

Gimnazija Vič

Hana Perman

ARNOLDOVA MAČKA

Mentor: Klemen Bajec, prof., Tjaša Bajc, prof.

Ljubljana, 2025

Kazalo

1	Uvod	4
2	Teoretično ozadje	5
2.1	Matrike in determinante	5
2.2	Linearne preslikave v \mathbb{R}^2 in matrike	8
2.3	Lastne vrednosti in lastni vektorji	12
3	Preslikava Arnoldove mačke	14
4	Potrebščine za nadaljnje raziskovanje	15
5	Kako generirati hiperbolične matrike?	16
5.1	Matrike po formuli za hiperbolično matriko	16
5.2	Matrike z Evklidovim algoritmom	16
6	Kako priti do periode?	19
6.1	Ugotavljanje periode s ponavljanjem preslikave	19
6.2	Ugotavljanje periode s potenciranjem matrike	22
7	Skrivanje sporočil	23
8	Analiza period	26
8.1	Grafi period v odvisnosti od stranice mreže	26
8.2	Grafi period v odvisnosti od matrike	31
9	Ali je perioda napovedljiva?	33
9.1	Zveza med stranico mreže in periodo	33
9.2	Množica koeficientov	34
9.3	K-means clustering	36
9.4	Raznolikost period	37
9.5	Uporabnost v steganografiji	37
10	Zaključek	38

Arnoldova mačka

POVZETEK

V raziskovalni nalogi smo raziskali preslikavo Arnoldove mačke. Preslikava je hiperbolični avtomorfizem torusa. Ko s to preslikavo dovoljkrat preslikamo celoštevilsko mrežo samo vase, se mreža preslika nazaj v svoje prvotno stanje. V raziskovalni nalogi smo se ukvarjali s tem, kolikšno je najmanjše število ponovitev, ki so potrebne, da bo preslikava identiteta. Imenovali smo ga perioda. Periode se ne da izračunati, lahko pa jo s poskušanjem določimo pri dani mreži in matriki. Raziskovali smo, kako kaotične so periode. Opazovali smo, kako na periodo vplivata velikost mreže in matrika, pri katerih je bila perioda določena. Pogledali smo razmerja med periodo in stranico mreže in ugotovili, kako pogosta so določena razmerja. Z metodo k-means clustering smo poskusili grafe period razporediti v skupine a nismo našli smiselne razvrstitve. Raziskali smo tudi, kako raznolike so periode pri dani matriki ali mreži. Jasne povezave med periodo in matriko nismo našli. Preslikava je torej kaotična, periode ni mogoče napovedati. Zaradi tega je preslikava primerna za skrivanje sporočil. Pogledali smo tudi uporabnost preslikave v steganografiji in napisali program, ki skrije sporočilo.

Arnold's cat map

ABSTRACT

In this research paper, we investigated the mapping of Arnold's cat. The mapping is a hyperbolic automorphism of the torus. When we use this mapping to map the integer grid into itself enough times, the mesh is mapped back to its original state. In our research, we dealt with the minimum number of repetitions needed for the image to be the identity. We call it the period. The period cannot be calculated, but it can be determined by experiment for a given grid and matrix. We investigated how chaotic periods are. We observed how the period is affected by the size of the grid, and the matrix for which the period is determined. We looked at the relationship between the period and the size of the grid, and found out how common these relationships are. Using k-means clustering method, we tried to arrange the graphs of periods into groups, but we did not find a meaningful classification. We also investigated how diverse the periods are for a given matrix or grid. The final conclusion is that no clear connection between the period and the matrix exists. The conclusion is that the mapping is chaotic and the period cannot be predicted, which means that the mapping is suitable for hiding messages. We also looked at the usefulness of mappings in steganography and wrote a program that can hide a message.

Math. Subj. Class. (2020): 37E30, 37D45, 39B12

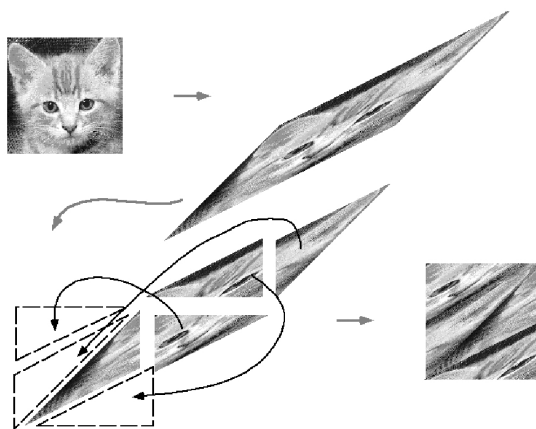
Ključne besede: perioda, preslikava

Keywords: period, mapping

1 Uvod

Preslikava Arnoldove mačke je dobila ime iz knjige *Problèmes ergodiques de la mécanique classique*, ki jo je skupaj z Andréjem Avezom napisal ukrajinski matematik Vladimir Igorjevič Arnold leta 1967. V knjigi je bil obris mačke uporabljen za ponazoritev delovanja preslikave na torusu. V Arnoldovi domači ruščini je preslikava znana kot "okroška (hladna juha)", kar se nanaša na mešalne lastnosti preslikave in tvori besedno igro. Arnold je kasneje zapisal, da se mu zdi ime "Arnoldova mačka", pod katerim je preslikava znana v angleščini in drugih jezikih, čudno. [4]

Opis preslikave Arnoldove mačke smo zasledili v knjigi *Diskretni dinamični sistemi* [1], v poglavju *Hiperbolični avtomorfizmi torusa* (str. 89). V knjigi je s sliko (glej sliko 1) ponazorjeno delovanje preslikave Arnoldove mačke.



Slika 1: Grafična ponazoritev preslikave, vir: [1]

Preslikava Arnoldove mačke je podana z matriko

$$M = \begin{bmatrix} 2 & 1 \\ 1 & 1 \end{bmatrix}.$$

Poglavje *Hiperbolični avtomorfizem torusa* govori o tem, kolikšno je najmanjše število ponovitev preslikave, ki jih moramo izvesti, da bo slika, podana s končno piksli, spet izgledala tako kot na začetku. Najmanjše število ponovitev poimenujemo perioda. Perioda je pri določeni preslikavi in velikosti slike (ali mreže) celo število n . V članku [2] je predstavljena uporaba preslikave Arnoldove mačke v steganografiji. Cilj steganografije je prikrivanje podatkov. V steganografiji podatek, ki ga imata pošiljatelj in prejemnik ter omogoča dešifriranje sporočila, imenujemo ključ. Če neko sliko (predstavljeno z mrežo točk) slikamo z matriko A po vzoru preslikave Arnoldove mačke, bo po določenem številu ponovitev preslikava identiteta. Če na katerega od n iteratov skrijemo sporočilo, se slednjega ne bo dalo razbrati, dokler se slika spet preslika v stanje v katerem smo skrili sporočilo. Torej jo moramo ponoviti tolikokrat, kot je perioda.

Tudi mi smo poskusili raziskati preslikavo in njene matematične značilnosti ter poskusili napisati program, ki bo poiskal periodo. Vprašali smo se, ali lahko pri dani matriki in stranici mreže določimo periodo brez poskušanja. Z raziskovanjem smo poskusili priti tudi do svojega programa za skrivanje sporočil.

2 Teoretično ozadje

Vir za teorijo je spletna stran Algebra 1 avtorja Tomaža Koširja. Dostopna je na [3] Začnimo z definicijami osnovnih pojmov.

2.1 Matrike in determinante

Definicija 2.1. Realna matrika M dimenzije 2×2 je shema realnih števil

$$M = \begin{bmatrix} a & b \\ c & d \end{bmatrix}.$$

Množico vseh realnih matrik dimenzije 2×2 označimo z $\mathbb{R}^{2 \times 2}$. Take matrike lahko seštevamo in množimo ter jih množimo s skalarjem po naslednjih pravilih. Naj bosta A in B matriki, ter r realno število, torej

$$A = \begin{bmatrix} a & b \\ c & d \end{bmatrix}, \quad B = \begin{bmatrix} e & f \\ g & h \end{bmatrix} \quad \text{in} \quad r \in \mathbb{R}.$$

Potem je

$$A + B = \begin{bmatrix} a + e & b + f \\ c + g & d + h \end{bmatrix}, \quad rA = \begin{bmatrix} ra & rb \\ rc & rd \end{bmatrix} \quad \text{in} \quad AB = \begin{bmatrix} ae + bg & af + bh \\ ce + dg & cf + dh \end{bmatrix}.$$

Na matrikah definiramo tudi operacijo transponiranja, ki ohrani diagonalne elemente, zamenja pa mesti elementov b in c ,

$$A^T = \begin{bmatrix} a & c \\ b & d \end{bmatrix}.$$

Opomnimo, da množenje matrik ni komutativno, torej produkta AB in BA praviloma nista enaka. Nalednje trditve opisujejo lastnosti seštevanja, množenja in transponiranja matrik.

Trditev 2.2. Za $A, B \in \mathbb{R}^{2 \times 2}$ velja

1. komutativnost seštevanja: $A + B = B + A$,
2. distributivnost: $A(B + C) = AB + AC$ in $(A + B)C = AC + BC$,
3. asociativnost množenja: $A(BC) = (AB)C$ in
4. $(AB)^T = B^T A^T$.

Zgornjo trditev lahko v celoti dokažemo z direktnim računom, zato bomo to izpustili. Naslednja trditev ni presenetljiva, saj podobno velja za realna števila.

Trditev 2.3. Realne 2×2 matrike so grupa za seštevanje. Nevtralni element je ničelna matrika

$$0 = \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix},$$

naprotna matrika matriki A pa je matrika $-A$.

Definicija 2.4. Matrica I je nevtralni element za množenje ali enota za množenje (imenujemo jo *identiteta*) in zanjo velja

$$I = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}.$$

Diagonalna matrica D ima neničelne elemente samo na glavni diagonalni.

$$D = \begin{bmatrix} d_1 & 0 \\ 0 & d_2 \end{bmatrix}.$$

Matrica A je *obrnljiva*, če obstaja taka matrica X , da je $AX = XA = I$. Matrico X imenujemo *inverzna matrica* matrike A .

Identiteta je torej takšna matrica, da za vsako matrico A velja $AI = IA = A$. Poglejmo lastnosti inverza in kdaj inverz obstaja. Pri tem je pomemben pojem determinante matrike.

Obrnljive matrike so grupa za množenje, kjer je matrica I identiteta, inverzni element elementa pa je njegova inverzna matrica.

Definicija 2.5. Naj bo

$$A = \begin{bmatrix} a & b \\ c & d \end{bmatrix}.$$

Determinanta matrike A je definirana kot

$$\det(A) = ad - bc.$$

Ker gre pri reševanju matrične enačbe $AX = XA = I$, s katero iščemo inverzno matrico matrike A za matrično množenje, nas zanima, kako je z determinanto produkta matrik.

Trditev 2.6. Naj bosta $A, B \in \mathbb{R}^{2 \times 2}$. Potem velja

$$\det(AB) = \det(A) \det(B) \text{ in } \det(A)^T = \det(A).$$

Dokaz. Tudi to lahko računsko dokažemo. V skladu s prej navedenimi definicijami računamo

$$A = \begin{bmatrix} a & b \\ c & d \end{bmatrix}, \quad B = \begin{bmatrix} e & f \\ g & h \end{bmatrix} \text{ in } r \in \mathbb{R}.$$

Dokazujemo, da velja $\det(AB) = \det(A) \det(B)$.

$$\det \begin{bmatrix} ae + bg & af + bh \\ ce + dg & cf + dh \end{bmatrix} = \det \begin{bmatrix} a & b \\ c & d \end{bmatrix} \det \begin{bmatrix} e & f \\ g & h \end{bmatrix}$$

Uporabimo formulo za determinanto in poračunajmo.

$$(ae + bg)(cf + dh) - (af + bh)(ce + dg) = (ad - bc)(eh - fg)$$

Poračunajmo najprej levo stran enačbe:

$$\begin{aligned} & (ae + bg)(cf + dh) - (af + bh)(ce + dg) = \\ & acef + adeh + bcfg + bdgh - acef - adfg - bceh - bdgh = \\ & \qquad\qquad\qquad adeh - adfg + bcfg - bceh. \end{aligned}$$

Zdaj poračunajmo še desno stran

$$\begin{aligned} & (ad - bc)(eh - fg) = \\ & = adeh - adfg + bcfg - bceh. \end{aligned}$$

Leva in desna stran enačbe sta enaki s čimer smo dokazali trditev.
Dokažimo še $\det(A^T) = \det(A)$. Matrika A naj ostane

$$A = \begin{bmatrix} a & b \\ c & d \end{bmatrix}.$$

Njena transponiranka A^T bo zato

$$A = \begin{bmatrix} a & c \\ b & d \end{bmatrix}.$$

Dokažimo da imata matriki enako determinanto.

$$\begin{aligned} \det(A) &= \det(A^T) \\ \det \begin{bmatrix} a & b \\ c & d \end{bmatrix} &= \det \begin{bmatrix} a & c \\ b & d \end{bmatrix} \end{aligned}$$

Če upoštevamo izraz za determinanto je rezultat

$$ad - bc = ad - bc.$$

Leva in desna stran enačbe sta enaki s čimer zaključimo dokaz. □

Sedaj lahko navedemo še pogoj za obstoj inverzne matrike.

Trditev 2.7. Naj bo $A \in \mathbb{R}^{2 \times 2}$. Matrika A obrnljiva natanko tedaj, ko $\det(A) \neq 0$.

Dokaz. Naj bo A obrnljiva in X njena inverzna matrika. Po trditvi 2.6. je $AX = I$, torej je tudi $\det(AX) = \det A \det X = \det I = 1$, kar pomeni, da je $\det A$ različna od 0. Dokažimo še obratno. Če $\det A \neq 0$, je inverzna matrika podana s formulo

$$X = A^{-1} = \frac{1}{\det A} \begin{bmatrix} d & -b \\ -c & a \end{bmatrix}.$$

Enostaven račun pokaže, da enačba drži, zato ga bomo izpustili. □

Opomba 2.8. Če je A celoštevilška matrika z determinanto 1 ali -1 , je tudi inverzna matrika celoštevilška.

2.2 Linearne preslikave v \mathbb{R}^2 in matrike

V uvodu smo zapisali, da je preslikava Arnoldove mačke tesno povezana z matriko

$$M = \begin{bmatrix} 2 & 1 \\ 1 & 1 \end{bmatrix}.$$

Poglejmo, kako so v splošnem povezane linearne preslikave in matrike. Za začetek definirajmo vektorski prostor \mathbb{R}^2 , linearno preslikavo in nekaj osnovnih pojmov o vektorjih.

Definicija 2.9. Z \mathbb{R}^2 označimo množico vseh vektorjev v ravnini,

$$\mathbb{R}^2 = \left\{ \begin{bmatrix} a \\ b \end{bmatrix}, a, b \in \mathbb{R} \right\}.$$

Definicija 2.10. Preslikava $\mathcal{A} : \mathbb{R}^2 \rightarrow \mathbb{R}^2$ je *linearna*, če za vsaka dva vektorja $\vec{v}_1, \vec{v}_2 \in \mathbb{R}^2$ in za vsaki dve realni števili m, n velja

$$\mathcal{A}(m\vec{v}_1 + n\vec{v}_2) = m\mathcal{A}(\vec{v}_1) + n\mathcal{A}(\vec{v}_2).$$

Zapisu $m\vec{v}_1 + n\vec{v}_2$ pravimo linearna kombinacija vektorjev \vec{v}_1 in \vec{v}_2 .

Definicija 2.11. Množica $\{\vec{v}_1, \vec{v}_2\}$ je *baza*, če lahko vsak vektor $\vec{v} \in \mathbb{R}^2$ na en sam način zapišemo kot

$$\vec{v} = m\vec{v}_1 + n\vec{v}_2.$$

Trditev 2.12. Množica $\{\vec{v}_1, \vec{v}_2\}$ je baza natanko tedaj, ko \vec{v}_1, \vec{v}_2 nista vzporedna.

Dokaz. Vektorja \vec{v}_1, \vec{v}_2 sta baza natanko tedaj, ko lahko vsak vektor \vec{v} zapišemo na en sam način kot $\vec{v} = m\vec{v}_1 + n\vec{v}_2$. Da lahko vsak vektor zapišemo kot linearno kombinacijo pa mora veljati $m\vec{v}_1 + n\vec{v}_2 = 0 \Leftrightarrow n, m = 0$. Če vektorja \vec{v}_1, \vec{v}_2 nista vzporedna za vsak vektor \vec{v} , po paralelogramskem pravilu obstajata natančno določena m, n , da je $\vec{v} = m\vec{v}_1 + n\vec{v}_2$. Če bi bila vektorja \vec{v}_1 in \vec{v}_2 vzporedna, bi lahko z njuno linearno kombinacijo napisali samo vse njima vzporedne vektorje. \square

Definicija 2.13. Množica

$$\left\{ \vec{e}_1 = \begin{bmatrix} 1 \\ 0 \end{bmatrix}, \vec{e}_2 = \begin{bmatrix} 0 \\ 1 \end{bmatrix} \right\}$$

je *standardna baza* prostora \mathbb{R}^2 .

Trditev 2.14. Vektorja

$$\vec{v}_1 = \begin{bmatrix} a \\ c \end{bmatrix}, \vec{v}_2 = \begin{bmatrix} b \\ d \end{bmatrix},$$

nista vzporedna natanko tedaj, ko

$$\det \begin{bmatrix} a & b \\ c & d \end{bmatrix} \neq 0.$$

Dokaz. Privzamimo, da sta vektorja $\vec{v}_1, \vec{v}_2 \neq 0$, ker je ničelni vektor po definiciji vzporeden vsakemu vektorju. Dokazali bomo, da sta vektorja vzporedna natanko tedaj, ko je zgornja determinanta enaka nič. Naj bo

$$\vec{v}_1 = \begin{bmatrix} a \\ c \end{bmatrix}, \quad \vec{v}_2 = \begin{bmatrix} b \\ d \end{bmatrix} \quad \text{in} \quad M = \begin{bmatrix} a & b \\ c & d \end{bmatrix}.$$

Če sta vektorja \vec{v}_1 in \vec{v}_2 vzporedna, velja $b = ka$ in $d = kc$ za neki $k \in \mathbb{R}$. Če slednje uporabimo v izrazu za determinantno, ugotovimo, da je

$$\det \begin{bmatrix} a & c \\ ka & kc \end{bmatrix} = kac - kac = 0.$$

Vidimo torej, da za vzporedna vektorja velja, da je opazovana determinanta enaka nič. Poglejmo še obrat, torej da iz tega, da je determinanta enaka nič, sledi, da sta vektorja vzporedna. Če je $\det(M) = 0$, velja $ad - bc = 0$, iz česar sledi $ad = bc$. Če to enačbo pomnožimo z $\frac{1}{ab}$, dobimo zvezo

$$\frac{a}{c} = \frac{b}{d} = k \quad \text{za} \quad a, b, c, d \neq 0.$$

Če bi na primer veljalo $c = 0$, bi iz tega sledilo $ad = 0$. Ker je \vec{v}_1 neničelni vektor, a po definiciji ni 0, zato je edina možna rešitev $d = 0$. Vektorja lahko zato zapišemo kot

$$\vec{v}_1 = \begin{bmatrix} a \\ 0 \end{bmatrix} \quad \vec{v}_2 = \begin{bmatrix} b \\ 0 \end{bmatrix}.$$

Taka vektorja sta vzporedna ne glede na a in b . Podobno lahko dokažemo tudi v primeru, da bi veljalo $a = 0$.

Če iz enačbe izrazimo b in d , lahko vektor \vec{v}_2 zapišemo kot

$$\vec{v}_2 = \begin{bmatrix} a \frac{b}{d} \\ b \frac{a}{c} \end{bmatrix} = \begin{bmatrix} ak \\ bk \end{bmatrix}.$$

Iz tega sledi, da je vektor \vec{v}_2 vzporeden \vec{v}_1 , ko je $\det(M) = 0$, s čimer zaključimo dokaz. \square

Trditev 2.15. *Linearna preslikava je enolično določena s slikama dveh nevzporednih vektorjev.*

Dokaz. Izberimo si nevzporedna vektorja \vec{v}_1 in \vec{v}_2 . Vsak vektor \vec{v} lahko zapišemo kot linearno kombinacijo teh dveh vektorjev, torej $\vec{v} = x_1\vec{v}_1 + x_2\vec{v}_2$. Naj bo $\mathcal{A}(\vec{v}_1) = a\vec{v}_1 + c\vec{v}_2$ in $\mathcal{A}(\vec{v}_2) = b\vec{v}_1 + d\vec{v}_2$. Ker je \mathcal{A} linearna preslikava, velja

$$\begin{aligned} \mathcal{A}(\vec{v}) &= \mathcal{A}(x\vec{v}_1 + y\vec{v}_2) = x\mathcal{A}(\vec{v}_1) + y\mathcal{A}(\vec{v}_2) \\ &= x(a\vec{v}_1 + c\vec{v}_2) + y(b\vec{v}_1 + d\vec{v}_2). \end{aligned}$$

\square

Poglejmo, kako lahko linearno preslikavo \mathcal{A} zapišemo v bazi $\{\vec{v}_1, \vec{v}_2\}$. Če je

$$\vec{v} = x_1\vec{v}_1 + x_2\vec{v}_2, \quad \mathcal{A}(\vec{v}_1) = a\vec{v}_1 + c\vec{v}_2 \text{ in } \mathcal{A}(\vec{v}_2) = b\vec{v}_1 + d\vec{v}_2, \quad (2.1)$$

potem podobno kot v dokazu zadnje trditve velja

$$\begin{aligned} \mathcal{A}(\vec{v}) &= x_1\mathcal{A}(\vec{v}_1) + x_2\mathcal{A}(\vec{v}_2) \\ &= x_1(a\vec{v}_1 + c\vec{v}_2) + x_2(b\vec{v}_1 + d\vec{v}_2) \\ &= \vec{v}_1(x_1a + x_2b) + \vec{v}_2(x_1c + x_2d) \\ &= y_1\vec{v}_1 + y_2\vec{v}_2. \end{aligned}$$

Dogovorimo se, da bomo vsak vektor $\vec{v} = x_1\vec{v}_1 + x_2\vec{v}_2$ identificirali z vektorjem komponent v baznih smereh. Zato je

$$\vec{v}_1 \simeq \begin{bmatrix} 1 \\ 0 \end{bmatrix}, \quad \vec{v}_2 \simeq \begin{bmatrix} 0 \\ 1 \end{bmatrix}, \quad \vec{v} \simeq \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}, \quad \mathcal{A}(\vec{v}) \simeq \begin{bmatrix} y_1 \\ y_2 \end{bmatrix}.$$

Če sedaj linearni preslikavi \mathcal{A} priredimo matriko

$$A = \begin{bmatrix} a & b \\ c & d \end{bmatrix},$$

dobimo zvezo

$$\begin{bmatrix} y_1 \\ y_2 \end{bmatrix} = \begin{bmatrix} a & b \\ c & d \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}.$$

Linearni preslikavi \mathcal{A} v bazi $\{\vec{v}_1, \vec{v}_2\}$ pripada matrika A . V drugih bazah isti linearni preslikavi pripada drugačna matrika.

Trditev 2.16. Če imamo linearno preslikavo \mathcal{A} , ki ji v bazi $\{\vec{v}_1, \vec{v}_2\}$ pripada matrika A , in linearno preslikavo \mathcal{B} , ki ji v bazi $\{\vec{v}_1, \vec{v}_2\}$ pripada matrika B , potem preslikavi $\mathcal{A} \circ \mathcal{B}$ v isti bazi pripada matrika AB .

Dokaz. Definirajmo vektor \vec{v} ter matriki A in B .

$$\vec{v} = \begin{bmatrix} x \\ y \end{bmatrix}, \quad A = \begin{bmatrix} a & b \\ c & d \end{bmatrix}, \quad B = \begin{bmatrix} e & f \\ g & h \end{bmatrix}.$$

Z matriko B slikamo vektor \vec{v} .

$$B\vec{v} = \begin{bmatrix} ex + fy \\ gx + hy \end{bmatrix}$$

Z matriko A slikamo vektor $B\vec{v}$.

$$\begin{aligned} A(B\vec{v}) &= \begin{bmatrix} a & b \\ c & d \end{bmatrix} \begin{bmatrix} ex + fy \\ gx + hy \end{bmatrix} \\ &= \begin{bmatrix} a(ex + fy) + b(gx + hy) \\ c(ex + fy) + d(gx + hy) \end{bmatrix} = \begin{bmatrix} x(ae + bg) + y(af + bh) \\ x(ce + dg) + y(cf + dh) \end{bmatrix} \end{aligned}$$

Enak rezultat dobimo, če vektor \vec{v} slikamo z matriko AB , ki je produkt matrik A in B .

$$AB\vec{v} = \begin{bmatrix} ae + bg & af + bh \\ ce + dg & cf + dh \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} x(ae + bg) + y(af + bh) \\ x(ce + dg) + y(cf + dh) \end{bmatrix}$$

□

2.3 Lastne vrednosti in lastni vektorji

Če sta A in B diagonalni matriki, je njun produkt enostavno izračunati, saj je za $A = \begin{bmatrix} a & 0 \\ 0 & b \end{bmatrix}$ in $B = \begin{bmatrix} c & 0 \\ 0 & d \end{bmatrix}$ njun produkt kar

$$AB = \begin{bmatrix} ac & 0 \\ 0 & bd \end{bmatrix}.$$

Če želimo izračunati večkratni kompozitum linearne preslikave \mathcal{A} same s seboj, je zato smiselno zapisati linearno preslikavo v taki bazi, da ji bo pripadala diagonalna matrika (če je to mogoče). Denimo, da je baza $\{\vec{v}_1, \vec{v}_2\}$ taka baza in da je pripadajoča diagonalna matrika enaka

$$D = \begin{bmatrix} \lambda_1 & 0 \\ 0 & \lambda_2 \end{bmatrix} \begin{bmatrix} x_1 \\ y_2 \end{bmatrix} = \begin{bmatrix} \lambda_1 x_1 \\ \lambda_2 x_2 \end{bmatrix} = \begin{bmatrix} y_1 \\ y_2 \end{bmatrix}$$

To pomeni, da mora veljati $\mathcal{A}(\vec{v}_1) = \lambda_1 \vec{v}_1$ in $\mathcal{A}(\vec{v}_2) = \lambda_2 \vec{v}_2$. Poglejmo, kdaj je enačba $\mathcal{A}(\vec{v}) = \lambda \vec{v}$ rešljiva za $\vec{v} \neq 0$. Naj linearni preslikavi \mathcal{A} v standardni bazi pripada matrika

$$A = \begin{bmatrix} a & b \\ c & d \end{bmatrix}.$$

Naj bo vektor \vec{v} enak

$$\vec{v} = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix},$$

Ko ga slikamo z matriko A , mora veljati

$$A\vec{v} = \begin{bmatrix} a & b \\ c & d \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} ax_1 + bx_2 \\ cx_1 + dx_2 \end{bmatrix} = \begin{bmatrix} \lambda x_1 \\ \lambda x_2 \end{bmatrix} = \begin{bmatrix} y_1 \\ y_2 \end{bmatrix}$$

Potem se enačba po komponentah prepíše kot

$$\begin{aligned} y_1 = ax_1 + bx_2 = \lambda x_1, & \quad y_2 = cx_1 + dx_2 = \lambda x_2, \\ x_1(a - \lambda) + bx_2 = 0, & \quad cx_1 + x_2(d - \lambda) = 0. \end{aligned}$$

Ko uredimo sistem, dobimo

$$x_1((a - \lambda)(d - \lambda) - bc) = 0, \quad x_2((a - \lambda)(d - \lambda) - bc) = 0.$$

Če je $P(\lambda) := ((a - \lambda)(d - \lambda) - bc) \neq 0$ dobimo $x_1, x_2 = 0$, zato je $\vec{v} = 0$, torej \vec{v} ni bazni vektor. Polinom $P(\lambda)$ mora zato biti enak nič. Iz tega sledi, da:

$$P(\lambda) = \lambda^2 - (a + d)\lambda + ad - bc = 0, \quad \lambda_{1,2} = \frac{a + d \pm \sqrt{(a + d)^2 - 4(ad - bc)}}{2}.$$

Opazimo, da

$$\lambda_1 + \lambda_2 = a + d, \quad \lambda_1 \lambda_2 = ad - bc = \det A \text{ in } P(\lambda) = \det(A - \lambda I).$$

Definicija 2.19. Naj bo \mathcal{A} linearna preslikava. Neničelni vektor \vec{v} je *lastni vektor* \mathcal{A} , če obstaja tako realno število λ , imenovano *lastna vrednost*, da velja

$$\mathcal{A}(\vec{v}) = \lambda\vec{v}.$$

Par (λ, \vec{v}) imenujemo *lastni par*, polinom $P(\lambda) = \det(A - \lambda I)$ pa *karakteristični polinom*.

Opomba: karakteristični polinom je neodvisen od izbire baze

Lastni vrednosti sta ničli karakterističnega polinoma. Nas bodo zanimale linearne preslikave, ki ohranjajo ploščino in nimajo lastnih vrednosti z absolutno vrednostjo 1.

Definicija 2.20. *Hiperbolična linearna preslikava* je preslikava, ki ohranja ploščino in ima lastni vrednosti, ki ne ležita na enotski krožnici v kompleksni ravnini.

Trditev 2.21. *Hiperbolična linearna preslikava nima kompleksno konjugiranih lastnih vrednosti in ima dve realni lastni vrednosti λ_1, λ_2 , za kateri velja*

$$0 < |\lambda_1| < 1 < |\lambda_2|.$$

Dokaz. Poznamo naslednje

$$\det \begin{bmatrix} a & b \\ c & d \end{bmatrix} = ad - bc = 1$$

in

$$\lambda_1\lambda_2 = ad - bc = 1.$$

Da bo produkt $\lambda_1\lambda_2 = 1$, mora veljati $\lambda_1^{-1} = \lambda_2$. Ker sta lastni vrednosti zunaj enotske krožnice, izločimo rešitev rešitev $\lambda_1 = \lambda_2 = \pm 1$. Iz istega razloga izločimo tudi rešitev $\lambda_1 = z$ in $\lambda_2 = \bar{z}$, saj je $\lambda_1\lambda_2 = 1 \Rightarrow z\bar{z} = |z| = 1$, kar pomeni, da sta z in \bar{z} iz enotske krožnice. Edina ostala možna rešitev je $0 < |\lambda_1| < 1 < |\lambda_2|$. \square

3 Preslikava Arnoldove mačke

Naslednje poglavje je povzeto po virih [2], [5]. Prostor \mathbb{R}^2 je dvodimenzionalen evklidski prostor, v katerem vsako točko definiramo s parom realnih števil (x, y) , prostor \mathbb{Z}^2 pa predstavlja mrežo točk s celoštevilskimi koordinatami, torej vseh točk oblike (m, n) , kjer sta m in n celi števili. Če v ravnini \mathbb{R}^2 identificiramo vse točke, katerih koordinate se razlikujejo za celo število, dobimo torus T . Poglejmo, zakaj. Ekvivalenčni razred točke (x, y) je enak

$$[x, y] = \{(x_1, y_1), x_1 = x + m, y_1 = y + n, m, n \in \mathbb{N}\}.$$

Ker so točke s celoštevilskimi koordinatami ekvivalentne točki $(0, 0)$, lahko tudi za predstavnika ekvivalenčnega razreda $[x, y]$ izberemo točko, ki leži v zaprtem enotskem kvadratu, pri čemer identificiramo nasproti ležeče robove. Če bi imeli kos papirja, bi to pomenilo, da bi zlepili zgornji in spodnji rob ter levi in desni rob. Rezultat bi bil torus.

Preslikava Arnoldove mačke je linearna preslikava torusa nase, ki ohranja ploščino. Ta preslikava je poseben primer *hiperboličnega automorfizma torusa*. Dana je s predpisom:

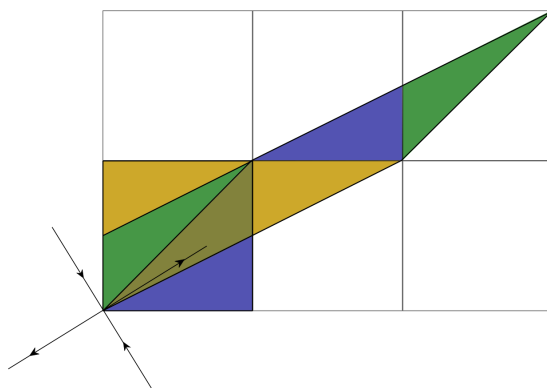
$$P : T \rightarrow T,$$

$$P(x, y) = (2x + y, x + y) \pmod{1}.$$

Taki preslikavi pripada matrika

$$A = \begin{bmatrix} 2 & 1 \\ 1 & 1 \end{bmatrix}.$$

Njeno delvanje lahko ponazorimo tudi s sliko



Slika 3: Grafična ponazoritev preslikave, vir: [5]

Z matriko smo mrežo najprej raztegnili v paralelogram, nato pa smo koordinate točk poračunali po modulu velikosti stranice. Mreža se je po tem postopku preslikala sam nase.

Z matriko A slikamo mrežo M_n oblike $\{(\frac{k}{n}, \frac{l}{n}); 0 \leq k, l < n\}$ za naravna števila k, l in n . Preslikavo smo opravljali samo na neki izbrani množici točk, zato je preslikava

v resnici permutacija te množice točk. Izkaže se, da preslikava Arnoldove mačke slika mrežo samo nase, torej je permutacija n^2 točk. Preslikava Arnoldove mačke ohranja vse mreže, če računamo po modulu 1. Permutacija ima periodo, preslikava Arnoldove mačke pa v celoti ni periodična, ker ima na različnih mrežah različne periode.

4 Potrebščine za nadaljnje raziskovanje

V nadaljem raziskovanju smo opazovali obnašanje period pri opravljanju preslikave z drugimi hiperboličnimi matrikami, torej matrikami, ki pripadajo drugim hiperboličnim avtomorfizmom torusa.

Pojmi, ki jih bomo uporabljali v nadaljevanju, so:

1. **matrika** - v nadaljevanju bomo z izrazom matrika poimenovali vse hiperbolične matrike.
2. **stranica mreže** - velikost stranice mreže (ali velikost stranice slike), ki jo slikamo. Mreže (ali slike) bodo predstavljene dolžino svoje stranice.
3. **perioda** - najmanjše število ponovitev preslikave, da je kompozitum teh preslikav identiteta.

5 Kako generirati hiperbolične matrike?

Vemo, da imajo hiperbolične matrike determinanto enako ena in lastni vrednosti, ki nista kompleksni števili in nista z enotske krožnice.

5.1 Matrike po formuli za hiperbolično matriko

V članku [2] je predstavljena formula za hiperbolično matriko.

$$M = \begin{bmatrix} 1 + ab & a \\ b & 1 \end{bmatrix}$$

S spreminjanjem realnih parametrov a in b lahko pridemo do neskončno mnogo hiperboličnih matrik.

Opomba 5.1. S formulo za hiperbolično matriko ne moremo priti do vseh hiperboličnih matrik.

5.2 Matrike z Evklidovim algoritmom

Še en način za iskanje koeficientov matrike, za katere bo veljalo $ad - bc = \pm 1$, je Evklidov algoritem. Evklidov algoritem sloni na naslednji zvezi

$$\gcd(a, b) = \gcd(b, a \bmod b), a > b.$$

Oznaka \gcd pomeni *greatest common divisor*. [6] Predstavljen je s preprostim rekurzivnim programom iz [6].

```
function gcd(a, b)
    if b = 0 return a
    else return gcd(b, a mod b)
```

Program po zgornji enačbi spreminja a in b dokler $b = 0$. Ko je $b = 0$ je a največji skupni delitelj števil a in b .

V skladu s [7] lahko program razširimo tako, da bo za dani par tujih si števil a in b vrnil par števil x in y , da bo $ax + by = 1$, a in b morata biti tuji si števili sicer bi bila determinata deljiva z njunimi skupnimi delitelji. Dobimo matriko oblike

$$M = \begin{bmatrix} a & -y \\ b & x \end{bmatrix}$$

Izvedba v Pythonu

```
def razširjen_gcd(a, b):
    """
    Razširjen Evklidov algoritem.
    Najde x in y, tako da velja a*x + b*y = gcd(a, b).
    Vrne gcd, x in y.
    """
    if b == 0:
```

```

    return a, 1, 0
gcd, x1, y1 = razširjen_gcd(b, a % b)
x = y1
y = x1 - (a // b) * y1
return gcd, x, y

```

Če za tuji si števili a, b iščemo par števil x, y , da bo veljalo $ax + by = 1$, vemo, da velja $ax + by = \gcd(a, b)$ in da je največji skupni delitelj tujih si števil 1. Za reševanje enačbe $ax + by = \gcd(a, b)$ lahko uporabimo razširjen Evklidov algoritem:

$$\begin{aligned}
 ax + by &= \gcd(a, b) \\
 bx_1 + (a \bmod b)y_1 &= \gcd(b, a \bmod b).
 \end{aligned}$$

Iz zgornje enačbe lahko izrazimo tudi x in y , če upoštevamo zvezo

$$a \bmod b = a - b\left(\left\lfloor \frac{a}{b} \right\rfloor\right).$$

$$\begin{aligned}
 bx_1 + (a \bmod b)y_1 &= \gcd(b, a \bmod b) \\
 bx_1 + (a - b\left(\left\lfloor \frac{a}{b} \right\rfloor\right))y_1 &= \gcd(a, b) = \gcd(b, a \bmod b) \\
 bx_1 + ay_1 - y_1b\left(\left\lfloor \frac{a}{b} \right\rfloor\right) &= \gcd(a, b) \\
 ay_1 + b(x_1 - y_1\left(\left\lfloor \frac{a}{b} \right\rfloor\right)) &= \gcd(a, b)
 \end{aligned}$$

Iz preurejene enačbe lahko preberemo, da je novi $x = y_1$ in da je novi $y = x_1 - y_1\left(\left\lfloor \frac{a}{b} \right\rfloor\right)$. Ko je $b = 0$, je a največji skupni delitelj a in b torej je a enak 1. Veljati mora

$$ax + by = 1,$$

zato je x lahko samo 1, za y pa obstaja neskončno mnogo rešitev. Izberemo si rešitev $y = 0$. Ostale rešitve dobimo tako, da x prištejemo kb in od y odštejemo ka (ali obratno) za $k \in \mathbb{R}$. Utemeljitev:

$$\begin{aligned}
 a(x + kb) + b(y - ka) &= ax + by \\
 ax + kba + by - kba &= ax + by \\
 ax + by + k(ba - ba) &= ax + by \\
 ax + by + 0k &= ax + by \\
 ax + by &= ax + by.
 \end{aligned}$$

Funkcija `razširjen_gcd` vrne vse matrike z determinanto ena. Ker poznamo zvezo med lastnimi vrednostmi in koeficienti matrike

$$P(\lambda) = \lambda^2 - (a + d)\lambda + ad - bc = 0, \quad \lambda_{1,2} = \frac{a + d \pm \sqrt{(a + d)^2 - 4(ad - bc)}}{2},$$

vemo, da ima polinom P samo realne rešitve zunaj enotske krožnice. Matrike oblike

$$M = \begin{bmatrix} a & b \\ c & d \end{bmatrix}$$

ki ne ustrezajo pogoju $(a + d)^2 - 4(ad - bc) > 0$, lahko zato izločimo. Če bi veljalo $(a + d)^2 - 4(ad - bc) = 0$, bi polinom imel dve enaki rešitvi, absolutna vrednost katerih je ena. Če bi veljalo $(a + d)^2 - 4(ad - bc) < 0$, bi polinom imel dve kompleksno konjugirani rešitvi, katerih produkt je 1.

Z Evklidovim algoritmom lahko najdemo vse hiperbolične matrike, a jih nismo poiskali, saj smo si izbrali rešitev $y = 0$.

6 Kako priti do periode?

Kot smo že povedali v poglavju 4 je perioda najmanjše število ponovitev preslikave, da je kompozitum teh preslikav identiteta. Periode ne moremo izračunati, do nje lahko pridemo samo s poskušanjem, torej jo lahko določimo pri dani mreži in matriki. Obstajata dva načina:

1. Dano mrežo lahko slikamo z dano matriko, dokler se mreža ne preslika v svoje prvotno stanje (tj. stanje, preden smo izvedli prvo preslikavo).
2. Dano matriko potenciramo, dokler niso koeficienti matrike po modulu stranice mreže enaki tistim matrike identitete. Če se to zgodi pri eksponentu n , je perioda enaka n .

Periode je praktično nemogoče določiti ročno. Za potrebe raziskovanja smo periode določali s programi v Pythonu. V nadaljevanju poglavja bomo predstavili tri glavne funkcije in njihovo delovanje. Vse funkcije, uporabljene za raziskovanje, so objavljene na githubu [9].

6.1 Ugotavljanje periode s ponavljanjem preslikave

Ponazorimo iskanje periode na primeru. Izberimo si matriko $A = \begin{bmatrix} 3 & 1 \\ 2 & 1 \end{bmatrix}$ in mrežo razsežnosti 3×3 .

Pri preslikavi bomo obravnavali samo točke A, B, C, D, E, F, G, H in I (glej sliko 3), saj gledamo preslikavo na torusu. Ker smo na torusu, ima mreža razsežnosti n točno n^2 celoštevilskih točk.

Primer preslikave

Izberimo si točko $F(2, 1)$. Stanje točke pred preslikavo označimo kot stanje 0.

- Stanje 0: $F(2, 1)$
- Izvedemo preslikavo:

1. Koordinate točke F preslikamo z matriko A : $\begin{bmatrix} 3 & 1 \\ 2 & 1 \end{bmatrix} \times \begin{bmatrix} 2 \\ 1 \end{bmatrix} = \begin{bmatrix} 7 \\ 5 \end{bmatrix}$

2. Nove koordinate točke izračunamo po modulu 3: $\begin{bmatrix} 7 \\ 5 \end{bmatrix} \bmod 3 = \begin{bmatrix} 1 \\ 2 \end{bmatrix}$

- Stanje 1: $F(1, 2)$.
- Ponavljamo postopek, dokler se F ne vrne nazaj v $F(2, 1)$.

Celoten postopek:

Stanje 0 : $F(2, 1)$

Stanje 1 : $F(1, 2)$

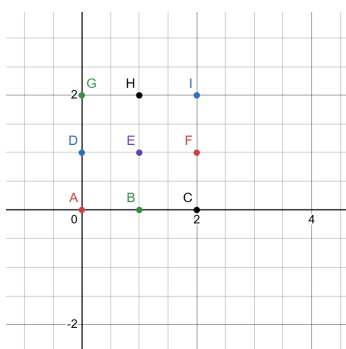
Stanje 2 : $F(2, 1)$

Preslikavo smo torej morali ponoviti dvakrat, da se je točka F preslikala nazaj sama vase. To pa ne pomeni, da je perioda enaka 2. Perioda je število ponovitev, po katerem se vse točke $A, B, C, D, E, F, G, H, I$ vrnejo nazaj v stanje 0.

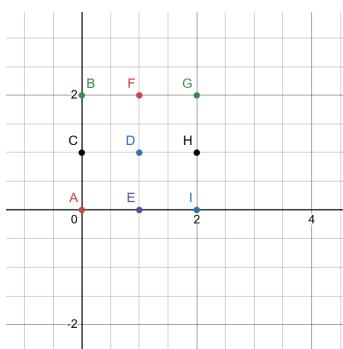
Tabela preslikav

	A	B	C	D	E	F	G	H	I
Stanje 0	(0,0)	(1,0)	(2,0)	(0,1)	(1,1)	(2,1)	(0,2)	(1,2)	(2,2)
Stanje 1	(0,0)	(0,2)	(0,1)	(1,1)	(1,0)	(1,2)	(2,2)	(2,1)	(2,0)
Stanje 2	(0,0)	(2,2)	(1,1)	(1,0)	(0,2)	(2,1)	(2,0)	(1,2)	(0,1)
Stanje 3	(0,0)	(2,0)	(1,0)	(0,2)	(2,2)	(1,2)	(0,1)	(2,1)	(1,1)
Stanje 4	(0,0)	(0,1)	(0,2)	(2,2)	(2,0)	(2,1)	(1,1)	(1,2)	(1,0)
Stanje 5	(0,0)	(1,1)	(2,2)	(2,0)	(0,1)	(1,2)	(1,0)	(2,1)	(0,2)
Stanje 6	(0,0)	(1,0)	(2,0)	(0,1)	(1,1)	(2,1)	(0,2)	(1,2)	(2,2)

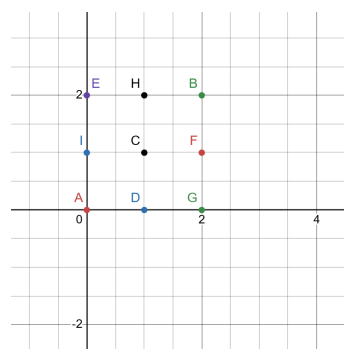
Tabela 1: Stanja točke v posameznih korakih preslikave



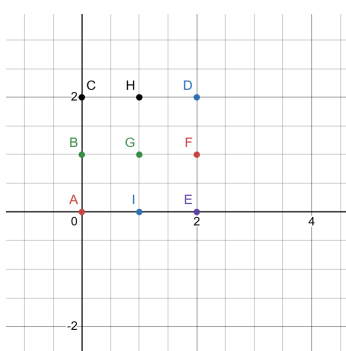
Slika 4: Stanje 0



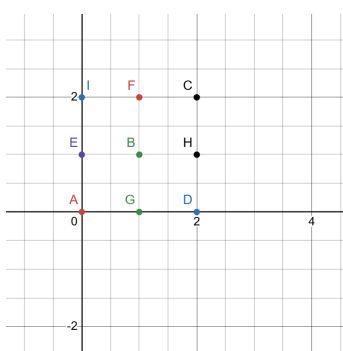
Slika 5: Stanje 1



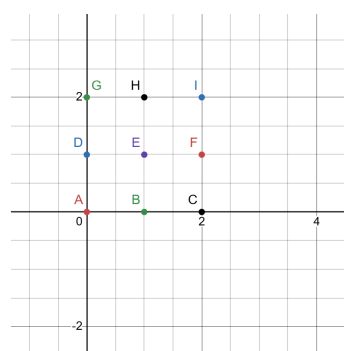
Slika 6: Stanje 2



Slika 7: Stanje 0



Slika 8: Stanje 1



Slika 9: Stanje 2

Iz tabele (glej tabelo 1) vidimo, da se vse točke razen A , F in H šele po šestih korakih vrnejo v stanje 0. Če se točki F in H vrneta v prvotno stanje že po 2 poskusih to pomeni, da se bosta po sodem številu poskusov vedno vrnila v stanje 0. Točka A se ne premika, kar je očitno že iz matričnega zapisa.

Ugotovitev je, da je perioda (torej število poskusov, ki so potrebni, da se točke vrnejo v stanje 0) enaka 6.

Predstavljeni postopek lahko izvajamo tudi s programom.
Izvedba v Pythonu

```
def premesaj(mreza, A):
    '''Funkcija sprejme mrezo n x n in preslikavo A
    vrednost vsakega mesta v mrezi prestavi na novo mesto
    tocka SPREMENI KOORDINATE ampak OHRANI VREDNOST'''
    n = len(mreza)
    nova_mreza = mreza_0(n)
    for i in range(n):
        for j in range(n):
            vrednost = mreza[i][j]
            y0, y1 = preslikava(A, (i,j), n)
            nova_mreza[y0][y1] = vrednost
    return nova_mreza
```

Zgornji program dobi mrežo, ki je v programu predstavljena kot dvodimenzionalna tabela z imenom *mreža*. Vsako mesto v tabeli ima neko vnaprej določeno vrednost, kar je enako, kot če imajo točke imena. Dano vrednost v tabeli si program zapomni, s preslikavo pa točki določi nove koordinate. Vrednost točke nato napiše v prazno dvodimenzionalno tabelo *nova_mreža*. Ta predstavlja stanje obravnavane mreže, ko jo enkrat preslikamo z matriko *A*.

Izvedba v Pythonu

```
def identiteta(n, A):
    '''funkcija sprejme velikost mreže n in preslikavo A
    na mreži n x n meša točke, dokler preslikana mreža ni
    enaka bazni mreži.
    funkcija vrne število iteracij do tega A^števec == I
    '''
    števec = 1
    bazna_mreza = mreza_x(n)
    mreža = mreza_x(n)
    while True:
        nova_mreza = premešaj(mreza, A)
        if nova_mreza == bazna_mreza:
            return števec
        else:
            mreža = nova_mreza
            števec += 1
```

Funkcija *identiteta* uporablja funkcijo *premešaj*. Stanje 0 shrani kot bazno mrežo. Program ponavlja zanko, dokler ni stanje mreže (*nova_mreža*) enako bazni mreži. Če pogoj ni izpolnjen, shrani novo mrežo kot *mreža*, da jo lahko še enkrat premeša.

Tak program je zelo počasen, saj mora z večanjem mreže obravnavati vse več podatkov.

6.2 Ugotavljanje periode s potenciranjem matrike

Ponovno ponazorimo iskanje periode na primeru $A = \begin{bmatrix} 3 & 1 \\ 2 & 1 \end{bmatrix}$ in mreži velikosti 3×3 .

Namesto da bi opazovali spreminjanje lege točk na mreži, bomo opazovali spreminjanje matrike. Vemo, da je ponavljanje preslikave pravzaprav kompozitum preslikave same s seboj. Če mrežo n -krat preslikamo z matriko A , je to enako, kot če bi jo enkrat preslikali z A^n . Ko bo izbrana matrika $A^n \pmod 3$ enaka matriki I (tj. identiteti), bomo iz potence lahko razbrali število ponovitev, ki je potrebno, da se točke, ki sestavljajo mrežo, vrnejo nazaj v stanje 0.

Načina iskanja period sta enakovredna, saj vrstni red računanja po modulu ter množenja ni pomemben. To lahko računsko pokažemo.

$$\begin{aligned} (nk + k_1)(nl + l_1) \pmod n &= (n^2kl + 2nkl_1 + k_1l_1) \pmod n \\ &= k_1l_1 \pmod n \end{aligned}$$

Izračunajmo še v drugem vrstnem redu in se prepričajmo, da je rezultat enak.

$$\begin{aligned} (nk + k_1)(nl + l_1) \pmod n &= ((nk + k_1) \pmod n)((nl + l_1) \pmod n) \\ &= k_1l_1 \pmod n \end{aligned}$$

Računanje po modulu je komutativna in asociativna operacija, zato je vseeno, ali računamo ostanke koeficientov matrike ali ostanke koordinat točk ter ali koeficiente matrike naprej potenciramo in potem računamo ostanek ali obratno.

Drugi korak preslikave bi v matričnem zapisu izgledal tako:

$$\begin{aligned} A^2 \pmod 3 &= \begin{bmatrix} 3 & 1 \\ 2 & 1 \end{bmatrix}^2 \pmod 3 \\ &= \begin{bmatrix} 11 & 4 \\ 8 & 3 \end{bmatrix} \pmod 3 \\ &= \begin{bmatrix} 2 & 1 \\ 2 & 0 \end{bmatrix}. \end{aligned}$$

Preslikavo moramo ponoviti tolikokrat, da bo matrika A enaka matriki I , tj. identiteti, torej matriki, ki vedno in v enem koraku preslika mrežo samo vase.

Komponente matrike se spreminjajo z vsakim korakom preslikave:

$$A = \begin{bmatrix} 3 & 1 \\ 2 & 1 \end{bmatrix}, A^2 = \begin{bmatrix} 2 & 1 \\ 2 & 0 \end{bmatrix}, A^3 = \begin{bmatrix} 2 & 0 \\ 0 & 2 \end{bmatrix}, A^4 = \begin{bmatrix} 0 & 2 \\ 1 & 2 \end{bmatrix}, A^5 = \begin{bmatrix} 1 & 2 \\ 1 & 0 \end{bmatrix}, A^6 = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}.$$

Tak način iskanja matrike lahko tudi sprogramiramo. Funkcijo smo poimenovali `identiteta_optimal`.

Izvedba v Pythonu

```
def identiteta_optimal(n, A):
    '''Funkcija s potencira (in računa po modulu) matriko, dokler ta
    ni enaka matriki identiteta. Potenca=perioda'''
    i_A = [[1, 0], [0, 1]]
    m_A = modul(A,n)
    števec = 1
    while True:
        if m_A == i_A:
            return števec
        else:
            m_A= modul(multiply_matrices(m_A, A),n)
            števec += 1
```

Funkcija v vhodu dobi podatek o velikosti mreže in matriki. Funkcija najprej koeficiente matrike poračuna z ostankom in preveri, ali je matrika identiteta. Če ni, jo potencira. Program ne zna potencirati matrik, zato je A matrika v stanju 0, ki se ne spreminja, m_A pa predstavlja matriko v vseh sledečih stanjih. Namesto da m_A potencira, jo pomnoži z A in koeficiente poračuna z ostankom. Program si potenco matrike zapomni s spremenljivko `števec`, ki je enaka potenci matrike.

Drugi način iskanja matrike je veliko boljši, saj lahko periodo poiščemo tako, da matriko potenciramo in pri tem spreminjamo samo koeficiente matrike. Način iskanja matrike s potenciranjem matrike tudi ni bistveno omejen z velikostjo mreže, saj je ta le številka, s katero računamo.

Funkcija `identiteta_optimal` je podlaga za funkcije za risanje grafov in nam je omogočila raziskovanje na mrežah zelo velikih razsežnosti. S tem smo pridobili podatke o več tisoč periodah.

7 Skrivanje sporočil

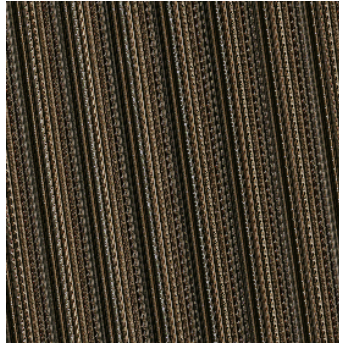
Preslikava Arnoldove mačke in drugi hiperbolični avtomorfizmi torusa so primerni za skrivanje sporočil, saj je njihovo delovanje nepredvidljivo. Računanje po modulu je v enosmer preprosto, saj je preprosto izračunati ostanek pri deljenju. Iz ostanka pa je veliko težje priti nazaj do številke, saj imajo različna števila lahko enak ostanek. Z računanjem po modulu torej izgubimo velik del informacije. Zaradi tega je sporočilo, brez da bi poznali matriko, praktično nemogoče dešifrirati. Če bi prejemnik slike s sporočilom poznal matriko in index iterata, na katerem je skrito sporočilo, bi lahko sporočilo dešifriral brez poznavanja periode. S formulo predstavljeno v trditvi 2.7, bi poiskal inverzno matriko matriki A , tj. X in z X^{index} mod stranica slike preslikal sliko in tako razkril sporočilo.

S pomočjo spodnjih slik prikažimo skrivanje sporočila. Sporočilo bomo skrili na sliko veliko 280×280 pikslov. Uporabili bomo matriko

$$A = \begin{bmatrix} 50 & 7 \\ 7 & 1 \end{bmatrix}.$$



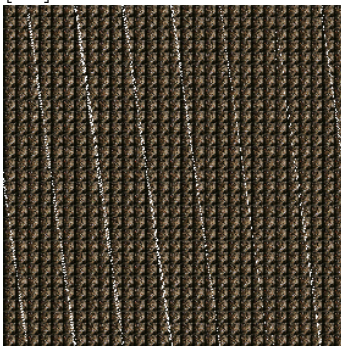
Slika 10: Stanje 0, vir: [10]



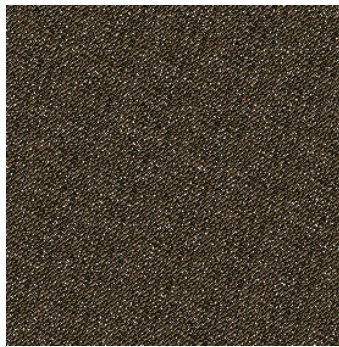
Slika 11: Stanje 1



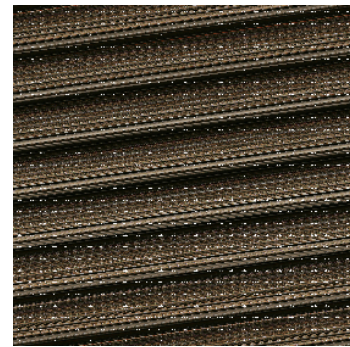
Slika 12: Stanje 2



Slika 13: Stanje 3



Slika 14: Stanje 4



Slika 15: Stanje 5



Slika 16: Stanje 6

Na drugi iterat smo skrili sporočilo "mačka" in nato nadaljevali preslikavo. Če bi prejemnik skritega sporočila prejel sliko stanje 6 in ključ, ki je matrika in potenca, bi sliko v stanju 6 preprosto preslikal z $X^{potenca}$ mod stranica mreže slike, če je $X = A^{-1}$, ter našel sporočilo.

Poglejmo, kako deluje funkcija `skrij_sporočilo`.

Izvedba v Pythonu

```
def skrij_sporočilo(image_path, A, index, sporočilo, color):
    '''A je matrika, ki slika; n je stranica mreže, index pove,
    kateri iterat želim da se prikaže'''
    with Image.open(image_path) as im:
        prej = im
```

```

n = im.size[0] # Get the width of the image
print('Mreža je: ',n)
for k in range(1, identiteta_optimal(n, A) + 1):
    print('začenjam preslikavo: ')
    potem = preslikava(prej, A, n)
    print('Končal sem')
    display_image(prej,1)
    if k == index:
        modified_potem = napiši_txt(potem, sporočilo, color)
        modified_potem.save('slika{}.png'.format(k), 'PNG')
        print(f"Saved image for index {k}")
        prej = modified_potem
        #display_image(prej,5)
    else:
        potem.save('slika{}.png'.format(k), 'PNG')
        print(f"Saved image for index {k}")
        prej = potem

```

Program dobi pot do mape, kjer je slika, sliko, matriko, nek indeks in podatke o sporočilu. Sliko preslika z dano matriko. Preslikavo ponovi tolikokrat, da je ta identiteta. Indeks pove, v katerem koraku poleg tega, da sliko preslika, nanjo napiše tudi sporočilo z zeleno barvo. Po tem ko program skrije sporočilo, nadaljuje preslikavo. Postopek skrivanja sporočila je viden na slikah med sliko 10 in sliko 16.

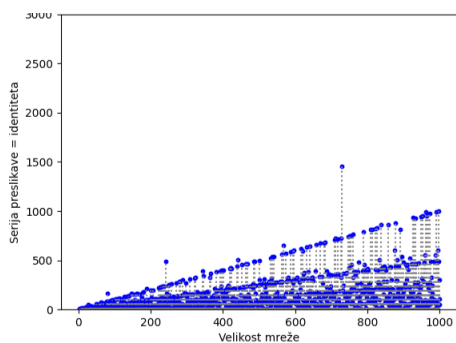
8 Analiza period

S funkcijo `identiteta_optimal` smo lahko poiskali veliko različnih period pri veliko različnih mrežah in matrikah. Začeli smo se spraševati ali lahko med navidez kaotičnimi periodami najdemo kakšen red.

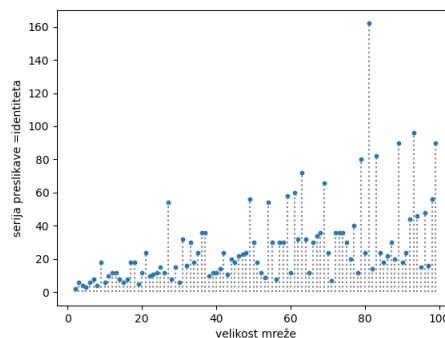
8.1 Grafi period v odvisnosti od stranice mreže

Z matriko smo po zgledu iz članka [2] narisali graf. Na x -os smo postavili velikosti stranice mreže, na y -osi pa periode, določene pri dani mreži in matriki. Vse periode na grafu so dolčene z isto matriko, zato lahko rečemo, da graf pripada določeni matriki. Na grafih lahko opazujemo, kako se perioda spreminja, ko povečujemo stranico mreže. Poglejmo grafe, ki pripadajo matriki

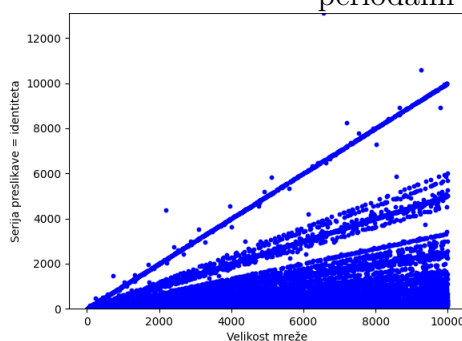
$$M = \begin{bmatrix} 3 & 1 \\ 2 & 1 \end{bmatrix}.$$



Slika 17: Graf pri dani matriki s 1000 periodami



Slika 18: Graf pri dani matriki s 100 periodami



Slika 19: Graf pri dani matriki z 10000 periodami

Na grafih s 1000 in 10.000 periodami opazimo jasni premici $y = x$ in $y = \frac{x}{2}$, eno zelo izstopajočo vrednost ter območje, kjer so točke zelo zgoščene. Premici sta jasno vidni, saj so preostale periode zgoščene pod njima. Na grafu s 1000 periodami premico $y = x$ sestavlja 1,2 % vseh točk, torej 12 točk, premico $y = \frac{x}{2}$ pa 210. Razmerje $\frac{\text{perioda}}{\text{stranica}} = \frac{1}{2}$ je najpogostejše razmerje med periodo in stranico na tem

grafu, je modus. Območje zgoščenih točk na grafu s 1000 periodami bi lahko opisali kot območje pod premico $y = x$, saj se pod njo nahaja 925 točk, tj. natanko 92,5 %vseh period. Iz grafa ni jasno razvidno, kako je teh 925 točk razporejenih, saj so zelo zgoščene. Manjše vrednosti se zato na grafu s samo 100 periodami bolje vidijo.

Za grafe povejmo osnovne statistike.

1. Graf s 100 periodami:
 - (a) Povprečna perioda: 28
 - (b) Najpogostejša perioda (modus): 12
 - (c) Mediana: 20
 - (d) Največja vrednost: 162
 - (e) Najmanjša vrednost: 2
2. Graf s 1000 periodami:
 - (a) Povprečna perioda: 168
 - (b) Najpogostejša perioda (modus): 36
 - (c) Mediana: 104
 - (d) Največja vrednost: 1458 izračunana pri stranici mreže 949
 - (e) Najmanjša vrednost: 2
3. Graf s 10000 periodami:
 - (a) Povprečna perioda: 1255
 - (b) Najpogostejša perioda (modus): 60
 - (c) Mediana: 612
 - (d) Največja vrednost: 13122
 - (e) Najmanjša vrednost: 2

Grafe smo risali z dvema družinama matrik. Prva družina matrik je bila generirana s pomočjo formule za hiperbolično matriko, ki je predstavljena v članku [2]. Druga družina matrik je bila generirana z uporabo Evklidovega algoritma.

Grafi matrik pridobljenih s pomočjo formule za hiperbolično matriko
 Po vzorcu formule za hiperbolično matriko smo tako ustvarili množico matrik, imenujmo jo *množica matrik*, v kateri so vse matrike, za katere velja $a, b \in [1, 10]$, $b > a$ in $a, b \in \mathbb{Z}$. Če se ne bi omejili na $b > a$, bi množica vsebovala tako matriko

$$M_1 = \begin{bmatrix} 1 + ab & a \\ b & 1 \end{bmatrix}$$

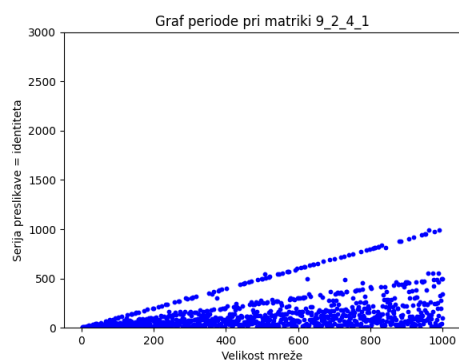
kot njeno transponiranko

$$M_2 = \begin{bmatrix} 1 + ab & b \\ a & 1 \end{bmatrix}.$$

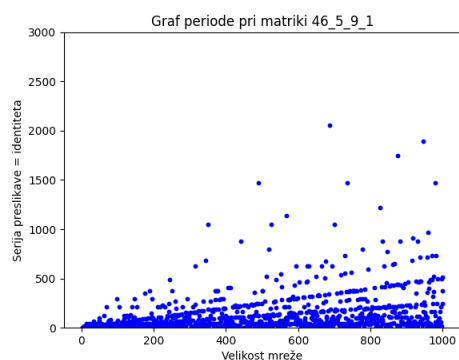
Matriki M_1 in M_2 imata enaki lastni vrednosti, zato je dovolj, da opazujemo samo eno od niju. Za $a = 1$ dobimo 10 matrik, za $a = 2$ jih dobimo 9 itd., torej *množica matrik* skupaj vsebuje 55 matrik. Pripravili smo tudi *množico mrež*, ki vsebuje vse mreže s stranico veliko med 2 in 1001, kar pomeni, da imajo mreže med 4 in 1.002.001 elementov. Opomnimo, da bodo na grafih in v nadaljnjem besedilu mreže predstavljene z dolžino svoje stranice, imenovano *stranica mreže*. Za vsako matriko iz *množice matrik* smo poiskali periodo za vsako tako mrežo. Zbrali smo jih v *množico period*, ki vsebuje 55.000 celoštevilskih elementov. Periode, ki smo jih določili z dano matriko A pri vseh mrežah iz *množice mrež*, smo ponazorili na grafu. To smo naredili za vse matrike. Rezultat je 55 grafov. Vsaki matriki torej pripada en graf. Grafi so poimenovani po matriki, s katero smo slikali mreže (v imenu si sledijo koeficienti $1+ab_a_b_1$).

Grafi in njihove posebnosti.

Nekateri grafi imajo eno do dve jasni premici ter nobene izstopajoče vrednosti. Drugi nimajo jasnih premic, imajo praktično samo izstopajoče vrednosti.

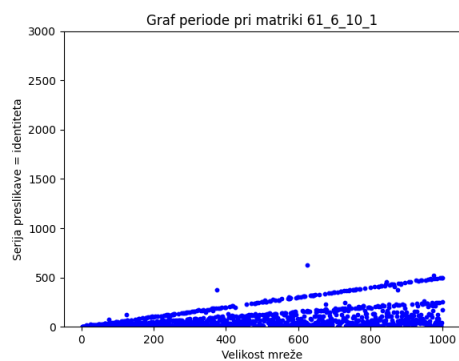


Slika 20: Graf brez izstopajočih vrednosti

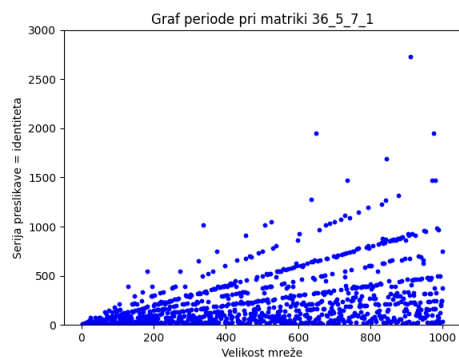


Slika 21: Graf z veliko izstopajočimi vrednostmi

Primerjajmo graf z najmanjšim povprečjem period in graf z največjim povprečjem period.

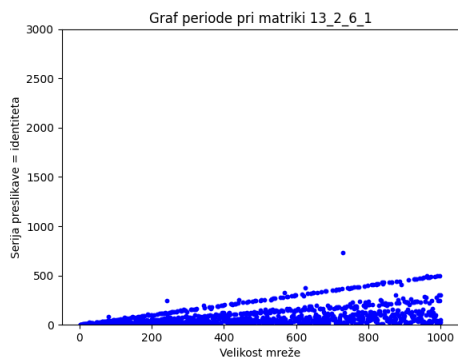


Slika 22: Graf z najmanjšim povprečjem

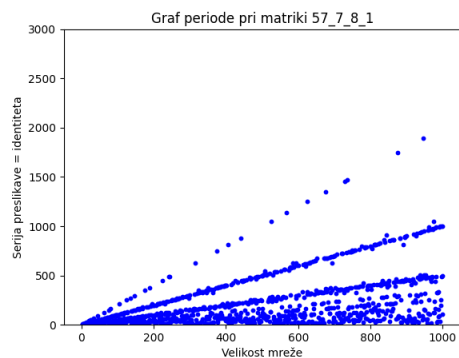


Slika 23: Graf z največjim povprečjem

Primerjajmo graf z največjim deležem period, ki ležijo pod premico $y = x$ in graf najmanjšim deležem period, ki ležijo pod premico $y = x$.

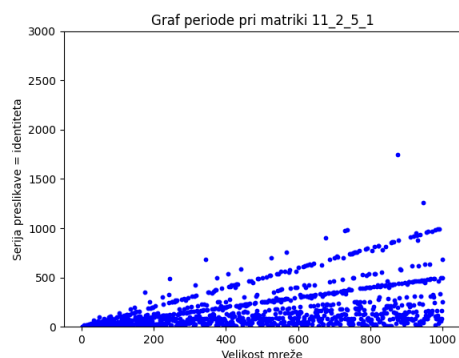
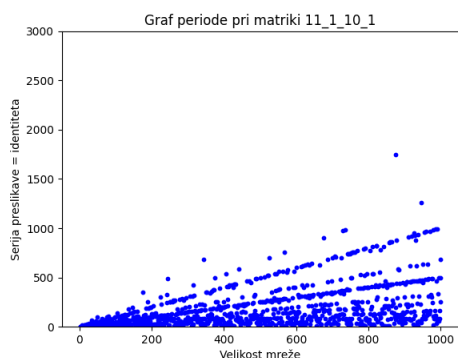
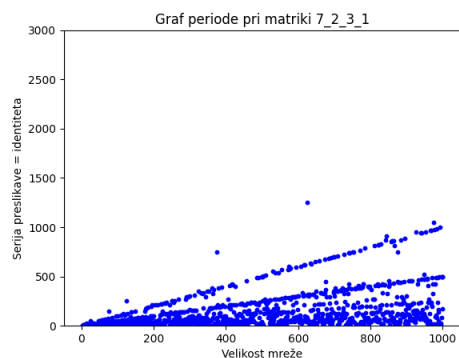
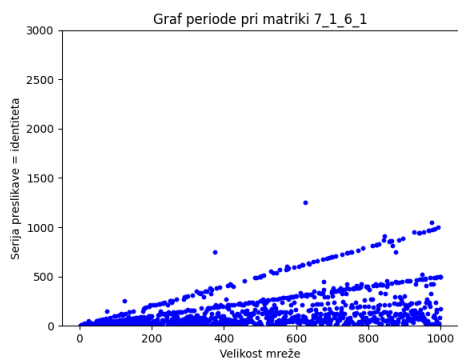


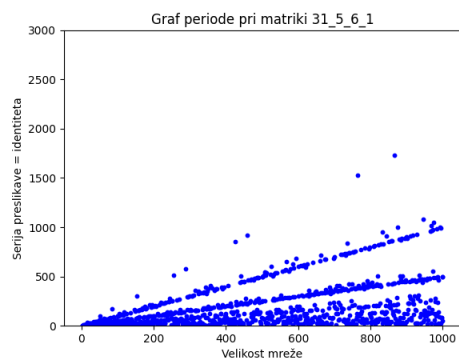
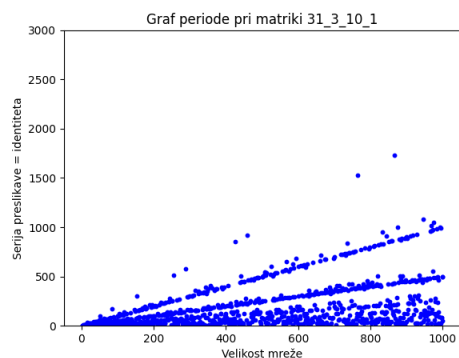
Slika 24: Graf z največjim procentom period pod $y = x$



Slika 25: Graf z najmanjšim procentom period pod $y = x$

Pokaže se tudi, da imajo določene matrike v množici matrik popolnoma enak graf (vsaka perioda, določena pri neki mreži, se ujema). Skupno vsem parom grafov je, da imata obe matriki v paru enak produkt a in b koeficientov matrike $A = \begin{bmatrix} 1 + ab & a \\ b & 1 \end{bmatrix}$ (seveda obstajajo tudi pari matrik, ki ustrezajo danemu pogoju, vendar nimajo enakega grafa). Poglejmo take grafe s 1000 periodami:





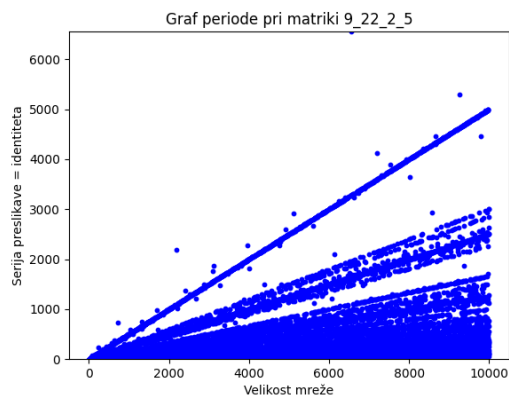
Poglejmo nekaj statističnih podatkov o *množici period*:

1. Mediana: 90
2. Modus: 12
3. Prvi kvartil (Q1): 36
4. Tretji kvartil (Q3): 216
5. Minimalna vrednost: 1
6. Maksimalna vrednost: 2850

Opozorimo na zanimivost, da obstajajo samo 4 grafi s 1000 periodami, ki prikazujejo periode v odvisnosti od mreže, kjer modus ni večkratnik števila 6.

Grafi matrik, pridobljenih z Evklidovim algoritmom

Narišemo lahko tudi grafe matrik, pridobljenih z Evklidovim algoritmom. Poglejmo si primer grafa (glej sliko 26)



Slika 26: graf od 9_22_2_5.

Pripravili smo množico matrik, pridobljenih z Evklidovim algoritmom, imenujmo jo *e-množica matrik*. To so celoštevilске matrike, dobljene s sledečim algoritmom

```

for a in range (18):
    for b in range (a,17):
        if sta_tuji_stevili(a,b) :
            x,y = find_xy(a,b)
            A=[[a,b], [-y,x]]
            if (((a+x)*(a+x))-4)>0):
                tabela_matrik.append(A)

```

Program najde 48 hiperboličnih matrik. Množica mrež ostane enaka. *E-množico period* dobimo tako, da z vsako matriko iz *e-množice matrik* slikamo vsako mrežo iz *množice mrež*.

Poglejmo, kakšna je *e-množica period*:

1. Mediana: 92
2. Modus: 60
3. Prvi kvartil (Q1): 40
4. Tretji kvartil (Q3): 220
5. Minimalna vrednost: 1
6. Maksimalna vrednost: 2850

V primerjavi s prvotno *množico period* ima *e-množica period* petkrat večji modus in podobno mediano, Q_1 ter Q_3 . Edina večja sprememba torej je drugačen modus.

8.2 Grafi period v odvisnosti od matrike

Tudi v tem poglavju bodo matrike, potrebne za risanje grafov določene na dva načina.

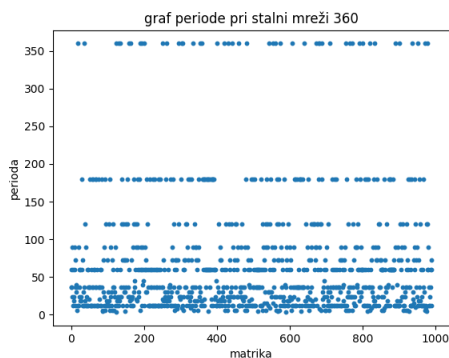
Grafi v odvisnosti od matrik, pridobljenih s formulo za hiperbolično matriko

Pripravimo si novo množico matrik, imenujmo jo kar *nova množica matrik* po vzorcu formule za hiperbolično matriko, v kateri so vse hiperbolične matrike oblike

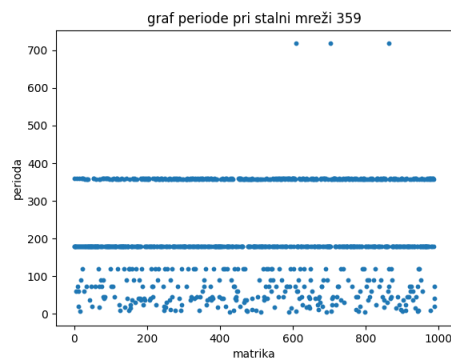
$$M = \begin{bmatrix} 1 + ab & a \\ b & 1 \end{bmatrix},$$

za katere velja $a, b \in [1, 45]$, $b > a$ in $a, b \in \mathbb{N}$. *Nova množica matrik* vsebuje 990 matrik. Narisali bomo graf, ki bo pripadal določeni stranici mreže, imenujemo jo *stalna stranica mreže*. Na x -osi bodo matrike iz *nove množice matrik*, prikazane s števili (število pove, katera po vrsti je bila matrika generirana), na y -osi pa perioda, določena pri stalni stranici in trenutni matriki. Na grafu bo 990 period.

Primer grafov, kjer so vse matrike generirane po formuli za hiperbolično matriko:



Slika 27: Graf pri stalni stranici mreže 360



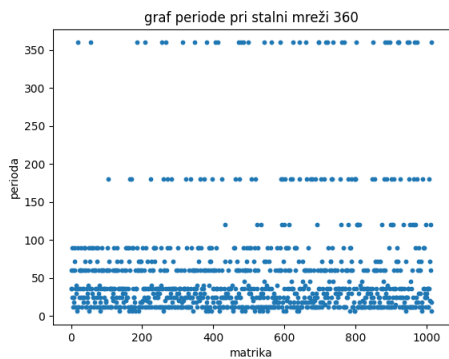
Slika 28: Graf pri stalni stranici mreže 359

Najpogostejša perioda je iz tega grafa veliko bolj očitna, saj je tam najjasnejša črta. Iz takih grafov je jasno razvidno, da če neko mrežo slikamo z različnimi matrikami, opazimo, da se ne pojavi veliko različnih period. Število različnih period tudi ni pogojeno z velikostjo stranice mreže, kar je očitno iz grafov. Na grafu (glej sliko 27) pri stranici mreže 360 je 21 različnih vrednosti, na grafu (glej sliko 28) pri stranici mreže 359 pa 24. Na grafu se kot "luknje" vidijo matrike, ki slikajo drugače kot večina matrik.

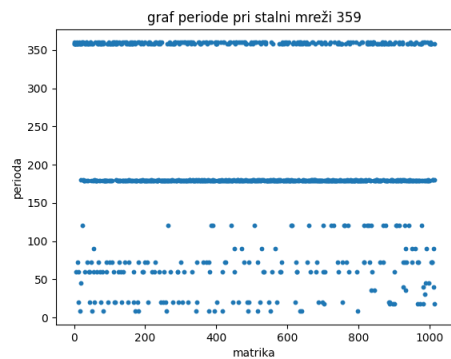
Grafi v odvisnosti od matrik pridobljenih z Evklidovim algoritmom

Množico matrik, ki jih postavimo na x -os, je lahko pridobljena tudi z Evklidovim algoritmom. V *e-novi množici matrik*, so vse celoštevilске matrike pridobljene z že predstavljenim algoritmom, kjer velja $a \in (0, 61)$ in $b \in (a, 62)$. Rezultat je 1015 matrik. Spet smo narisali graf, ki pripada določeni stranici mreže. Na x -osi so matrike iz *e-nove množice matrik* prikazane s števili (število pove, katera po vrsti je bila matrika generirana), na y -osi pa perioda, določena pri stalni stranici in trenutni matriki. Na grafu bo prav tako 1015 period.

Primer grafa, kjer so vse matrike generirane z Evklidovim algoritmom:



Slika 29: Graf pri stalni stranici mreže 360



Slika 30: Graf pri stalni stranici mreže 359

9 Ali je perioda napovedljiva?

V poglavju 8 smo si ogledali kakšna je perioda glede na mrežo in matriko. V tem poglavju pa bomo opazovali, kako je perioda odvisna od mreže in matrike ter ali lahko iz ugotovljenega sklepamo na kakšno pravilo.

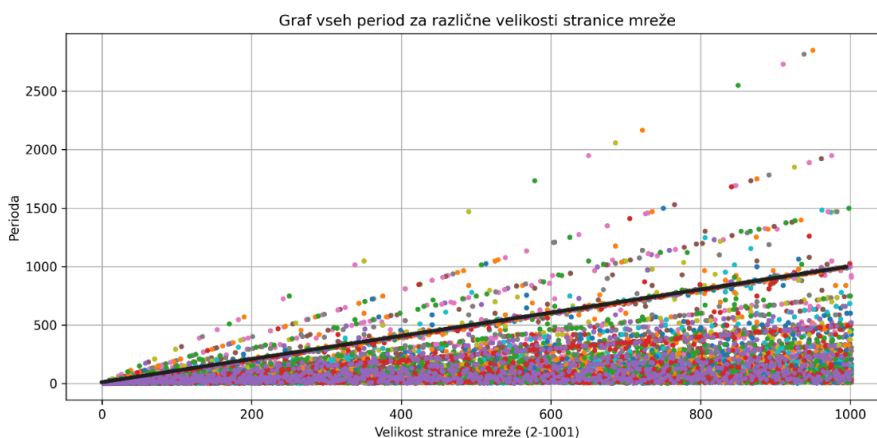
9.1 Zveza med stranico mreže in periodo

V članku [2] so predstavljene določene zveze med periodo in mrežo, dokazane za matriko

$$M = \begin{bmatrix} 2 & 1 \\ 1 & 1 \end{bmatrix}.$$

Ena izmed dokazanih zvez je, da perioda nikoli ni več kot trikratnik stranice mreže, pri kateri je bila določena. Pri vseh opazovanih periodah tudi mi nismo našli periode, ki bi bila več kot 3-kratnik stranice mreže, pri kateri bi bila določena.

Da bi si lažje predstavljali zveze med periodami in stranicami mrež, ponazorimo vse periode iz množice period na enem grafu (glej slika 31). Za dan x na x -os postavimo stranico mreže, na y -os pa vseh 55 period, vsaka določena z eno od matrik iz množice matrik. Lahko si predstavljamo, da smo vseh 55 grafov ponazorili na enem grafu.



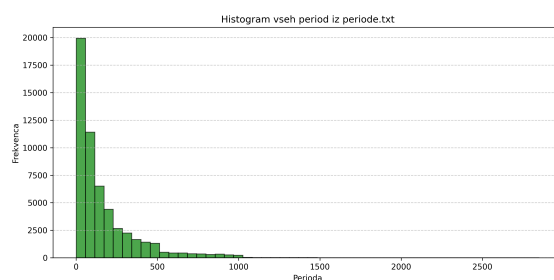
Slika 31: Graf vseh period

Črna črta na grafu (glej 31) je ponazoritev naraščanja stranic mrež kot premice. Zaradi črte zelo jasno opazimo, kako večina period sčasoma pade pod to premico. Nekaj točk sicer sestavlja premice, ki jih vidimo na grafu, ampak izpostavimo, da se polovica period nahaja pod premico $y = 90$ in kar 90,5 % pod premico $y = x$.

Majhnost period ponazorimo še s histogramom (glej sliko 32)

Točke na grafu (glej 31) ležijo znotraj nekega trikotnika. Iz tega lahko sklepamo, da se velike periode pojavijo pri velikih mrežah. Zanima nas, kakšna so razmerja med periodami in stranicami mrež, pri kateri so bile periode določene. Uvedimo novo količino *koeficient*, ki je definirana kot

$$\text{koeficient} = \frac{\text{perioda}}{\text{stranica mreže}}$$



Slika 32: Graf vseh period

9.2 Množica koeficientov

Množico koeficientov dobimo tako, da vsako periodo v *množici period* delimo s stranico mreže, pri kateri je bila določena. *Množica koeficientov* je tako enakih razsežnosti kot *množica period*. Z raziskovanjem koeficientov lahko raziskujemo neodvisno od tega, da imajo večje mreže večje periode.

Poglejmo si nekaj podatkov o *množici koeficientov*:

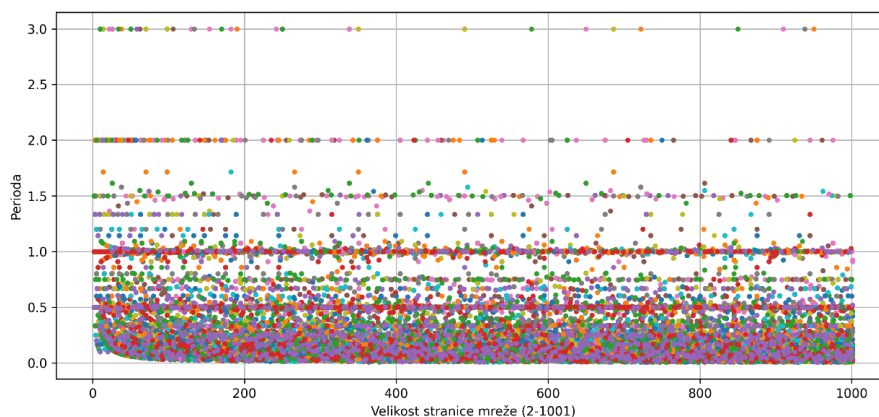
1. Mediana: 0,25
2. Modus: 1,0
3. Q1: 0,11, Q3: 0,50, IQR: 0,39
4. Najmanjši koeficient: 0,01
5. Največji koeficient: 3,0

Iz kvartilov razberemo, da so koeficienti v glavnem majhni. Poglejmo kako majhni so zares v tabeli koeficientov (glej tabelo 2). Opomba: deleži so zaokroženi na dve decimalki.

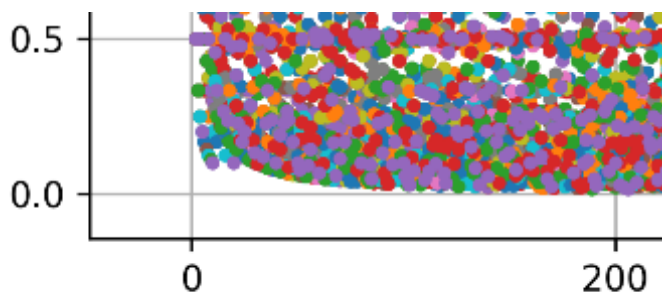
Koliko koeficientov k ustreza nekemu pogoju		
Pogoj	število	delež
$k < 0,01$	0	0,00 %
$k < 0,1$	12444	22,71 %
$k < 1$	49429	90,19 %
$1 < k < 2$	3830	6,96 %
$2 < k < 3$	0	0,00 %
$k = 0,5$	770	1,40 %
$k = 1$	1120	2,04 %
$k = 2$	379	0,69 %
$k = 3$	47	0,09 %

Tabela 2: Tabela koeficientov

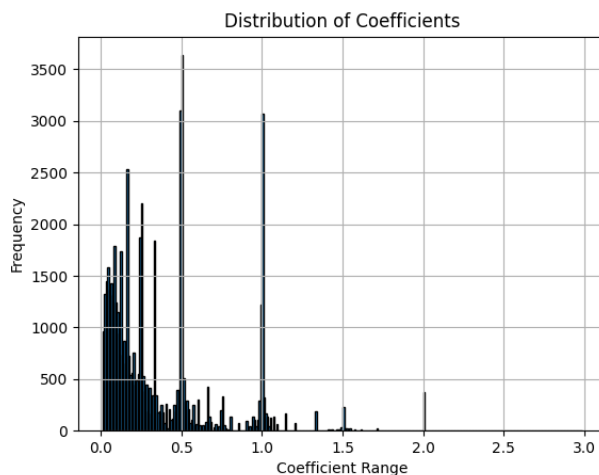
Iz podatkov v tabeli koeficientov (glej tabela 2) lahko splošimo, da večje izstopajoče vrednosti sestavljajo premice. Samo 2,04 % vseh koeficientov je enakih



Slika 33: Graf vseh koeficientov



Slika 34: Graf vseh koeficientov



Slika 35: Histogram

1, kljub temu da premico $y = x$ vidimo na skoraj vsakem grafu. Večina period je manjših od svoje stranice mreže. Iz podatkov razberemo, da je polovica koeficientov manjših ali enakih 0,25. V množici koeficientov je 90 % koeficientov manjših od 1 in 22 % manjših od 0,1. Vemo tudi, da za vsak posamičen graf periode v odvisnosti od stranice mreže velja, da je zagotovo vsaj 79 % period pod premico $y = x$. Koeficient nikoli ni k -kratnik stranice mreže, če je $k \in (2, 3)$. Povemo lahko tudi, da je so

koeficienti v glavnem večji pri manjših mrežah. To se lepo vidi na grafu (glej sliko 33 in sliko 34), kjer je pri majhnih mrežah prostor med 0 in približno 0,25 prazen, medtem ko v nadaljevanju ni. Opozorimo, da je najmanjši koeficient enak 0,01 (zakroženo). Koeficient je izračunan po formuli iz katere sledi, da je perioda produkt koeficienta in stranice mreže. Perioda mora biti celo število, zato mora biti stranica mreže, pri kateri bi lahko določili tak koeficient, vsaj 100. Perioda je enaka 1 samo če je matrika identiteta, zato mora stranica mreže biti vsaj 200 ali več. Zaradi tega razmisleka vemo, da zelo majhni koeficienti ne morejo nastopati pri majhnih mrežah. Če pogledamo prvih 500 period iz vseh 55 grafov skupaj, je povprečje koeficientov enako 0,69, če pa pogledamo drugih 500 period, je povprečje koeficientov 0,32, kar je več kot dvakrat manj. Pravzaprav grejo povprečja od večjih proti manjšim. Na spodnjem seznamu piše povprečje prvih 50 koeficientov, torej prvih 50 koeficientov, dobljenih iz prvih 50 period vsakega od 55 grafov skupaj.

1. povprečje prvih 50 koeficientov : 0,69
2. povprečje prvih 100 koeficientov: 0,59
3. povprečje prvih 500 koeficientov: 0,43
4. povprečje zadnjih 500 koeficientov: 0,32
5. povprečje zadnjih 100 koeficientov: 0,30
6. povprečje zadnjih 50 koeficientov: 0,31

Iz histograma (glej 35) lahko razberemo, da so koeficienti zelo neenakomerno razporejeni, a venadrlje zgoščene na intervalu $[0, 1]$.

9.3 K-means clustering

Z metodo **k-mean clustering** smo poskušali matrike, določene s formulo za hiperbolično matriko, razvrstiti v skupine glede na njihove periode in koeficiente. Periode in koeficienti grafa s 1000 periodami ene matrike so predstavljeni kot 1000-dimenzionalni vektor. Ta metoda po nekem algoritmu razvrsti vektorje v zelene skupine, sešteje razdalje med vsemi vektorji v neki skupini in vsoto deli z njihovim številom. To naredi za vsako skupino. Recimo temu, da določi vsoto skupine. Na koncu sešteje vsoto razdalj v vsaki skupini. Poskuša vse možne razvrstitve v skupine in vrne tisto razvrstitev, kjer je vsota razdalj v vsaki skupini najmanjša. [8]

Program je matrike glede na periode razporedil v tri skupine. Matrike in njihovi grafi znotraj iste skupine na videz nimajo nobenih opaznih podobnosti. Lahko izpostavimo, da matrike oblike

$$M = \begin{bmatrix} 1 + ab & a \\ b & 1 \end{bmatrix},$$

za katere velja, da je produkt ab enak, pogosto pristanejo v isti skupini.

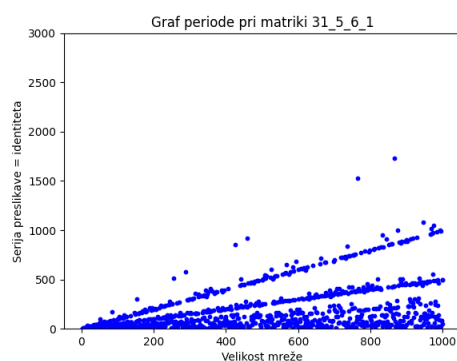
Matrike smo glede na koeficiente razporedili v 2, 5 in 10 skupin. Med grafi, ki pripadajo matrikam iste skupine, ni očitnih podobnosti. Natančneje smo pogledali mediane, moduse, delež koeficientov, manjših od 1, pri grafih matrik znotraj neke

skupine, a nismo našli nobene podobnosti. Edina zanimivost je, da ko smo matrike razporedili v 5 skupin, sta v eni skupini pristala samo dva popolnoma enaka grafa, določena pri matrikah, ki se ujemata v produktu ab .

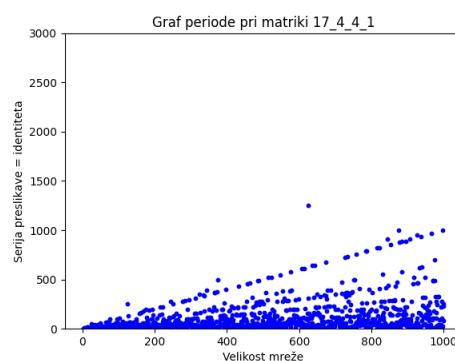
9.4 Raznolikost period

Opazili smo, da neka perioda lahko nastopi več kot enkrat. Če pri določenih pogojih nastopa malo različnih period, bi lahko periodo lažje napovedali. Pogledajmo, kako raznolike so periode v danih okoliščinah.

Če pogledamo 55 grafov periode v odvisnosti od mreže, kjer stranice mrež tečejo od 2 do 1001, in pogledamo, koliko različnih period je na grafu, ugotovimo, da je na grafu s 1000 periodami najmanj 215 in največ 314 različnih period.



Slika 36: Graf z 314 različnimi periodami



Slika 37: Graf z 215 različnimi vrednostmi

Če pogledamo grafe v odvisnosti od matriki, pa ugotovimo, da če pogledamo mreže velikosti med 50 in 105 (izbrano popolnoma naključno), na posamičnem grafu nastopa najmanj 9 in največ 25 različnih vrednosti.

Ko primerjamo grafe period v odvisnosti od mreže in grafe period v odvisnosti od matrike, ko je na obeh grafih 1000 period, ugotovimo, da obstaja občutno manj različnih period na grafih periode v odvisnosti od matrike kot na grafih periode v odvisnosti od mreže. Zaključek je, da lahko periodo lažje napovemo, če poznamo stranico mreže, kot če poznamo matriko.

9.5 Uporabnost v steganografiji

Navežimo ugotovitve na uporabnost preslikave v steganografiji. Če dobimo sliko s skritim sporočilom, ugotovimo da obstaja neko relativno majhno število možnih period, ki so določene z recimo eno izmed 1000 matrik iz množice matrik. S pravilno uganjeno periodo in stranico mreže ne moremo določiti matrike, brez matrike pa skritega sporočila ne moremo dešifrirati.

10 Zaključek

V raziskovanju smo si ogledali matematične značilnosti preslikave Arnoldove mačke. Ugotovili smo, da je preslikava hiperbolični avtomorfizem torusa, ki je lahko ponazorjen s celoštevilsko matriko, ki ima realne lastne vrednosti izven enotske krožnice. Ogledali smo si, kako preslikava slika celoštevilsko mrežo in kakšno je najmanjše število ponovitev preslikave (perioda), da bo kompozitum preslikave same s seboj identiteta. Opisali smo dva načina, kako pridemo do periode.

Glavno vprašanje je bilo, ali lahko periodo napovemo, če poznamo matriko in stranico mreže. Opazovali smo razmerja med stranico mreže in periodo ter ugotovili, da je najpogostejše razmerje med periodo in stranico mreže enako $1 : 1$. Nismo našli povezave med razmerji med periodo in stranico mreže ter matriko, pri kateri je bila perioda določena

Ugotovili smo, da se v množici period, določenih pri stalni mreži, a različnih matrikah matrikami, pojavi občutno manj različnih period, kakor pri v množici period določenih s stalno matriko, a različnimi mrežami. Najpomembnejša ugotovitev je, da nismo našli povezave med periodo in matriko.

Če iščemo sporočilo, skrito na neko sliko, in poznamo periodo in velikost slike, še zmeraj obstaja več različnih matrik, pri katerih je bila ta perioda lahko določena. Skritega sporočila ne moremo dešifrirati, ne da bi sliko s skritim sporočilom slikali z določeno matriko. Podatek o velikosti mreže nas sicer nekoliko omeji. Vemo na primer, da perioda nikoli ni več kot trikratnik stranice mreže, vendar opomnimo, da imamo na opazovanih grafih periode v odvisnosti od matrike veliko različnih matrik, a malo različnih period. Ne pozabimo tudi, da smo se omejili na neko izbrano množico matrik, množica vseh hiperboličnih matrik pa je neskončna. Ker povezave med matriko, velikostjo mreže in periodo nismo našli, lahko zaključimo, da je preslikava kaotična.

Literatura

- [1] Mitja Lakner, Peter Petek, Marjeta Škrapin: Diskretni dinamični sistemi, str. 89
- [2] Mitja Lakner, Peter Petek, Marjeta Škrapin: Vrnitev Arnoldove mačke, Obzor-
nik mat. fiz. 62 (2015) 2
- [3] <https://users.fmf.uni-lj.si/kosir/poucevanje/0910/alg1-fm.html>
- [4] https://en.wikipedia.org/wiki/Vladimir_Arnold
- [5] https://en.wikipedia.org/wiki/Arnold%27s_cat_map
- [6] https://sl.wikipedia.org/wiki/Evklidov_algoritem
- [7] https://en.wikipedia.org/wiki/Euclidean_algorithm
- [8] https://en.wikipedia.org/wiki/K-means_clustering
- [9] [https://github.com/Hana-Perman/Arnoldova-macka/blob/main/
arnoldova_macka](https://github.com/Hana-Perman/Arnoldova-macka/blob/main/arnoldova_macka)
- [10] stock.snap.io