 2(q + 1) *Discrete Mathematics (Impact Factor: 0.57). 01/1977; 18(1):79-85.*

[10] J. Seberry, M. Yamada Hadamard matrices, sequences, and block designs. *University of Wollongong, Research Online (1992)*

[11] Y.X Yang, Theory and Applications of Higher-Dimensional Hadamard Matrices, Second Edition *Chapman and Hall/CRC; 2 edition*

[12] Python Sofware Fundation, `https://www.python.org/`

[13] Numpy, `http://www.numpy.org/`

[14] SciPy, `http://www.scipy.org/`

[15] Matplotlib, `http://matplotlib.org/`