UNIVERSIDADE FEDERAL DE PERNAMBUCO

CENTRO DE TECNOLOGIA E GEOCIÊNCIAS

DEPARTAMENTO DE ELETRÔNICA E SISTEMAS

PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA ELÉTRICA

ARQUIMEDES JOSÉ DE ARAÚJO PASCHOAL

TRANSFORMADAS EM CORPOS FINITOS E CÓDIGOS DE BLOCO LINEARES: NOVAS DEFINIÇÕES E CENÁRIOS DE APLICAÇÃO

Recife

2018

ARQUIMEDES JOSÉ DE ARAÚJO PASCHOAL

TRANSFORMADAS EM CORPOS FINITOS E CÓDIGOS DE BLOCO LINEARES: NOVAS DEFINIÇÕES E CENÁRIOS DE APLICAÇÃO

Tese de Doutorado apresentada ao Programa de Pós-graduação em Engenharia Elétrica da Universidade Federal de Pernambuco como requisito parcial para a obtenção do grau de Doutor em Engenharia Elétrica. Área de concentração: Comunicações.

Orientador: Prof. Ricardo Campello de Souza, PhD.

Recife

2018

Catalogação na fonte
Bibliotecária Maria Luiza de Moura Ferreira, CRB-4 / 1469

P279t Paschoal, Arquimedes José de Araújo.
Transformadas em corpos finitos e códigos de bloco lineares: novas definições e cenários de aplicação / Arquimedes José de Araújo Paschoal. - 2018.

130 folhas, il.; tab., abr. sigl. e simb.

Orientador: Prof. Ricardo Campello de Souza, PhD.
Tese (Doutorado) – Universidade Federal de Pernambuco. CTG. Programa de Pós-graduação em Engenharia Elétrica, 2018.
Inclui Referências e Apêndices.

1. Engenharia Elétrica. 2. Triângulo de Pascal. 3. Transformadas numéricas. 4. Transformada numérica de Pascal. 5. Transformada de Hamming. 6. Transformada de Golay. 7. Cifragem de imagens. I. Souza, Ricardo Campello de (Orientador). II. Título.

UFPE

621.3 CDD (22. ed.) BCTG/2018-240

Universidade Federal de Pernambuco
Pós-Graduação em Engenharia Elétrica

PARECER DA COMISSÃO EXAMINADORA DE DEFESA DE TESE DE DOUTORADO DE

ARQUIMEDES JOSÉ DE ARAÚJO PASCHOAL

TÍTULO

"TRANSFORMADAS EM CORPOS FINITOS E CÓDIGOS DE BLOCO LINEARES: NOVAS DEFINIÇÕES E CENÁRIOS DE APLICAÇÃO"

A comissão examinadora composta pelos professores: RICARDO MENEZES CAMPELLO DE SOUZA, DES/UFPE; VALDEMAR CARDOSO DA ROCHA JÚNIOR, DES/UFPE; JULIANO BANDEIRA LIMA, DES/UFPE; HÉLIO MAGALHÃES DE OLIVEIRA; DE/UFPE e LUIZ FELIPE DE QUEIROZ SILVEIRA, DECA/UFRN, sob a presidência do primeiro, consideram o candidato **ARQUIMEDES JOSÉ DE ARAÚJO PASCHOAL APROVADO.**

Recife, 27 de fevereiro de 2018.

MARCELO CABRAL CAVALCANTI
Coordenador do PPGEE

RICARDO MENEZES CAMPELLO DE SOUZA
Orientador e Membro Titular Interno

HÉLIO MAGALHÃES DE OLIVEIRA
Membro Titular Externo

VALDEMAR CARDOSO DA ROCHA JÚNIOR
Membro Titular Interno

LUIZ FELIPE DE QUEIROZ SILVEIRA
Membro Titular Externo

JULIANO BANDEIRA LIMA
Membro Titular Interno

Dedico este trabalho a Deus que me criou simples e ignorante e me deu o livre arbítrio para que eu escolhesse meus caminhos, certos ou errados. Agradeço a ti, pai amado, pelos anjos da dor, do medo, da insegurança, da necessidade e tantos outros que me enviaste para que eu me fortalecesse e aprendesse o verdadeiro valor da vida e do respeito.

AGRADECIMENTOS

Gostaria de agradecer primeiramente a Deus pela família que me concedeu, pelas oportunidades de crescimento, pela força que me inspirou em tantos momentos de minha vida. Em especial agradeço a meus pais, Maria Ester Corrêa de Araújo Paschoal e Paulo Aristarcho de Melo Paschoal *(in memoriam)*, minha esposa Gisélia Maria da Rocha Paschoal e minha filha Rebeca Maria da Rocha Paschoal.

Não posso deixar de agradecer ao meu orientador e amigo pessoal, Professor Ricardo Menezes Campello de Souza, pela presença sempre constante ao meu redor, tal como um irmão mais experiente cuja confiança depositada em minha pessoa foi fundamental para a continuidade das pesquisas. Igualmente, agradeço ao meu amigo, Professor Hélio Magalhães de Oliveira, pelos inúmeros artigos enviados, pelas conversas estimulantes, pela vibração a cada passo dado. Igualmente agradeço ao Professor Juliano Bandeira Lima pelos programas fornecidos, pela parceria no Grupo de Processamento Digital de Sinais. Não poderia deixar de citar o Professor Gilson Jerônimo da Silva Júnior que colaborou muito nas discussões dos temas da tese, nas provas e discussões salutares. Também agradeço ao Professor Luiz Felipe da UFRN pelos comentários e sugestões durante a qualificação e na leitura desta Tese. At last, but not at least, gostaria de agradecer ao Professor Valdemar Cardozo da Rocha Júnior pelos comentários feitos principalmente durante as conversas no corredor do Departamento de Eletrônica e Sistemas.

Também não poderia deixar de agradecer aos amigos (barqueiros) mais próximos da pós-graduação pelas conversas, idéias compartilhadas e pela travessia do Hades (sem nada cobrar!), além da força que nos manteve unidos: Bergson José do Nascimento, Bruna Palm, Carlos Eduardo Correia de Souza, Diego Ramos Canterle, Jamile Tuanne Dantas Alves, José Antonio Pérez Morales Artiles, Paulo Hugo Espírito Santo Lima, Ravi Barreto Dória Figueiredo, Verusca Severo de Lima e Vilmar Vaz da Silva. Aos outros amigos que se distanciaram por trabalharem em outras linhas de pesquisa, fica aqui o meu abraço e o meu muito obrigado pelo companheirismo.

Impossível deixar de fora meus alunos do Instituto Federal de Educação, Ciência e Tecnologia de Pernambuco/Campus Caruaru que sempre me apoiaram e vibraram juntamente comigo. Agradeço também ao diretor geral do campus Caruaru, George Mello, por ter concedido meu afastamento, facilitando meus estudos. Finalmente, aos amigos professores do IFPE/Caruaru pelo incentivo e apoio em todos os momentos, em especial Professor Dalton Cezane, Professor Dr. Luciano Cabral, Professor Dr. Alexander Sena e o Professor José Alci pela leitura do original, sugestões, críticas e revisão matemática.

RESUMO

O trabalho desenvolvido nesta Tese versa sobre a área de Processamento de Sinais por meio de transformadas definidas sobre corpos finitos. A principal ferramenta usada é o triângulo de Pascal definido sobre o corpo finito $GF(p)$. Uma nova transformada baseada neste triângulo de Pascal, a transformada numérica de Pascal, é introduzida e suas propriedades são investigadas. A estrutura do triângulo de Pascal modular é explorada de forma única nesta Tese, revelando novas propriedades. Uma nova família de códigos de bloco lineares multiníveis baseada em tal transformada, os códigos de Pascal, é definida. A construção de novas transformadas definidas em corpos finitos baseadas em famílias conhecidas de códigos corretores de erros é investigada. Em particular as transformadas numéricas de Hamming e de Golay, as quais se baseiam nos respectivos códigos corretores de erros, são introduzidas. Um caso especial é considerado, a saber, a versão cíclica destas transformadas, e suas propriedades são investigadas. A estrutura de autossimilaridade da matriz do triângulo de Pascal modular é considerada, e uma aplicação das transformadas introduzidas nesta Tese, como uma ferramenta de pré-processamento para cifragem de imagens, é sugerida. Algoritmos rápidos para a computação da transformada numérica de Pascal são propostos.

Palavras-chave: Triângulo de Pascal. Transformadas numéricas. Transformada numérica de Pascal. Transformada de Hamming. Transformada de Golay. Cifragem de imagens.

ABSTRACT

The work developed in this thesis concerns signal processing by means of finite field transforms. The main tool used in the research is Pascal's triangle defined over the finite field $GF(p)$. A new transform based on this Pascal's triangle, the Pascal Number Theoretic Transform, is introduced and its properties are investigated. The structure of the modular Pascal's triangle is exploited uniquely, revealing new properties. A new family of multilevel linear block codes based on such transforms, the Pascal Codes, is defined and its parameters determined. The construction of new finite field transforms based on established families of error correcting codes is investigated. In particular, the Hamming number-theoretic transform and the Golay number-theoretic transform, which are based on the respective error correcting codes, are introduced. A special case is considered, namely, the cyclic version of these transforms, and its properties are investigated. Pascal's triangle self-similarity structure is investigated and a possible application, concerning image encryption, is considered. Fast algorithms for computing the Pascal Number Theoretic Transform are proposed.

Keywords: Pascal's triangle. Number-theoretic transforms. Hamming transforms. Golay transforms. Image encryption.

LISTA DE ILUSTRAÇÕES

LISTA DE TABELAS

LISTA DE ABREVIATURAS E SIGLAS

TNGC	Transformada numérica de Golay cíclica
NPCR	Number of Pixels Changing Rate
UACI	Unified Averaged Changed Intensity
TNHC	Transformada numérica de Hamming cíclica
TNP	Transformada numérica de Pascal

LISTA DE SÍMBOLOS

$\lfloor \cdot \rfloor$:	Função Piso
$\lceil \cdot \rceil$:	Função Teto
$\phi(\cdot)$	Função Totiente de Euler
λ	Autovalor
C_i^k	Combinação de i elementos tomados k a k
d	Distância mínima do código de bloco linear
F^*	Conjunto F retirando-se o elemento 0 (zero)
G	Matriz geradora do código de bloco linear
$GF(p)$	Campo de Galois de ordem p
$GL(N, GF(p))$	Grupo linear geral formado pelas matrizes $N \times N$ sobre o corpo finito $GF(p)$
H	Matriz de verificação de paridade do código de bloco linear
$h(x)$	Polinômio de verificação de paridade
k	Dimensão do código de bloco linear
L_N	Matriz triangular inferior de ordem N
$MDC(a, b)$	Máximo divisor comum entre a e b
n	Comprimento do código de bloco linear
P_N	Matriz de Pascal de ordem N
P_∞	Matriz de Pascal infinita
$p(\lambda)$	Polinômio característico
$T_{H,p}^{(\lambda)}$	Matriz da transformada numérica obtida a partir da matriz de verificação de paridade H, sobre o corpo finito $GF(p)$, com autovalor λ
U_N	Matriz triangular superior de ordem N
$v = \{v_0, \cdots, v_{N-1}\}$	Sequência de componentes no domínio do tempo

$V = \{V_0, \cdots, V_{N-1}\}$	Sequência de componentes no domínio da frequência
$\triangleq$	Igual por definição
α	Elemento primitivo do corpo finito $GF(p)$
$g(x)$	Polinômio gerador
$C(n, k, d)$	Código de bloco linear corretor de erros
$\circledast_p$	Convolução cíclica módulo p
$*$	Convolução linear
$r_p(l_i(x), l_j(x))$	Correlação cíclica entre a i-ésima e a j-ésima linhas da matriz de Pascal
$r_n(l_i(x), l_j(x))$	Correlação linear entre a i-ésima e a j-ésima linhas da matriz de Pascal
$l_i(x)$	Polinômio correspondente à i-ésima linha da matriz de Pascal
$\pi(x)$	Polinômio gerador do corpo finito $GF(p^r)$

SUMÁRIO

1 INTRODUÇÃO

Uma das principais razões de se pesquisar transformadas numéricas é o fato de se obter uma precisão "infinita", ou seja, para tais transformadas não existe o chamado erro de arredondamento ou truncagem, uma vez que toda a aritmética se efetua no conjunto dos números inteiros. Neste sentido, sempre que se define alguma transformada, é natural perguntar: "Existe uma versão numérica para a mesma?" A próxima pergunta natural é: "Existe uma forma eficiente de se calcular tal transformada, explorando-se as propriedades da estrutura matemática que lhe dá suporte?" A resposta a esta segunda pergunta nos leva ao mundo dos algoritmos rápidos. Assim, encontramos versões numéricas de transformadas originalmente definidas no *continuum* (Transformada de Fourier, Transformada de Hilbert, Transformadas do seno e do cosseno, Transformada de Hartley, etc.) e transformadas definidas diretamente sobre os corpos finitos, tal como a Transformada de Pascal apresentada aqui. As versões numéricas, em geral, são definidas de modo a preservar as mesmas propriedades de suas versões contínuas. Seja como for, qualquer transformada deve sempre (de alguma forma) traduzir o problema a ser resolvido em um problema mais simples de ser tratado. Assim, alguma transformada projetada para se resolver um determinado problema pode se mostrar ineficiente na solução de um outro problema. Por exemplo, a Transformada de Fourier não é a ferramenta adequada quando se deseja explorar a resolução temporal/frequencial, para isto existem as transformadas Wavelets (DEOLIVEIRA, 2007) e a Transformada de Gabor (*Short Time Fourier Transform*) (GABOR, 1946). Uma questão relevante, de um modo geral, é a complexidade aritmética (entendida aqui como o número de multiplicações e adições) necessária ao cálculo da transformada. Muitos algoritmos eficientes têm sido desenvolvidos visando reduzir esta complexidade aritmética. Estes algoritmos eficientes implementados em Processadores Digitais de Sinais (DSP), em FPGA ou em circuitos integrados de aplicação específica (ASIC), permitem o desenvolvimento de equipamentos capazes de processar informações em tempos "tão curtos" que podemos considerar o processamento como sendo feito em tempo real. Desta forma, uma tomografia computadorizada, por exemplo, é realizada e observada "instantaneamente" na tela de um monitor permitindo ao operador julgar se as imagens obtidas são adequadas ou não, e então efetuar os ajustes necessários à obtenção dos detalhes requeridos no exame.

Transformadas definidas sobre corpos finitos têm sido empregadas tanto na área de processamento digital de sinais, quanto nas áreas de códigos corretores de erros e criptografia, entre outras (POLLARD, 1971), (SOUZA; FREIRE; DEOLIVEIRA, 2009), (LIMA; LIMA; MADEIRO, 2013).

1.1 UM POUCO DE HISTÓRIA: UMA LINHA DO TEMPO

Hoje sabe-se que Gauss desenvolveu um algoritmo similar[1] à transformada rápida de Fourier (FFT, do inglês Fast Fourier Transform) para o cálculo dos coeficientes da série finita de Fourier em 1805 (HEIDEMAN; JOHNSON; BURRUS, 1984). É interessante perceber aqui que este trabalho de Gauss antecede, inclusive, ao próprio trabalho de Fourier sobre análise harmônica (1807). Aparentemente, Gauss não teria avaliado a complexidade computacional de seu algoritmo.

Em 1971, J. M. Pollard (POLLARD, 1971) definiu a primeira transformada sobre um corpo finito. No artigo de Pollard, é abordada a computação da convolução de sequências longas de inteiros por meio de aritmética modular.

Em 1972, C. M. Rader (RADER, 1972) propôs a transformada numérica de Mersenne cuja computação, em certos casos, só requer adições e deslocamentos cíclicos.

Em 1974, R. C. Agarwal e C. S. Burrus (AGARWAL; BURRUS, 1974) definiram a transformada numérica de Fermat cuja computação, em certos casos, também não envolve multiplicações, apenas adições e deslocamentos cíclicos.

Em 1993, (CAIRE; GROSSMAN; POOR, 1993) introduziram a transformada wavelet sobre corpos finitos.

No início dos anos 1980, Victor Namias (NAMIAS, 1980) introduziu a transformada fracionária de Fourier.

Em 1998, (SOUZA et al., 1998) introduziram a transformada de Hartley em um corpo finito apresentando suas principais propriedades e apontando possíveis aplicações para a mesma. Na definição desta transformada foi empregada a recém introduzida trigonometria sobre corpos finitos (PASCHOAL; SOUZA; DEOLIVEIRA, 1993). A importante família de transformadas sobre $GF(p)$ quando p é um primo de Mersenne foi considerada. Para tais transformadas, o comprimento de bloco é uma potência de 2 e isto possibilita a utilização de algoritmos rápidos de base 2 para sua computação.

Em 2004, (SOUZA et al., 2004) introduziram a transformada numérica do cosseno usando trigonometria sobre corpos finitos.

Em 2005, (SOUZA et al., 2005) introduziram a transformada numérica do seno. O caso em que p é um primo de Mersenne foi considerado o que permite a utilização de algoritmos rápidos de base 2 para sua computação.

Em 2008, (LIMA, 2008) completou a família de transformadas trigonométricas estendendo para 16 o número de transformadas trigonométricas sobre corpos finitos conhecidas até então (três).

[1] O algoritmo de Gauss corresponde à FFT de Cooley-Tukey com dizimação na frequência.

Em 2012, (LIMA; SOUZA, 2012) introduziram a transformada fracionária de Fourier usando conceitos de trigonometria sobre corpos finitos.

Em 2013, (LIMA; LIMA; MADEIRO, 2013) introduziram as transformadas fracionárias do cosseno e do seno a partir da transformada fracionária de Fourier.

Em 2014, Kak (KAK, 2014) introduziu a transformada numérica de Hilbert sugerindo aplicações na área de criptografia.

Em 2015, P. Lima et al. (LIMA, 2015) introduziram a transformada fracionária de Hartley, a qual se baseia em funções de matrizes sobre corpos finitos. Foi desenvolvida uma aplicação em cifragem de imagens para demonstrar a flexibilidade e aplicabilidade de tal transformada.

Em 2015, (PASCHOAL; DEOLIVEIRA; SOUZA, 2015) introduziram a transformada numérica de Pascal considerada neste trabalho, usando como base a matriz de Pascal sobre o corpo finito $GF(p)$.

A Tabela 1 lista as transformadas de corpo finito conhecidas na literatura.

Tabela 1 – Transformadas Sobre Corpos Finitos

Transformada	Aplicação Sugerida
Transformada de Fourier (1971)	Codif. de Canal e Processamento Digital de Sinais
Transformada Numérica de Mersenne (1972)	Processamento Digital de Sinais
Transformada Numérica de Fermat (1974)	Processamento Digital de Sinais
Transformada Wavelet (1993)	Codif. de Canal e Processamento Digital de Sinais
Transfiormada de Hartley (1998)	Codificação de Canal e Multiplexação Digital
Transformada do Cosseno (2004)	Multiplexação Digital e Marca dágua
Transformada do Seno (2005)	Marca dágua
Transformadas Trigonométricas (2008)	Codif. de Canal, Criptograf. e Multiplexação Digital
Transformada Fracionária de Fourier (2012)	Codificação de Canal e Criptografia
Transformada Fracionária do Cosseno (2013)	Criptografia e Multiplexação Digital
Transformada Fracionária do Seno (2013)	Criptografia e Multiplexação Digital
Transformada Numérica de Hilbert (2014)	Criptografia
Transformada Fracionária de Hartley (2015)	Criptografia
Transformada Numérica de Pascal (2015)	Criptografia

As transformadas do cosseno e do seno na Tabela 1 são conhecidas como transformadas trigonométricas e só foram definidas após a introdução de uma trigonometria sobre corpos finitos (PASCHOAL; SOUZA; DEOLIVEIRA, 1993)(SOUZA et al., 1998).

1.2 CONTEÚDO DA TESE

No Capítulo 2 introduzimos os preliminares matemáticos necessários ao desenvolvimento dos conceitos abordados nesta Tese.

No Capítulo 3 são introduzidas novas relações na matriz de Pascal modular (PASCHOAL; SOUZA; OLIVEIRA, 2018), especialmente úteis na determinação de propriedades da Transfor-

mada Numérica de Pascal (TNP). A fatoração das matrizes de Pascal de determinadas ordens, por meio de um produto de Kronecker, é apresentada.

No Capítulo 4 a transformada numérica de Pascal (TNP) é definida, suas principais propriedades são enumeradas e o algoritmo de fatoração de Cholesky é empregado para a determinação da transformada inversa (PASCHOAL; DEOLIVEIRA; SOUZA, 2015). Diversas proposições são enunciadas, com suas respectivas provas, discute-se a estrutura da matriz de Pascal adotada na definição da TNP e, finalmente, introduz-se uma definição de periodicidade para a matriz de Pascal infinita. Esta estrutura está relacionada ao comportamento autossimilar da matriz de Pascal que, por sua vez, conecta a matriz de Pascal à teoria dos fractais. A ordem da matriz de Pascal P_N é determinada e expessões matemáticas para os autovalores associados à matriz de transformação da TNP de comprimento $N = p^r$ e suas multiplicidades são obtidas.

No Capítulo 5 são apresentados algoritmos rápidos para a computação da transformada numérica de Pascal (PASCHOAL; SOUZA, 2017).

No Capítulo 6 são apresentadas aplicações da transformada numérica de Pascal. Introduz-se a família de códigos corretores de erros de Pascal a partir da definição da transformada numérica de Pascal, de acordo com a técnica introduzida por Campello (SOUZA; FREIRE; DEOLIVEIRA, 2009). Os parâmetros de alguns códigos são determinados. São introduzidas as transformadas numéricas de Hamming e Golay a partir dos respectivos códigos corretores de erros, usando a mesma técnica. Algumas propriedades destas novas transformadas são discutidas. Apresenta-se a utilização das mesmas como uma ferramenta de pré-processamento para a cifragem de imagens.

No Capítulo 7 apresentamos as conclusões deste trabalho e as sugestões para trabalhos futuros.

O Apêndice A contém o código fonte em Mathematica® dos programas usados para se obter as matrizes de paridade dos códigos de Pascal sobre corpos finitos.

O Apêndice B contém uma análise comparativa das transformadas de Hamming e de Golay, definidas nesta Tese, como possíveis ferramentas em aplicações de cifragem de imagens por meio de um script em Matlab® cuja listagem encontra-se no Apêndice C.

O Apêndice C contém a listagem dos programas, desenvolvidos em Matlab®, usados para o processamento de imagens por meio das transformadas definidas nesta Tese.

O Apêndice D contém o código fonte do programa em Matlab® usado para desenvolver os chamados tapetes de Pascal.

1.3 CONTRIBUIÇÕES

As principais contribuições desta Tese são

a) Demonstração de novas relações na matriz de Pascal modular (Cap. 3).

b) Definição da transformada numérica de Pascal (TNP) e análise de suas propriedades (Cap. 4).
c) Análise da autoestrutura da matriz de transformação da TNP (Cap. 4).
d) Construção de algoritmos rápidos para a computação da TNP (Cap. 5).
e) Definição da família de códigos corretores de erros de Pascal (Cap. 6).
f) Definição das transformadas numéricas de Hamming e de Golay, obtidas a partir dos respectivos códigos corretores de erros (Cap. 6).
g) Análise da aplicabilidade das transformadas numéricas introduzidas na Tese como uma ferramenta de pré-processamento na cifragem de imagens (Cap. 6).

2 PRELIMINARES MATEMÁTICOS

Transformadas definidas sobre corpos finitos têm sido aplicadas em diversas áreas da Engenharia Eletrônica, tais como criptografia (LIMA; LIMA; MADEIRO, 2013), processamento de imagens (CINTRA et al., 2009), multiplexação digital (DEOLIVEIRA; SOUZA; KAUFFMAN, 1999), (SOUZA; OLIVEIRA, 2006) e codificação de canal (SOUZA; FREIRE; DEOLIVEIRA, 2009), (FREIRE, 2009), dentre outras áreas.

Neste capítulo é feita uma breve introdução à teoria dos corpos finitos. Posteriormente, apresentamos a matriz de Pascal adotada neste trabalho, o método de fatoração de Cholesky[1] ((CONTE; BOOR, 1980), (EDELMAN; STRANG, 2004)) e exploramos algumas propriedades desta matriz.

2.1 CORPOS FINITOS

Corpos finitos foram introduzidos por Évariste Galois [⋆25/10/1811;†31/05/1832] em 1830 na sua prova da inexistência de soluções por radicais para equações algébricas de grau maior do que 4.

Definição 1. *Um corpo $< F, \oplus, \otimes >$ é uma estrutura matemática formada por um conjunto F e duas operações, aqui denotadas $\oplus$ e $\otimes$, obedecendo aos seguintes axiomas:*

(a) A estrutura $< F, \oplus >$ é um grupo Abeliano.
(b) A estrutura $< F - \{0\}, \otimes >$ é um grupo Abeliano.
(c) Se a, b e c $\in$ F, então

$$(a \bigoplus b) \bigotimes c = (a \bigotimes c) \bigoplus (b \bigotimes c),$$
$$c \bigotimes (a \bigoplus b) = (c \bigotimes a) \bigoplus (c \bigotimes b).$$

Por uma questão de simplificação de notação, iremos nos referir ao conjunto F como sendo o próprio corpo. Ademais, denotamos a $\oplus$ b e a $\otimes$ b por a+b e ab, respectivamente.

Definição 2. *Um corpo formado por um número finito de elementos é chamado corpo finito; caso contrário, ele é chamado de corpo infinito. A quantidade de elementos de um corpo é chamada de cardinalidade do corpo ou ordem do corpo.*

Teorema 1. *Só existem corpos finitos de ordem igual a uma potência de um primo (MCELIECE, 1987).*

Definição 3. *A característica de um corpo finito F é o menor número inteiro positivo m tal que, para qualquer elemento $a \in F$, não nulo, tem-se*

[1] André-Louis Cholesky [⋆15/10/1875;†31/08/1918] foi um cartógrafo francês. A fatoração de Cholesky é muito útil na resolução de problemas de ortogonalização de sinais.

$$ma = \underbrace{a + a + \cdots + a}_{\textit{m vezes}} = 0, \ \textit{em F.}$$

Exemplo 1. *O conjunto $F = Z_5 = \{0, 1, 2, 3, 4\}$, munido das operações de adição e multiplicação módulo 5, é um corpo finito de ordem 5 e característica 5.*

■

Definição 4. *A ordem multiplicativa de um elemento $a \in F$ é o menor número inteiro positivo r tal que $a^r = 1$.*

De uma maneira geral, $Z_p = \{0, 1, \cdots, p-1\}$, o conjunto dos inteiros módulo p, munido das operações de adição e multiplicação módulo p, forma o corpo finito denotado por $GF(p)$ (do inglês *Galois Field* ou Campo de Galois).

Os corpos finitos de ordem p^m são chamados de corpos de extensão do corpo base $GF(p)$. Os elementos de $GF(p^m)$ podem ser representados por (MCELIECE, 1987)

(a) $m - uplas$ de elementos de $GF(p)$;
(b) Polinômios, sobre $GF(p)$, de grau menor do que m;
(c) Potências de um elemento primitivo do corpo, isto é, um elemento cuja ordem multiplicativa é $p^m - 1$.

A representação em (c) facilita a implementação da operação de multiplicação em um corpo de extensão. As operações usadas no corpo $GF(p^m)$ são adição e multiplicação módulo um polinômio $p(x)$. Este polinômio precisa atender a algumas condições.

Definição 5. *O polinômio irredutível $p(x)$ de grau m sobre $GF(p)$ é dito pertencer ao expoente **e**, quando este for o menor inteiro positivo tal que $p(x)$ divida $(x^e - 1)$. Quando **e** $= (p^m - 1)$, $p(x)$ é chamado polinômio primitivo (MCELIECE, 1987).*

Polinômios primitivos permitem a representação dos elementos de um corpo finito por meio de potências de um elemento do corpo, chamado de elemento primitivo (representação da letra (c)).

Exemplo 2. *Construção do corpo $GF(2^3)$. Para representarmos os elementos deste corpo como potências de um elemento $\alpha \in GF(2^3)$, precisamos de um polinômio primitivo, ou seja, precisamos de um polinômio irredutível sobre $GF(2)$ de grau $m = 3$ que pertença ao expoente $2^3 - 1 = 7$. Vamos representar este polinômio por $\pi(x)$. Um polinômio que atende à estes requesitos é $\pi(x) = x^3 + x + 1$. Se α é uma raiz de $\pi(x)$ sobre $GF(2^3)$, então*

$$\pi(\alpha) = \alpha^3 + \alpha + 1 = 0 \Rightarrow \alpha^3 = \alpha + 1. \tag{2.1}$$

Outra forma de representar os elementos deste corpo, sem a necessidade de se ter um polinômio primitivo, é por meio do conjunto de todos os polinômios com coeficientes sobre

$GF(2)$ de grau menor ou igual a 2, isto é,

$$0, 1, x, x+1, x^2, x^2+1, x^2+x, x^2+x+1. \tag{2.2}$$

A Tabela 2 lista os elementos do corpo $GF(2^3)$ construído usando-se o polinômio primitivo $\pi(x)$.

Tabela 2 – Representações dos elementos do corpo $GF(2^3)$.

Potências de α	**Representação Binária**	**Representação Polinomial**
$*$	000	0
α^0	001	1
α	010	x
α^2	100	x^2
$\alpha^3 = \alpha + 1$	011	$x+1$
$\alpha^4 = \alpha^2 + \alpha$	110	x^2+x
$\alpha^5 = \alpha^2 + \alpha + 1$	111	x^2+x+1
$\alpha^6 = \alpha^2 + 1$	101	x^2+1

■

2.2 MATRIZES DE PASCAL

Existem, pelo menos, doze definições para a matriz de Pascal (BIRREGAH; DOH; ADJALLAH, 2006). Neste trabalho adotamos a matriz de Pascal apresentada na Definição 6.

Definição 6. *A matriz de Pascal de ordem N, denotada por P_N, é a matriz quadrada de elementos*

$$[P_N]_{i,k} \triangleq C_{i+k}^{i}, \tag{2.3}$$

em que C_{i+k}^{i} denota a combinação de $(i+k)$, tomada i a i. Quando $N \to \infty$, a matriz é denominada matriz de Pascal infinita e é denotada por P_∞.

Definição 7. *Uma matriz A é dita ser positiva definida quando*

(a) A matriz A é simétrica;
(b) $x^T \cdot A \cdot x > 0$, para todo $x \neq 0$.

Quando $N < \infty$, P_N é simétrica positiva definida, podendo ser decomposta como o produto de uma matriz triangular inferior por uma matriz triangular superior. Existem várias formas de efetuar tal decomposição, por exemplo, eliminação de Gauss, decomposição PLU, decomposição de Crout, decomposição LR, decomposição QR, decomposição de Cholesky, entre outras (CONTE; BOOR, 1980). Aqui usamos a decomposição de Cholesky, pois a matriz de Pascal é uma matriz positiva definida e este é um método eficiente neste caso (EDELMAN; STRANG, 2004). A decomposição de Cholesky é um caso específico da decomposição LU,

que, por sua vez, é uma variante do método de eliminação de Gauss. No chamado método de Cholesky tem-se

$$P_N = L_N \cdot L_N^T, \tag{2.4}$$

em que L_N é a matriz de elementos L_{ik} dados por

$$L_{ik} = \begin{cases} C_i^k, & \text{se} \quad i \geq k, \\ 0, & \text{se} \quad i < k. \end{cases} \tag{2.5}$$

Mostra-se que, para todo N, a fatoração de Cholesky é sempre possível e que os elementos da diagonal principal de L_N são todos iguais a 1 (EDELMAN; STRANG, 2004), (LV; HUANG; REN, 2009), consequentemente, o determinante de P_N é igual a 1. Mostra-se também que os autovalores das matrizes P_N e P_N^{-1} são reais e positivos (BACHER; CHAPMAN, 2004).

2.3 O MÉTODO DE CHOLESKY

Aqui não iremos deduzir o método de Cholesky, por existir vasta literatura sobre o mesmo (CONTE; BOOR, 1980), (CALL; VELLEMAN, 1993), (EDELMAN; STRANG, 2004). Os elementos L_{ik} da matriz triangular inferior, pelo método de Cholesky, são dados por:

(i) Termos na diagonal principal

$$L_{00} = \sqrt{p_{00}}, \tag{2.6}$$

$$L_{kk} = \sqrt{p_{kk} - \sum_{j=0}^{k-1} L_{kj}^2}, \tag{2.7}$$

em que $k = 1, \cdots, N-1$.

(ii) Termos fora da diagonal principal

$$L_{j0} = p_{j0}/L_{00}, \tag{2.8}$$

$$L_{kj} = \frac{p_{kj} - \sum_{i=0}^{j-1} L_{ki} L_{ji}}{L_{jj}}. \tag{2.9}$$

Exemplo 3. *Considere a matriz de Pascal de ordem 5,*

$$P_5 = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 \\ 1 & 2 & 3 & 4 & 5 \\ 1 & 3 & 6 & 10 & 15 \\ 1 & 4 & 10 & 20 & 35 \\ 1 & 5 & 15 & 35 & 70 \end{bmatrix}.$$

A decomposição de Cholesky da mesma é obtida seguindo-se os passos:

(a) Escreva a matriz a seguir, colocando apenas os elementos da diagonal principal e a primeira coluna (todos iguais a 1),

$$L_5 = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 \\ 1 & \square & 1 & 0 & 0 \\ 1 & \square & \square & 1 & 0 \\ 1 & \square & \square & \square & 1 \end{bmatrix};$$

(b) Construa as outras linhas observando que os elementos faltantes podem ser obtidos a partir dos elementos iniciais, usando-se a relação de Stiefel diretamente; ou seja

$$C_i^k = C_{i-1}^k + C_{i-1}^{k-1}$$

e

$$L_5 = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 \\ 1 & 2 & 1 & 0 & 0 \\ 1 & 3 & 3 & 1 & 0 \\ 1 & 4 & 6 & 4 & 1 \end{bmatrix}.$$

Portanto,

$$P_5 = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 \\ 1 & 2 & 3 & 4 & 5 \\ 1 & 3 & 6 & 10 & 15 \\ 1 & 4 & 10 & 20 & 35 \\ 1 & 5 & 15 & 35 & 70 \end{bmatrix} = L_5 \cdot L_5^T = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 \\ 1 & 2 & 1 & 0 & 0 \\ 1 & 3 & 3 & 1 & 0 \\ 1 & 4 & 6 & 4 & 1 \end{bmatrix} \begin{bmatrix} 1 & 1 & 1 & 1 & 1 \\ 0 & 1 & 2 & 3 & 4 \\ 0 & 0 & 1 & 3 & 6 \\ 0 & 0 & 0 & 1 & 4 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix}.$$

■

3 NOVAS RELAÇÕES NO TRIÂNGULO DE PASCAL MODULAR

Neste capítulo apresentamos novas relações no triângulo de Pascal modular que serão especialmente úteis na determinação de propriedades da transformada numérica de Pascal (TNP) a ser definida no Capítulo 4 desta Tese.

3.1 A MATRIZ DE PASCAL MODULAR

A Transformada de corpo finito introduzida nesta Tese, no Capítulo 4, usa a matriz de Pascal modular definida a seguir.

Definição 8. *A matriz de Pascal modular sobre $GF(p)$, denotada por $P_N^{(p)}$, é a matriz quadrada de ordem N, de elementos*

$$\left[P_N^{(p)}\right]_{i,k} \triangleq C_{i+k}^i \ (\textit{mod}\ p).$$

No restante desta Tese a matriz de Pascal modular será chamada simplesmente de matriz de Pascal e representada por P_N.

Proposição 1. *Se p é um número primo, então, $C_{p+t}^i \equiv 0 \ \ (\textit{mod}\ p)$, $t = 0, 1, 2, \cdots, i-1$ e $i = 1, 2, \cdots, p-1$.*

Demonstração. Considere o cálculo de C_{p+t}^i.

$$C_{p+t}^i = \frac{(p+t)!}{i!(p+t-i)!} = \frac{(p+t)(p+t-1)\cdots(p+t-i+1)(p+t-i)!}{i!(p+t-i)!}$$
$$C_{p+t}^i = \frac{(p+t)(p+t-1)\cdots(p+t-i+1)}{i!}.$$

Então,

$$i!C_{p+t}^i = (p+t)(p+t-1)\cdots(p+t-i+1).$$

Como $0 \leq t \leq i-1$, percebe-se que o lado direito da expressão anterior é um múltiplo de p. Resulta, então, que

$$(p+t)(p+t-1)\cdots(p+t-i+1) \equiv 0 \ (\textit{mod}\ p),$$

ou seja,

$$i!C_{p+t}^{i} \equiv 0 \ (mod\ p)$$

e, portanto, $p|i!$ ou $p|C_{p+t}^{i}$. Se $p|i!$, então, $p|j$ para algum j tal que $1 \leq j \leq i \leq p-1$, o que não é possível. Portanto, $p|C_{p+t}^{i}$ e o resultado segue.

□

Proposição 2. *A matriz de Pascal, definida sobre $GF(p)$, cuja ordem N é igual a p, é triangular superior, em relação à sua diagonal secundária.*

Demonstração. Por definição, o elemento $[P_N]_{i,k}$ da matriz de Pascal é dado por C_{i+k}^{i}. Os elementos abaixo da diagonal secundária são tais que $i+k \geq N(=p)$, ou seja, $i+k = p+t$, para algum t tal que $0 \leq t \leq k-1$. Portanto, para estes elementos, podemos escrever

$$C_{i+k}^{i} = C_{p+t}^{p+t-k} = C_{p+t}^{k}$$

e, da Proposição 3.1, o resultado segue. □

Proposição 3. *A matriz de Pascal P_N, em que $N = L \cdot p$, sobre o corpo finito $GF(p)$, pode ser obtida a partir do produto de Kronecker $P_N = P_L \otimes P_p$, em que P_L e P_p são matrizes de Pascal de ordem L e p, respectivamente.*

Demonstração. Pela definição do produto de Kronecker, tem-se que os elementos da matriz $A \otimes B$, em que A é uma matriz $m \times n$ e B é uma matriz $p \times q$, são obtidos multiplicando-se cada elemento da matriz A pela matriz B, obtendo-se assim uma matriz de ordem $mp \times nq$. Como as matrizes de Pascal são quadradas, tem-se $m = n = L$ e $p = q$. Então, a matriz resultante do produto de Kronecker possui ordem Lp. Considere o produto de Kronecker $P_L \otimes P_p$, em que

$$P_L = \begin{bmatrix} 1 & 1 & \cdots & 1 \\ 1 & C_2^1 & \cdots & C_L^1 \\ \vdots & \vdots & \ddots & \vdots \\ 1 & C_L^{L-1} & \cdots & C_{2L-2}^{L-1} \end{bmatrix} \quad e \quad P_p = \begin{bmatrix} 1 & 1 & \cdots & 1 \\ 1 & C_2^1 & \cdots & C_p^1 \\ \vdots & \vdots & \ddots & \vdots \\ 1 & C_p^{p-1} & \cdots & C_{2p-2}^{p-1} \end{bmatrix}.$$

Tem-se,

$$P_L \bigotimes P_p = \begin{bmatrix} P_p & P_p & \cdots & P_p \\ P_p & C_2^1 P_p & \cdots & C_L^1 P_p \\ \vdots & \vdots & \ddots & \vdots \\ P_p & C_L^{L-1} P_p & \cdots & C_{2L-2}^{L-1} P_p \end{bmatrix}.$$

Sejam r e s os índices que identificam os blocos formados pelas cópias da matriz P_p multiplicada pelos termos binomiais de P_L. Então $r, s = 0, 1, ..., L-1$ e $[P_L]_{r,s} = C_{r+s}^{r}$. A

matriz $P_N = P_L \otimes P_p$ é formada pelos elementos obtidos pela multiplicação $C^r_{r+s} C^{\widehat{j}}_{\widehat{i}+\widehat{j}}$, em que $\widehat{i}, \widehat{j} = 0, 1, ..., p-1$. Note que os índices das linhas (i) e das colunas (j) da matriz P_N são dados por

$$i = \widehat{i} + rp, \quad j = \widehat{j} + sp,$$

em que[1] $r = \left\lfloor \frac{i}{p} \right\rfloor$ e $s = \left\lfloor \frac{j}{p} \right\rfloor$. Assim,

$$(P_L \bigotimes P_p)_{i,j} = C^r_{r+s} C^{\widehat{j}}_{\widehat{i}+\widehat{j}} (\text{mod } p) = \frac{(r+s)!}{r!s!} \cdot \frac{[(i+j)-(r+s)p]!}{(i-rp)!(j-sp)!}.$$

Deseja-se provar que

$$(P_L \bigotimes P_p)_{i,j} = \frac{(r+s)!}{r!s!} \cdot \frac{[(i+j)-(r+s)p]!}{(i-rp)!(j-sp)!} = C^i_{i+j} (\textit{mod } p). \tag{3.1}$$

Fixando os valores de r e s, a prova é feita por indução em i. Por simetria, a prova por indução em j é semelhante.
i) Passo Base: $i = 0 \Rightarrow r = 0$. Então

$$\frac{(0+s)!}{0!s!} \cdot \frac{[(0+j)-(0+s)p]!}{(0-0p)!(j-sp)!} = 1 = C^0_{0+j}. \tag{3.2}$$

ii) Passo da Indução:

$$(P_L \bigotimes P_p)_{i+1,j} = \frac{[(i+1+j)-(r+s)p]}{(i+1-rp)} \cdot \frac{(r+s)!}{r!s!} \cdot \frac{[(i+j)-(r+s)p]!}{(i-rp)!(j-sp)!}. \tag{3.3}$$

Supondo que a equação (3.1) é verdadeira, então

$$(P_L \bigotimes P_p)_{i+1,j} = \frac{[(i+1+j)-(r+s)p]}{(i+1-rp)} \cdot C^i_{i+j} \equiv C^{i+1}_{i+j+1} \ (\textit{mod } p). \tag{3.4}$$

□

Esta proposição possui consequências que resultam das propriedades do produto de Kronecker; a saber, se A e B são duas matrizes quaisquer, então (ZHANG, 2017)

(a) Se A e B são matrizes triangulares superiores (inferiores), então $A \otimes B$ é uma matriz triangular superior (inferior).
(b) Se A e B são matrizes simétricas, então $A \otimes B$ é uma matriz simétrica.
(c) Se A e B são matrizes positivas definidas, então $A \otimes B$ é uma matriz positiva definida.
(d) Se A possui autovalores λ_i $(i = 1, \cdots, n)$ e B possui autovalores $\beta_j (j = 1, \cdots, m)$, então $A \otimes B$ possui autovalores $\lambda_i \beta_j$.

[1] $\lfloor x \rfloor$ denota a função piso de x.

(e) Associatividade: $A \otimes B \otimes C = (A \otimes B) \otimes C = A \otimes (B \otimes C)$.

Exemplo 4. *Considere a matriz P_9 sobre GF(3),*

$$P_9 = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 2 & 0 & 1 & 2 & 0 & 1 & 2 & 0 \\ 1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 \\ 1 & 1 & 1 & 2 & 2 & 2 & 0 & 0 & 0 \\ 1 & 2 & 0 & 2 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 2 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} = \begin{bmatrix} P_3 & P_3 & P_3 \\ P_3 & 2P_3 & 0P_3 \\ P_3 & 0P_3 & 0P_3 \end{bmatrix}$$

$$= \begin{bmatrix} 1 & 1 & 1 \\ 1 & 2 & 0 \\ 1 & 0 & 0 \end{bmatrix} \bigotimes \begin{bmatrix} 1 & 1 & 1 \\ 1 & 2 & 0 \\ 1 & 0 & 0 \end{bmatrix} = P_3 \bigotimes P_3.$$

Da mesma forma, nota-se que $P_6 = P_2 \otimes P_3$, $P_{18} = P_6 \otimes P_3$. Assim, por meio das propriedades do produto de Kronecker, podemos escrever

$$P_{18} = P_6 \bigotimes P_3 = (P_2 \bigotimes P_3) \bigotimes P_3 = P_2 \bigotimes (P_3 \bigotimes P_3) = P_2 \bigotimes P_9.$$

■

Note que o resultado do Exemplo 3.1, juntamente com a propriedade (d) do produto de Kronecker, mostra que é possível encontrar os autovalores das matrizes de Pascal P_{18} e P_6, sobre $GF(3)$, por meio dos autovalores das matrizes de Pascal P_2 e P_3, sobre $GF(3)$.

Proposição 4. *Se a matriz de Pascal P_p, sobre $GF(p)$, possui todos os autovalores sobre $GF(p)$, então, a matriz de Pascal P_{pL} possui px autovalores sobre $GF(p)$ e py autovalores sobre um corpo de extensão, em que x e y são as quantidades de autovalores de P_L sobre o corpo base e sobre o corpo de extensão, respectivamente.*

Demonstração. A propriedade (d) do produto de Kronecker, garante que os autovalores de P_{pL} são obtidos a partir do produto dos autovalores de P_L pelos autovalores de P_p. Assim, como todos os autovalores de P_p estão sobre $GF(p)$, o resultado segue. □

Proposição 5. *Se a matriz de Pascal P_p, sobre $GF(p)$, possui todos os autovalores sobre $GF(p)$, então, a matriz de Pascal P_r, em que $r = p^m$, sobre $GF(p)$, também possui todos os autovalores sobre $GF(p)$.*

Demonstração. Pela Proposição 3, $P_{p^m} = P_p \otimes P_p \otimes \cdots \otimes P_p$. Como os autovalores de P_p pertencem a $GF(p)$, então, pela Propriedade de Associatividade do produto de Kronecker, resulta que os autovalores de P_{p^m} pertencem a $GF(p)$. □

Proposição 6. *Se $\beta = \alpha^{\left(\frac{p^m-1}{p-1}\right)}$, em que α é um elemento primitivo do corpo de extensão $GF(p^m)$, então $\beta \in GF(p)$.*

Demonstração. Note que $\beta^{(p-1)} = \alpha^{(p^m-1)} = 1$. Portanto, $\beta \in GF(p)$. □

Corolário 1. *Os elementos β^k, em que $k = 1, 2, \cdots, p-1$ pertencem a $GF(p)$.*

Demonstração. Como as potências de um elemento no corpo base estão no corpo base, e os elementos do corpo base possuem ordens que são divisores de $(p-1)$, segue-se que $\beta^k \in GF(p), k = 1, 2, \cdots, p-1$. □

Exemplo 5. *Considere o corpo finito $GF(5^2)$. O elemento $\beta = \alpha^{(\frac{5^2-1}{5-1})} = \alpha^6$ deste corpo está no corpo base, ou seja, $GF(5)$. Ademais, de acordo com o Corolário 1, os elementos α^{12}, α^{18} e α^{24} também estão no corpo base. Em verdade, estes elementos assumem valores diferentes no corpo base, dependendo do polinômio primitivo escolhido para definir a aritmética do corpo. Assim, para $\pi(x) = x^2 + 2x + 3$ tem-se $\alpha^6 = 3$, $\alpha^{12} = 4$, $\alpha^{18} = 2$ e $\alpha^{24} = 1$. Por outro lado, para $\pi(x) = x^2 + x + 2$ tem-se $\alpha^6 = 2, \alpha^{12} = 4, \alpha^{18} = 3$ e $\alpha^{24} = 1$.* ■

Proposição 7. *Se $\pi(x)$ é o polinômio gerador de $GF(p^m)$, então, pelas relações de Girard pode-se identificar qual o valor da primeira potência do elemento primitivo que se encontra no corpo base, a saber*

$$\beta = \alpha^{\left(\frac{p^m-1}{p-1}\right)} = (-1)^m \pi(0). \tag{3.5}$$

Demonstração. Seja $\pi(x) = a_m x^m + a_{m-1} x^{m-1} + \cdots + a_1 x + a_0$ o polinômio gerador de $GF(p^m)$, em que $a_m = 1$. Então se α e os seus conjugados são as raízes de $\pi(x)$, pelas relações de Girard tem-se

$$\alpha\alpha^p \cdots \alpha^{p^{m-1}} = (-1)^m \frac{a_0}{a_m} = (-1)^m \pi(0),$$
$$\alpha^{1+p+p^2+\cdots+p^{m-1}} = \alpha^{\left(\frac{p^m-1}{p-1}\right)} = (-1)^m \pi(0)$$

e o resultado segue. □

Exemplo 6. *Considere $N = 54 = 18 \times 3$. Portanto, sobre $GF(3)$, $P_{54} = P_{18} \otimes P_3$ e, assim, os autovalores de P_{54} são obtidos pelo produto dos autovalores de P_{18} pelos autovalores de P_3. As raízes distintas do polinômio característico associado a P_{18} estão sobre o corpo de extensão $GF(9)$ e são α^2 e α^6, ambas com multiplicidade 9. Todas as raízes do polinômio característico associado a P_3 estão no corpo base. Logo, todos os autovalores de P_{54} estão no*

corpo de extensão. Os autovalores de P_{54} são: α^2 e α^6, ambos com multiplicidade 27. Note que para $N = 54 \times 54 = 4 \times 3^6$ não podemos escrever $P_{54\times54} = P_{54} \otimes P_{54}$. Em verdade, pela propriedade (d) do poduto de Kronecker, $P_{54\times54} = P_4 \otimes P_{3^6}$. Como P_3 tem todos os autovalores sobre o corpo base, então, P_{3^6} possui todos os autovalores no corpo base. Da mesma forma, P_4 possui todos os autovalores no corpo base. Portanto, $P_{54\times54}$ possui todos os autovalores no corpo base. ■

Proposição 8. *Se p é um número primo, então, $C^i_{p^r+t} \equiv 0 \ (mod\, p),\ t = 0, 1, \cdots, i-1$ e $i = 1, 2, \cdots, p^r - 1$.*

Demonstração. Note que $C^i_{p^r+t}(mod\, p),\ t = 0, 1, \cdots, i-1$ e $i = 1, 2, \cdots, p^r - 1$ corresponde aos valores da matriz P_{p^r} que se encontram abaixo da diagonal secundária. Mas, podemos escrever

$$P_{p^r} = P_p \otimes P_p \otimes \cdots \otimes P_p.$$

Portanto, de acordo com a Proposição 1 e em função da propriedade (a) do Produto de Kronecker, vista anteriormente, o resultado segue. □

Proposição 9. *A matriz de Pascal, definida sobre $GF(p)$, cuja ordem N é uma potência de um número primo, p, tem todos os elementos abaixo de sua diagonal secundária iguais a zero.*

Demonstração. O resultado segue diretamente a partir da Proposição 8. □

3.2 SOBRE A ESTRUTURA DAS MATRIZES DE PASCAL

Uma observação mais atenta revela que algumas linhas da matriz de Pascal possuem um padrão repetitivo. Por exemplo, considere a matriz de Pascal de ordem $N = 16$, sobre $GF(2)$,

$$P_{16} = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 0 & 1 & 1 & 0 & 0 & 1 & 1 & 0 & 0 & 1 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

e observe as oito primeiras linhas.

Definição 9. *Dizemos que uma dada linha da matriz de Pascal de ordem N é periódica de período t quando nesta linha forem identificados exatamente N/t períodos, em que $N/t \in N^*$ e $1 \leq t \leq N/2$.*

Proposição 10. *Se P é a matriz de Pascal infinita (P_∞), sobre $GF(p)$, então*

(a) Todas as linhas são periódicas e os períodos das mesmas são sempre potências da característica do corpo.

(b) Existem exatamente $\phi(p^k)$ linhas de período p^k, em que $k \in N$ e $\phi(\cdot)$ denota a função de Euler (BURTON, 2006).

(c) O período da linha i é $p^{\lceil \log_p i \rceil}$, em que $\lceil \cdot \rceil$ denota a função teto.

$$\underbrace{11\ldots1}_{p}\underbrace{0000\ldots000000}_{(p^2-p)}$$

Figura 1 – "Invasão de zeros."
Fonte: O autor.

Demonstração. Observe a matriz de Pascal infinita sobre $GF(p)$,

$$P_\infty = \begin{bmatrix} 1 & 1 & 1 & \ldots & 1 & 1 & 1 & 1 & 1 & \ldots \\ 1 & 2 & 3 & \ldots & p-1 & p\equiv 0 & 1 & 2 & 3 & \ldots \\ 1 & 3 & 6 & \ldots & C_p^2 & C_{p+1}^2 & 1 & 3 & 6 & \ldots \\ \vdots & \vdots & \vdots & \ldots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 1 & C_{i+1}^i & C_{i+2}^i & \ldots & C_{i+3}^i & C_{i+4}^i & C_{i+p}^i & C_{i+p+1}^i & C_{i+p+2}^i & \ldots \\ \vdots & \vdots & \vdots & \ldots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 1 & 0 & 0 & \ldots & 0 & 0 & 1 & 0 & 0 & \ldots \\ 1 & 1 & 1 & \ldots & 1 & 1 & 2 & 2 & 2 & \ldots \\ \vdots & \vdots & \vdots & \ldots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \end{bmatrix}.$$

A primeira linha de P_∞ possui período igual a 1, conforme Definição 9. Observe que existem, em seguida, $(p-1)$ linhas de período igual a p. A última linha de período igual a p é a p-ésima linha $(100\ldots00)$. Devido à relação de Stiefel (RS), a $(p+1)$-ésima linha tem período p^2. Observe que a última linha de período p^2 será a linha composta por 1 seguido por (p^2-1) zeros. Considere a $(p+1)$-ésima linha. Observe que ocorre uma "invasão" de zeros nas linhas seguintes até se chegar à linha $10000\ldots0$, contendo (p^2-1) zeros. Porém, em algum ponto anterior, temos uma sequência de p uns seguidos por (p^2-p) zeros.

A primeira linha de período p^2 possui p zeros no seu final. A "invasão de zeros" começa da linha seguinte e vai até a linha mostrada na Figura 1. Neste ponto, tem-se $(p-2)$ blocos de tamanho p que serão "invadidos" mais $(p-1)$ posições do bloco mais à esquerda mais 1 (correspondente a primeira linha de período p^2) até se chegar na última linha, a saber $10000\ldots0$ composta por 1 seguido por (p^2-1) zeros. Assim, o número de linhas de período p^2 é $[(p-2)p+(p-1)]+1 = p^2-p = \phi(p^2)$. Vamos usar indução finita para mostrar que o número de linhas de período p^i é dado por $\phi(p^i)$.

Passo Base: $i=1$, tem-se $(p-1)$ linhas de período p,

$$p-1 = \phi(p).$$

Passo da Indução: Supor que para $i=k$, existam $\phi(p^k)$ linhas. A primeira linha de período p^{k+1} é formada por p blocos de tamanho p^k preenchidos todos com uns. A última linha de período p^{k+1} é formada por um bloco de tamanho $(p^{k+1}-1)$ contendo um 1 à esquerda e $(p^{k+1}-1)$ zeros à direita. Então, a quantidade de linhas de período p^{k+1} é

$$(p-2)p^k + (p^k-1) + 1 = p^{k+1} - p^k = \phi(p^{k+1}).$$

Observe que devido à relação de Stiefel, os períodos são sempre potências da característica. Observe as linhas e seus respectivos períodos mostrados na Tabela 3 a seguir.

Tabela 3 – Linhas da matriz de Pascal, seus períodos e $\lceil \log_p linha \rceil$.

Linha	**Período**	$\lceil \log_p linha \rceil$
1	1	0
2	$p-1$	$\lceil \log_p 2 \rceil = 1$
...	...	...
p	$\phi(p)$	$\lceil \log_p p \rceil = 1$
$p+1$	$\phi(p^2)$	2
...	...	...
p^2	$\phi(p^2)$	2
...	...	...

Observe que cada bloco periódico possui o mesmo índice. Assim, todas as linhas de período p possuem o índice 1, todas as linhas de período p^2 possuem índice 2, e assim por diante. Logo o período da linha i é $p^{\lceil \log_p i \rceil}$. □

Proposição 11. *Se P é a matriz de Pascal de ordem N, P_N, sobre $GF(p)$, então*

(a) Nem todas as linhas são periódicas. Os períodos das linhas periódicas são sempre potências da característica do corpo.

(b) Se existe uma linha de período p^k, então existem exatamente $\phi(p^k)$ linhas de período p^k, em que $k \in \mathbb{N}$.

(c) Sempre que $MDC(N,p) = 1$, o número de linhas periódicas e o número de períodos distintos são, ambos, iguais a 1. Caso contrário, o número de linhas periódicas é $p^{\lfloor \log_p \frac{N}{2} \rfloor}$.

Demonstração. A letra (a) já foi provada anteriormente. Diferentemente da matriz infinita, apenas algumas linhas apresentam periodicidade. Note que a matriz de Pascal de ordem N é uma matriz simétrica. Assim, se existir alguma linha de determinado período, então, todas as linhas correspondentes a este período estão na matriz de Pascal de ordem N. Pela Definição 9, o período deve ser um divisor da ordem da matriz de Pascal de ordem N e por (a) os períodos são sempre potências da característica do corpo. Então, quando $MDC(N,p) = 1$, resulta que $p \nmid N$. Assim, o único período possível é $p^0 = 1$. □

Exemplo 7. *Na matriz P_{16}, a linha $11001100\cdots$ é periódica de período $t = 4$. Observe que as linhas têm período conforme a Tabela 4. Note também que, pela Definição 9, podemos afirmar que todas as linhas a partir da nona linha não são periódicas.* ■

Tabela 4 – Períodos das linhas da matriz de Pascal de ordem 16, obtidos a partir da Proposição 11.

Linha (i)	**Período** (t)
1	1
2	2
3	4
4	4
5	8
6	8
7	8
8	8

Exemplo 8. *Considere a matriz P_{18} sobre $GF(3)$,*

$$P_{18} = \begin{bmatrix}
1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\
1 & 2 & 0 & 1 & 2 & 0 & 1 & 2 & 0 & 1 & 2 & 0 & 1 & 2 & 0 & 1 & 2 & 0 \\
1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 \\
1 & 1 & 1 & 2 & 2 & 2 & 0 & 0 & 0 & 1 & 1 & 1 & 2 & 2 & 2 & 0 & 0 & 0 \\
1 & 2 & 0 & 2 & 1 & 0 & 0 & 0 & 0 & 1 & 2 & 0 & 2 & 1 & 0 & 0 & 0 & 0 \\
1 & 0 & 0 & 2 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 2 & 0 & 0 & 0 & 0 & 0 \\
1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\
1 & 2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 2 & 2 & 2 & 2 & 2 & 2 & 2 & 2 & 2 \\
1 & 2 & 0 & 1 & 2 & 0 & 1 & 2 & 0 & 2 & 1 & 0 & 2 & 1 & 0 & 2 & 1 & 0 \\
1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 2 & 0 & 0 & 2 & 0 & 0 & 2 & 0 & 0 \\
1 & 1 & 1 & 2 & 2 & 2 & 0 & 0 & 0 & 2 & 2 & 2 & 1 & 1 & 1 & 0 & 0 & 0 \\
1 & 2 & 0 & 2 & 1 & 0 & 0 & 0 & 0 & 2 & 1 & 0 & 1 & 2 & 0 & 0 & 0 & 0 \\
1 & 0 & 0 & 2 & 0 & 0 & 0 & 0 & 0 & 2 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\
1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 2 & 2 & 2 & 0 & 0 & 0 & 0 & 0 & 0 \\
1 & 2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 2 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0
\end{bmatrix}.$$

O número de linhas de período 1 é $\phi(1) = 1$, o número de linhas de período 3 é $\phi(3) = 2$ e o número de linhas de período 3^2 é $\phi(3^2) = 6$. ■

Proposição 12. *Em se tratando das matrizes de Pascal sobre $GF(2)$, escrevendo-se a ordem da matriz como $N = 2^k \cdot (2m + 1)$, então*

$$\textit{O número de períodos} = \begin{cases} k, & \textit{se} \quad m = 0, \\ k + 1, & \textit{se} \quad m \neq 0. \end{cases}$$

Demonstração. Observe que os períodos devem ser divisores de N (menores do que N). Para $m = 0$, os valores de t estabelecidos pela Definição 9 são $2^0, 2^1, \cdots, 2^{k-1}$. Portanto, existem k períodos. No caso em que $m \neq 0$ podemos considerar todas as potências de 2 no intervalo de 0 até k; ou seja, $k + 1$ potências. □

Exemplo 9. *Considere a matriz de Pascal P_{28} sobre $GF(2)$,*

$$P_{28} = \begin{bmatrix}
1&1&1&1&1&1&1&1&1&1&1&1&1&1&1&1&1&1&1&1&1&1&1&1&1&1&1&1\\
1&0&1&0&1&0&1&0&1&0&1&0&1&0&1&0&1&0&1&0&1&0&1&0&1&0&1&0\\
1&1&0&0&1&1&0&0&1&1&0&0&1&1&0&0&1&1&0&0&1&1&0&0&1&1&0&0\\
1&0&0&0&1&0&0&0&1&0&0&0&1&0&0&0&1&0&0&0&1&0&0&0&1&0&0&0\\
1&1&1&1&0&0&0&0&1&1&1&1&0&0&0&0&1&1&1&1&0&0&0&0&1&1&1&1\\
1&0&1&0&0&0&0&0&1&0&1&0&0&0&0&0&1&0&1&0&0&0&0&0&1&0&1&0\\
1&1&0&0&0&0&0&0&1&1&0&0&0&0&0&0&1&1&0&0&0&0&0&0&1&1&0&0\\
1&0&0&0&0&0&0&0&1&0&0&0&0&0&0&0&1&0&0&0&0&0&0&0&1&0&0&0\\
1&1&1&1&1&1&1&1&0&0&0&0&0&0&0&0&1&1&1&1&1&1&1&1&0&0&0&0\\
1&0&1&0&1&0&1&0&0&0&0&0&0&0&0&0&1&0&1&0&1&0&1&0&0&0&0&0\\
1&1&0&0&1&1&0&0&0&0&0&0&0&0&0&0&1&1&0&0&1&1&0&0&0&0&0&0\\
1&0&0&0&1&0&0&0&0&0&0&0&0&0&0&0&1&0&0&0&1&0&0&0&0&0&0&0\\
1&1&1&1&0&0&0&0&0&0&0&0&0&0&0&0&1&1&1&1&0&0&0&0&0&0&0&0\\
1&0&1&0&0&0&0&0&0&0&0&0&0&0&0&0&1&0&1&0&0&0&0&0&0&0&0&0\\
1&1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&1&1&0&0&0&0&0&0&0&0&0&0\\
1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&1&0&0&0&0&0&0&0&0&0&0&0\\
1&1&1&1&1&1&1&1&1&1&1&1&1&1&1&1&0&0&0&0&0&0&0&0&0&0&0&0\\
1&0&1&0&1&0&1&0&1&0&1&0&1&0&1&0&0&0&0&0&0&0&0&0&0&0&0&0\\
1&1&0&0&1&1&0&0&1&1&0&0&1&1&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
1&0&0&0&1&0&0&0&1&0&0&0&1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
1&1&1&1&0&0&0&0&1&1&1&1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
1&0&1&0&0&0&0&0&1&0&1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
1&1&0&0&0&0&0&0&1&1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
1&0&0&0&0&0&0&0&1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
1&1&1&1&1&1&1&1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
1&0&1&0&1&0&1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
1&1&0&0&1&1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
1&0&0&0&1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0
\end{bmatrix}.$$

Ora, $28 = 2^2(2 \cdot 3 + 1)$; portanto, tem-se $k + 1 = 3$ períodos cujos valores são 1, 2 e 4. Logo, o número de linhas periódicas da matriz é $\phi(1) + \phi(2) + \phi(4) = 1 + 1 + 2 = 4$. ■

A estrutura periódica da matriz de Pascal pode ser explorada no sentido de se obter algoritmos rápidos e também ser associada a algoritmos que explorem a natureza fractal do triângulo de Pascal, no sentido de produzir, por exemplo, antenas multibandas, tal como ocorre com a antena baseada no triângulo de Sierpinski (MOHANAMURALIM; SHANMUGANANTHAM, 2012).

3.3 NOVAS RELAÇÕES NA MATRIZ DE PASCAL MODULAR

Proposição 13. *(Hockey Stick modular)*

$$\sum_{k=0}^{r} C_{i+k}^{i} \equiv C_{i+r+1}^{i+1} (mod\, p).$$

Demonstração. Existem diversas provas para esta propriedade do triângulo de Pascal usual (não modular), que também se verifica na matriz de Pascal adotada aqui. Até onde saibamos, a prova apresentada nesta proposição, para o triângulo de Pascal modular, é original. A prova é por indução em r.

Passo Base: Para $r = 1$,

$$\sum_{k=0}^{1} C_{i+k}^{i} = 1 + (i+1) = i + 2 = C_{i+1+1}^{i+1}.$$

Passo da Indução:

$$\sum_{k=0}^{r+1} C_{i+k}^{i} = C_{i+r+1}^{i} + \sum_{k=0}^{r} C_{i+k}^{i} = C_{i+r+1}^{i} + C_{i+r+1}^{i+1}.$$

Note que

$$\begin{aligned} C_{i+r+1}^{i} + C_{i+r+1}^{i+1} &= \frac{(i+r+1)!}{i!(r+1)!} + \frac{(i+r+1)!}{(i+1)!r!} \\ &= \frac{(i+1)(i+r_1)! + (r+1)(i+r+1)!}{(i+1)!(r+1)!} \\ &= \frac{(i+r+2)(i+r+1)!}{(i+1)!(r+1)!} = \frac{(i+r+2)!}{(i+1)!(r+1)!} = C_{i+r+2}^{i+1}. \end{aligned}$$

□

Proposição 14.

$$\sum_{k=0}^{p-1} C_{i+k}^{i} \equiv \begin{cases} 1 \ (\textit{mod}\ p), & \textit{se} \quad i = p-1, \\ 0 \ (\textit{mod}\ p), & \textit{se} \quad i \neq p-1. \end{cases}$$

Demonstração. Pela Proposição 13, tem-se que

$$\sum_{k=0}^{p-1} C_{i+k}^{i} \equiv C_{i+p}^{i+1} \ (\textit{mod}\ p).$$

Caso 1: $i = p - 1$.

$$\sum_{k=0}^{p-1} C_{(p-1)+k}^{(p-1)} = C_{(2p-1)}^{p},$$

em que

$$\begin{aligned} C_{(2p-1)}^{p} &= \frac{(2p-1)!}{p!(p-1)!} = \frac{(2p-1)(2p-2)\cdots(2p-(p-1))p!}{p!(p-1)!} \\ &= \frac{(2p-1)(2p-2)\cdots(p+1)}{(p-1)!}. \end{aligned}$$

Note que $(2p-1)(2p-2)\cdots(p+1) \equiv (p-1)(p-2)\cdots(1)$ (*mod p*) e o resultado segue.

Caso 2: $0 \leq i \leq p-2$.

Pela Proposição 1, o primeiro elemento nulo na linha i da matriz de Pascal modular de ordem p ocorre na coluna $p-i$. Então, pela Proposição 13, tem-se

$$\sum_{k=0}^{p-i-1} C_{i+k}^{i} = C_{i+(p-i-1)+1}^{i+1} = C_{p}^{i+1} \equiv 0 \ (\textit{mod } p).$$

□

Proposição 15.

$$C_{2(p-1)-i}^{p-1} \equiv \begin{cases} 1(\textit{mod } p), & \textit{se} \quad i = p-1, \\ 0(\textit{mod } p), & \textit{se} \quad i \neq p-1. \end{cases}$$

Demonstração. Para $i = p-1$,

$$C_{2(p-1)-i}^{(p-1)} = C_{(p-1)}^{(p-1)} = 1.$$

Para $0 \leq i \leq p-2$, a prova é por indução em i.

Passo Base: $i = 0$,

$$C_{2(p-1)-i}^{(p-1)} = \frac{(2p-2)!}{(p-1)!(p-1)!}.$$

Mas, note que

$$(2p-2)! = (2p-2)(2p-3)\cdots(p+1)p(p-1)!$$

e

$$C_{2(p-1)}^{(p-1)} = \frac{(2p-2)(2p-3)\cdots(p+1)p}{(p-1)!}.$$

Como $MDC(p,(p-1)!) = 1$, então, pelo Lema de Euclides (BURTON, 2006), segue-se que

$$C_{2(p-1)}^{(p-1)} = \frac{(2p-2)(2p-3)\cdots(p+1)p}{(p-1)!} = kp \equiv 0 \ (\textit{mod } p).$$

Passo da Indução: Assumindo que

$$C_{2(p-1)-i}^{(p-1)} \equiv 0 \ (\textit{mod } p)$$

e observando que

$$C_{2(p-1)-(i+1)}^{(p-1)} = \frac{(2(p-1)-(i+1))!)}{(p-1)!((p-1)-(i+1))!},$$

então

$$C_{2(p-1)-i}^{(p-1)} = \frac{(2(p-1)-i)!}{(p-1)!((p-1)-i)!},$$

em que

$$(2(p-1)-i)! = (2(p-1)-i)(2(p-1)-(i+1))!$$

e

$$((p-1)-i)! = ((p-1)-i)((p-1)-(i+1))!.$$

Assim, resulta

$$\begin{aligned} C_{2(p-1)-i}^{(p-1)} &= \frac{(2(p-1)-i)(2(p-1)-(i+1))!}{(p-1)!((p-1)-i)((p-1)-(i+1))!} \\ &= \frac{(2(p-1)-i)}{((p-1)-i)} C_{2(p-1)-(i+1)}^{(p-1)}, \end{aligned}$$

ou seja,

$$C_{2(p-1)-(i+1)}^{(p-1)} = \frac{((p-1)-i)}{(2(p-1)-i)} C_{2(p-1)-i}^{(p-1)}.$$

Mas, por hipótese,

$$C_{2(p-1)-i}^{(p-1)} \equiv 0 \ (\textit{mod}\, p), \quad i = 0, 1, \cdots, p-3.$$

Note que o termo

$$\frac{((p-1)-i)}{(2(p-1)-i)}$$

é não-nulo, módulo p, para a faixa especificada. Resulta

$$C_{2(p-1)-(i+1)}^{(p-1)} \equiv 0 \ (\textit{mod}\, p).$$

□

Proposição 16.

$$\sum_{k=0}^{p-1} C_{i+k}^{i} \equiv C_{2(p-1)-i}^{p-1} (\textit{mod}\, p).$$

Demonstração. Pelas Proposições 14 e 15, o resultado segue imediatamente. □

Proposição 17.

$$\sum_{k=0}^{p-1} C_{i+k}^{i} C_{j+k}^{j} = C_{2(p-1)-(i+j)}^{(p-1)-i} \ (mod\ p).$$

Demonstração. A prova é feita por indução em i. A prova por indução em j é similar.

Passo Base: $i = 0$,

$$\sum_{k=0}^{p-1} C_{i+k}^{i} C_{j+k}^{j} = \sum_{k=0}^{p-1} C_{j+k}^{j} = C_{2(p-1)-j}^{p-1}$$

pela Proposição 16.

Passo da Indução: Vamos assumir que

$$\sum_{k=0}^{p-1} C_{i+k}^{i} C_{j+k}^{j} \equiv C_{2(p-1)-(i+j)}^{(p-1)-i} (mod\ p).$$

Então, fazendo-se $i + 1 = t$ em

$$S = \sum_{k=0}^{p-1} C_{i+k+1}^{i+1} C_{j+k}^{j}$$

tem-se

$$S = \sum_{k=0}^{p-1} C_{t+k}^{t} C_{j+k}^{j} = C_{2(p-1)-(t+j)}^{(p-1)-t} \ (mod\ p) = C_{2(p-1)-(i+j)-1}^{(p-1)-i-1} \ (mod\ p).$$

Note que o resultado acima é válido para $i = 0, 1, \cdots, p-2$. Assim, precisamos verificar o resultado para $i = p - 1$. Se $i = p - 1$, então $i + 1 = p$. Portanto,

$$S = \sum_{k=0}^{p-1} C_{i+k+1}^{i+1} C_{j+k}^{j} = \sum_{k=0}^{p-1} C_{k+p}^{p} C_{j+k}^{j}.$$

Mas, $C_{k+p}^{p} \equiv 1 \ (mod\ p)$ e assim

$$S = \sum_{k=0}^{p-1} C_{j+k}^{j},$$

em que pela Proposição 14,

$$\sum_{k=0}^{p-1} C_{j+k}^{j} = \begin{cases} 1, & \text{se} \quad j = p-1, \\ 0, & \text{se} \quad j \neq p-1. \end{cases}$$

Portanto,

$$C^{(p-1)-i}_{2(p-1)-(i+j)} \ (mod\, p) = C^{(p-1)-(p-1)}_{2(p-1)-2(p-1)} \ (mod\, p) = C^0_0 = 1$$

e o resultado segue.

□

4 A TRANSFORMADA NUMÉRICA DE PASCAL

4.1 CONCEITOS BÁSICOS

O triângulo de Pascal tem sido utilizado em aplicações envolvendo Processamento de Sinais (GARCÍA-UGALDE; PSENICKA; JIMÉNEZ-SALINAS, 2011), (GUDVANGEN; BUSKERUD, 1999), Processamento de Imagem (GOODMAN; ABURDENE, 2006), Análise Numérica (ACETO, 2005), Antenas Fractais (ROMEU; SOLER, 2001), entre outras áreas.

Neste capítulo apresenta-se a Transformada Numérica de Pascal (TNP), baseada na versão modular do triângulo de Pascal, introduzida em (PASCHOAL; DEOLIVEIRA; SOUZA, 2015). Algumas possibilidades de aplicação desta transformada incluem, por exemplo, (1) a definição de novos códigos corretores de erros, (2) a cifragem de imagens e (3) o projeto de antenas fractais.

Definição 10. *A Transformada Numérica de Pascal da sequência* $v = (v_0, \cdots, v_{N-1})$, $v_i \in GF(p)$, *é a sequência* $V = (V_0, V_1, \cdots, V_{N-1}), V_k \in GF(p)$, *em que*

$$V_k \triangleq \sum_{i=0}^{N-1} C_{i+k}^{i} v_i \ (mod\ p). \tag{4.1}$$

Em formato matricial, escreve-se $V = Pv$, em que os elementos da matriz P são $[P]_{i,k} = C_{i+k}^{i}$. O par transformado da TNP é denotado por v $\leftrightarrow$ V.

Exemplo 10. *Considere a TNP de comprimento 5, sobre* $GF(5)$, *dada por*

$$V_k = \begin{bmatrix} V_0 \\ V_1 \\ V_2 \\ V_3 \\ V_4 \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 \\ 1 & 2 & 3 & 4 & 0 \\ 1 & 3 & 1 & 0 & 0 \\ 1 & 4 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} v_0 \\ v_1 \\ v_2 \\ v_3 \\ v_4 \end{bmatrix}.$$

A TNP da sequência $v = (1, 2, 3, 4, 0)$ *é a sequência* $V = (0, 0, 0, 4, 1)$.

■

Teorema 2. *(Transformada Inversa) A TNP inversa da sequência* $V = (V_0, ..., V_{N-1})$, $V_k \in GF(p)$, *é a sequência* $v = (v_0, ..., v_{N-1}), v_i \in GF(p)$, *em que*

$$v_i = \sum_{k=0}^{N-1} \left[(-1)^{i+k} \sum_{j=Max(i,k)}^{N-1} C_j^i C_j^k \right] V_k \ (mod\ p). \tag{4.2}$$

Demonstração. Seja $P_N = L_N \cdot U_N$ a matriz de Pascal de comprimento N, fatorada pelo método de Cholesky. A partir desta fatoração, tem-se $P_N^{-1} = U_N^{-1} \cdot L_N^{-1}$. Mas, sabemos que, pelo método de Cholesky, $U_N = L_N^T$. Portanto, resulta em $P_N^{-1} = (L_N^T)^{-1} \cdot L_N^{-1}$, em que

$$\left[L_N^{-1}\right]_{i,k} = \begin{cases} (-1)^{i-k} C_i^k, & \text{se} \quad i \geq k, \\ 0, & \text{se} \quad i < k. \end{cases} \tag{4.3}$$

A demonstração da relação anterior é relativamente simples e usa como argumento o fato de que a inversa de uma matriz triangular inferior (superior) é também uma matriz triangular inferior (superior). Então, resulta em

$$\left[P_N^{-1}\right]_{i,k} = (-1)^{i+k} \sum_{j=Max(i,k)}^{N-1} C_j^i C_j^k \tag{4.4}$$

e o resultado segue. □

Proposição 18. *As componentes da sequência* V *podem ser escritas em função da componente* V_0 *e dos valores da sequência* $v = (v_0, v_1, \cdots, v_{N-1})$ *por meio de*

$$V_k = V_0 + \sum_{i=0}^{N-2} \left[\sum_{r=i+1}^{k} C_{i+r}^r \right] v_{i+1}, \tag{4.5}$$

em que

$$V_0 = \sum_{i=0}^{N-1} v_i. \tag{4.6}$$

Demonstração. Partindo-se da relação de Stiefel, temos que

$$C_{i+k}^i = C_{i+k-1}^{i-1} + C_{i+k-1}^i, \quad i = 1, 2, \cdots$$

Multiplicando-se ambos os membros por v_i, resulta em

$$C_{i+k}^i v_i = C_{i+k-1}^{i-1} v_i + C_{i+k-1}^i v_i, \quad i = 1, 2, \cdots$$

e

$$\sum_{i=1}^{N-1} C_{i+k}^i v_i = \sum_{i=1}^{N-1} C_{i+k-1}^{i-1} v_i + \sum_{i=1}^{N-1} C_{i+k-1}^i v_i.$$

Mas, note que

$$v_0 + \sum_{i=1}^{N-1} C_{i+k-1}^i v_i = V_{k-1}$$

e, portanto,

$$V_k = V_{k-1} + \sum_{i=1}^{N-1} C_{i+k-1}^{i-1} v_i.$$

Aqui, podemos fazer a mudança de índices $j = i - 1$, para obter

$$V_k = V_{k-1} + \sum_{j=0}^{N-2} C_{j+k}^{j} v_{j+1}.$$

Vejamos agora os valores de $V_k, \quad k = 1, 2, 3, \cdots,$

$$\begin{aligned}
V_1 &= V_0 + \sum_{j=0}^{N-2} C_{j+1}^{j} v_{j+1} = V_0 + \sum_{j=0}^{N-2} (j+1) v_{j+1}, \\
V_2 &= V_1 + \sum_{j=0}^{N-2} C_{j+2}^{j} v_{j+1} = V_1 + \frac{1}{2} \sum_{j=0}^{N-2} (j+2)(j+1) v_{j+1}, \\
V_2 &= V_0 + \sum_{j=0}^{N-2} (j+1) v_{j+1} + \sum_{j=0}^{N-2} \frac{(j+2)(j+1)}{2} v_{j+1}, \\
V_2 &= V_0 + \sum_{j=0}^{N-2} \left[(j+1) + \frac{(j+2)(j+1)}{2} \right] v_{j+1}.
\end{aligned}$$

Note que a soma interna pode ser escrita como

$$(j+1) + \frac{(j+2)(j+1)}{2} = \sum_{r=1}^{2} C_{j+r}^{r}$$

e, então,

$$V_2 = V_0 + \sum_{j=0}^{N-2} \left[\sum_{r=1}^{2} C_{j+r}^{r} \right] v_{j+1}.$$

Por indução, chega-se a

$$V_k = V_0 + \sum_{j=0}^{N-2} \left[\sum_{r=j+1}^{k} C_{j+r}^{r} \right] v_{j+1}, \quad k = 1, 2, \cdots$$

□

Para uma TNP cujo comprimento é um número primo, observa-se que

(a) A TNP de um impulso é uma constante.

(b) A TNP de uma constante é um impulso deslocado.

(c) Uma dada componente V_k depende apenas das componentes $v_i,\ 0 \leq i \leq p - 1 - k$.

(d) A inversa da matriz P_p é triangular inferior em relação à diagonal secundária. Seus elementos são os mesmos de P_p porém aparecem refletidos em relação à esta diagonal.

(e) A soma dos elementos das linhas de P_p, com exceção da última linha, é congruente a zero módulo p.

(f) As complexidades aditiva e multiplicativa para se computar a TNP são, respectivamente,

$$A(p) = \frac{p(p-1)}{2} \quad \text{e} \quad M(p) = \frac{p(p+1)}{2}.$$

Note que a complexidade aritmética, entendida aqui como a soma das complexidades aditiva e multiplicativa, vale $A(p) + M(p) = p^2$.

4.1.1 Decompondo um vetor na base de Pascal

Uma vez que as linhas da matriz de Pascal modular P_p são linearmente independentes, as mesmas formam uma base do espaço vetorial formado por todas as p^p p-uplas sobre $GF(p)$, chamada base de Pascal e denotada por B_p. Neste contexto, o problema de decompor um vetor arbitrário b, sobre $GF(p)$, na base de Pascal equivale a encontrar o vetor a tal que $b = P_p a$, ou seja, o vetor a pode ser obtido por meio da TNP inversa.

Exemplo 11. *Considere o problema de decomposição do vetor $b = (b_0, b_1, \ldots, b_4)$ na base de Pascal B_5. Este problema é equivalente a*

$$\begin{bmatrix} 1 & 1 & 1 & 1 & 1 \\ 1 & 2 & 3 & 4 & 0 \\ 1 & 3 & 1 & 0 & 0 \\ 1 & 4 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} a_0 \\ a_1 \\ a_2 \\ a_3 \\ a_4 \end{bmatrix} = \begin{bmatrix} b_0 \\ b_1 \\ b_2 \\ b_3 \\ b_4 \end{bmatrix}.$$

Resolvendo-se o sistema de equações deste exemplo, chega-se a

$$\begin{aligned} a_0 &= b_4, \\ a_1 &= 4b_3 + b_4, \\ a_2 &= b_2 + 3b_3 + b_4, \\ a_3 &= 4b_1 + 3b_2 + 2b_3 + b_4, \\ a_4 &= b_0 + b_1 + b_2 + b_3 + b_4, \end{aligned}$$

ou seja

$$a = P_5^{-1} b = \begin{bmatrix} 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 4 & 1 \\ 0 & 0 & 1 & 3 & 1 \\ 0 & 4 & 3 & 2 & 1 \\ 1 & 1 & 1 & 1 & 1 \end{bmatrix} \begin{bmatrix} b_0 \\ b_1 \\ b_2 \\ b_3 \\ b_4 \end{bmatrix}.$$

Observa-se que a matriz inversa pode ser obtida diretamente da matriz P_5 por meio de uma rotação. Este resultado é válido para qualquer matriz P_p, sobre $GF(p)$, tal como descrito na Proposição 20 mais adiante.

4.2 PROPRIEDADES DA TRANSFORMADA NUMÉRICA DE PASCAL

P1: Linearidade. Se $v \leftrightarrow V$ e $u \leftrightarrow U$, então, para quaisquer $\alpha, \beta \in GF(p)$,

$$\alpha v + \beta u \leftrightarrow \alpha V + \beta U.$$

Demonstração. A prova segue diretamente da definição da TNP. □

P2: Deslocamento no Tempo. Considere a sequência $b = (b_0, b_1, \cdots, b_{N-1})$, em que $b_i = v_{i-m}$. Então $b \leftrightarrow B$, em que

$$B_k = \sum_{i=-m}^{N-1-m} C_{i+k+m}^{i+m} v_i \ (\textit{mod } p).$$

Demonstração.

$$\sum_{i=0}^{N-1} C_{i+k}^{i} b_i = \sum_{i=0}^{N-1} C_{i+k}^{i} v_{i-m}.$$

Fazendo $i - m = j$, resulta em

$$\sum_{i=0}^{N-1} C_{i+k}^{i} v_{i-m} = \sum_{j=-m}^{N-1-m} C_{j+k+m}^{j+m} v_j = \sum_{i=-m}^{N-1-m} C_{i+k+m}^{i+m} v_i.$$

□

P3: Impulso. A TNP da sequência $\delta[n] = [10 \cdots 00]$ é a sequência $V = [V_0 V_1 \cdots V_{N-1}]$, em que $V_k = 1, \ \forall k$.

Demonstração. Partindo-se da própria definição da transformada, tem-se que

$$V_k = \sum_{i=0}^{N-1} C_{i+k}^{i} v_i \ (\textit{mod } p).$$

Mas, note que apenas $v_0 \neq 0$. Assim, resulta imediatamente que

$$V_k = \sum_{i=0}^{N-1} C_{i+k}^{i} v_i = C_{0+k}^{0} v_0 = 1.$$

□

P4: TNP de uma linha da matriz P_p. A TNP da linha l_i da matriz P_p, sobre $GF(p)$, é a linha $p - 1 - i$ invertida.

Demonstração. Considere a TNP da sequência $\delta[n-i] = (0, \cdots, 0, 1, 0, \cdots, 0)$. A partir da definição da TNP, verifica-se que

$$TNP(\delta[n-i]) = l_i,$$

em que l_i é a i-ésima linha da matriz P_p. Uma vez que a matriz de transformação da TNP possui ordem igual a 3, conforme Proposição 20 provada mais adiante, resulta que

$$TNP^{-1}(\delta[n-i]) = TNP(l_i).$$

Assim, como as linhas da matriz P_p^{-1} são as linhas da matriz P_p escritas de forma inversa e refletidas em relação a linha central, ou seja, a linha i da matriz P_p é levada na posição $p-1-i$ e escrita de forma inversa. Usando notação polinomial, resulta

$$TNP(l_i(x)) = x^{p-1} l_{p-1-i}(x^{-1}).$$

□

P5: Constante. A TNP da sequência constante unitária $v = (1, 1, \cdots, 1)$ é a sequência $V = [V_0 V_1 \cdots V_{N-1}]$ em que $V_k = C_{N+k}^{k+1}$ $(mod\ p)$. Para $N = p$, a sequência $V_k = C_{N+k}^{k+1}$ é um impulso deslocado.

Demonstração. Percebe-se aqui que

$$V_k = \sum_{i=0}^{N-1} C_{i+k}^{i} v_i = \sum_{i=0}^{N-1} C_{i+k}^{i} = C_k^0 + C_{k+1}^1 + C_{k+2}^2 + \cdots + C_{k+N-1}^{N-1}.$$

O somatório anterior corresponde à soma das diagonais do triângulo de Pascal, cujo resultado é

$$C_n^0 + C_{n+1}^1 + C_{n+2}^2 + \cdots + C_{n+k}^k = C_{n+k+1}^k.$$

Portanto, resulta em

$$V_k = C_{N+k}^{N-1} = C_{N+k}^{k+1} \ (mod\ p).$$

Para $N = p$, observe que

$$V_k = C_{p+k}^{p-1} = C_{p+k}^{k+1} \equiv 0 \ (mod\ p), k = 0,\ 1,\ \cdots,\ p-2,$$

conforme Proposição 8. Para $k = p-1$, pode-se escrever

$$V_{p-1} = C_{2p-1}^{p} = \frac{(2p-1)!}{p!(p-1)!} = \frac{(2p-1)(2p-2)\cdots p(p-1)!}{p!(p-1)!}.$$

Então,

$$V_{p-1} = \frac{(2p-1)(2p-2)\cdots(p+1)p}{1\cdot 2\cdots(p-1)\cdot p} = \frac{(2p-1)(2p-2)\cdots(p+1)}{1\cdot 2\cdots(p-1)}.$$

Reduzindo-se o numerador módulo p, resulta

$$V_{p-1} \equiv \frac{1\cdot 2\cdots(p-1)}{1\cdot 2\cdots(p-1)} \,(\textit{mod}\, p) \equiv 1.$$

□

P6: Convolução Linear. A TNP da convolução das sequências $v = (v_0, v_1, \cdots, v_{N-1})$ e $u = (u_0, u_1 \cdots, u_{N-1})$ corresponde à sequência W_k em que

$$W_k = \sum_{r=0}^{N-1} v_r \sum_{i=0}^{2N-2} C_{i+k}^{i} u_{i-r}.$$

Demonstração. Sabe-se que a convolução linear discreta de duas sequências é dada por

$$w(n) = v(n) * u(n) = \sum_{i} v_i u_{n-i}.$$

Aqui vamos considerar $u[n]$ e $v[n]$ sequências de mesmo comprimento N e definidas apenas para índices positivos. Neste sentido, tem-se que

$$W(k) = TNP(v(n) * u(n)) = \sum_{i=0}^{2N-2} C_{i+k}^{i} \sum_{r=0}^{N-1} v_r u_{i-r}, \quad k = 0, 1, 2, \cdots, 2N-2.$$

Como $u_{i-r} = 0, \quad i - r < 0$ ou $i - r \geq N$, então

$$W(k) = \sum_{r=0}^{N-1} v_r \sum_{i=0}^{2N-2} C_{i+k}^{i} u_{i-r}.$$

□

A Proposição P6 mostra que a TNP da convolução linear entre as sequências $u[n]$ e $v[n]$ corresponde à soma das transformadas numéricas de Pascal da sequência $u[n]$ deslocada e ponderada pela sequência $v[n]$.

Exemplo 12. *Considere a TNP sobre $GF(5)$ da convolução entre as sequências $v[n] = (1, 2, 3)$ e $u[n] = (4, 0, 1)$. A convolução entre $v[n]$ e $u[n]$ é a sequência $w[n] = (4, 3, 3, 2, 3)$ cuja TNP vale $(0, 4, 1, 1, 4)$. Vejamos a TNP da sequência $u[n]$ e de suas versões deslocadas.*

$$\begin{bmatrix} 1 & 1 & 1 & 1 & 1 \\ 1 & 2 & 3 & 4 & 0 \\ 1 & 3 & 1 & 0 & 0 \\ 1 & 4 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 \end{bmatrix} \cdot \begin{bmatrix} 4 \\ 0 \\ 1 \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} 0 \\ 2 \\ 0 \\ 4 \\ 4 \end{bmatrix},$$

$$\begin{bmatrix} 1 & 1 & 1 & 1 & 1 \\ 1 & 2 & 3 & 4 & 0 \\ 1 & 3 & 1 & 0 & 0 \\ 1 & 4 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 \end{bmatrix} \cdot \begin{bmatrix} 0 \\ 4 \\ 0 \\ 1 \\ 0 \end{bmatrix} = \begin{bmatrix} 0 \\ 2 \\ 2 \\ 1 \\ 0 \end{bmatrix},$$

$$\begin{bmatrix} 1 & 1 & 1 & 1 & 1 \\ 1 & 2 & 3 & 4 & 0 \\ 1 & 3 & 1 & 0 & 0 \\ 1 & 4 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 \end{bmatrix} \cdot \begin{bmatrix} 0 \\ 0 \\ 4 \\ 0 \\ 1 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 4 \\ 0 \\ 0 \end{bmatrix}.$$

Efetuando a soma ponderada pela sequência $v[n] = (1, 2, 3)$*, resulta*

$$1 \cdot \begin{bmatrix} 0 \\ 2 \\ 0 \\ 4 \\ 4 \end{bmatrix} + 2 \cdot \begin{bmatrix} 0 \\ 2 \\ 2 \\ 1 \\ 0 \end{bmatrix} + 3 \cdot \begin{bmatrix} 0 \\ 0 \\ 4 \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} 0 \\ 4 \\ 1 \\ 1 \\ 4 \end{bmatrix}.$$

■

P7: TNP da Convolução Cíclica entre as linhas da matriz P_p.

Teorema 3. *A TNP da convolução cíclica, sobre $GF(p)$, das linhas l_i e l_j da matriz P_p, em sua forma polinomial, é dada por*

$$TNP(l_i(x) \circledast_p l_j(x)) = \begin{cases} (0 \ldots 0), & \textit{se } i + j + 1 < p, \\ x^{p-(i+j+1)} l_{(p-(i+j+1))}(x^{-1}), & \textit{caso contrário}. \end{cases}$$

Demonstração. Inicialmente, mostra-se que a convolução cíclica entre as linhas l_i e l_j da matriz P_p é dada por

$$l_i \circledast_p l_j = \begin{cases} (0 \ldots 0), & \text{se } i + j < (p-1), \\ l_{(i+j+1)(mod\, p)}, & \text{caso contrário.} \end{cases}$$

Escrevendo-se as linhas i e j como polinômios, tem-se

$$l_i(x) \triangleq C_i^i + xC_{i+1}^i + x^2 C_{i+2}^i + \ldots + x^{p-1-i} C_{p-1}^i,$$
$$l_j(x) \triangleq C_j^j + xC_{j+1}^j + x^2 C_{j+2}^j + \ldots + x^{p-1-j} C_{p-1}^j.$$

A convolução cíclica entre a sequência formada pelos elementos da linha i e a sequência formada pelos elementos da linha j corresponde à sequência associada ao produto dos polinômios $l_i(x)$

e $l_j(x)$ reduzidos módulo $(x^p - 1)$. Assim, se $grau(l_i(x)l_j(x)) < p$ então não haverá redução modular e, portanto, o resultado será dado pelo produto (não-nulo) $l_i(x)l_j(x)$. Isto ocorre quando

$$2(p-1) - (i+j) < p,$$

ou seja,

$$i + j + 1 \geq p.$$

Assim, se montarmos uma tabela com todas as possíveis convoluções cíclicas entre as linhas da matriz de transformação P_p, iremos observar que na diagonal secundária e abaixo dela, teremos apenas convoluções não-nulas. Por outro lado, se $i + j + 1 \geq p$, então

$$\begin{aligned} i + j + 1 &> p - 1, \\ p - 2 &< i + j, \\ (p - 1 - i) + (p - 1 - j) &< p. \end{aligned}$$

Portanto, o grau do polinômio obtido pelo produto dos polinômios $l_i(x)$ por $l_j(x)$ é menor do que p e, assim, não há redução modular e o produto é não-nulo. Logo, a convolução é não-nula se, e somente se, $i + j + 1 \geq p$. Finalmente, observe que

$$(1-x)^{p-1-i} = \sum_{k=0}^{p-1-i} C^k_{p-1-i}(-x)^k.$$

Mas,

$$\begin{aligned} C^k_{p-1-i} &= \frac{[p-(i+1)][p-(i+2)]\ldots[p-(i+k)][p-(i+k+1)]!}{k![p-(i+k+1)]!} \\ &\equiv \frac{(-1)^k(i+k)\ldots(i+2)(i+1)}{k!}\frac{i!}{i!} \\ &= (-1)^k C^k_{i+k}. \end{aligned}$$

Portanto,

$$(1-x)^{p-1-i} = \sum_{k=0}^{p-1-i} (-1)^k C^k_{i+k}(-x)^k = l_i(x),$$

em que $l_i(x)$ é o polinômio que representa a i-ésima linha da matriz de Pascal. Note que, na matriz P_p, $l_{p-2}(x) = (1-x)$; assim,

$$l_i(x) = [l_{p-2}(x)]^{p-1-i}.$$

Este é um resultado marcante, pois mostra que qualquer linha da matriz de Pascal P_p pode ser obtida como uma potência da linha l_{p-2}; ou seja, a linha l_{p-2} contém toda informação necessária para se construir a matriz P_p, funcionando como uma linha *geradora* da matriz.

Como estamos considerando $i+j+1 \geq p$, podemos escrever $i+j+1 = p+r$, em que $0 \leq r \leq p-1$. Assim, a convolução entre as linhas i e j pode ser obtida do produto

$$\begin{aligned} l_i(x)l_j(x) &= (l_{p-2}(x))^{p-1-i}(l_{p-2}(x))^{p-1-j} = (l_{p-2}(x))^{(p-1)-(i+j+1-p)} \\ &= (l_{p-2}(x))^{(p-1-r)} = l_r(x) = l_{(i+j+1)(mod\,p)}(x), \end{aligned}$$

portanto, a TNP da convolução cíclica, sobre $GF(p)$, entre as linhas l_i e l_j da matriz de Pascal, em sua forma polinomial, é dada por

$$TNP(l_i(x) \circledast_p l_j(x)) = \begin{cases} (0 \ldots 0), & \text{se } i+j+1 < p, \\ x^{p-(i+j+1)} l_{(p-(i+j+1))}(x^{-1}), & \text{caso contrário.} \end{cases}$$

Note que se $i+j = i'+j'$, então $l_i(x)l_j(x) = l_{i'}(x)l_{j'}(x)$. Assim, a matriz das convoluções cíclicas é uma matriz Hankel. □

A Tabela a seguir, ilustra o cálculo das convoluções cíclicas entre as linhas da matriz P_5.

Tabela 5 – Convoluções Cíclicas entre as linhas de P_5.

$\circledast_p$	l_0	l_1	l_2	l_3	l_4
l_0	00000	00000	00000	00000	11111
l_1	00000	00000	00000	11111	12340
l_2	00000	00000	11111	12340	13100
l_3	00000	11111	12340	13100	14000
l_4	11111	12340	13100	14000	10000

Exemplo 13. *Considere a convolução cíclica módulo 5 das sequências, definidas sobre* $GF(5)$, $b^{(1)} = (2,0,4,4,0)$ *e* $b^{(2)} = (2,1,3,4,0)$. *De acordo com a Subseção 4.1.1, a decomposição das sequências* $b^{(1)}$ *e* $b^{(2)}$ *na base* $B_5 = (l_0, l_1, l_2, l_3, l_4)$ *resulta em* $a^{(1)} = (0,1,1,0,0)$ *e* $a^{(2)} = (0,1,0,1,0)$, *ou seja,*

$$\begin{aligned} b^{(1)} &= l_1 + l_2, \\ b^{(2)} &= l_1 + l_3. \end{aligned}$$

Portanto,

$$\begin{aligned} b^{(1)} \circledast_5 b^{(2)} &= (l_1 + l_2) \circledast_5 (l_1 + l_3) \\ &= l_1 \circledast_5 l_1 + l_1 \circledast_5 l_3 + l_2 \circledast_5 l_1 + l_2 \circledast_5 l_3. \end{aligned}$$

Usando a Propriedade P7, resulta

$$\begin{aligned} b^{(1)} \circledast_5 b^{(2)} &= l_1 \circledast_5 l_3 + l_2 \circledast_5 l_3 \\ &= l_0 + l_1 = (1,1,1,1,1) + (1,2,3,4,0) = (2,3,4,0,1). \end{aligned}$$

P8: Teorema da Convolução Cíclica

Teorema 4. *A TNP da convolução cíclica entre as sequências $b^{(1)}$ e $b^{(2)}$ de comprimento p, sobre $GF(p)$, pode ser obtida por*

$$TNP\{b^{(1)} \circledast_p b^{(2)}\} = \sum_{\substack{i,j,\\ i+j+1\geq p}} a_i^{(1)} a_j^{(2)} l^{(inv)}_{(p-1)-(i+j+1)},$$

em que $l_i^{(inv)}$ representa a i-ésima linha da matriz P_p invertida, $a^{(1)} \leftrightarrow b^{(1)}$ e $a^{(2)} \leftrightarrow b^{(2)}$.

■

Demonstração. A demonstração deste Teorema decorre diretamente da decomposição das sequências $b^{(1)}$ e $b^{(2)}$ na base de Pascal (Seção 4.1.1), da Propriedade P7 e da Propriedade P4. □

Existem várias formas de se definir a correlação cíclica entre duas sequências definidas sobre $GF(p)$.

P9: TNP da Correlação Cíclica entre linhas da matriz P_p

Definição 11. *A correlação cruzada cíclica entre as sequências $u = (u_0,\ u_1,\ \cdots,\ u_{N-1})$ e $v = (v_0,\ v_1,\ \cdots,\ v_{N-1})$, definidas sobre $GF(p)$, é a sequência $r = (r_0,\ r_1,\ \cdots,\ r_{N-1})$, em que r_n é dada por*

$$r_n \triangleq \sum_{k=0}^{N-1} u_{(n+k)(mod\ p)} v_k.$$

Da mesma forma que acontece na correlação usual, a correlação aqui definida não é comutativa. Considerando-se a versão polinomial da Definição 11, tem-se

$$r(x) = u(x^{-1})v(x)(\textit{mod}\ x^p\ \textit{-1}).$$

Como as sequências u e v podem ser decompostas na base de Pascal, resulta que a correlação cíclica r é, na verdade, uma correlação entre combinações lineares de linhas da matriz de Pascal. Mas, isto corresponde a uma soma de correlações ponderadas pelos respectivos coeficientes das combinações lineares.

Teorema 5. *A TNP da correlação cruzada cíclica entre as linhas l_i e l_j da matriz de Pascal P_p, sobre $GF(p)$, é dada por*

$$TNP(r_n(l_i; l_j)) = \begin{cases} (0 \ldots 0), & \textit{se } i+j+1 < p, \\ TNP((p-1)^{i+2} x^{i+1} l_{i+j+1(mod\ p)}), & \textit{caso contrário}, \end{cases}$$

em que o produto $x^{i+1} l_{i+j+1}$ representa a $(i+j+1)$-ésima linha da matriz de Pascal deslocada, à direita, por $(i+1)$ posições.

■

Demonstração. Inicialmente, mostra-se que a correlação cíclica entre as linhas l_i e l_j da matriz P_p é dada por

$$r_n(l_i; l_j) = \begin{cases} (0 \dots 0), & \text{se } i+j+1 < p, \\ (p-1)^{i+2} x^{i+1} l_{i+j+1 (mod\, p)}, & \text{caso contrário.} \end{cases}$$

Vimos que

$$l_i(x) = [l_{p-2}(x)]^{p-1-i}.$$

Então, podemos escrever

$$l_i(x^{-1}) = l_i(x^{p-1}) = [l_{p-2}(x^{p-1})]^{p-1-i},$$

de modo que, como $l_{p-2}(x) = (1 + (p-1)x)$, resulta

$$\begin{aligned} l_{p-2}(x^{p-1}) &= (1 + (p-1)x^{p-1}) = (x^p + (p-1)x^{p-1}) \\ &= (p-1)x^{p-1}(1 + (p-1)x) = (p-1)x^{p-1} l_{p-2}(x). \end{aligned}$$

Portanto,

$$\begin{aligned} r_n(l_i; l_j) &= l_i(x^{-1}) l_j(x) = [(p-1)x^{p-1} l_{p-2}(x)]^{p-1-i} [l_{p-2}(x)]^{p-1-j} \\ &= [(p-1)x^{p-1}]^{p-1-i} l_i(x) l_j(x), \end{aligned}$$

de modo que, pela Propriedade P6, quando $i + j + 1 < p$, tem-se $r_n(l_i; l_j) = (0, \cdots, 0)$. Caso contrário, ainda pela Propriedade P6, tem-se

$$\begin{aligned} r_n(l_i; l_j) &= (p-1)^{p-1-i} x^{(p-1)(p-1-i)} [l_{p-2}(x)]^{p-1-(i+j+1)} \\ &= (p-1)^{p-1-i} x^{(p-1)(p-1-i)} l_{i+j+1} \\ &= (p-1)^{i+2} x^{i+1} l_{i+j+1} \end{aligned}$$

e o resultado segue. □

Note que, pela Propriedade P6, sempre que $i + j + 1 < p$, a convolução cíclica e a correlação cíclica, ambas, se anulam. A Tabela 6 apresenta as correlações cíclicas entre as linhas da matriz de Pascal P_5.

Tabela 6 – Correlação Cruzada Cíclica entre as linhas de P_5

$r_n(l_i; l_j)$	l_0	l_1	l_2	l_3	l_4
l_0	00000	00000	00000	00000	11111
l_1	00000	00000	00000	44444	10432
l_2	00000	00000	11111	34012	10013
l_3	00000	44444	32104	24004	10004
l_4	11111	12340	13100	14000	10000

A Tabela 7 apresenta as mesmas correlações mostradas na Tabela 6, sendo indicado o papel de linha geradora da linha $l_{p-2} = l_3$ da matriz de Pascal P_5.

Tabela 7 – Correlação Cruzada Cíclica entre as linhas de P_5 em função do polinômio $l_3(x)$.

$r_n(l_i; l_j)$	$l_0(x)$	$l_1(x)$	$l_2(x)$	$l_3(x)$	$l_4(x)$
$l_0(x)$	00000	00000	00000	00000	$[l_3(x)]^4$
$l_1(x)$	00000	00000	00000	$4\,[l_3(x)]^4$	$[l_3(x^4)]^3$
$l_2(x)$	00000	00000	$[l_3(x)]^4$	$x^3\,[l_3(x)]^3$	$[l_3(x^4)]^2$
$l_3(x)$	00000	$4\,[l_3(x)]^4$	$x^2\,[l_3(x^4)]^3$	$4\,[l_3(x^4)]^2$	$[l_3(x^4)]^1$
$l_4(x)$	$[l_3(x)]^4$	$[l_3(x)]^3$	$[l_3(x)]^2$	$[l_3(x)]$	$[l_3(x)]^0$

Exemplo 14. *Considere o cálculo da correlação cíclica na forma polinomial, sobre $GF(5)$, entre os vetores $u = (3, 3, 0, 4, 0)$ e $v = (4, 0, 1, 0, 0)$. Em primeiro lugar calculam-se as TNPs inversas destes vetores para se obter os coeficientes das combinações lineares que deram origem aos mesmos:*

$$\begin{bmatrix} 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 4 & 1 \\ 0 & 0 & 1 & 3 & 1 \\ 0 & 4 & 3 & 2 & 1 \\ 1 & 1 & 1 & 1 & 1 \end{bmatrix} \cdot \begin{bmatrix} 3 \\ 3 \\ 0 \\ 4 \\ 0 \end{bmatrix} = \begin{bmatrix} 0 \\ 1 \\ 2 \\ 0 \\ 0 \end{bmatrix};$$

logo, $u = l_1 + 2l_2$.

$$\begin{bmatrix} 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 4 & 1 \\ 0 & 0 & 1 & 3 & 1 \\ 0 & 4 & 3 & 2 & 1 \\ 1 & 1 & 1 & 1 & 1 \end{bmatrix} \cdot \begin{bmatrix} 4 \\ 0 \\ 1 \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 1 \\ 3 \\ 0 \end{bmatrix},$$

logo, $v = l_2 + 3l_3$. Portanto, resulta

$$\begin{aligned} r_n(l_1 + 2l_2; l_2 + 3l_3) &= r_n(l_1; l_2) + r_n(l_2; l_3) + 2r_n(l_2; l_2) + 3r_n(l_1; l_3) \\ &= r_n(l_2; l_3) + 2r_n(l_2; l_2) + 3r_n(l_1; l_3). \end{aligned}$$

Note que a primeira correlação cíclica é nula porque a convolução cíclica entre as mesmas linhas é nula. As outras correlações cíclicas podem ser encontradas usando-se o Teorema 5:

$$\begin{aligned} r_n(l_2; l_3) &= x^3 l_1 = (3, 4, 0, 1, 2); \\ r_n(l_2; l_2) &= x^3 l_0 = l_0 = (1, 1, 1, 1, 1); \\ r_n(l_1; l_3) &= 4x^2 l_0 = (4, 4, 4, 4, 4). \end{aligned}$$

Assim, resulta $r_n(u; v) = (3, 4, 0, 1, 2) + 2(1, 1, 1, 1, 1) + 3(4, 4, 4, 4, 4) = (2, 3, 4, 0, 1)$.

■

A Tabela 8 a seguir resume as propriedades da Transformada Numérica de Pascal.

Tabela 8 – Propriedades da TNP

Propriedade	Domínio do Tempo	Domínio da TNP
Par Transformado	$v = (v_i)$	$V = (V_k)$
Linearidade	α v + β u	α V+ β U
Deslocamento no tempo	v_{i-m}	$\sum_{i=-m}^{N-1-m} C_{i+k+m}^{i+m} v_i$
Impulso	$\delta[n] = [10\cdots00]$	$V = [11\cdots1]$
Linha i_0 de P_p	l_{i_0}	l_{p-1-i_0} refletida
Linha $p-1-i_0$ refletida de P_p	l_{p-1-i_0} refletida	$\delta[i-i_0]$
Impulso deslocado	$\delta[n-i_0] = [0\ \cdots 1\ \cdots 0]$	l_{i_0}
Constante	$v = [1, 1, \cdots 1]$	$V_k = C_{N+k}^{N-1} = C_{N+k}^{k+1}$ $N = p \Rightarrow V_k = [00\cdots1]$
Convolução Linear	$v(n) * u(n)$	$\sum_{r=0}^{N-1} v_r \sum_{i=0}^{2N-2} C_{i+k}^{i} u_{i-r}$
Convolução Cíclica entre linhas de P_p	$l_i \circledast_p l_j$	$\begin{cases} (0\ldots0), \text{ se } i+j+1 < p, \\ x^{p-(i+j+1)} l_{(p-(i+j+1))}(x^{-1}), \text{ c.c.} \end{cases}$
Teorema da Convolução Cíclica	$b^{(1)} \circledast_p b^{(2)}$	$\sum_{\substack{i,j,\\ i+j+1\geq p}} a_i^{(1)} a_j^{(2)} l_{(p-1)-(i+j+1)}^{(inv)}$,
Correlação Cruzada Cícli-ca entre linhas de P_p	$r_n(l_i; l_j)$	$TNP((p-1)^{i+2} x^{(i+1)} (l_i \circledast_p l_j))$

Considerando a matriz P_5 sobre $GF(5)$, a Tabela 9 a seguir mostra alguns pares transformados da TNP de comprimento $N = p = 5$.

$$P_5 = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 \\ 1 & 2 & 3 & 4 & 0 \\ 1 & 3 & 1 & 0 & 0 \\ 1 & 4 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 \end{bmatrix},$$

Tabela 9 – Pares transformados, sobre $GF(5)$, da TNP de comprimento 5.

$11111 \leftrightarrow 00001 \leftrightarrow 10000 \leftrightarrow 11111.$
$12340 \leftrightarrow 00041 \leftrightarrow 01000 \leftrightarrow 12340.$
$13100 \leftrightarrow 00131 \leftrightarrow 00100 \leftrightarrow 13100.$
$14000 \leftrightarrow 04321 \leftrightarrow 00010 \leftrightarrow 14000.$
$10000 \leftrightarrow 11111 \leftrightarrow 00001 \leftrightarrow 10000.$

4.3 A AUTOESTRUTURA DA TRANSFORMADA NUMÉRICA DE PASCAL

Proposição 19. *A matriz de Pascal cujo comprimento é $N = p^r$ satisfaz $P_N^2 = LP_NL$, em que L é a matrix $N \times N$*

$$L = \begin{bmatrix} 0 & 0 & 0 & \dots & 0 & 1 \\ 0 & 0 & 0 & \dots & 1 & 0 \\ \vdots & \vdots & \vdots & \iddots & \vdots & \vdots \\ 0 & 1 & 0 & \dots & 0 & 0 \\ 1 & 0 & 0 & \dots & 0 & 0 \end{bmatrix}.$$

Demonstração. A Proposição 17 nos fornece a identidade

$$\sum_{k=0}^{p-1} C_{i+k}^{i} C_{j+k}^{j} \equiv C_{2(p-1)-(i+j)}^{(p-1)-i} \ (\textit{mod}\ p),$$

quando $N = p$. O lado esquerdo desta identidade é P_N^2 para $N = p$. Observe que o produto LP_N tem o efeito de reescrever a matriz P_N trocando as posições das linhas da seguinte forma, a i-ésima linha é levada na $(p-1-i)$-ésima linha. Ou seja, produz o mapeamento, $i \to p-1-i$. O produto à direita pela matriz L, de forma similar, produz o mapeamento $j \to p-1-j$. Assim, para $N = p$ podemos escrever que os elementos da matriz produto LP_NL são dados por $C_{p-1-i+p-1-j}^{p-1-i}$, ou seja, $C_{2(p-1)-(i+j)}^{p-1-i}$. Os resultados acima são válidos quando $N = p^r$ pois a matriz P_N^2 é dada pela mesma expressão do lado esquerdo com $(p-1)$ substituído por $p^r - 1$ e o efeito da matriz L sobre a matriz P_N não muda quando o comprimento muda, seja multiplicando à direita ou à esquerda. □

Proposição 20. *A matriz de Pascal sobre $GF(p)$, em que p é um primo ímpar, cuja ordem é $N = p^r$, satisfaz $[P_N]^3 = I_N$.*

Demonstração. A matriz de Pascal cujo comprimento é $N = p^r$ possui a propriedade

$$P_N^2 = LP_NL,$$

em que a matriz L é uma matriz de ordem 2. Assim, note que

$$P_N^4 = LP_NLLP_NL = LP_N^2L$$

e

$$LP_N^2L = LLP_NLL = P_N.$$

Então,

$$P_N^4 = P_N \Rightarrow P_N^3 = I_N.$$

□

Proposição 21. *Os autovalores associados à matriz de transformação da TNP de ordem $N = p^r$, sobre $GF(p)$, em que p é um primo ímpar, satisfazem $\lambda^3 = 1$.*

Demonstração. Note que se v é um autovetor da TNP com autovalor associado λ, então $P_N v = \lambda v$. Da Proposição 20, pode-se escrever $P_N^3 v = \lambda^3 v = v$ e, portanto, $\lambda^3 = 1$. □

Proposição 22. *O traço da matriz de Pascal de ordem $N = p$, P_N, sobre $GF(p)$, $Tr(P_N) \triangleq \sum_{i=0}^{p-1} C_{2i}^i = (-3)^{(\frac{p-1}{2})}$, satisfaz*

$$Tr(P_N) = \begin{cases} 1, & \textit{se} \quad 3|(p-1), \\ -1, & \textit{se} \quad 3 \nmid (p-1). \end{cases}$$

Demonstração. Aqui usamos a identidade (BACHER; CHAPMAN, 2004)

$$C_{2i}^i = (-4)^i C_{-1/2}^i = (-4)^i C_{(\frac{p-1}{2})}^i.$$

Assim, do binômio de Newton, resulta

$$\sum_{i=0}^{p-1} C_{2i}^i = \sum_{i=0}^{p-1} (-4)^i C_{-1/2}^i = \sum_{i=0}^{\frac{p-1}{2}} (-4)^i C_{-1/2}^i = (-4+1)^{(\frac{p-1}{2})} = (-3)^{(\frac{p-1}{2})}.$$

Aqui usamos o fato de que $C_{2i}^i \equiv 0 \ (\textit{mod}\ p)$, para $i = \frac{p+1}{2}, \ldots, (p-1)$. Note que podemos expressar $(-3)^{(\frac{p-1}{2})}$ por meio do símbolo de Legendre e usar a Lei da Reciprocidade Quadrática (BURTON, 2006) para escrever

$$Tr(P_N) = \sum_{i=0}^{p-1} C_{2i}^i = (-3)^{(\frac{p-1}{2})} = (-3/p) = (p/3) = \begin{cases} 1, & \text{se} \quad 3|(p-1), \\ -1, & \text{se} \quad 3 \nmid (p-1). \end{cases}$$

□

Proposição 23. *A matriz de transformação da TNP de ordem $N = p > 3$, sobre $GF(p)$, é diagonalizável com autovalores que são as raízes cúbicas da unidade, a saber, 1, λ e λ^2. No caso em que $3 \mid (p-1)$, λ e λ^2 pertencem ao corpo base. As multiplicidades dos autovalores 1, λ e λ^2 são $\frac{p+2}{3}$, $\frac{p-1}{3}$ e $\frac{p-1}{3}$, respectivamente. Caso $3 \nmid (p-1)$, os autovalores λ e λ^2 estão no corpo de extensão $GF(p^2)$ gerado por $\pi(x) = 1 + x + x^2$ e as multiplicidades de 1, λ e λ^2 são $\frac{p-2}{3}$, $\frac{p+1}{3}$ e $\frac{p+1}{3}$, respectivamente.*

Demonstração. Sabemos que os autovalores da matriz de transformação satisfazem à equação $x^3 = 1$. Esta equação possui três raízes, uma delas é o valor unitário. Note que se λ é uma raiz, então, λ^2 também é raiz, pois $(\lambda^2)^3 = \lambda^6 = (\lambda^3)^2 = 1$. Portanto, 1, λ e λ^2 são as três raízes de $x^3 = 1$. Mas, podemos escrever $x^3 - 1 = (x-1)(x^2 + x + 1)$. No caso em que $3 \mid (p-1)$, pela Lei da Reciprocidade Quadrática, as raízes do polinômio $x^2 + x + 1$ estão sobre $GF(p)$. Caso contrário, as outras raízes de $x^3 - 1 = 0$ estão no corpo de extensão quadrático gerado por

$x^2 + x + 1$. Note que as matrizes P_N e P_N^2 possuem o mesmo polinômio característico, uma vez que, pela Proposição 19, tem-se que $P_N^2 = LP_NL$ e como os autovalores de P_N são obtidos a partir de

$$det(P_N - \lambda I) = 0,$$

então, se

$$A = P_N - \lambda I,$$

pode-se escrever

$$LAL = LP_NL - L\lambda IL = P_N^2 - \lambda I.$$

Mas,

$$det(LAL) = det(A).$$

Logo, P_N e P_N^2 possuem o mesmo polinômio característico.

Considere que a expressão para o traço da matriz de transformação P_N seja $Tr(P_N) = a + b\lambda + c\lambda^2$, em que a, b e c são as multiplicidades das raízes 1, λ e λ^2, respectivamente. Como as multiplicidades de λ e λ^2 são iguais, resulta que $b = c$. Ademais as raízes de $\lambda^2 + \lambda + 1$ satisfazem a $\lambda^2 + \lambda = -1$. Então, podemos escrever

$$\begin{cases} a + b + c = p, \\ b = c. \end{cases}$$

Pela Proposição 22, se $3|(p-1)$, então, $Tr(P_N) = 1$, caso contrário $Tr(P_N) = -1$. Para o caso em que $3 \mid (p-1)$ tem-se

$$\begin{cases} a + b + c = p, \\ b = c, \\ Tr(P_N) = a + b\lambda + c\lambda^2 = 1. \end{cases}$$

Somando as duas primeiras equações, resulta

$$a + 2b = p \Rightarrow b = \frac{p-a}{2}.$$

Mas, como $b = c, Tr(P_N) = a + b(\lambda + \lambda^2) = 1$ e lembrando que $\lambda^2 + \lambda + 1 = 0$, resulta

$$a - b = 1 \Rightarrow b = a - 1.$$

Então,

$$\frac{p-a}{2} = a - 1 \Rightarrow a = \frac{p+2}{3} \Rightarrow b = c = \frac{p-1}{3}.$$

Para o caso em que $3 \nmid (p-1)$, tem-se $Tr(P_N) = -1$. Daí, segue-se que

$$\begin{cases} a + b + c = p, \\ b = c, \\ Tr(P_N) = a + b\lambda + c\lambda^2 = -1. \end{cases}$$

Somando as duas primeiras equações, resulta

$$a + 2b = p \Rightarrow b = \frac{p-a}{2}.$$

Mas, como $b = c, Tr(P_N) = a + b(\lambda + \lambda^2) = -1$ e lembrando que $\gamma^2 + \lambda + 1 = 0$, resulta

$$a - b = -1 \Rightarrow b = a + 1.$$

Assim,

$$\frac{p-a}{2} = a + 1 \Rightarrow a = \frac{p-2}{3} \Rightarrow b = c = \frac{p+1}{3}.$$

□

Exemplo 15. *Considere a matriz de Pascal sobre $GF(7)$ de ordem $N = 7$. Como $3 \mid (p-1)$, resulta*

a) Todos os autovalores estão sobre o corpo base, a saber 1, 2 e $2^2 = 4$.
b) As multiplicidades são $a = \frac{p+2}{3} = 3$ e $b = \frac{p-1}{3} = c = 2$.

De fato, o polinômio característico para P_7 é $p(x) = 1 + 6x + 5x^3 + 2x^4 + x^6 + 6x^7$, cuja fatoração é $p(x) = 6(3+x)^2(5+x)^2(6+x)^3$. ■

Exemplo 16. *Considere a matriz de Pascal sobre $GF(11)$ de ordem $N = 11$. Como $3 \nmid (p-1)$, resulta*

a) Existem dois autovalores no corpo de extensão $GF(11^2)$, α^{40} e α^{80}.
b) As multiplicidades dos autovalores são $a = (p-2)/3 = 3$ e $b = (p+1)/3 = c = 4$.

De fato, o polinômio característico para P_{11} é $p(x) = 1 + x + x^2 + 8x^3 + 8x^4 + 8x^5 + 3x^6 + 3x^7 + 3x^8 + 10x^9 + 10x^{10} + 10x^{11}$, cuja fatoração é $p(x) = 10(10+x)^3(1+x+x^2)^4$. ■

Quando $N = p > 3$ for a ordem da matriz de Pascal, então as multiplicidades de seus autovalores estão distribuídas conforme a Tabela 10.

Tabela 10 – Multiplicidade dos autovalores da matriz de Pascal de comprimento $N = p$.

N	1	λ	λ^2
$3k$+1	k+1	k	k
$3k$+2	k	k+1	k+1

Proposição 24. *A matriz de transformação da TNP de ordem $N = p^r, (p > 3)$, sobre $GF(p)$, é diagonalizável com autovalores que são as raízes cúbicas da unidade, a saber, 1, λ e λ^2. No caso em que $3 \mid (p-1)$, λ e λ^2 pertencem ao corpo base. As multiplicidades dos autovalores 1, λ e λ^2 para $p = 3k+1$ $(\forall r)$ ou $p = 3k+2$ (r par) são $\frac{p^r+2}{3}$, $\frac{p^r-1}{3}$ e $\frac{p^r-1}{3}$, respectivamente. Caso $p = 3k+2$ (r ímpar), os autovalores λ e λ^2 estão no corpo de extensão $GF(p^2)$ gerado pelo polinômio $1 + x + x^2$ e as multiplicidades de 1, λ e λ^2 são $\frac{p^r-2}{3}$, $\frac{p^r+1}{3}$ e $\frac{p^r+1}{3}$, respectivamente.*

Demonstração. Uma vez que

$$Tr(P_{p^r}) = Tr(P_p \bigotimes P_p \ldots \bigotimes P_p) = Tr(P_p)Tr(P_p)\ldots Tr(P_p) = (Tr(P_p))^r$$

e como

$$Tr(P_p) = \sum_{i=0}^{p-1} C_{2i}^i = (-3)^{(\frac{p-1}{2})} = \begin{cases} 1, & \text{se } 3|(p-1), \\ -1, & \text{se } 3 \nmid (p-1), \end{cases}$$

segue-se que

$$Tr(P_{p^r}) = \begin{cases} 1, \text{ se } p = 3k+1, \forall r, \\ -1, \text{ se } p = 3k+2, r \text{ ímpar }, \\ 1 \text{ se } p = 3k+2, r \text{ par }. \end{cases}$$

Da mesma forma que antes,

$$\begin{cases} a+b+c = p^r, \\ b = c. \end{cases}$$

Como ocorreu antes, as matrizes P_N e P_N^2 possuem o mesmo polinômio característico.

Como $\lambda^3 - 1 = (\lambda - 1)(\lambda^2 + \lambda + 1)$, as outras raízes de $\lambda^3 - 1$ estão no corpo de extensão quadrático gerado por $(\lambda^2 + \lambda + 1)$. Para o caso em que $3 \mid (p-1)$ ou $p = 3k+2$ (r par) tem-se:

$$\begin{cases} a+b+c = p^r, \\ b = c, \\ Tr(P_N) = a + b\lambda + c\lambda^2 = 1. \end{cases}$$

Somando as duas primeiras equações, resulta

$$a + 2b = p^r \Rightarrow b = \frac{p^r - a}{2}.$$

Mas, como $b = c$, $Tr(P_N) = a + b(\lambda + \lambda^2) = 1$ e lembrando que $\lambda^2 + \lambda + 1 = 0$, resulta

$$a - b = 1 \Rightarrow b = a - 1.$$

Então,

$$\frac{p^r - a}{2} = a - 1 \Rightarrow a = \frac{p^r + 2}{3} \Rightarrow b = c = \frac{p^r - 1}{3}.$$

Para o caso em que $p = 3k+2$ (r ímpar), tem-se $Tr(P_N) = -1$. Daí, segue-se que

$$\begin{cases} a+b+c = p^r, \\ b = c, \\ Tr(P_N) = a + b\lambda + c\lambda^2 = -1. \end{cases}$$

Somando as duas primeiras equações, resulta

$$a + 2b = p^r \Rightarrow b = \frac{p^r - a}{2}.$$

Mas, como $b = c,\ Tr(P_N) = a + b(\lambda + \lambda^2) = -1$ e lembrando que $\lambda^2 + \lambda + 1 = 0$, resulta

$$a - b = -1 \Rightarrow b = a + 1.$$

Assim,

$$\frac{p^r - a}{2} = a + 1 \Rightarrow a = \frac{p^r - 2}{3} \Rightarrow b = c = \frac{p^r + 1}{3}.$$

□

Exemplo 17. *Os autovalores de P_{25} são 1, α^8 e α^{16}. Como $p = 3k+2$ (r par), então, $Tr(P_N) = 1$. Logo,*

$$a = \frac{5^2 + 2}{3} = 9, \quad b = c = \frac{5^2 - 1}{3} = 8.$$

De fato, o polinômio característico de P_{25} é $P(x) = 1 + 4x + 2x^3 + 3x^4 + 3x^6 + +2x^7 + +4x^9 + x^{10} + 4x^{15} + x^{16} + 3x^{18} + 2x^{19} + 2x^{21} + 3x^{22} + x^{24} + 4x^{25}$, que pode ser fatorado como $4(4 + x)^9(1 + x + x^2)^8$. ■

Proposição 25. *Os autovetores associados à matriz de transformação da transformada numérica de Pascal de comprimento $N = p$, sobre $GF(p)$, são dados por*

$$v(x) = \lambda \sum_{i=0}^{p-1} a_i x^{p-1-i}(x-1)^i,$$

em que

$$a_n = \lambda(p-1)^n \left[\sum_{k=0}^{n} a_{p-n-1+k} C^k_{p-n-1+k} \right].$$

Demonstração. A transformada numérica de Pascal de um vetor qualquer, pode ser escrita como uma combinação linear das linhas da matriz de transformação. Em particular, os autovetores satisfazem

$$\lambda v(x) = \sum_{i=0}^{p-1} a_i l_i(x).$$

Aplicando-se a transformada numérica de Pascal a ambos os lados da equação anterior, resulta

$$TNP(\lambda v(x)) = \lambda^2 v(x) = \sum_{i=0}^{p-1} a_i TNP(l_i(x)).$$

Mas, sabe-se que

$$TNP(l_i(x)) = x^{p-1} l_{p-1-i}(x^{-1}).$$

Assim, resulta

$$\lambda^2 v(x) = \sum_{i=0}^{p-1} a_i x^{p-1} l_{p-1-i}(x^{-1}).$$

Mas,

$$l_i(x) = [l_{p-2}(x)]^{p-1-i}.$$

Logo, pode-se escrever

$$\begin{aligned}\lambda^2 v(x) &= \sum_{i=0}^{p-1} a_i x^{p-1} \left[l_{p-2}(x^{-1})\right]^{p-1-(p-1-i)} \\ &= \sum_{i=0}^{p-1} a_i x^{p-1} \left[l_{p-2}(x^{-1})\right]^{i}.\end{aligned}$$

Usando o fato de que $\lambda^3 = 1$ e $l_{p-2}(x) = 1 - x$, segue-se que

$$\begin{aligned}v(x) &= \lambda \sum_{i=0}^{p-1} a_i x^{p-1} \left[1 - x^{-1}\right]^{i} \\ &= \lambda \sum_{i=0}^{p-1} a_i x^{p-1-i} \left[x - 1\right]^{i}.\end{aligned}$$

Para se encontrar os coeficientes a_n, procedemos da seguinte forma:

$$a_n = \frac{1}{n!} \frac{d^n v(x)}{dx^n}\bigg|_{x=0}.$$

A aplicação sucessiva do procedimento descrito, produz

$$\begin{aligned}a_0 &= \lambda a_{p-1}, \\ a_1 &= \lambda(p-1)\left[a_{p-2} + (p-1)a_{p-1}\right], \\ a_2 &= \lambda\left[a_{p-3} + a_{p-2}(p-2) + a_{p-1}\right], \\ a_3 &= \lambda(p-1)\left[a_{p-4} + (p-3)a_{p-3} + \frac{(p-2)(p-3)}{2} a_{p-2} + (p-1)a_{p-1}\right],\end{aligned}$$

cuja expressão geral é

$$a_n = \lambda(p-1)^n \left[\sum_{k=0}^{n} a_{p-n-1+k} C_{p-n-1+k}^{k}\right].$$

□

Uma característica dos autovetores da matriz de Pascal, P_p, é que, quando $\lambda = 1$, $x_0 = x_{p-1}$, qualquer que seja p, pois se $x = (x_0, \cdots, x_{p-1})$ é um autovetor com autovalor associado λ, então

$$\begin{bmatrix} 1 & 1 & \cdots & 1 \\ 1 & 2 & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots \\ 1 & 0 & \cdots & 0 \end{bmatrix} \begin{bmatrix} x_0 \\ x_1 \\ \vdots \\ x_{p-2} \\ x_{p-1} \end{bmatrix} = \lambda \begin{bmatrix} x_0 \\ x_1 \\ \vdots \\ x_{p-2} \\ x_{p-1} \end{bmatrix}.$$

Além disso, note que se $x_1 \equiv x_2 \equiv \cdots \equiv x_{p-2} (mod\, p)$, podemos escrever

$$\begin{bmatrix} x_0 \\ x_1 \\ \vdots \\ x_{p-2} \\ x_{p-1} \end{bmatrix} = \begin{bmatrix} 0 & 0 & \cdots & 0 & 1 \\ 0 & 0 & \cdots & (p-1) & 1 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 1 & 1 & \cdots & 1 & 1 \end{bmatrix} \begin{bmatrix} x_0 \\ x_1 \\ \vdots \\ x_{p-2} \\ x_{p-1} \end{bmatrix}$$

e, portanto,

$$x_1 = (p-1)x_{p-2} + x_{p-1} = (p-1)x_1 + x_0.$$

Logo, $x_1 = \dfrac{x_0}{2} \equiv x_0 \left(\dfrac{p+1}{2}\right)$ $(mod\, p)$. Então, podemos escrever o autovetor x como

$$x = x_0 \begin{bmatrix} 1 \\ \left(\dfrac{p+1}{2}\right) \\ \vdots \\ \left(\dfrac{p+1}{2}\right) \\ 1 \end{bmatrix}.$$

Uma vez que a matriz de Pascal P_{p^r} é uma matriz simétrica, pelo Teorema Espectral (SAFF; SNIDER, 2015), decorre que

a) A matriz de Pascal é diagonalizável com p^r autovalores, contando as multiplicidades.
b) Os autovalores são reais.
c) As multiplicidades algébrica e geométrica de autovalores distintos são iguais.
d) Autovetores associados a autovalores distintos são ortogonais, ou seja, os autoespaços são mutuamente ortogonais.

Proposição 26. *Se $v = (v_0, v_1, \cdots, v_{p-1})$ é um autovetor de P_p associado ao autovalor λ, então, v é autovetor de P_p^2 com autovalor λ^2.*

Demonstração. Se v é autovetor de P_p, então, $P_p v = \lambda v$. Assim, podemos escrever

$$P_p(P_p v) = P_p(\lambda v) = \lambda P_p v = \lambda^2 v = P_p^2 v$$

□

Exemplo 18. *Sabe-se que o vetor $v = (2, 2, 1, 2, 3, 0, 1)$ é autovetor de P_7 com autovalor*

associado $\lambda = 2$*. Ou seja,*

$$Pv = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 2 & 3 & 4 & 5 & 6 & 0 \\ 1 & 3 & 6 & 3 & 1 & 0 & 0 \\ 1 & 4 & 3 & 6 & 0 & 0 & 0 \\ 1 & 5 & 1 & 0 & 0 & 0 & 0 \\ 1 & 6 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} 2 \\ 2 \\ 1 \\ 2 \\ 3 \\ 0 \\ 1 \end{bmatrix} = 2 \begin{bmatrix} 2 \\ 2 \\ 1 \\ 2 \\ 3 \\ 0 \\ 1 \end{bmatrix}$$

e

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

A Tabela 11 a seguir mostra a autoestrutura (autovalores e autovetores) da matriz de Pascal P_7*:*

Tabela 11 – Autoestrutura da matriz de Pascal P_7.

λ	Autovetores (Base)
1	[1 0 0 4 1 1 1] [0 1 0 2 5 6 0] [0 0 1 5 1 0 0]
2	[1 0 3 2 1 2 2] [0 1 0 5 3 5 0]
4	[1 0 2 0 5 4 4] [0 1 2 1 0 3 0]

■

5 ALGORITMOS RÁPIDOS PARA COMPUTAÇÃO DA TNP

Sempre que uma transformada é definida, procura-se uma forma eficiente de implementá-la. Um dos aspectos importantes a considerar neste contexto é a complexidade aritmética, medida pelo número de multiplicações e adições para computar a transformada. Chama-se algoritmo rápido a qualquer procedimento computacional que permita uma redução desta complexidade. Neste capítulo, vamos investigar a construção de algoritmos rápidos para reduzir a complexidade multiplicativa para o cálculo da Transformada Numérica de Pascal, definida no Capítulo 4.

5.1 A TNP DE COMPRIMENTO PRIMO

Considerando a Definição 4.1, a complexidade multiplicativa do cálculo da TNP de comprimento N é $M(N) = N^2$. Nesta seção consideramos a TNP, sobre $GF(p)$, cujo comprimento é $N = p$. Pode-se mostrar que para este comprimento, a complexidade multiplicativa, incluindo-se as multiplicações triviais, é

$$M(N) = \frac{p}{2}(p+1). \tag{5.1}$$

De acordo com a Proposição 3.2, neste caso, a matriz de Pascal tem todos os elementos abaixo de sua diagonal secundária iguais a zero.

Exemplo 19. *Considere a TNP da sequência* $v = (v_0, v_1, \cdots, v_6), v_i \in GF(7)$, $V = (V_0, V_1, \cdots, V_6), V_k \in GF(7)$, *em que*

$$V_k = \sum_{i=0}^{6} C_{i+k}^{i} v_i \ (mod\ 7).$$

Em formato matricial, tem-se $V = P_7 v$, *ou seja,*

$$\begin{bmatrix} V_0 \\ V_1 \\ V_2 \\ V_3 \\ V_4 \\ V_5 \\ V_6 \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 2 & 3 & 4 & 5 & 6 & 0 \\ 1 & 3 & 6 & 3 & 1 & 0 & 0 \\ 1 & 4 & 3 & 6 & 0 & 0 & 0 \\ 1 & 5 & 1 & 0 & 0 & 0 & 0 \\ 1 & 6 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} v_0 \\ v_1 \\ v_2 \\ v_3 \\ v_4 \\ v_5 \\ v_6 \end{bmatrix}.$$

Note que: i) Os coeficientes não nulos da segunda linha da matriz P_7 *são congruentes, módulo 7, aos inteiros 1, 2, 3, -3, -2 e -1. Assim,* V_1 *pode ser escrita como* $V_1 = (v_0 - v_5) + 2(v_1 - v_4) +$

$3(v_2 - v_3)$, *reduzindo-se de seis para três o número de multiplicações para sua computação. O mesmo raciocínio pode ser aplicado às outras linhas pares (Figura 2). ii) Os coeficientes não nulos das linhas ímpares são simétricos. Explorando-se esta simetria,* V_2*, por exemplo, pode ser computada como* $V_2 = (v_0 + v_4) + 3(v_1 + v_3) + 6v_2$[1]*, reduzindo-se o número de multiplicações de cinco para três. O mesmo raciocínio pode ser empregado às outras linhas ímpares (Figura 3).* ■

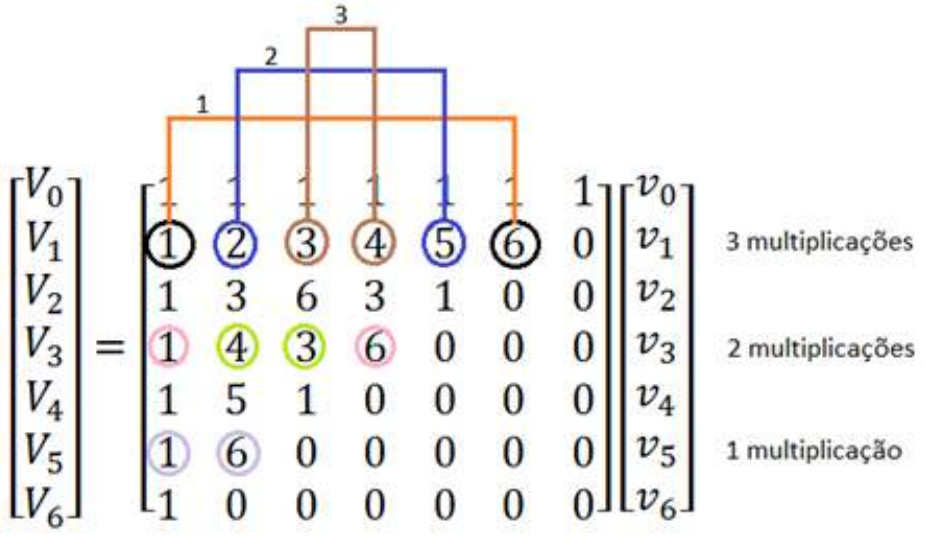


Figura 2 – Agrupamento dos termos nas linhas pares.
Fonte: O autor.

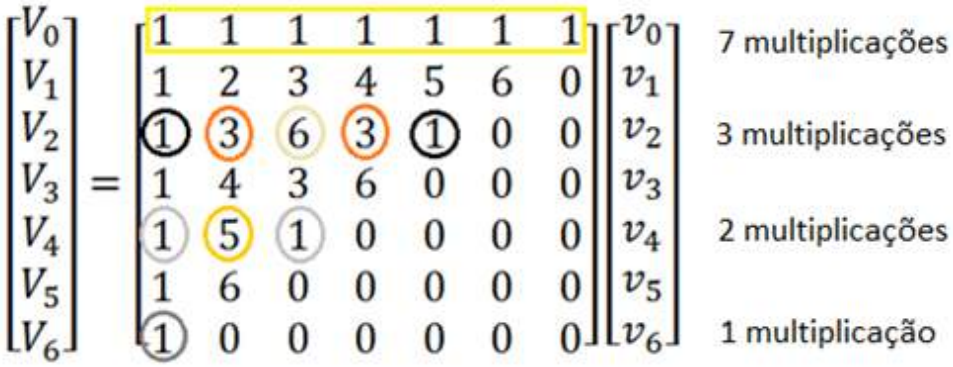


Figura 3 – Agrupamento dos termos nas linhas ímpares.
Fonte: O autor.

Em geral, o número de multiplicações para computar a TNP de comprimento $N = p$ por meio do algoritmo proposto no Exemplo 19, pode ser calculado como:

Linhas pares:

$$M_{par} = 1 + 2 + \cdots + \frac{p-1}{2} = \frac{p^2 - 1}{8}.$$

[1] Veja que para esta linha pode-se fazer melhor ainda: $V_2 = (v_0 + v_4 - v_2) + 3(v_1 + v_3)$.

Linhas ímpares:

$$M_{impar} = 1 + 2 + \cdots + \frac{p-1}{2} + p = \frac{p^2-1}{8} + p.$$

Assim, a complexidade aritmética multiplicativa de acordo com o algoritmo proposto é dada por

$$M(N) = M(p) = 2 \cdot \frac{p^2-1}{8} + p = \frac{p^2+4p-1}{4}. \tag{5.2}$$

Observe que a primeira linha requer apenas multiplicações triviais, uma vez que

$$V_0 = \sum_{i=0}^{p-1} v_i.$$

Desta forma, para computarmos a complexidade multiplicativa excluindo-se as multiplicações triviais devemos subtrair p da expressão anterior (primeira linha) e, como retiramos uma multiplicação trivial de cada linha restante, devemos subtrair $p-1$ do resultado. Assim,

$$M(p) = \frac{p^2+4p-1}{4} - p - (p-1) = \frac{p^2-4p+3}{4} = \frac{(p-1)(p-3)}{4}. \tag{5.3}$$

Observe, no Exemplo 19, que a quantidade de multiplicações foi reduzida de $\frac{p}{2}(p+1) - 7 - 6 = 15$ para $\frac{p^2-4p+3}{4} = 6$. Em verdade, o valor para $M(p)$ encontrado aqui é uma cota superior, uma vez que existe a possibilidade de se ter outras multiplicações triviais na matriz de transformação da TNP. Para se obter uma expressão da complexidade multiplicativa que não inclua multiplicações triviais, é necessário contabilizar quantos termos são congruentes, módulo p, com ± 1. Assim, por exemplo, para o caso $N = 11$, a Equação 5.3 nos fornece 20 multiplicações. Todavia, uma análise desta matriz nos revela que o número de multiplicações não triviais é 17. A Tabela 12 fornece as complexidades multiplicativas do cálculo da TNP de comprimento p, sobre $GF(p)$, obtidas a partir das Equações 5.1, 5.2 e 5.3. A título de referência são indicados também os valores de $Nlog_2N$.

Tabela 12 – Comparativo da complexidade multiplicativa da TNP de comprimento p, sobre $GF(p)$, de acordo com as Equações 5.1, 5.2 e 5.3.

Comprimento	Método			
	Eq. (5.1)	Eq. (5.2)	Eq. (5.3)	$N \log_2 N$
7	28	19	6	19
11	66	41	20	38
13	91	55	30	48
17	153	89	56	69
19	190	109	72	80
23	276	155	110	104
29	435	239	182	140

À medida em que o comprimento aumenta, a relação entre as complexidades multiplicativas Eq.(5.3)/Eq.(5.1) se aproxima de 50%.

5.2 A TNP DE COMPRIMENTO $N = kp$

Na Seção 5.1 vimos o cálculo da TNP sobre $GF(p)$ para um comprimento $N = p$. Agora, vamos considerar o cálculo da TNP sobre $GF(p)$ para comprimentos do tipo $N = kp$, k um número inteiro ≥ 1. Neste caso, observa-se uma estrutura que permite aproveitar o estudo feito anteriormente.

Proposição 27. *A computação da TNP de comprimento $N = 3k$, sobre GF(3), só requer multiplicações triviais.*

Demonstração. Os elementos do corpo finito $GF(3)$ são congruentes a 0, 1 e -1 módulo 3. Assim, todas as multiplicações são triviais. Note que o mesmo ocorre para a TNP definida sobre $GF(2)$, para qualquer comprimento. □

Teorema 6. *A TNP de comprimento $N = kp$, em que p é um número primo ímpar maior do que 3, pode ser computada por meio de um algoritmo de complexidade multiplicativa dada por*

$$M(N) = M(kp) = k\left(\frac{p^2 - 4p + 3}{4}\right) + p\,(k-1)^2\,. \tag{5.4}$$

Demonstração. De acordo com a Proposição 27, quando $p = 2$ ou $p = 3$ só existem multiplicações triviais. A prova deste Teorema pode ser feita considerando-se $P_N = P_k \otimes P_p$, ou seja,

$$V = P_N v = \begin{bmatrix} \widehat{V}_0 \\ \widehat{V}_1 \\ \vdots \\ \widehat{V}_{k-1} \end{bmatrix} = \begin{bmatrix} P_p & P_p & \cdots & P_p \\ P_p & C_2^1 P_p & \cdots & C_k^1 P_p \\ \vdots & \vdots & \ddots & \vdots \\ P_p & C_k^{k-1} P_p & \cdots & C_{2k-2}^{k-1} P_p \end{bmatrix} \begin{bmatrix} \widehat{v}_0 \\ \widehat{v}_1 \\ \vdots \\ \widehat{v}_{k-1} \end{bmatrix}.$$

Note que o vetor coluna v possui k componentes, em que cada uma possui dimensão p. Armazenando-se todos os produtos resultantes da multiplicação da primeira linha da matriz P_N pelo vetor v, evitam-se multiplicações adicionais no cálculo das outras componentes de V. Cada produto requer $M(p)$ multiplicações, conforme Eq.(5.3). Note a existência de uma submatriz $(k-1) \times (k-1)$ em que as únicas multiplicações necessárias são pelos termos binomiais e envolve p multiplicações cada. O resultado segue. □

Na Tabela 13 é apresentada a complexidade multiplicativa $M(N)$, dada pela Eq.(5.4), para o cálculo da TNP, sobre $GF(5)$, de comprimento $N = 5k$. Para efeito de comparação é mostrada a complexidade multiplicativa direta, bem como a redução nesta complexidade proporcionada pelo algoritmo rápido e os valores de $Nlog_2N$.

Tabela 13 – Complexidade multiplicativa da TNP, sobre $GF(5)$, para $N = 5k, k > 1$.

N	10	15	20	30	35
$M(N)$	9	26	53	137	194
N^2	100	225	400	900	1225
$Nlog_2N$	33	59	86	147	179
Redução (%)	91	88,44	86,75	84,77	84,16

Exemplo 20. *Considere a matriz da TNP sobre GF(7) de comprimento N=21,*

$$P_{21} = \begin{bmatrix}
1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\
1 & 2 & 3 & 4 & 5 & 6 & 0 & 1 & 2 & 3 & 4 & 5 & 6 & 0 & 1 & 2 & 3 & 4 & 5 & 6 & 0 \\
1 & 3 & 6 & 3 & 1 & 0 & 0 & 1 & 3 & 6 & 3 & 1 & 0 & 0 & 1 & 3 & 6 & 3 & 1 & 0 & 0 \\
1 & 4 & 3 & 6 & 0 & 0 & 0 & 1 & 4 & 3 & 6 & 0 & 0 & 0 & 1 & 4 & 3 & 6 & 0 & 0 & 0 \\
1 & 5 & 1 & 0 & 0 & 0 & 0 & 1 & 5 & 1 & 0 & 0 & 0 & 0 & 1 & 5 & 1 & 0 & 0 & 0 & 0 \\
1 & 6 & 0 & 0 & 0 & 0 & 0 & 1 & 6 & 0 & 0 & 0 & 0 & 0 & 1 & 6 & 0 & 0 & 0 & 0 & 0 \\
1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\
1 & 1 & 1 & 1 & 1 & 1 & 1 & 2 & 2 & 2 & 2 & 2 & 2 & 2 & 3 & 3 & 3 & 3 & 3 & 3 & 3 \\
1 & 2 & 3 & 4 & 5 & 6 & 0 & 1 & 2 & 3 & 4 & 5 & 6 & 0 & 1 & 2 & 3 & 4 & 5 & 6 & 0 \\
1 & 3 & 6 & 3 & 1 & 0 & 0 & 2 & 6 & 5 & 6 & 2 & 0 & 0 & 3 & 2 & 4 & 2 & 3 & 0 & 0 \\
1 & 4 & 3 & 6 & 0 & 0 & 0 & 2 & 1 & 6 & 5 & 0 & 0 & 0 & 3 & 5 & 2 & 4 & 0 & 0 & 0 \\
1 & 5 & 1 & 0 & 0 & 0 & 0 & 2 & 3 & 2 & 0 & 0 & 0 & 0 & 3 & 1 & 3 & 0 & 0 & 0 & 0 \\
1 & 6 & 0 & 0 & 0 & 0 & 0 & 2 & 5 & 0 & 0 & 0 & 0 & 0 & 3 & 4 & 0 & 0 & 0 & 0 & 0 \\
1 & 0 & 0 & 0 & 0 & 0 & 0 & 2 & 0 & 0 & 0 & 0 & 0 & 0 & 3 & 0 & 0 & 0 & 0 & 0 & 0 \\
1 & 1 & 1 & 1 & 1 & 1 & 1 & 3 & 3 & 3 & 3 & 3 & 3 & 3 & 6 & 6 & 6 & 6 & 6 & 6 & 6 \\
1 & 2 & 3 & 4 & 5 & 6 & 0 & 3 & 6 & 2 & 5 & 1 & 4 & 0 & 6 & 5 & 4 & 3 & 2 & 1 & 0 \\
1 & 3 & 6 & 3 & 1 & 0 & 0 & 3 & 2 & 4 & 2 & 3 & 0 & 0 & 6 & 4 & 1 & 4 & 6 & 0 & 0 \\
1 & 4 & 3 & 6 & 0 & 0 & 0 & 3 & 5 & 2 & 4 & 0 & 0 & 0 & 6 & 3 & 4 & 1 & 0 & 0 & 0 \\
1 & 5 & 1 & 0 & 0 & 0 & 0 & 3 & 1 & 3 & 0 & 0 & 0 & 0 & 6 & 2 & 6 & 0 & 0 & 0 & 0 \\
1 & 6 & 0 & 0 & 0 & 0 & 0 & 3 & 4 & 0 & 0 & 0 & 0 & 0 & 6 & 1 & 0 & 0 & 0 & 0 & 0 \\
1 & 0 & 0 & 0 & 0 & 0 & 0 & 3 & 0 & 0 & 0 & 0 & 0 & 0 & 6 & 0 & 0 & 0 & 0 & 0 & 0
\end{bmatrix}.$$

Dado que

$$P_7 = \begin{bmatrix}
1 & 1 & 1 & 1 & 1 & 1 & 1 \\
1 & 2 & 3 & 4 & 5 & 6 & 0 \\
1 & 3 & 6 & 3 & 1 & 0 & 0 \\
1 & 4 & 3 & 6 & 0 & 0 & 0 \\
1 & 5 & 1 & 0 & 0 & 0 & 0 \\
1 & 6 & 0 & 0 & 0 & 0 & 0 \\
1 & 0 & 0 & 0 & 0 & 0 & 0
\end{bmatrix},$$

então podemos escrever P_{21} na forma

$$P_{21} = \begin{bmatrix} P_7 & P_7 & P_7 \\ P_7 & 2P_7 & 3P_7 \\ P_7 & 3P_7 & 6P_7 \end{bmatrix}.$$

Assim, $k = 3$ e $p = 7$, o que importa em um número de multiplicações igual a

$$M(21) = M(3 \cdot 7) = 3\left(\frac{7^2 - 4 \times 7 + 3}{4}\right) + 7\,(3-1)^2 = 46.$$

Compare este valor com o valor da complexidade multiplicativa pelo método direto, a saber, 147 multiplicações (não incluindo as multiplicações triviais e as multiplicações por zero). Ou seja, uma redução de aproximadamente 31% no número de multiplicações. ■

5.3 A TNP DE COMPRIMENTO $N = p^r$

Teorema 7. *A TNP de comprimento $N = p^r$, em que p é um número primo ímpar maior do que 3, pode ser computada por meio de um algoritmo de complexidade multiplicativa dada por*

$$M(p^r) = p^{r-1}M(p) + (r-1)p^{r-1}\left(\frac{p^2 - 3p + 2}{2}\right), \tag{5.5}$$

em que $M(p)$ é dada por

$$M(p) = \left[\frac{p^2 - 4p + 3}{4}\right].$$

Demonstração. A prova é feita por indução em r.

Passo Base: Fazendo-se $r = 1$, na Eq.(5.5), resulta em $M(p) = M(p)$.

Passo da Indução: Considera-se a Eq.(5.5) verdadeira e faz-se $N = p^{r+1}$. Expressando $P_{p^{r+1}}$ na forma $P_{p^{r+1}} = P_p \otimes P_{p^r}$, tem-se

$$V = \begin{bmatrix} \hat{V}_0 \\ \hat{V}_1 \\ \vdots \\ \hat{V}_{p-1} \end{bmatrix} = \begin{bmatrix} P_{p^r} & P_{p^r} & \cdots & P_{p^r} & P_{p^r} \\ P_{p^r} & 2P_{p^r} & \cdots & (p-1)P_{p^r} & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ P_{p^r} & 0 & \cdots & 0 & 0 \end{bmatrix} \begin{bmatrix} \hat{v}_0 \\ \hat{v}_1 \\ \vdots \\ \hat{v}_{p-1} \end{bmatrix},$$

em que os vetores $\hat{V}_k$ e $\hat{v}_i$ possuem p^r componentes, $i, k = 0, 1 \cdots, p-1$. Observe que

$$\hat{V}_0 = P_{p^r}\hat{v}_0 + P_{p^r}\hat{v}_1 + \cdots + P_{p^r}\hat{v}_{p-1},$$

em que cada uma das p parcelas contribui com $M(p^r)$ multiplicações, resultando em uma quantidade de multiplicações igual a $pM(p^r)$. As linhas restantes só contêm multiplicações por

coeficientes binomiais. Devido à estrutura triangular superior da matriz $P_{p^{r+1}}$, a quantidade de coeficientes binomiais é dada por

$$1 + 2 + \cdots + (p-2) = \left(\frac{p^2 - 3p + 2}{2}\right),$$

com p^r multiplicações para cada componente. Assim, resulta

$$\begin{aligned} M(p^{r+1}) &= pM(p^r) + p^r \left(\frac{p^2 - 3p + 2}{2}\right) \\ &= p^r M(p) + (r-1)p^r \left(\frac{p^2 - 3p + 2}{2}\right) + p^r \left(\frac{p^2 - 3p + 2}{2}\right) \\ &= p^r M(p) + rp^r \left(\frac{p^2 - 3p + 2}{2}\right) \end{aligned}$$

e o resultado segue. □

Na Tabela 14 é apresentada a complexidade multiplicativa $M(N)$, dada pela Eq. (5.5), para o cálculo da TNP de comprimento $N = p^r$, em que p é um primo maior do que 3. Para efeito de comparação é mostrada a complexidade multiplicativa direta, bem como a redução nesta complexidade proporcionada pelo algoritmo rápido, e os valores de $Nlog_2N$.

Tabela 14 – Complexidade multiplicativa da TNP de comprimento $N = 5^r, r > 1$.

N	25	125	625	3.125	15.625
$M(N)$	40	350	2.500	16.250	100.000
$5^r\left(\frac{5^r+1}{2}\right)$	325	7.875	195.625	4.884.375	122.078.125
$Nlog_2N$	116	871	5.805	36.280	217.681
Redução (%)	87,69	95,55	98,72	99,66	99,92

6 APLICAÇÕES DAS TRANSFORMADAS NUMÉRICAS DE PASCAL, HAMMING E GOLAY

6.1 OBTENDO CÓDIGOS CORRETORES DE ERROS A PARTIR DE TRANSFORMADAS DIGITAIS

Sabe-se que a matriz de paridade H de um código de bloco linear satisfaz à relação

$$vH^T = 0, \tag{6.1}$$

em que v é uma palavra do código (LIN; COSTELLO, 2004). Para qualquer transformação linear T, tem-se que seus autovetores satisfazem à relação

$$[T - \lambda I]v = 0, \tag{6.2}$$

em que λ é o autovalor associado ao autovetor v. Identificando-se a matriz de paridade H com a forma escalonada padrão da matriz $[T - \lambda I]$, percebe-se que, a partir de qualquer transformada digital, é possível definir um código de bloco linear (FREIRE, 2009). Especificamente, esta técnica foi usada para construir as famílias dos códigos de Fourier e de Hartley (SOUZA; FREIRE; DEOLIVEIRA, 2009), (SOUZA; BRITTO; DEOLIVEIRA, 2011). Nesta Tese, uma nova família de códigos de bloco lineares, denominada Códigos de Pascal, é construída a partir da Transformada Numérica de Pascal.

6.1.1 Códigos de Pascal

Usando como matriz de transformação a matriz de Pascal, sobre $GF(p)$, P_N, na Equação (6.1), podemos construir os códigos de Pascal, $CP^{(\lambda)}(n, k, d)$. Determinando os autovalores da matriz P_N, a matriz $[P_N - \lambda I]$ é encontrada. Esta matriz, quando reduzida à forma escalonada, resulta na matriz de verificação de paridade, $H_p^{(\lambda)}$, do código. A existência de um código de Pascal sobre $GF(p)$ requer que o polinômio característico da matriz P_N possua, pelo menos, um autovalor neste corpo.

Exemplo 21. *Construção do código de Pascal de comprimento $N = 13$ sobre $GF(3)$. Considere a matriz de Pascal, sobre GF(3),*

$$P_{13} = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 2 & 0 & 1 & 2 & 0 & 1 & 2 & 0 & 1 & 2 & 0 & 1 \\ 1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 \\ 1 & 1 & 1 & 2 & 2 & 2 & 0 & 0 & 0 & 1 & 1 & 1 & 2 \\ 1 & 2 & 0 & 2 & 1 & 0 & 0 & 0 & 0 & 1 & 2 & 0 & 2 \\ 1 & 0 & 0 & 2 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 2 \\ 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 0 \\ 1 & 2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 2 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 2 & 2 & 2 & 2 \\ 1 & 2 & 0 & 1 & 2 & 0 & 1 & 2 & 0 & 2 & 1 & 0 & 2 \\ 1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 2 & 0 & 0 & 2 \\ 1 & 1 & 1 & 2 & 2 & 2 & 0 & 0 & 0 & 2 & 2 & 2 & 1 \end{bmatrix},$$

cujo polinômio característico é $p(x) = 1 + 2x + x^3 + 2x^4 + x^9 + 2x^{10} + x^{12} + 2x^{13} = 2(1+x)^{12}(2+x)$. *Os autovalores são* $\lambda_1 = 2$, *com multiplicidade* $m = 12$, *e* $\lambda_2 = 1$, *com multiplicidade* $m = 1$. *A matriz* $[P_{13} - \lambda_1 I]$, *reduzida à forma escalonada, é*

$$H_{13}^{(2)} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 2 & 2 & 2 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 2 & 0 & 0 & 2 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 2 & 0 & 2 & 2 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 2 & 2 & 2 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 2 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 2 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 2 & 2 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 2 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 1 & 1 \end{bmatrix}.$$

O código de Pascal gerado é o código $CP^{(2)}(13, 4, 4)$ *e sua matriz geradora é*

$$G_{13}^{(2)} = \begin{bmatrix} 2 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 & 2 & 0 & 1 & 0 & 2 & 0 & 1 & 0 & 0 \\ 1 & 0 & 1 & 1 & 0 & 0 & 0 & 2 & 2 & 0 & 0 & 1 & 0 \\ 1 & 1 & 1 & 0 & 2 & 0 & 0 & 0 & 2 & 0 & 0 & 0 & 1 \end{bmatrix}.$$

Para o autovalor $\lambda_2 = 1$, *obtém-se o código* $CP^{(1)}(7, 1, 7)$. *Este resultado ilustra o fato de que é possível se obter códigos de comprimentos diferentes do comprimento da transformada.* ■

Códigos de Pascal foram construídos por meio da técnica apresentada anteriormente. As Tabelas 15 e 16 apresentam os parâmetros obtidos para estes códigos. Especificamente, são mostrados os valores do comprimento (N) da TNP, os autovalores (λ) encontrados e suas multiplicidades (m), a taxa do código (R) e seus parâmetros (N_C, k, d), para os corpos $GF(2)$ e $GF(3)$, respectivamente. A distância mínima foi obtida usando-se o programa gratuito SAGEMATH 8.0 ®. Tabelas similares para os corpos $GF(5)$, $GF(7)$, $GF(11)$ e $GF(13)$ são apresentadas no Apêndice A. Nestas tabelas não são apresentados os parâmetros relacionados aos

códigos de Pascal (i) cujas taxas são menores do que 1/5 e (ii) cujos autovalores encontram-se em corpos de extensão. Neste último caso, os códigos obtidos estão definidos sobre corpos de extensão. A investigação de tais códigos faz parte das propostas indicadas para continuidade desta Tese.

Tabela 15 – Parâmetros associados aos códigos de Pascal $CP^{(\lambda)}(N_C, k, d)$ sobre $GF(2)$.

Comprimento		Autovalores	Multiplicidade	Dimensão	Taxa	Distância Mínima
N	N_C	λ	m	k	R	d
3	2	1	3	1	50%	2
4	4	1	2	2	50%	2
5	2	1	5	1	50%	2
8	8	1	2	2	25%	5
11	4	1	11	1	25%	4
12	8	1	6	2	25%	4
13	10	1	9	3	30%	4
14	9	1	4	4	44,4%	3
15	14	1	7	5	35,7%	4
16	16	1	6	6	37,5%	4
17	14	1	9	5	35,7%	4
18	12	1	4	4	33,3%	4
19	10	1	15	3	30%	4
20	8	1	10	2	25%	4
21	8	1	21	2	25%	4
24	7	1	6	2	28,7%	3
25	11	1	9	3	27,3%	4
30	28	1	8	8	28,6%	8

Tabela 16 – Parâmetros associados aos códigos de Pascal $CP^{(\lambda)}(N_C, k, d)$ sobre GF(3).

Comprimento		Autovalores	Multiplicidade	Dimensão	Taxa	Distância Mínima
N	N_C	λ	m	k	R	d
3	3	1	3	1	33,3%	3
4	3	2	4	1	25%	3
5	4	1	1	1	25%	4
	3	2	4	1	33,3%	3
7	3	1	3	1	33,3%	3
8	7	1	6	2	28,6%	3
9	9	1	9	3	33,3%	3
10	7	1	6	2	28,6%	3
11	5	1	3	1	20%	3
12	12	2	12	3	25%	6
13	9	2	12	4	44,4%	4
14	13	2	12	4	30,8%	6
21	15	1	8	3	20%	9
22	17	1	16	4	23,5%	9
24	21	1	18	6	28,6%	9
26	26	1	24	8	30,8%	9
27	27	1	27	9	33,3%	9
28	25	1	24	8	32%	9
29	23	1	21	7	30,4%	9
42	38	2	36	11	28,9%	12

Pelas Tabelas 15 e 16 percebe-se que os códigos de Pascal construídos possuem taxas menores do que 50%. Outro ponto é que nem sempre a transformada numérica de Pascal gera um código de Pascal sobre o corpo base. Os códigos de Pascal que estamos considerando são códigos definidos sobre o corpo $GF(p)$ e o polinômio característico $p(x)$ associado com as matrizes de transformação da TNP pode ter suas raízes em um corpo de extensão, ou seja, $p(x)$ pode ser irredutível sobre $GF(p)$. Considere, por exemplo, a obtenção de um código de Pascal sobre $GF(2)$ para um comprimento de transformada $N = 22$. O polinômio característico para este caso é $p(x) = (1 + x + x^2)^{11}$, que é irredutível sobre $GF(2)$. De forma análoga, para o comprimento $N = 6$ também não existe código de Pascal sobre $GF(2)$. O polinômio característico para este comprimento de transformada é $p(x) = (1 + x + x^2)^3$.

Exemplo 22. *Considere a TNP de comprimento $N = 2$ sobre $GF(2)$. O polinômio característico associado à matriz de transformação, $p(x) = 1 + x + x^2$, pertence ao expoente 3 (MCELIECE, 1987). Os autovalores estão todos no corpo $GF(2^m)$, em que m é o menor inteiro tal que $3|(2^m - 1)$. Como $m = 2$, as raízes de $p(x)$, α e α^2, estão sobre $GF(4) = \{0, 1, \alpha, \alpha^2\}$.* ■

O produto dos autovalores do polinômio característico sempre resulta em +1 ou -1, independentemente da característica do corpo. Tal fato decorre das relações de Girard[1] (GIRARD,

[1] Albert Girard (França, 1595-Holanda,1632) foi um matemático francês que escreveu o livro "Invention nouvelle en l'algèbre" no qual demonstra as relações entre as raízes e os coeficientes de uma equação, admitindo a

1884). De fato, a versão sobre corpos finitos da relação de Girard para o produto de todas as raízes do polinômio característico, $p(x) = a_0 + a_1x + \cdots + a_nx^n$, é $\frac{a_n}{a_0}(-1)^{N\,mod\,p}$, em que a_0 é igual ao determinante da matriz P_N e é igual a 1, e $a_n = 1$, pois o polinômio característico é mônico (STRANG, 2006). As Tabelas 17 e 18 listam os polinômios característicos da matriz P_N sobre $GF(2)$ e $GF(3)$, respectivamente, sua fatoração e seus autovalores.

Tabela 17 – Polinômio Característico $p(x)$, sua fatoração, autovalores (λ) sobre $GF(2)$ e suas multiplicidades (m), comprimento do código gerado (N_C) sobre $GF(2)$ e comprimento (N) da TNP.

N	N_C	$p(x)$	λ	m
3	2	$(1+x)^3$	1	3
4	4	$(1+x)^2(1+x+x^2)$	1	2
5	2	$(1+x)^5$	1	5
7	6	$(1+x+x^2)^2(1+x)^3$	1	3
8	8	$(1+x+x^2)^3(1+x)^2$	1	2
9	6	$(1+x+x^2)^2(1+x)^5$	1	5
11	4	$(1+x)^{11}$	1	11
12	8	$(1+x)^6(1+x+x^2)^3$	1	6
13	10	$(1+x)^9(1+x+x^2)^2$	1	9
14	9	$(1+x)^4(1+x+x^2)^5$	1	4
15	14	$(1+x)^7(1+x+x^2)^4$	1	7
16	16	$(1+x)^6(1+x+x^2)^5$	1	6
17	14	$(1+x)^9(1+x+x^2)^4$	1	9
18	12	$(1+x)^4(1+x+x^2)^7$	1	4
19	10	$(1+x)^{15}(1+x+x^2)^2$	1	15
20	8	$(1+x)^{10}(1+x+x^2)^5$	1	10
21	8	$(1+x)^{21}$	1	21
23	12	$(1+x)^{11}(1+x+x^2)^6$	1	11
24	7	$(1+x)^6(1+x+x^2)^9$	1	6
25	11	$(1+x)^9(1+x+x^2)^8$	1	9
30	28	$(1+x)^8(1+x+x^2)^{11}$	1	8

existência das raízes negativas. Girard introduziu as abreviaturas sin, cos e tan para as respectivas funções trigonométricas.

Tabela 18 – Polinômio Característico $p(x)$, sua fatoração, autovalores (λ) sobre $GF(3)$ e suas multiplicidades (m), comprimento do código gerado (N_C) sobre $GF(3)$ e comprimento (N) da TNP.

N	N_C	$p(x)$	λ	m
3	3	$2(2+x)^3$	1	3
4	3	$(1+x)^4$	2	4
5	4	$2(1+x)^4(2+x)$	1	1
	3		2	4
7	3	$2(2+x)^3(2+x+x^2)(2+2x+x^2)$	1	3
8	7	$(2+x)^6(1+x^2)$	1	6
9	9	$2(2+x)^9$	1	8
10	7	$(1+x)^4(2+x)^6$	1	6
	9		2	4
11	5	$2(1+x)^6(2+x)^3(1+x^2)$	1	3
	11		2	6
12	12	$(1+x)^{12}$	2	12
13	7	$2(1+x)^{12}(2+x)$	1	1
	9		2	12
14	13	$(1+x)^{12}(1+x^2)$	2	12
15	12	$2(1+x)^{12}(2+x)^3$	1	3
	12		2	12
16	11	$(1+x)^6(1+x^2)^3(2+x+x^2)(2+2x+x^2)$	2	6
17	16	$2(1+x)^4(2+x)(1+x^2)^6$	1	1
	9		2	4
19	9	$2(2+x)^3(1+x^2)^6(2+x+x^2)(2+2x+x^2)$	1	3
20	13	$(2+x)^6(1+x^2)^3(2+x^2+x^4)(2+2x^2+x^4)$	1	6
21	15	$2(2+x)^9(2+x+x^2)^3(2+2x+x^2)^3$	1	9
22	17	$(1+x)^4(2+x)^{16}(1+x^2)$	1	16
	15		2	4
23	19	$2(1+x)^4(2+x)^{19}$	1	19
	15		2	4
24	24	$(2+x)^{18}(1+x^2)^3$	1	18
25	25	$2(2+x)^{21}(2+x+x^2)(2+2x+x^2)$	1	21
26	26	$(2+x)^{24}(1+x^2)$	1	24
27	27	$2(2+x)^{27}$	1	27
28	28	$(1+x)^4(2+x)^{24}$	1	24
	28		2	4
29	25	$2(1+x)^6(2+x)^{21}(1+x^2)$	1	21
	29		2	6
42	38	$(1+x)^{36}(1+x^2)^3$	2	36
43	12	$2(1+x)^{36}(2+x)^3(2+x+x^2)(2+2x+x^2)$	1	3
	27		2	36

6.1.1.1 Decodificação dos Códigos de Pascal

Os códigos de Pascal sendo códigos lineares de bloco podem ser decodificados usando-se as técnicas usuais de decodificação de tais códigos. Todavia, uma forma alternativa de decodificação pode ser pensada explorando-se as características específicas da matriz de transformação de Pascal.

Considere que a palavra recebida seja escrita como

$$r = v + e,$$

em que v é a palavra-código transmitida e e é o vetor erro possivelmente introduzido pelo canal. Como as palavras-código do código de Pascal são autovetores associados à matriz de transformação de Pascal, resulta que a aplicação da TNP em ambos os lados da equação anterior produz

$$\begin{aligned} TNP(r) &= TNP(v) + TNP(e) \\ &= \lambda v + TNP(e). \end{aligned}$$

Sem perda de generalidade, vamos considerar que o autovalor seja $\lambda = 1$. Assim, caso a palavra recebida não contenha erros, $TNP(r) = v = r$. Caso contrário, a palavra recebida contém erros.

Considere inicialmente a ocorrência de um único erro na posição $i,\ 0 \leq i \leq (p-1)$. Então $TNP(e) = l_i$, em que l_i é a i-ésima linha da matriz de transformação da TNP. Desta forma, podemos escrever

$$TNP(r) = v + l_i.$$

Denotando $TNP(r)$ por $R = [R_0\ R_1\ \cdots\ R_{p-1}]$ e $v = [v_0\ v_1\ \cdots\ v_{p-1}]$, resulta

$$TNP(r) = \begin{bmatrix} R_0 \\ R_1 \\ \vdots \\ R_{p-1} \end{bmatrix} = \begin{bmatrix} \textit{Matriz Geradora} \\ \textit{de Autovetores} \\ \textit{(MGA)} \end{bmatrix} + \begin{bmatrix} l_i^{(0)} = 1 \\ l_i^{(1)} \\ \vdots \\ l_i^{(p-1)} \end{bmatrix},$$

em que $l_i^{(j)}$ correponde à j-ésima componente da i-ésima linha da matriz de Pascal.

Exemplo 23. *Considere o código de Pascal de comprimento $N = 7$, sobre GF(7). Suponha que a palavra código transmitida foi $v = [1\,0\,0\,4\,1\,1\,1]$, mas foi recebida a palavra $r = [1\,0\,0\,5\,1\,1\,1]$. Como $TNP(r)$=[2 4 3 3 1 1 1] e $\lambda = 1$, então*

$$\begin{bmatrix} 2 \\ 4 \\ 3 \\ 3 \\ 1 \\ 1 \\ 1 \end{bmatrix} = \begin{bmatrix} v_6 \\ 6v_5 + v_6 \\ v_4 + 5v_5 + v_6 \\ 5v_4 + 2v_5 + 4v_6 \\ v_4 \\ v_5 \\ v_6 \end{bmatrix} + \begin{bmatrix} 1 \\ l_i^{(1)} \\ l_i^{(2)} \\ l_i^{(3)} \\ l_i^{(4)} \\ l_i^{(5)} \\ l_i^{(6)} \end{bmatrix},$$

em que a matriz geradora de autovetores foi obtida como solução do sistema

$$\begin{bmatrix} v_0 \\ v_1 \\ v_2 \\ v_3 \\ v_4 \\ v_5 \\ v_6 \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 6 & 1 \\ 0 & 0 & 0 & 0 & 1 & 5 & 1 \\ 0 & 0 & 0 & 6 & 3 & 4 & 1 \\ 0 & 0 & 1 & 3 & 6 & 3 & 1 \\ 0 & 6 & 5 & 4 & 3 & 2 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 \end{bmatrix} \begin{bmatrix} v_0 \\ v_1 \\ v_2 \\ v_3 \\ v_4 \\ v_5 \\ v_6 \end{bmatrix}.$$

Então, $v_6 = 1$ *e, portanto,* $l_i^{(6)} = 0$ *(Isto exclui a linha* l_0 *como possível candidata). Mas, note que podemos escrever*

$$1 = v_5 + l_i^{(5)},$$

em que $l_i^{(5)}$ *é igual a 0 ou 6. Supondo* $l_i^{(5)} = 6$*, resulta em* $v_5 = 2$*, o que não é possível pois*

$$\begin{aligned} l_i^{(1)} + 6v_5 + v_6 &= 4, \\ l_i^{(1)} + 6.2 + 1 &= 4, \\ l_i^{(1)} &= 5 \end{aligned}$$

e a linha da matriz de Pascal seria $(1\ 5\ 1\ 0\ 0\ 0\ 0)$*, cuja componente* $l_i^{(5)} \neq 6$*. Logo,* $l_i^{(5)} = 0$ *e, portanto,* $v_5 = 1$*. Assim, usando este valor de* v_5 *na equação* $4 = 6v_5 + v_6 + l_i^{(1)}$*, resulta que* $l_i^{(1)} = 4$ *e, portanto a linha da matriz de Pascal é a linha* $(1\ 4\ 3\ 6\ 0\ 0\ 0)$ *e o erro inserido pelo canal foi* $(0\ 0\ 0\ 1\ 0\ 0\ 0)$*. Portanto,* $v_3 = r_3 - 1 = 5 - 1 = 4$ *e a palavra-código transmitida estimada é* $\hat{v} = [1\ 0\ 0\ 4\ 1\ 1\ 1]$*.* ■

Exemplo 24. *No exemplo anterior, consideramos um único erro de amplitude unitária. Agora, iremos considerar um erro único de amplitude* k*, ou seja,* $e = (0\ 0\ \cdots k\ \cdots 0)$*. Considere o mesmo código de Pascal do exemplo anterior supondo a mesma palavra transmitida mas, com palavra recebida* $r = [1\ 0\ 0\ 0\ 1\ 1\ 1]$*. Como* $TNP(r) = [4\ 5\ 2\ 1\ 1\ 1\ 1]$ *e* $\lambda = 1$*, então*

$$\begin{bmatrix} 4 \\ 5 \\ 2 \\ 1 \\ 1 \\ 1 \\ 1 \end{bmatrix} = \begin{bmatrix} v_6 \\ 6v_5 + v_6 \\ v_4 + 5v_5 + v_6 \\ 5v_4 + 2v_5 + 4v_6 \\ v_4 \\ v_5 \\ v_6 \end{bmatrix} + k \begin{bmatrix} 1 \\ l_i^{(1)} \\ l_i^{(2)} \\ l_i^{(3)} \\ l_i^{(4)} \\ l_i^{(5)} \\ l_i^{(6)} \end{bmatrix}.$$

As seguintes equações são obtidas do sistema acima:

$$\begin{aligned} 4 &= v_6 + k, \\ 1 &= v_6 + k l_i^{(6)}. \end{aligned}$$

Mas, $l_i^{(6)} \in \{0, 1\}$. *Note que se* $l_i^{(6)} = 1$, *resulta*

$$v_6 + k = 4,$$
$$v_6 + k = 1,$$

o que não é possível. Então, $l_i^{(6)} = 0$ *e, portanto,* $k = 3$.

Uma vez determinado k, *a decodificação segue os mesmos passos de antes.* ■

Algorithm 1 Decodificação dos Códigos de Pascal para um único erro

procedure DECODEPASCAL(R)
  if ($R = r$) **then**
    Não houve erro ou erro não detectável.
  else if ($R^{(0)} = R^{(p-1)}$) **then**
    Erro na posição zero
    k é solução do sistema $R = MGA + kl_0$
  else
    $k \leftarrow R^{(0)} - R^{(p-1)}$
    $l_i^{(p-1)} \leftarrow 0$
    $v_{p-1} \leftarrow R^{(p-1)}$
    $j \leftarrow 2$
    repeat
      l_i^{p-j} é solução do sistema $R = MGA + kl_i$
      $j \leftarrow j + 1$
    until ($l_i^{p-j-1} \neq 0$)
    A linha está determinada
  end if
end procedure

Vamos considerar agora erros duplos por meio de um exemplo.

Exemplo 25. *Considere o mesmo código de Pascal dos exemplos anteriores, com a mesma palavra transmitida, admitindo-se erro duplo com amplitude unitária. Suponha que a palavra recebida seja* $r = [2\ 0\ 0\ 4\ 1\ 1\ 2]$. *Como* $TNP(r) = [3\ 1\ 1\ 5\ 2\ 2\ 2]$ *e* λ*=1, então*

$$\begin{bmatrix} 3 \\ 1 \\ 1 \\ 5 \\ 2 \\ 2 \\ 2 \end{bmatrix} = \begin{bmatrix} v_6 \\ 6v_5 + v_6 \\ v_4 + 5v_5 + v_6 \\ 5v_4 + 2v_5 + 4v_6 \\ v_4 \\ v_5 \\ v_6 \end{bmatrix} + \begin{bmatrix} 1 \\ l_i^{(1)} \\ l_i^{(2)} \\ l_i^{(3)} \\ l_i^{(4)} \\ l_i^{(5)} \\ l_i^{(6)} \end{bmatrix} + \begin{bmatrix} 1 \\ l_j^{(1)} \\ l_j^{(2)} \\ l_j^{(3)} \\ l_j^{(4)} \\ l_j^{(5)} \\ l_j^{(6)} \end{bmatrix}$$

e, de acordo com o sistema anterior, pode-se escrever

$$3 = v_6 + 2 \Rightarrow v_6 = 1.$$

Sabe-se que as seguintes condições podem ocorrer em relação às linhas i e j no sistema de equações anterior:

$l_i^{(6)} l_j^{(6)}$	Equação	Solução
00	$2 = v_6 + 0$	$v_6 = 5$
01/10	$2 = v_6 + 1$	$v_6 = 1$
11	$2 = v_6 + 2$	$v_6 = 0$

Portanto, se $l_i^{(6)} = 0$, então, $l_j^{(6)} = 1$ e a linha j é a linha l_0 da matriz de Pascal e assim podemos escrever

$$\begin{bmatrix} 3 \\ 1 \\ 1 \\ 5 \\ 2 \\ 2 \\ 2 \end{bmatrix} = \begin{bmatrix} v_6 \\ 6v_5 + v_6 \\ v_4 + 5v_5 + v_6 \\ 5v_4 + 2v_5 + 4v_6 \\ v_4 \\ v_5 \\ v_6 \end{bmatrix} + \begin{bmatrix} 1 \\ l_i^{(1)} \\ l_i^{(2)} \\ l_i^{(3)} \\ l_i^{(4)} \\ l_i^{(5)} \\ 0 \end{bmatrix} + \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \end{bmatrix}.$$

Do sistema anterior tem-se as seguintes equações:

$$\begin{aligned} v_5 + l_i^{(5)} + 1 &= 2, \\ v_5 + l_i^{(5)} &= 1, \\ v_5 &= 6l_i^{(5)} + 1. \end{aligned}$$

Por outro lado,

$$\begin{aligned} 6v_5 + 1 + l_i^{(1)} + 1 &= 1, \\ v_5 &= l_i^{(1)} + 1. \end{aligned}$$

Então, $l_i^{(1)} = 6l_i^{(5)}$, cuja única solução é $l_i^{(1)} = l_i^{(5)} = 0$. Portanto, $l_i = (1\,0\,0\,0\,0\,0\,0)$. Assim, os dois erros foram nas posições 0 e 6 e a palavra-código transmitida estimada é $\hat{v} = (1\,0\,0\,4\,1\,1\,1)$.

■

6.2 TRANSFORMADAS PERFEITAS

As transformadas aqui apresentadas são obtidas a partir dos códigos de Hamming e de Golay, que são códigos perfeitos, sendo, por esta razão, denominadas de transformadas perfeitas. Aplicações destas transformadas em Processamento de Imagem são propostas na Seção 6.3.

6.2.1 A Transformada Numérica de Hamming

Nesta seção é definida uma nova transformada numérica a partir do código de Hamming (HAMMING, 1950). Conforme comentado na seção anterior, qualquer matriz de transformação T tem seus autovetores v satisfazendo à relação

Algorithm 2 Decodificação dos Códigos de Pascal para erro duplo

procedure DECODEPASCAL2(R)
  if ($R = r$) **then**
    Não houve erro ou erro não detectável.
  else if ($R^{(0)} - 2 = v_{(p-1)}$) **then**
    Houve erro duplo.
    if ($R^{(p-1)} - R^{(0)} + 2 = 1$) **then**
      A componente 0 possui erro
      $l_i^{(p-1)} = 0$, $l_j^{(p-1)} = 1$
    else
      $l_i^{(p-1)} = l_j^{(p-1)} = 0$
      $R^{(p-1)} = v_{(p-1)} = R_0 - 2$
    end if
    Some os dois vetores e use Algorithm 1.
  else
    Um único erro. Use Algorithm 1.
  end if
end procedure

$$[T - \lambda I]\, v = 0,$$

em que λ denota o autovalor associado a v. Na seção 6.1.1 construiu-se a matriz de verificação de paridade, H, de um código de bloco linear, escalonando-se a matriz quadrada singular $[T - \lambda I]$. Neste capítulo, seguimos no "sentido oposto", ou seja, partindo da matriz de verificação de paridade de um código de bloco linear $C(n, k, d)$, acrescentamos, à mesma, k linhas de modo a obter uma matriz quadrada singular de ordem n, H_e, correspondente à matriz $[T - \lambda I]$. Este procedimento leva à construção de uma nova transformada cuja matriz de transformação é $T = H_e + \lambda I$. Tal transformada recebe o nome do código de bloco linear usado na sua construção. Assim, por exemplo, se $C(n, k, d)$ representa o código de Hamming sobre $GF(p)$, então T representa a matriz de transformação da Transformada Numérica de Hamming (TNH) sobre $GF(p)$.

6.2.1.1 A Transformada Numérica de Hamming sobre $GF(2)$.

Iniciamos considerando o código de Hamming binário $C(7, 4, 3)$, com matriz de verificação de paridade

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

Neste caso, para construir a matriz de transformação da Transformada Numérica de Hamming binária, $T_H^{(\lambda)}$, precisamos adicionar 4 linhas à matriz H, obtendo a matriz H_e. Tais linhas são obtidas por meio da combinação linear das linhas da matriz H. À matriz H_e obtida por este procedimento, deveremos somar λI_N em que, neste caso específico, $\lambda = 1$ é um

autovalor e $I_N = I_7$ é a matriz identidade de ordem 7. Note que, em verdade, os autovalores são desconhecidos. Neste sentido, deve-se testar os possíveis candidatos a autovalor e eliminar aqueles que resultarem em uma matriz de transformação singular. Assim, é possível obter matrizes de transformação distintas. Como estamos considerando o código de Hamming binário, resulta

$$T_H^{(1)} = \begin{bmatrix} 0 & 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 0 & 1 & 1 & 1 & 0 \\ 1 & 0 & 1 & 1 & 1 & 1 & 0 \\ 0 & 1 & 1 & 1 & 1 & 1 & 0 \\ 1 & 1 & 0 & 1 & 0 & 0 & 0 \end{bmatrix}.$$

Como existem várias formas de se escrever a matriz de verificação de paridade do código de Hamming $C(7, 4, 3)$, então, a matriz da transformada obtida não é única. Por exemplo, se tivéssemos partido da matriz de verificação de paridade escrita em seu formato sistemático

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

chegaríamos à matriz

$$\hat{T}_H^{(1)} = \begin{bmatrix} 0 & 1 & 0 & 1 & 1 & 0 & 0 \\ 1 & 1 & 1 & 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 & 0 & 0 & 1 \\ 0 & 1 & 1 & 1 & 1 & 1 & 0 \\ 1 & 0 & 1 & 0 & 0 & 0 & 1 \\ 1 & 1 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 1 & 1 & 0 \end{bmatrix}.$$

Note que estamos caminhando para uma definição não-algébrica da transformada numérica de Hamming sobre $GF(p)$, uma vez que as matrizes de verificação de paridade adotadas até aqui não estão sendo representadas algebricamente. Certamente, esta abordagem dificulta a obtenção de expressões que representem o comportamento da transformada diante de determinadas operações, tal como o deslocamento de bits. Mesmo assim, vamos propor uma primeira definição.

Definição 12. *A transformada numérica de Hamming (TNH), cujo comprimento é* $N = (p^m - 1)/(p - 1)$*, da sequência* $v = (v_0, v_1, \cdots, v_{N-1})$*,* $v_i \in GF(p)$*, é a sequência* $V = (V_0, V_1, \cdots, V_{N-1}), V_k \in GF(p)$*, dada por*

$$V = T_H^{(\lambda)} \cdot v,$$

em que $T_H^{(\lambda)}$ *é a matriz de transformação de Hamming sobre* $GF(p)$*, parametrizada pelo autovalor* λ*.* ■

Aparentemente, a única propriedade natural é que tal transformada é linear, uma vez que a propriedade distributiva da multiplicação em relação à adição é satisfeita por matrizes.

6.2.1.2 A Transformada Numérica de Hamming Cíclica (TNHC).

Uma forma de representar algebricamente a transformada numérica de Hamming sobre $GF(p)$ é considerar a matriz de verificação de paridade H no formato

$$H = \begin{bmatrix} h(x) \\ xh(x) \\ x^2h(x) \\ \vdots \\ x^{n-k-1}h(x) \end{bmatrix},$$

em que $h(x)$ é o polinômio de paridade do código de Hamming cíclico sobre $GF(p)$ (MOON, 2005). As k linhas necessárias para compor a matriz H_e são obtidas deslocando-se ciclicamente o polinômio $h(x)$.

Exemplo 26. *Considere o código de Hamming cíclico binário $C(7,4,3)$ com polinômio de verificação de paridade dado por $h(x) = x^4 + x^2 + x + 1$. A matriz de verificação de paridade H é dada por*

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

A matriz da transformada, neste caso, é

$$T_H^{(\lambda)} = \begin{bmatrix} 1 & 0 & 1 & 1 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 & 1 & 1 & 0 \\ 0 & 0 & 1 & 0 & 1 & 1 & 1 \\ 1 & 0 & 0 & 1 & 0 & 1 & 1 \\ 1 & 1 & 0 & 0 & 1 & 0 & 1 \\ 1 & 1 & 1 & 0 & 0 & 1 & 0 \\ 0 & 1 & 1 & 1 & 0 & 0 & 1 \end{bmatrix} + \lambda \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}.$$

Como estamos sobre $GF(2)$, então $\lambda = 1$. Resulta

$$T_H^{(\lambda)} = \begin{bmatrix} 0 & 0 & 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 & 1 \\ 1 & 0 & 0 & 0 & 0 & 1 & 1 \\ 1 & 1 & 0 & 0 & 0 & 0 & 1 \\ 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 & 0 & 0 & 0 \end{bmatrix}.$$

■

Note que podemos considerar as linhas da matriz λI_N como formadas por deslocamentos cíclicos do vetor $[\lambda 00 \cdots 0]$; em forma polinomial a matriz λI_N pode ser escrita como

$$\begin{bmatrix} \lambda(x) \\ x\lambda(x) \\ x^2\lambda(x) \\ \vdots \\ x^{N-1}\lambda \end{bmatrix},$$

resultando em

$$T_H^{(\lambda)} = \begin{bmatrix} h(x) \\ xh(x) \\ x^2h(x) \\ \vdots \\ x^{N-1}h(x) \end{bmatrix} + \begin{bmatrix} \lambda(x) \\ x\lambda(x) \\ x^2\lambda(x) \\ \vdots \\ x^{N-1}\lambda(x) \end{bmatrix} = \begin{bmatrix} [h(x)+\lambda(x)] \\ x[h(x)+\lambda(x)] \\ x^2[h(x)+\lambda(x)] \\ \vdots \\ x^{N-1}[h(x)+\lambda(x)] \end{bmatrix}.$$

Definição 13. *A Transformada numérica de Hamming cíclica (TNHC), de comprimento* $N = (p^m-1)/(p-1)$*, da sequência* $v = (v_0, v_1, \cdots, v_{N-1})$*,* $v_i \in GF(p)$*, é a sequência* $V = (V_0, V_1, \cdots, V_{N-1})$*,* $V_k \in GF(p)$*, em que*

$$V = T_H^{(\lambda)} \cdot v$$

e

$$T_H^{(\lambda)} = \begin{bmatrix} [h(x)+\lambda(x)] \\ x[h(x)+\lambda(x)] \\ x^2[h(x)+\lambda(x)] \\ \vdots \\ x^{N-1}[h(x)+\lambda(x)] \end{bmatrix}.$$

■

Observe que esta definição abre a possibilidade de um tratamento algébrico das possíveis propriedades da TNHC.

Proposição 28. *A matriz* $T_H^{(\lambda)}$ *é circulante.*

O procedimento para gerar a matriz H_e, que considera os deslocamentos cíclicos do polinômio de paridade do código, $h(x)$, resulta numa matriz circulante. Este aspecto não é modificado ao se acrescentar H_e à matriz λI_N.

Propriedades da TNHC

i) **Linearidade**

ii) **Deslocamento no domínio do tempo**: Considere a sequência $\widehat{v} = (\widehat{v}_0, \cdots, \widehat{v}_{N-1})$ em que $\widehat{v}_i = v_{i-m}$. Então, $\widehat{v} \leftrightarrow \widehat{V}$, em que

$$\widehat{V} = x^m V.$$

Demonstração. Note que estamos considerando, sem perda de generalidade, um deslocamento cíclico de m posições para a direita. Assim, podemos escrever $\widehat{v} = x^m v$ *(mod* $x^N - 1$*)*. Portanto,

$$\widehat{V} = \begin{bmatrix} [h(x)+\lambda(x)] \\ x[h(x)+\lambda(x)] \\ x^2[h(x)+\lambda(x)] \\ \vdots \\ x^{N-1}[h(x)+\lambda(x)] \end{bmatrix} \begin{bmatrix} x^m v_0 \\ x^m v_1 \\ x^m v_2 \\ \vdots \\ x^m v_{N-1} \end{bmatrix} = x^m \begin{bmatrix} [h(x)+\lambda(x)] \\ x[h(x)+\lambda(x)] \\ x^2[h(x)+\lambda(x)] \\ \vdots \\ x^{N-1}[h(x)+\lambda(x)] \end{bmatrix} \begin{bmatrix} v_0 \\ v_1 \\ \vdots \\ v_{N-1} \end{bmatrix}$$
$$= x^m V.$$

□

iii) **Deslocamento no domínio da frequência:** Considere a sequência $\widehat{V} = (\widehat{V}_0, \cdots, \widehat{V}_{N-1})$ em que $\widehat{V}_k = V_{k-l}$. Então, $\widehat{v} = x^l v$.

Demonstração. Vamos calcular $\widehat{V}$ para $\widehat{v} = x^l v$.

$$\widehat{V} = \begin{bmatrix} [h(x)+\lambda(x)] \\ x[h(x)+\lambda(x)] \\ x^2[h(x)+\lambda(x)] \\ \vdots \\ x^{N-1}[h(x)+\lambda(x)] \end{bmatrix} \cdot \begin{bmatrix} x^l v_0 \\ x^l v_1 \\ x^l v_2 \\ \vdots \\ x^l v_{N-1} \end{bmatrix} = x^l \begin{bmatrix} [h(x)+\lambda(x)] \\ x[h(x)+\lambda(x)] \\ x^2[h(x)+\lambda(x)] \\ \vdots \\ x^{N-1}[h(x)+\lambda(x)] \end{bmatrix} \begin{bmatrix} v_0 \\ v_1 \\ \vdots \\ v_{N-1} \end{bmatrix}$$
$$= x^l V.$$

e o resultado segue. □

iv) **Transformada da sequência constante:** A transformada da sequência $v = (r, r, \cdots, r)$ é a sequência de componentes $V_k = r \cdot peso(h(x))$ *(mod p)*,$\forall k$.

v) **Transformada da sequência impulso:** A transformada da sequência $\delta = (1, 0, \cdots, 0)$, corresponde à primeira coluna da matriz $T_H^{(\lambda)}$, isto é, aos coeficientes do polinômio $x[h(x) + \lambda(x)]$.

Exemplo 27. *A matriz de transformação da TNHC de comprimento 15 sobre* $GF(2)$ *é*

$$T_H^{(1)} = \begin{bmatrix} 0 & 1 & 1 & 1 & 0 & 1 & 0 & 1 & 1 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 & 0 & 1 & 0 & 1 & 1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 0 & 1 & 0 & 1 & 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 1 & 0 & 1 & 1 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 1 & 0 & 1 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 1 & 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 1 & 0 & 1 & 1 \\ 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 1 & 0 & 1 \\ 1 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 1 & 0 \\ 0 & 1 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 1 \\ 1 & 0 & 1 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 0 \\ 0 & 1 & 0 & 1 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 1 & 1 \\ 1 & 0 & 1 & 0 & 1 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 1 \\ 1 & 1 & 0 & 1 & 0 & 1 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 \\ 1 & 1 & 1 & 0 & 1 & 0 & 1 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \end{bmatrix},$$

cuja inversa é

$$[T_H^{(1)}]^{-1} = \begin{bmatrix} 0 & 0 & 1 & 0 & 1 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 1 & 1 \\ 1 & 0 & 0 & 1 & 0 & 1 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 1 \\ 1 & 1 & 0 & 0 & 1 & 0 & 1 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 1 \\ 1 & 1 & 1 & 0 & 0 & 1 & 0 & 1 & 1 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 & 0 & 0 & 1 & 0 & 1 & 1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 & 0 & 0 & 1 & 0 & 1 & 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 1 & 0 & 1 & 1 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 1 & 0 & 1 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 1 & 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 1 & 0 & 1 & 1 \\ 1 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 1 & 0 & 1 \\ 1 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 1 & 0 \\ 0 & 1 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 1 \\ 1 & 0 & 1 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 \\ 1 & 1 & 0 & 1 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 1 & 1 & 0 \end{bmatrix}.$$

Usando a fórmula de inversão

$$v = \left[T_H^{(1)}\right]^{-1} V$$

para o vetor $V = (1, 0, 1, 1, 0, 0, 1, 1, 1, 0, 0, 0, 1, 0, 0)$*, no domínio transformado, obtém-se*

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

■

6.2.2 A Transformada de Golay

Existem dois códigos de Golay, o código binário $\mathscr{G}(23, 12, 7)$ $(h(x) = x^{12} + x^{11} + x^{10} + x^9 + x^8 + x^5 + x^2 + 1)$ e o código ternário $\mathscr{G}(11, 6, 5)$ $(h(x) = x^6 + 2x^5 + 2x^4 + 2x^3 + x^2 + 1)$. Nesta Tese iremos usar o código $\mathscr{G}(11, 6, 5)$ cuja matriz de verificação de paridade é

$$H_G = \begin{bmatrix} 1 & 1 & 1 & 2 & 2 & 0 & 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 2 & 1 & 0 & 2 & 0 & 1 & 0 & 0 & 0 \\ 1 & 2 & 1 & 0 & 1 & 2 & 0 & 0 & 1 & 0 & 0 \\ 1 & 2 & 0 & 1 & 2 & 1 & 0 & 0 & 0 & 1 & 0 \\ 1 & 0 & 2 & 2 & 1 & 1 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}.$$

Assim como foi feito com o código de Hamming, a matriz de paridade acima precisa ser expandida de modo a se ter uma matriz quadrada 11×11. Vamos usar as seguintes combinações lineares para completar a matriz: $l_1 + l_2,\ l_1 + l_3,\ l_1 + l_4,\ l_1 + l_5,\ l_2 + l_3$ e $l_2 + l_4$. Assim, resulta

$$H_{G_e} = \begin{bmatrix} 1 & 1 & 1 & 2 & 2 & 0 & 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 2 & 1 & 0 & 2 & 0 & 1 & 0 & 0 & 0 \\ 1 & 2 & 1 & 0 & 1 & 2 & 0 & 0 & 1 & 0 & 0 \\ 1 & 2 & 0 & 1 & 2 & 1 & 0 & 0 & 0 & 1 & 0 \\ 1 & 0 & 2 & 2 & 1 & 1 & 0 & 0 & 0 & 0 & 1 \\ 2 & 2 & 0 & 0 & 2 & 2 & 1 & 1 & 0 & 0 & 0 \\ 2 & 0 & 2 & 2 & 0 & 2 & 1 & 0 & 1 & 0 & 0 \\ 2 & 0 & 1 & 0 & 1 & 1 & 1 & 0 & 0 & 1 & 0 \\ 2 & 1 & 0 & 1 & 0 & 1 & 1 & 0 & 0 & 0 & 1 \\ 2 & 0 & 0 & 1 & 1 & 1 & 0 & 1 & 1 & 0 & 0 \\ 2 & 0 & 2 & 2 & 2 & 0 & 0 & 1 & 0 & 1 & 0 \end{bmatrix}.$$

À esta matriz somamos λI. Supondo $\lambda = 1$, resulta

$$T_{G_e}^{(1)} = \begin{bmatrix} 2 & 1 & 1 & 2 & 2 & 0 & 1 & 0 & 0 & 0 & 0 \\ 1 & 2 & 2 & 1 & 0 & 2 & 0 & 1 & 0 & 0 & 0 \\ 1 & 2 & 2 & 0 & 1 & 2 & 0 & 0 & 1 & 0 & 0 \\ 1 & 2 & 0 & 2 & 2 & 1 & 0 & 0 & 0 & 1 & 0 \\ 1 & 0 & 2 & 2 & 2 & 1 & 0 & 0 & 0 & 0 & 1 \\ 2 & 2 & 0 & 0 & 2 & 0 & 1 & 1 & 0 & 0 & 0 \\ 2 & 0 & 2 & 2 & 0 & 2 & 2 & 0 & 1 & 0 & 0 \\ 2 & 0 & 1 & 0 & 1 & 1 & 1 & 1 & 0 & 1 & 0 \\ 2 & 1 & 0 & 1 & 0 & 1 & 1 & 0 & 1 & 0 & 1 \\ 2 & 0 & 0 & 1 & 1 & 1 & 0 & 1 & 1 & 1 & 0 \\ 2 & 0 & 2 & 2 & 2 & 0 & 0 & 1 & 0 & 1 & 1 \end{bmatrix}.$$

Esta transformada possui polinômio característico $p(x) = 1+x^2+x^3+x^5+2x^6+2x^8+2x^9+2x^{11}$, autovalor $\lambda = 1$ e determinante igual a 1.

6.2.2.1 A Transformada de Golay Cíclica

A construção da matriz de transformação da Transformada de Golay Cíclica segue os mesmos passos da construção da Transformada de Hamming Cíclica. Assim, partindo-se de seu polinômio de verificação de paridade, $(h(x) = x^6+2x^5+2x^4+2x^3+x^2+1)$, e considerando-se $\lambda = 1$ como autovalor, constroi-se a matriz de transformação a seguir,

$$T_{GC}^{(1)} = \begin{bmatrix} 2 & 2 & 2 & 2 & 1 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 2 & 2 & 2 & 2 & 1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 2 & 2 & 2 & 2 & 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 2 & 2 & 2 & 2 & 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 2 & 2 & 2 & 2 & 1 & 0 & 1 \\ 1 & 0 & 0 & 0 & 0 & 2 & 2 & 2 & 2 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 2 & 2 & 2 & 2 & 1 \\ 1 & 0 & 1 & 0 & 0 & 0 & 0 & 2 & 2 & 2 & 2 \\ 2 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 2 & 2 & 2 \\ 2 & 2 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 2 & 2 \\ 2 & 2 & 2 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 2 \end{bmatrix}.$$

Verificou-se, por meio do Sagemath®, que a matriz de transformação definida anteriormente possui ordem multiplicativa igual a 242, polinômio característico $(2 + x)^6(1 + x + x^2 + x^3 + 2x^4 + x^5)$, tendo $\lambda = 1$ como autovalor no corpo base com multiplicidade 6.

Propriedades da TNGC

i) **Linearidade**

ii) **Deslocamento no domínio do tempo**: Considere a sequência $\widehat{v} = (\widehat{v}_0, \cdots, \widehat{v}_{N-1})$ em que $\widehat{v}_i = v_{i-m}$. Então, $\widehat{v} \leftrightarrow \widehat{V}$, em que

$$\widehat{V} = x^m V.$$

iii) **Deslocamento no domínio da frequência:** Considere a sequência $\widehat{V} = (\widehat{V}_0, \cdots, \widehat{V}_{N-1})$ em que $\widehat{V}_k = V_{k-l}$. Então, $\widehat{v} = x^l v$.

iv) **Transformada da sequência constante:** A transformada da sequência $v = (r, r, \cdots, r)$ é a sequência de componentes $V_k = r, \forall k$.

v) **Transformada da sequência impulso:** A transformada da sequência $\delta = (1, 0, \cdots, 0)$, corresponde à primeira coluna da matriz $T_{GC}^{(\lambda)}$, isto é, aos coeficientes do polinômio $x[h(x) + \lambda(x)]$.

6.3 PROCESSAMENTO DE IMAGENS

Uma possível aplicação das transformadas numéricas de Pascal, Hamming e Golay, desenvolvidas nesta Tese, é na cifragem de imagens. Neste cenário, a correlação entre pixels adjacentes é uma medida indicativa da capacidade da transformada em dispersar (difundir) informação; outras medidas desta capacidade são o histograma, a entropia e os valores do NPCR *(Number of Changing Pixel Rate)* e UACI *(Unified Averaged Changed Intensity)* . Todas estas medidas foram implementadas usando-se o Matlab como ferramenta de programação. As imagens utilizadas nos testes, mostradas na Figura 4, foram obtidas da base de dados <http://sipi.usc.edu/database/> e convertidas para tons de cinza.

Para efeito de processamento de imagens, a imagem original (512 $\times$ 512 pixels) foi dividida em 4096 blocos de 8 $\times$ 8 pixels. Como a imagem é convertida em tons de cinza, os valores dos pixels variam de 0 até 255. Portanto, usou-se o corpo finito $GF(257)$.

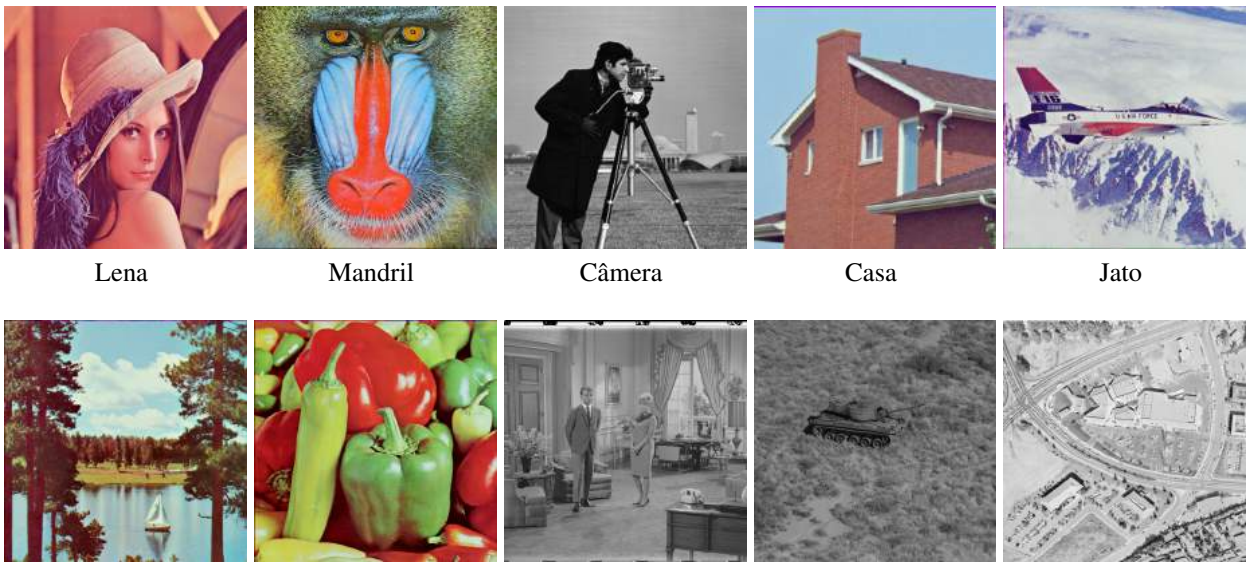


Figura 4 – Imagens utilizadas nos testes das transformadas de Pascal, Hamming e Golay.
Fonte: Obtidas em <http://sipi.usc.edu/database/>.

Neste sentido, caso o processamento produza um valor igual a 256, este valor não representa um pixel na escala de tons de cinza. Porém, isto não representa nenhum problema

quando não estamos interessados em exibir a imagem da transformada em tons de cinza. Caso desejemos visualizar a mesma, basta repetir o cálculo da transformada até que este valor não seja produzido. No caso da transformada numérica de Pascal de comprimento $N = 8$, sobre $GF(257)$, como a matriz de Pascal possui período 18.176.960 (calculado pelo Sagemath®), o risco de se repetirem os dados antes de se eliminar o valor 256 é *desprezível*. As transformadas numéricas de Hamming e de Golay usadas nos testes possuem períodos 5 e 12, respectivamente.

Denotando-se por C_1 e C_2 duas imagens cifradas, de dimensões $W \times H$, tais que suas imagens originais sejam diferentes em apenas um pixel, definem-se as métricas NPCR e UACI por

$$NPCR = \frac{\sum_{i,j} D(i,j)}{W \times H} \times 100\%,$$
$$UACI = \frac{1}{W \times H} \sum_{i,j} \frac{|C_1(i,j) - C_2(i,j)|}{F} \times 100\%,$$

em que $F = 255$ e $D(i,j) = 1$, se $C_1(i,j) = C_2(i,j)$ e $D(i,j) = 0$, caso contrário.

Pode-se mostrar (WU; NOONAN; AGAIAN, 2011) que a métrica NPCR pode ser vista como uma distribuição de probabilidades Binomial com valor médio $F/(F+1)$, enquanto que a métrica UACI pode ser vista como uma distribuição Normal com valor médio $(F+2)/(3F+3)$, em que $F = 2^n - 1$ é o maior valor possível em um esquema de codificação empregando n bits.

Para uma cifragem ideal (WU; NOONAN; AGAIAN, 2011) de 8 bits, resulta que o valor médio da métrica NPCR é, aproximadamente, 99,61%; ao passo que o valor médio da métrica UACI fica em torno de 33,46%.

Uma outra métrica utilizada foi a medida da entropia da imagem, definida por

$$H(S) = -\sum_{i=0}^{255} P(i) log_2 P(i),$$

em que $P(i)$ é a probabilidade de ocorrência do pixel i. No contexto de Processamento de imagens, a entropia nos dá uma medida da aleatoriedade da imagem. Assim, para uma fonte aleatória capaz de fornecer 256 símbolos, a entropia seria igual a 8.

6.3.1 Avaliação das Transformadas Numéricas de Pascal, Hamming e Golay

As transformadas numéricas de Pascal, Hamming e Golay são avaliadas por meio dos seguintes indicadores: histogramas, NPCR, UACI, r_{xx} (correlação horizontal entre pixels vizinhos), r_{yy} (correlação vertical entre pixels vizinhos), r_{xy} (correlação diagonal entre pixels vizinhos) e Entropia H(S).

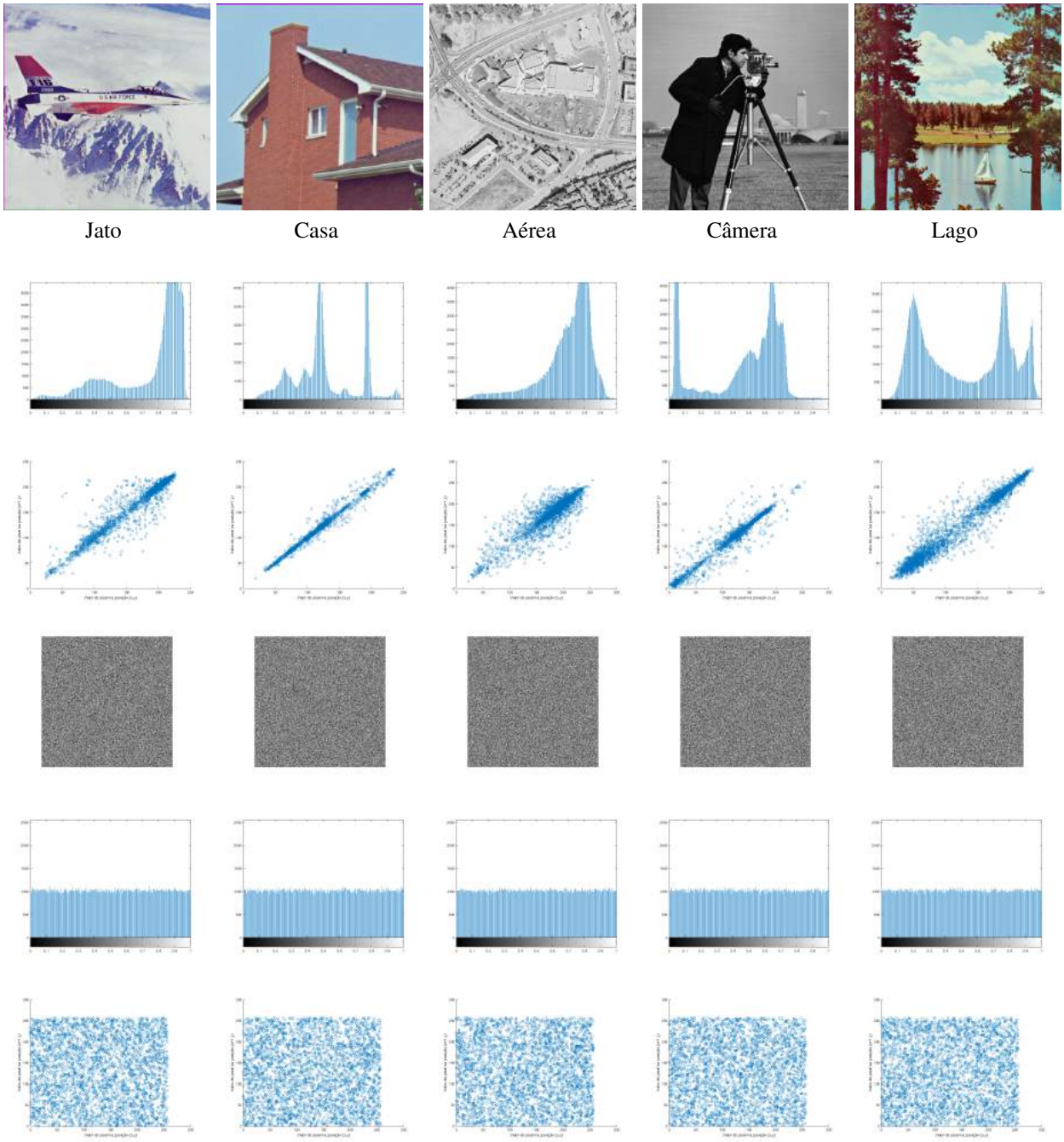


Figura 5 – Imagens utilizadas, seus histogramas e correlações verticais, TNP das imagens, histogramas das transformadas e correlação vertical das transformadas.
Fonte: O autor.

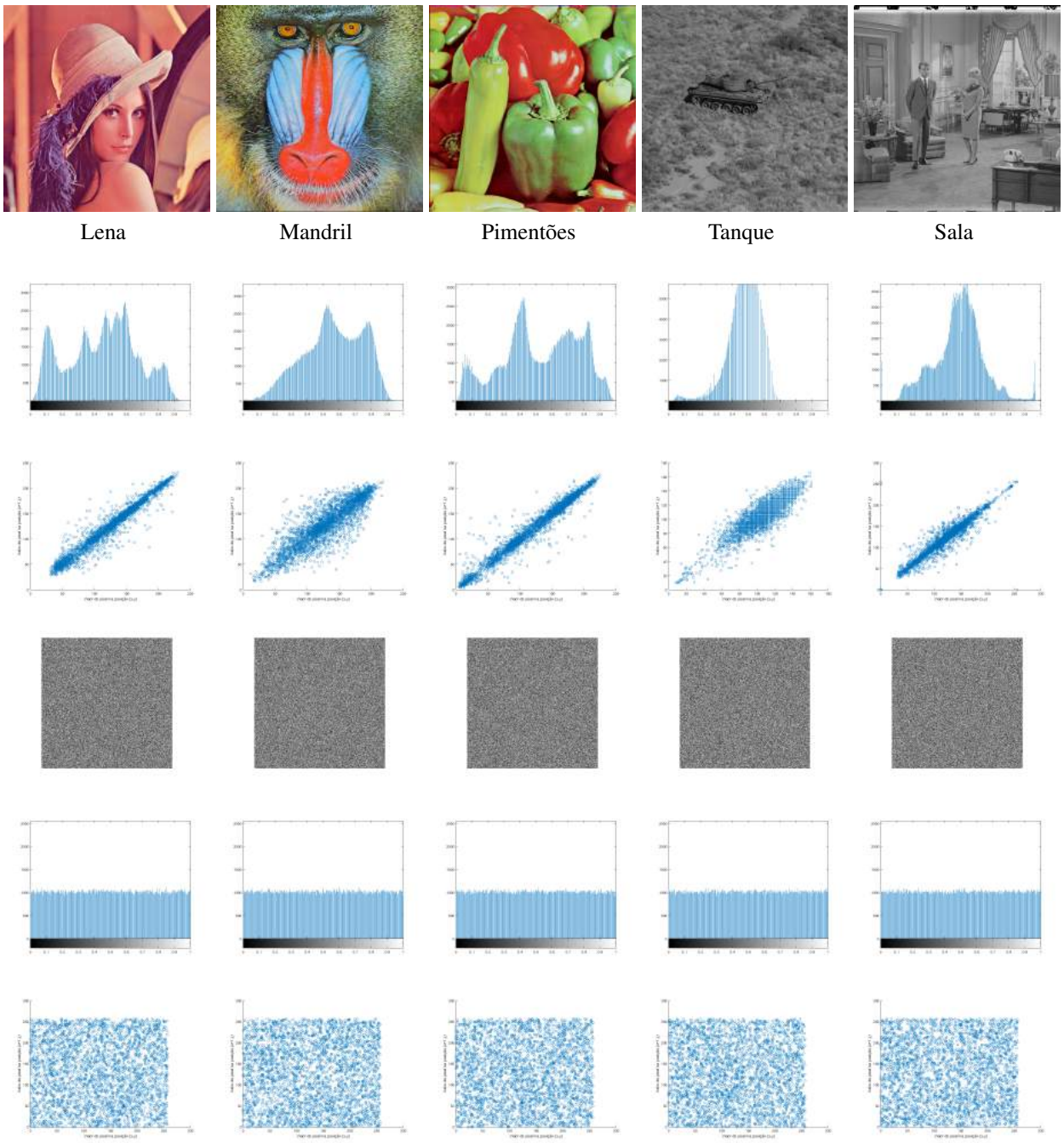


Figura 6 – Imagens utilizadas, seus histogramas e correlações verticais, TNP das imagens, histogramas das transformadas e correlação vertical das transformadas.
Fonte: O autor.

As figuras 5 e 6 mostram as imagens utilizadas nos testes, juntamente com alguns indicadores mencionados anteriormente. Note que a aplicação da TNP produz uma imagem difusa, apresentando um histograma que se *aproxima* de uma distribuição uniforme. As tabelas a seguir resumem os valores obtidos para as transformadas definidas nesta Tese. Em termos de correlação entre pixels, percebe-se claramente que as transformadas de Pascal e de Hamming apresentam resultados similares, enquanto que a transformada de Golay apresenta resultados inferiores, o que pode ser confirmado pelas imagens das transformadas (Apêndice B).

Os valores de NPCR e UACI mostrados nas tabelas 20, 22 e 24 foram obtidos comparando-se uma imagem e sua respectiva transformada. O cálculo destas métricas nos informa, respectivamente, a quantidade de pixels que foram modificados e o quanto cada pixel variou, em média, o que dá alguma informação sobre a capacidade de promover difusão das transformadas. Note que as métricas NPCR e UACI reportadas nestas tabelas não seguiram as definições originais das mesmas (WU; NOONAN; AGAIAN, 2011). No sentido de permitir uma comparação justa com os resultados definidos para tais métricas, as mesmas imagens foram processadas em conformidade com suas especificações. Assim, foi necessária uma modificação na maneira como as transformadas foram computadas. Para que a mudança de um único pixel na imagem provocasse um efeito de avalanche ao ser computada sua transformada, foi realizada a superposição de uma coluna nos blocos 8×8 que compõem a imagem, bem como o cálculo da transformada de cada imagem foi realizado N_T vezes, conforme a Tabela 26. A mesma metodologia foi utilizada para se avaliar a aproximação dos histogramas das transformadas por uma distribuição uniforme.

Tabela 19 – Coeficientes de correlação das imagens antes e depois da aplicação da TNP.

Métrica	Lena	Pimentões	Mandril	Lago	Sala	Jato	Casa	Aérea	Tanque	Camera
$r_{xy}(h)$	0,9848	0,9791	0,7580	0,9716	0,8909	0,9644	0,9902	0,8607	0,8765	0,9901
$\hat{r}_{xy}(h)$	0,0037	-0,0085	-0,0015	0,0048	-0,0019	0,0048	0,0031	-0,0024	-0,0008	-0,0052
$r_{xy}(v)$	0,9712	0,9767	0,8641	0,9750	0,9359	0,9660	0,9953	0,9020	0,8867	0,9829
$\hat{r}_{xy}(v)$	0,0039	0,0012	0,0009	0,0011	0,0031	-0,0009	-0,0019	-0,0027	-0,0025	0,0010
$r_{xy}(d)$	0,9586	0,9639	0,7251	0,9574	0,8554	0,9377	0,9858	0,8037	0,8377	0,9731
$\hat{r}_{xy}(d)$	0,0034	-0,0022	-0,0011	-0,0028	-0,0017	-0,0023	0,0007	-0,0027	-0,0010	0,0006

Tabela 20 – Entropia antes e depois da aplicação da TNP e métricas NPCR, UACI.

Métrica	Lena	Pimentões	Mandril	Lago	Sala	Jato	Casa	Aérea	Tanque	Camera
NPCR	99,99	99,99	100	100	99,99	100	99,99	99,99	100	99,99
UACI	30,03	30,71	28,74	32,75	27,62	36,48	29,49	31,45	26,32	31,16
H(S)	7,445	7,594	7,358	7,484	7,201	6,703	6,425	6,994	5,9916	7,048
$\hat{H}$(S)	7,997	7,997	7,997	7,997	7,997	7,997	7,997	7,997	7,997	7,997

Tabela 21 – Coeficientes de correlação das imagens antes e depois da aplicação da TNH.

Métrica	Lena	Pimentões	Mandril	Lago	Sala	Jato	Casa	Aérea	Tanque	Camera
$r_{xy}(h)$	0,9849	0,9789	0,7602	0,9717	0,8926	0,9636	0,9903	0,8612	0,8774	0,9901
$\hat{r}_{xy}(h)$	0,0068	0,0043	0,0027	0,00001	0,0073	0,0174	0,0281	0,0066	-0,0020	0,0091
$r_{xy}(v)$	0,9722	0,9768	0,8650	0,9752	0,9377	0,9659	0,9952	0,9011	0,8865	0,9834
$\hat{r}_{xy}(v)$	-0,0043	-0,0015	0,0011	0,0009	0,0087	0,0115	0,0260	0,0040	-0,0022	0,0129
$r_{xy}(d)$	0,9594	0,9637	0,7283	0,9579	0,8571	0,9362	0,9858	0,8034	0,8385	0,9737
$\hat{r}_{xy}(d)$	0,0004	0,0004	-0,0030	-0,0029	-0,0006	0,0027	0,0023	0,0003	-0,0016	0,0056

Tabela 22 – Entropia antes e depois da aplicação da TNH e métricas NPCR, UACI.

Métrica	Lena	Pimentões	Mandril	Lago	Sala	Jato	Casa	Aérea	Tanque	Camera
NPCR	99,60	99,62	99,60	99,60	99,62	99,60	99,59	99,60	99,62	99,61
UACI	30,20	30,74	28,81	32,73	27,60	36,48	29,57	31,48	26,35	31,07
H(S)	7,445	7,594	7,358	7,484	7,201	6,703	6,425	6,994	5,992	7,048
$\hat{H}$(S)	7,997	7,997	7,997	7,997	7,997	7,997	7,997	7,997	7,997	7,997

Tabela 23 – Coeficientes de correlação das imagens antes e depois da aplicação da TNG.

Métrica	Lena	Pimentões	Mandril	Lago	Sala	Jato	Casa	Aérea	Tanque	Camera
$r_{xy}(h)$	0,9865	0,9816	0,7879	0,9737	0,8822	96,47	0,9898	0,8661	0,8925	0,9915
$\hat{r}_{xy}(h)$	0,2150	0,1532	0,0282	0,0970	0,1717	0,1780	0,2329	0,1105	0,0673	0,2348
$r_{xy}(v)$	0,9740	0,9791	0,8786	0,9769	0,9403	0,9666	0,9950	0,9030	0,9085	0,9829
$\hat{r}_{xy}(v)$	0,1398	0,1323	0,0300	0,0991	0,0428	0,2048	0,3070	0,1371	0,0762	0,2787
$r_{xy}(d)$	0,9609	0,9655	0,7493	0,9588	0,8448	0,9370	0,9851	0,8073	0,8569	0,9743
$\hat{r}_{xy}(d)$	0,0664	0,0477	0,0090	0,0340	0,0428	0,0791	0,1042	0,0367	0,0133	0,1218

Tabela 24 – Entropia antes e depois da aplicação da TNG e métricas NPCR, UACI.

Métrica	Lena	Pimentões	Mandril	Lago	Sala	Jato	Casa	Aérea	Tanque	Camera
NPCR	100	100	99,98	100	99,99	100	100	99,99	100	100
UACI	29,78	30,97	28,47	32,70	31,08	36,06	29,22	38,33	30,41	32,95
H(S)	7,3905	7,591	7,353	7,475	6,718	6,705	6,427	6,137	6,090	6,788
$\hat{H}$(S)	7,997	7,997	7,997	7,997	7,997	7,997	7,997	7,997	7,997	7,997

A aproximação da distribuição do histograma da transformada por uma distribuição uniforme foi avaliada pelo teste χ^2 (HOEL; PORT; STONE, 1971). Considerando-se um nível de confiabilidade de 95% e para $N = 255$, o nível crítico situa-se em 293,25. A Tabela 25 mostra os valores médios obtidos para NPCR e UACI em 100 testes, em que a cada teste a TNP é aplicada 50 vezes. O valor χ^2 é calculado sobre a última transformada computada.

Tabela 25 – Métricas NPCR, UACI e χ^2 para a TNP.

Métrica	Lena	Pimentões	Mandril	Lago	Sala	Jato	Casa	Aérea	Tanque	Camera
NPCR	98,88	98,46	98,37	98,29	98,49	98,34	98,31	98,58	99,00	98,50
UACI	33,22	33,05	33,06	33,02	33,08	33,04	33,03	33,08	33,28	33,08
χ^2	225,03	252,68	229,68	251,41	239,81	234,20	254,73	247,10	220,54	276,77

A tabela 26 mostra os valores para o teste χ^2 e os valores médios para NPCR e UACI em função do número de vezes em que a TNP é computada.

Tabela 26 – Métricas NPCR, UACI e χ^2 para a TNP em função do número de transformadas.

	Lena				Mandril				Pimentões			
N_T	10	20	30	40	10	20	30	40	10	20	30	40
NPCR	67,02	84,69	93,55	94,27	66,59	82,70	91,06	95,59	69,09	79,48	92,41	96,63
UACI	22,50	28,46	31,44	31,66	22,38	27,79	30,60	32,12	23,20	26,69	31,05	32,46
χ^2	249,8	277,0	267,9	238,6	237,6	218,4	272,5	271,4	267,9	257,3	238,1	258,5
	Lago				Sala				Jato			
N_T	10	20	30	40	10	20	30	40	10	20	30	40
NPCR	68,81	80,36	91,53	94,80	70,84	84,20	92,37	95,85	68,79	80,24	92,36	96,73
UACI	23,12	27,01	30,75	31,85	23,82	28,30	31,01	32,17	23,09	26,95	31,01	32,50
χ^2	253,3	290,6	251,7	302,8	281,0	262,6	273,0	300,6	225,2	280,2	275,6	293,6

As tabelas correspondentes às tabelas 25 e 26, para as transformadas de Hamming e de Golay, são mostradas no Apêndice B, tabelas B1 e B2, respectivamente. Os resultados mostrados nesta seção apontam para a possibilidade de utilização das transformadas definidas nesta Tese como ferramentas de criptografia. A dispersão da correlação das imagens, obtida pela aplicação

da transformada, também indica que as mesmas poderiam ser utilizadas em uma etapa de pré-processamento na cifragem de imagens com o objetivo de aumentar a resistência do mesmo à ataques diferenciais e à ataques de correlação.

O equipamento utilizado nas simulações foi um laptop Dell Inspiron equipado com processador Intel® i7 2.50GHz, 16 GB de memória RAM.

6.4 ANTENAS FRACTAIS

A matriz de transformação da TNP pode ser aplicada no projeto de antenas baseadas em formas fractais (COHEN, August, 2005), tais como as antenas fractais de Sierpinski[2] (MOHANAMURALIM; SHANMUGANANTHAM, 2012) e Minkowski [3] (SUGANTHI, 2012) (as duas mais comuns). Tais antenas são compactas e multibandas com aplicações em telefonia celular e em microondas. Atualmente existem mais de 2000 publicações envolvendo pesquisas com antenas fractais. Nas Figuras 7 e 8 mostramos as antenas fractais de Sierpinski e Minkowski, respectivamente, encontradas muito frequentemente em aparelhos de telefonia celular.

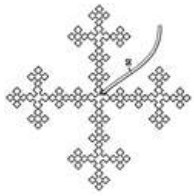


Figura 7 – Antena Fractal de Minkowski
Fonte: O autor.

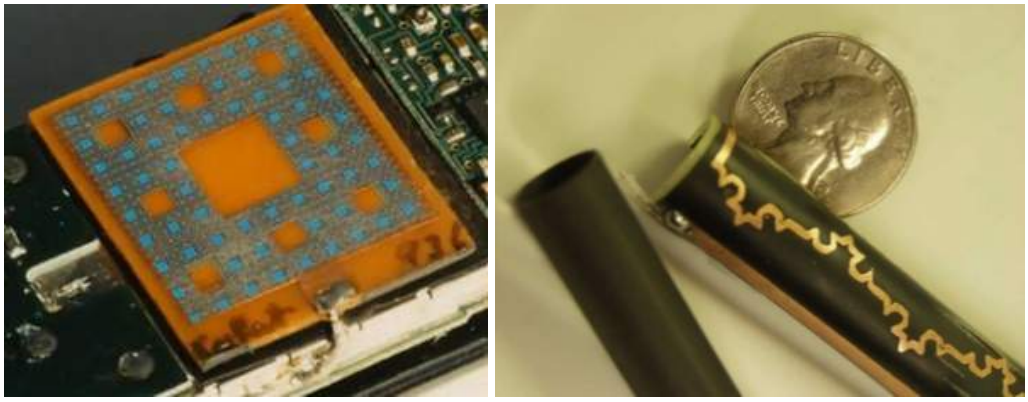


Figura 8 – Antena Fractal de Sierpinski e antena fractal monopolo de Koch
Fonte: O autor.

Uma característica muito atraente das antenas fractais, além das características citadas anteriormente, é a inexistência de componentes eletrônicos formando o circuito tanque. Ou

[2] Waclaw Sierpinski (⋆1882;†1969) foi um matemático polonês que em 1916 apresentou o triângulo de Sierpinski.
[3] Hermann Minkowski (⋆1864;†1909) foi um matemático alemão de ascendência judia-lituana, que criou e desenvolveu a geometria dos números e que usou métodos geométricos para resolver problemas difíceis em teoria dos números, física matemática e teoria da relatividade.

seja, uma antena fractal é um circuito radiante LC sem as partes componentes. Percebeu-se, neste trabalho, que as matrizes de Pascal apresentam estruturas autossimilares que permitem o desenvolvimento de verdadeiros “tapetes”, tal como o ilustrado [4] na Figura 9, produzido no Matlab®.

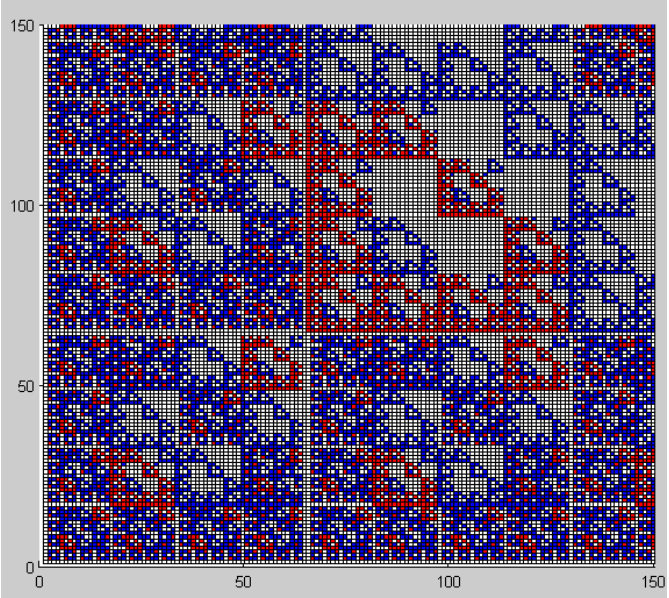


Figura 9 – Tapete de Pascal para N=5, p=5 com 512 iterações
Fonte: O autor.

Para se ter uma idéia da importância atual da Geometria Fractal, basta citar alguns de seus usos:

i) Na área de saúde (DEY, 2005): No estudo do sistema cardiovascular, em neurobiologia, em patologia e em biologia molecular. Na detecção de tumores cancerígenos.
ii) Em Telecomunicações: Antenas baseadas em formas fractais (tal como a antena de Sierpinski) são capazes de operar de forma otimizada em diversas frequências e com larguras de faixa ampliadas.
iii) Existem aplicações em Mineralogia, Ecologia, Economia, Aplicações industriais, etc.

6.4.1 Geometria Fractal

Fractais são padrões que se repetem indefinidamente em diferentes escalas. Em verdade, podemos dizer que existe uma repetição iterada de uma dada transformação. Por exemplo, considere o processo de construção do fractal conhecido como triângulo de Sierpinski ilustrado na Figura 10.

[4] O programa para gerar estes tapetes foi gentilmente cedido pelo Professor Hélio Magalhães de Oliveira do Departamento de Estatística da Universidade Federal de Pernambuco.

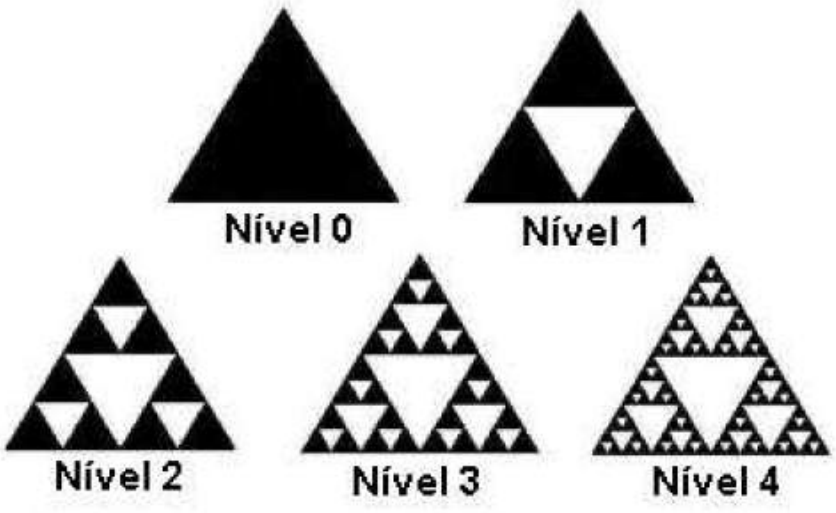


Figura 10 – Construção do triângulo de Sierpinski até a quarta iteração
Fonte: O autor.

No nível 0 temos o triângulo inicial. A partir deste triângulo, aplicamos a regra: "Conectar as medianas de cada segmento formando um triângulo central que deve ser removido." A repetição desta regra constitui o processo iterativo mostrado na Figura 10. Matematicamente denotamos

$$A = \lim_{n \Rightarrow 0} \tau^n(A_0),$$

em que A_0 denota o objeto inicial e τ^n denota n iterações da transformação τ.

Quando os fractais começaram a ser investigados, os mesmos foram denominados de "monstros matemáticos" e isto fez com que fossem "esquecidos" até serem redescobertos na década de 1960 por Benoît Mandelbrot (MANDELBROT, 1977).

Existem fractais que são gerados algebricamente, como, por exemplo, o conjunto de Mandelbrot, definido como o conjunto de pontos no plano complexo para o qual a sequência definida recursivamente por

$$Z_{novo} = Z^2_{anterior} + C$$

não tende ao infinito com C uma constante arbitrária. (MANDELBROT, 1977).

A Figura 11 ilustra o surgimento de frequências ressonantes em função do número de iterações. Note que, à medida em que o número de iterações cresce, surgem novas frequências de ressonância.

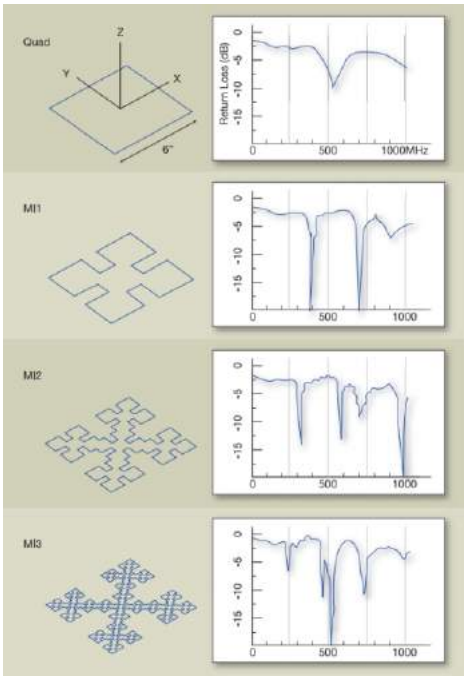


Figura 11 – Frequências ressonantes em função do número de iterações para a antena fractal de Minkowski.

Fonte: O autor.

Sabe-se que se todos os números ímpares do triângulo de Pascal forem pintados de preto, com os outros números pintados de branco, obtém-se o triângulo de Sierpinski (ROMEU; SOLER, 2001). Em outras palavras, o triângulo de Pascal pode ser usado como gerador de formas autossimilares, como o triângulo de Sierpinski, comumente usado no desenvolvimento de antenas multibandas. Na Figura 12(a) temos um triângulo de Pascal cujos múltiplos de 2 foram pintados na cor laranja, para um número de iterações igual a 10; na Figura 12(b) temos um triângulo de Pascal para 240 iterações.

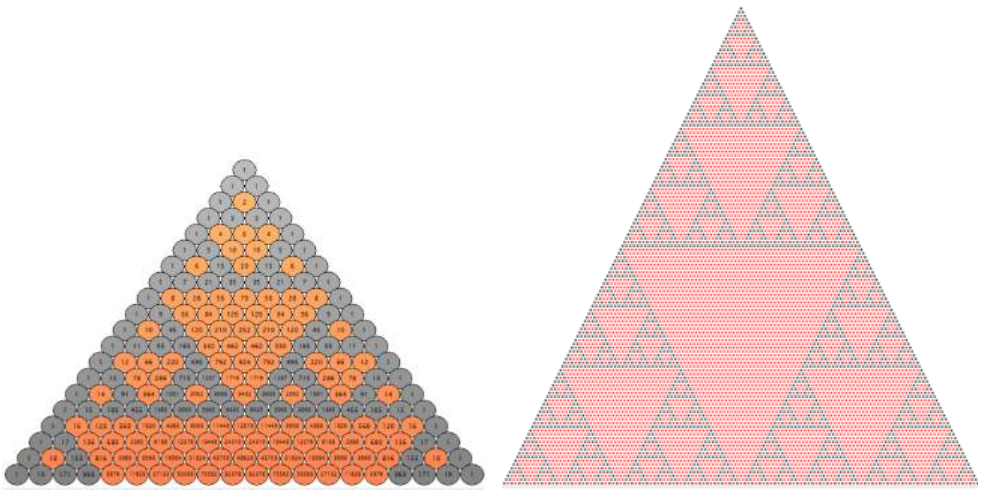


Figura 12 – Triângulo de Pascal Módulo 2: (a) Após 10 iterações; (b) Após 240 iterações.

Fonte: O autor.

Na Figura 13 temos um triângulo de Pascal módulo 5, em que cinco cores, corresponden-

tes aos inteiros módulo 5, conforme legenda, foram utilizadas para pintar o mesmo.

Figura 13 – Triângulo de Pascal Módulo 2: (a) Após 10 iterações; (b) Após 240 iterações.
Fonte: O autor.

Pelo que se observa no triângulo de Pascal, em termos de estrutura autossimilar, percebe-se que o mesmo tem potencial para ser utilizado no projeto de antenas fractais. Certamente, apenas com o desenvolvimento de uma antena prática baseada no mesmo é que suas características podem ser avaliadas; a saber, em termos de (COHEN, August, 2005):

i) Largura de faixa.
ii) Quantidade de bandas.
iii) Agilidade em frequência (*Frequency Agility*).
iv) Compactação.

Aplicações, especialmente militares (COHEN, August, 2005), são mais do que simples possibilidades. O conceito de invariância de frequência é bem conhecido e sabe-se hoje que determinadas condições devem ser atendidas para que uma antena seja invariante. Uma antena invariante é aquela cuja estrutura não depende da frequência que irá irradiar, mas dos ângulos desta estrutura. Seria uma antena fractal de Pascal uma antena invariante em termos de frequência? Antenas invariantes chegam a ter uma banda passante 200 vezes maior do que a banda passante de antenas de banda larga convencionais com menos da metade do comprimento de onda da menor frequência de operação. Atualmente, as antenas usadas em aplicações militares são grandes, de banda estreita e faz-se necessário uma grande quantidade destas antenas. Uma antena fractal, certamente, aponta para uma nova era neste sentido.

7 CONCLUSÕES E PROPOSTAS PARA CONTINUAÇÃO

Transformadas, em geral, são ferramentas matemáticas projetadas com o objetivo de resolver um problema de forma mais eficiente (e.g., com menor complexidade computacional) e/ou de fornecer informações sobre o mesmo, não disponíveis diretamente no domínio original. Neste contexto, esta Tese estuda transformadas definidas sobre estruturas algébricas finitas, as chamadas transformadas digitais. Especificamente, a mesma aborda as transformadas numéricas, definidas sobre o corpo finito $GF(p)$. Uma das principais razões de se pesquisar tais transformadas é o fato de se ter uma precisão "infinita", ou seja, para tais transformadas não existe erro de arredondamento ou truncagem, uma vez que as mesmas empregam aritmética módulo p.

Algumas áreas da Engenharia Eletrônica moderna, tais como Processamento Digital de Sinais, Códigos Corretores de Erros e Criptografia, entre outras, têm sido beneficiadas com o uso de transformadas definidas sobre corpos finitos. Este trabalho aborda a Transformada Numérica de Pascal (TNP), apresentando suas principais propriedades e propondo algumas aplicações da mesma.

7.1 CONTRIBUIÇÕES

As contribuições desta Tese encontram-se nos Capítulos 3 a 6. Após uma breve introdução, no Capítulo 1, são apresentados preliminares matemáticos, no Capítulo 2, necessários à compreensão do material apresentado nos capítulos seguintes.

No Capítulo 3 as matrizes de Pascal P_N (finita) e P_∞ (infinita), sobre $GF(p)$, foram definidas. Neste contexto, novas relações envolvendo estas matrizes foram determinadas. Em particular, mostrou-se a natureza triangular superior, em relação à diagonal secundária, da matriz de Pascal de ordem $N = p^r$, bem como a fatoração da matriz de ordem $N = kp^r$ por meio de um produto de Kronecker. Novas relações combinatórias foram estabelecidas envolvendo os elementos da matriz de Pascal P_p

No Capítulo 4 a Transformada Numérica de Pascal é definida, sua transformada inversa foi determinada usando-se a fatoração de Cholesky e suas principais propriedades foram estabelecidas. Especificamente, por meio de uma abordagem polinomial, foram encontradas expressões matemáticas para a convolução e a correlação cíclicas entre as linhas da matriz de Pascal P_p, o que levou ao resultado marcante da existência de uma linha geradora nesta matriz, a partir da qual todas as demais linhas podem ser determinadas. Os teoremas da convolução cíclica e da correlação cíclica foram apresentados. A multiplicidade dos autovalores da TNP, de ordem p^r, foi encontrada e, por meio do teorema espectral, dado que a matriz de Pascal adotada nesta Tese é uma matriz simétrica real, portanto, diagonalizável, as dimensões de seus autoespaços foram

determinadas.

No Capítulo 5, a periodicidade e as simetrias presentes na matriz de Pascal modular foram exploradas com o objetivo de construir algoritmos rápidos para a computação da TNP. Foram considerados os casos $N = p$, $N = kp$, k inteiro ≥ 1 tal que $p \nmid k$, e $N = p^r$, r inteiro ≥ 1. Tabelas de complexidade multiplicativa foram apresentadas para tais comprimentos e comparadas com a computação direta da transformada. Observou-se que as maiores reduções ocorreram, para um dado valor fixo de p, no caso em que o comprimento é uma potência de um primo.

No Capítulo 6 são sugeridos cenários de possíveis aplicações para as transformadas numéricas definidas nesta Tese. Em particular, a família de códigos de Pascal foi definida. Os parâmetros de alguns códigos de Pascal foram determinados e observou-se que, em alguns casos, o comprimento do código obtido pode ser menor do que o comprimento da transformada. Na obtenção da matriz de verificação de paridade destes códigos, que envolve a determinação dos autovalores λ da matriz de transformação da TNP, observou-se que tais autovalores podem se encontrar em corpos de extensão. Nesta Tese foram considerados apenas os códigos de Pascal associados aos autovalores pertencentes a $GF(p)$. A existência de autovalores sobre corpos de extensão abre a possibilidade para a definição de códigos de Pascal sobre tais corpos. Explorando-se as simetrias dos autovetores da TNP, algoritmos de decodificação para erros simples e duplos foram propostos.

A introdução das transformadas numéricas de Hamming (TNH) e de Golay (TNG) representa uma importante aplicação da teoria introduzida em (SOUZA; FREIRE; DEOLIVEIRA, 2009) (SOUZA; BRITTO; DEOLIVEIRA, 2011), em que se delineia a existência de um isomorfismo entre códigos de bloco lineares e transformadas digitais. Neste cenário, dada uma matriz de transformação sobre um corpo finito, pode-se obter a matriz de verificação de paridade H de um código de bloco linear, tal como o código de Pascal definido nesta Tese. De forma análoga, dado um código de bloco linear, descrito por sua matriz H, pode-se obter a matriz de transformação de uma transformada digital, tal como a TNH e a TNG, introduzidas nesta Tese. Uma versão cíclica da TNH, a Transformada Numérica de Hamming Cíclica (TNHC), é proposta nesta Tese, que permite uma análise algébrica de suas propriedades.

Por fim, a estrutura autossimilar da matriz de transformação da TNP abre possibilidades de aplicação no projeto de antenas multibandas baseadas em estruturas fractais. Tais antenas possuem largura de faixa que são até 200 vezes maior do que a largura de faixa das antenas de banda larga convencionais.

7.2 PROPOSTAS PARA CONTINUAÇÃO DO TRABALHO

Como propostas para dar continuidade à pesquisa relatada nesta Tese, enumeramos:

1. Investigar a concepção de novos algoritmos rápidos para a computação da TNP.

2. Construir algoritmos para a decodificação dos (novos) códigos introduzidos, explorando a estrutura autossimilar da matriz de transformação da TNP.
3. Construir novas classes de códigos de bloco lineares sobre corpos de extensão. Esta possibilidade ocorre quando os autovalores da TNP não se encontram em $GF(p)$.
4. Investigar propriedades e aplicações adicionais das transformadas numéricas de Hamming e de Golay.
5. Investigar a aplicação da técnica empregada na definição da TNHC para outras famílias de códigos cíclicos, e.g. códigos BCH.
6. Investigar a relação existente entre os coeficientes da matriz de transformação da TNP e a sequência de Fibonacci.
7. Simular o comportamento de antenas fractais de Pascal, construídas a partir da estrutura autossimilar da matriz de transformação da TNP, investigando-se a possibilidade de invariância de frequência para tais antenas.
8. Aprofundar o estudo da autoestrutura da TNP, considerando a matriz P_N sobre $GF(p)$ como um elemento do grupo linear $GL(N, GF(p))$.
9. Conceber novos criptossistemas baseados nas transformadas numéricas definidas nesta Tese.
10. Investigar a implementação da TNP usando-se FPGA e fazer um estudo comparativo do desempenho da mesma em relação às ferramentas numéricas tradicionais usadas para análise de sinais.

7.3 TRABALHOS PUBLICADOS

A. J. A. Paschoal, H. M. DeOliveira e R. M. Campello de Souza, A Transformada Numérica de Pascal. *Anais do XXXIII Simpósio Brasileiro de Telecomunicações*, Juiz de Fora, 2015.

A. J. A. Paschoal, R. M. Campello de Souza e H. M. DeOliveira Novas Relações na Matriz de Transformação da Transformada Numérica de Pascal, *Congresso Nacional de Matemática Aplicada e Computacional - CNMAC*, São José dos Campos, 2017.

A. J. A. Paschoal e R. M. Campello de Souza, Algoritmos Rápidos para o Cálculo da Transformada Numérica de Pascal, *Congresso Nacional de Matemática Aplicada e Computacional - CNMAC*, São José dos Campos, 2017.

7.4 TRABALHOS SUBMETIDOS

A. J. A. Paschoal and R. M. Campello de Souza, The Pascal Number Theoretic Transform. (Submetido) *Electronics Letters*.

REFERÊNCIAS

ACETO, L. Some applications of the Pascal matrix to the study of numerical methods for differential equations. *Bollettino dell'Unione Matematica Italiana*, Unione Matematica Italiana, v. 8, n. 3, p. 639–651, 2005. Citado na página 42.

AGARWAL, R.; BURRUS, C. Fast convolution using fermat number transforms with applications to digital filtering. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, Institute of Electrical and Electronics Engineers (IEEE), v. 22, n. 2, p. 87–97, apr 1974. Disponível em: <https://doi.org/10.1109/tassp.1974.1162555>. Citado na página 17.

BACHER, R.; CHAPMAN, R. Symmetric pascal matrices modulo p. *European Journal of Combinatorics*, Elsevier BV, v. 25, n. 4, p. 459–473, may 2004. Disponível em: <https://doi.org/10.1016/j.ejc.2003.06.001>. Citado 2 vezes nas páginas 24 e 57.

BIRREGAH, B.; DOH, P. K.; ADJALLAH, K. H. The twelve triangular matrix forms of the Pascal triangle: a systematic approach with the set of circulant operators. In: *10th WSEAS International Conference on Applied Mathemaics (MATH'06) Proceedings*. Dallas, Texas, USA: [s.n.], 2006. v. 3. Citado na página 23.

BURTON, D. M. *Elementary number theory*. [S.l.]: Tata McGraw-Hill Education, 2006. Citado 3 vezes nas páginas 32, 38 e 57.

CAIRE, G.; GROSSMAN, R.; POOR, H. Wavelet transforms associated with finite cyclic groups. *IEEE Transactions on Information Theory*, Institute of Electrical and Electronics Engineers (IEEE), v. 39, n. 4, p. 1157–1166, jul 1993. Disponível em: <https://doi.org/10.1109/18.243435>. Citado na página 17.

CALL, G. S.; VELLEMAN, D. J. Pascal's matrices. *The American Mathematical Monthly*, JSTOR, v. 100, n. 4, p. 372, apr 1993. Disponível em: <https://doi.org/10.2307/2324960>. Citado na página 24.

CINTRA, R. et al. Fragile watermarking using finite field trigonometrical transforms. *Signal Processing: Image Communication*, Elsevier BV, v. 24, n. 7, p. 587–597, aug 2009. Disponível em: <https://doi.org/10.1016/j.image.2009.04.003>. Citado na página 21.

COHEN, N. Fractals' new era in military antenna design. *Defense Electronics*, v. 7, p. 12–17, August, 2005. Citado 2 vezes nas páginas 96 e 100.

CONTE, S. D.; BOOR, C. de. *Elementary numerical analysis: an algorithmic approach*. [S.l.]: McGraw-Hill Higher Education, 1980. Citado 3 vezes nas páginas 21, 23 e 24.

DEOLIVEIRA, H. M. *Análise de Sinais para Engenheiros: Uma Abordagem via wavelets*. [S.l.]: Brasport, 2007. Citado na página 16.

DEOLIVEIRA, H. M.; SOUZA, R. M. Campello de; KAUFFMAN, A. N. Efficient multiplex for band-limited channels: Galois-field division multiple access. In: *Proceedings of the 1999 Workshop on Coding and Cryptography*. Paris: WCC-99, 1999. p. 235–241. Citado na página 21.

DEY, P. Fractal geometry: Basic principles and applications in pathology. *Anal Quant Cytol Histol*, v. 27, n. 5, p. 284–290, 2005. Citado na página 97.

EDELMAN, A.; STRANG, G. Pascal matrices. *The American Mathematical Monthly*, JSTOR, v. 111, n. 3, p. 189, mar 2004. Disponível em: <https://doi.org/10.2307/4145127>. Citado 3 vezes nas páginas 21, 23 e 24.

FREIRE, E. S. V. *Construção de códigos de bloco lineares via transformadas digitais*. Dissertação (Mestrado) — Universidade Federal de Pernambuco, 2009. Programa de Pós-Graduação em Engenharia Elétrica. Citado 2 vezes nas páginas 21 e 72.

GABOR, D. Theory of communication. part 1: The analysis of information. *Journal of the Institution of Electrical Engineers - Part III: Radio and Communication Engineering*, Institution of Engineering and Technology (IET), v. 93, n. 26, p. 429–441, nov 1946. Disponível em: <https://doi.org/10.1049/ji-3-2.1946.0074>. Citado na página 16.

GARCÍA-UGALDE, F. J.; PSENICKA, B.; JIMÉNEZ-SALINAS, M. O. Z transformation by Pascal matrix and its applications in the design of IIR filters. *Journal of applied research and technology*, UNAM, Centro de Ciencias Aplicadas y Desarrollo Tecnológico, v. 9, n. 3, p. 355–366, 2011. Citado na página 42.

GIRARD, A. *Invention nouvelle en l'algebre*. [S.l.]: D. Bierens de Haan, 1884. v. 1. Citado na página 76.

GOODMAN, T. J.; ABURDENE, M. F. A hardware implementation of the discrete Pascal transform for image processing. In: *Proceedings of SPIE*. [S.l.: s.n.], 2006. v. 6064, p. 148–155. Citado na página 42.

GUDVANGEN, S.; BUSKERUD, H. Practical applications of number theoretic transforms. *NORSIG-99, Norway*, 1999. Citado na página 42.

HAMMING, R. W. Error detecting and error correcting codes. *Bell System Technical Journal*, Institute of Electrical and Electronics Engineers (IEEE), v. 29, n. 2, p. 147–160, apr 1950. Disponível em: <https://doi.org/10.1002/j.1538-7305.1950.tb00463.x>. Citado na página 81.

HEIDEMAN, M.; JOHNSON, D.; BURRUS, C. Gauss and the history of the fast fourier transform. *IEEE ASSP Magazine*, Institute of Electrical and Electronics Engineers (IEEE), v. 1, n. 4, p. 14–21, oct 1984. Disponível em: <https://doi.org/10.1109/massp.1984.1162257>. Citado na página 17.

HOEL, P. G.; PORT, S. C.; STONE, C. J. *Introduction to Statiscal Theory*. [S.l.]: Houghton Mifflin, 1971. Citado na página 95.

KAK, S. The number theoretic hilbert transform. *Circuits, Systems, and Signal Processing*, Springer Nature, v. 33, n. 8, p. 2539–2548, mar 2014. Disponível em: <https://doi.org/10.1007/s00034-014-9759-8>. Citado na página 18.

LIMA, J.; LIMA, E.; MADEIRO, F. Image encryption based on the finite field cosine transform. *Signal Processing: Image Communication*, Elsevier BV, v. 28, n. 10, p. 1537–1547, nov 2013. Disponível em: <https://doi.org/10.1016/j.image.2013.05.008>. Citado 3 vezes nas páginas 16, 18 e 21.

LIMA, J.; SOUZA, R. C. de. The fractional fourier transform over finite fields. *Signal Processing*, Elsevier BV, v. 92, n. 2, p. 465–476, feb 2012. Disponível em: <https://doi.org/10.1016/j.sigpro.2011.08.010>. Citado na página 18.

LIMA, J. B. *Trigonometria sobre Corpos Finitos: Novas Definições e Cenários de Aplicação*. Tese (Tese de Doutorado) — Programa de Pós-Graduação em Engenharia Elétrica, UFPE, 2008. Citado na página 17.

LIMA, P. H. E. S. *Transformadas fracionais em corpos finitos: novas definições e cenários de aplicação*. Tese (Tese de Doutorado) — Programa de Pós-Graduação em Engenharia Elétrica, UFPE, 2015. Citado na página 18.

LIN, S.; COSTELLO, D. J. *Error Control Coding*. 2nd. ed. [S.l.]: Prentice Hall Englewood Cliffs, 2004. Citado na página 72.

LV, X.-G.; HUANG, T.-Z.; REN, Z.-G. A new algorithm for linear systems of the pascal type. *Journal of Computational and Applied Mathematics*, Elsevier BV, v. 225, n. 1, p. 309–315, mar 2009. Disponível em: <https://doi.org/10.1016/j.cam.2008.07.045>. Citado na página 24.

MANDELBROT, B. B. *Fractals*. [S.l.]: Wiley Online Library, 1977. Citado na página 98.

MCELIECE, R. *Finite Fields for Computer Scientists and Engineers, 1987*. [S.l.]: Kluwer Academic Publishers, Norwell, MA, 1987. Citado 3 vezes nas páginas 21, 22 e 75.

MOHANAMURALIM, R.; SHANMUGANANTHAM, T. Sierpinski carpet fractal antenna for multiband applications. *International Journal of Computer Applications*, International Journal of Computer Applications, 244 5 th Avenue,# 1526, New York, NY 10001, USA India, v. 39, n. 14, p. 19–23, 2012. Citado 2 vezes nas páginas 36 e 96.

MOON, T. K. *Error Correction Coding: Mathematical Methods and Algorithms*. [S.l.]: John Wiley and Sons, 2005. Citado na página 84.

NAMIAS, V. The fractional order fourier transform and its application to quantum mechanics. *IMA Journal of Applied Mathematics*, Oxford University Press (OUP), v. 25, n. 3, p. 241–265, 1980. Disponível em: <https://doi.org/10.1093/imamat/25.3.241>. Citado na página 17.

PASCHOAL, A. J. A.; DEOLIVEIRA, H. M.; SOUZA, R. M. Campello de. A transformada numérica de Pascal. In: *XXXIII Simpósio Brasileiro de Telecomunicações*. Juiz de Fora - Brasil: Anais..., 2015. Citado 3 vezes nas páginas 18, 19 e 42.

PASCHOAL, A. J. A.; SOUZA, R. M. C. D.; OLIVEIRA, H. M. D. Novas relações na matriz de transformação da transformada numérica de pascal. In: . SBMAC, 2018. Disponível em: <https://doi.org/10.5540/03.2018.006.01.0404>. Citado na página 18.

PASCHOAL, A. J. A.; SOUZA, R. M. C. de. Algoritmos rápidos para o cálculo da transformada numérica de pascal. In: . SBMAC, 2017. Disponível em: <https://doi.org/10.5540/03.2018.006.01.0310>. Citado na página 19.

PASCHOAL, A. J. A.; SOUZA, R. M. Campello de; DEOLIVEIRA, H. M. Novas ferramentas para o processamento de sinais em corpos finitos. In: *XI Simpósio Brasileiro de Telecomunicações 1993, Natal, pp. 704-707*. [S.l.]: Anais..., 1993. Citado 2 vezes nas páginas 17 e 18.

POLLARD, J. M. The fast fourier transform in a finite field. *Mathematics of Computation*, American Mathematical Society, v. 25, n. 114, p. 365–374, 1971. ISSN 00255718, 10886842. Disponível em: <http://www.jstor.org/stable/2004932>. Citado 2 vezes nas páginas 16 e 17.

RADER, C. Discrete convolutions via mersenne transrorms. *IEEE Transactions on Computers*, Institute of Electrical and Electronics Engineers (IEEE), C-21, n. 12, p. 1269–1273, dec 1972. Disponível em: <https://doi.org/10.1109/t-c.1972.223497>. Citado na página 17.

ROMEU, J.; SOLER, J. Generalized sierpinski fractal multiband antenna. *IEEE Transactions on Antennas and Propagation*, Institute of Electrical and Electronics Engineers (IEEE), v. 49, n. 8, p. 1237–1239, 2001. Disponível em: <https://doi.org/10.1109/8.943320>. Citado 2 vezes nas páginas 42 e 99.

SAFF, E. B.; SNIDER, A. D. *Fundamentals of Matrix Analysis with Applications*. [S.l.]: John Wiley & Sons, 2015. Citado na página 63.

SOUZA, M. M. CAMPELLO de et al. The discrete cosine transform over prime finite fields. In: *Telecommunications and Networking - ICT 2004*. Berlin, Heidelberg: Springer, 2004. p. 482–487. ISBN 978-3-540-27824-5. Disponível em: <$https://doi.org/10.1007/978-3-540-27824-5_65$>. Citado na página 17.

SOUZA, R. M. C. de; OLIVEIRA, H. M. de. Eigensequences for multiuser communication over the real adder channel. In: *2006 International Telecommunications Symposium*. IEEE, 2006. Disponível em: <https://doi.org/10.1109/its.2006.4433415>. Citado na página 21.

SOUZA, R. M. Campello de; BRITTO, R. M. C.; DEOLIVEIRA, H. M. Códigos de Hartley em corpos finitos. In: *XXIX Simpósio Brasileiro de Telecomunicações*. Curitiba, PR, Brasil: Anais..., 2011. Citado 2 vezes nas páginas 72 e 102.

SOUZA, R. M. Campello de et al. Trigonometry in finite fields and a new Hartley transform. In: *Proceedings. 1998 IEEE International Symposium on Information Theory (Cat. No.98CH36252)*. Cambridge, MA, USA: IEEE, 1998. p. 293. Disponível em: <https://doi.org/10.1109/isit.1998.708898>. Citado 2 vezes nas páginas 17 e 18.

SOUZA, R. M. Campello de et al. A transformada discreta do seno em um corpo finito. In: *Anais do XXVIII Congresso Nacional de Matemática Aplicada e Computacional*. São Paulo, Brasil: CNMAC, 2005. v. 53, p. 404–406. Citado na página 17.

SOUZA, R. M. Campello de; FREIRE, E. S. V.; DEOLIVEIRA, H. M. Fourier codes. In: *10th International Symposium on Communication Theory and Applications*. Ambleside, Lake District, UK: Proceedings..., 2009. Citado 5 vezes nas páginas 16, 19, 21, 72 e 102.

STRANG, G. *Introduction to Linear Algebra*. [S.l.]: Wellesley-Cambridge Press Wellesley, MA, 2006. 247 p. Citado na página 76.

SUGANTHI, S. Design and simulation of miniaturized multiband fractal antennas for microwave applications. *International Journal of Information and Electronics Engineering*, EJournal Publishing, 2012. Disponível em: <https://doi.org/10.7763/ijiee.2012.v2.217>. Citado na página 96.

WU, Y.; NOONAN, J. P.; AGAIAN, S. NPCR and UACI randomness tests for image encryption. *Cyber journals: multidisciplinary journals in science and technology, Journal of Selected Areas in Telecommunications (JSAT)*, p. 31–38, 2011. Citado 2 vezes nas páginas 91 e 94.

ZHANG, X.-D. *Matrix Analysis and Applications*. [S.l.]: Cambridge University Press, 2017. Citado na página 28.

APÊNDICES

APÊNDICE A - SIMULAÇÕES NO MATHEMATICA

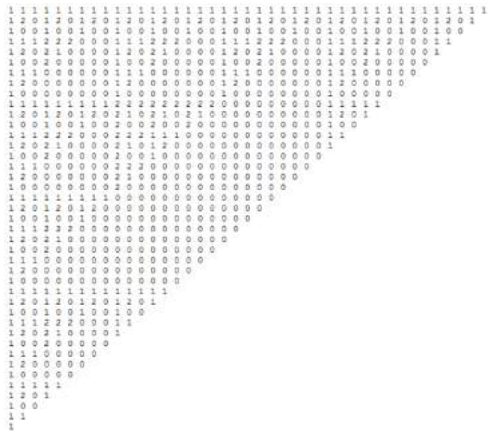
A1 - Códigos de Pascal

A fim de avaliar os códigos de Pascal foram considerados os casos p=2, 3, 5, 7, 11, 13, 17, 19, 23 e 29, para diversos comprimentos, totalizando 183 casos (34, 35, 28, 27, 25, 24, 3, 2, 1 e 4, respectivamente). Para cada caso, foram determinados:

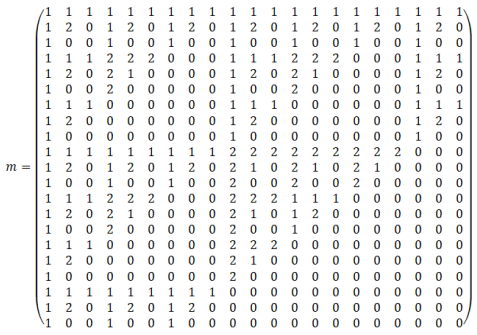
a) O polinômio característico associado à matriz de Pascal
b) Os autovalores associados (no corpo base e/ou no corpo de extensão)
c) A fatoração do polinômio característico
d) A matriz de paridade do código de Pascal obtido
e) Os parâmetros do código obtido (N_C, k, d e R)

O fato de as matrizes de Pascal, com ordens iguais a múltiplos da característica do corpo finito considerado, poderem ser fatoradas como um produto de Kronecker, facilitou a determinação dos seus autovalores.

```
(**********************************
Explorando códigos de Pascal
   N = 21, p = 3
**********************************)
PascalTriangleForm[Table[Mod[Binomial[n,Range[0,n]],3],{n,0,40}]]
```



$m =$



```
p=PolynomialMod[CharacteristicPolynomial[m,x],3]
1+2x^9+x^12+2x^21

Factor[p,Modulus→3]
```

```
2(2+x)^9(2+x+x^2)^3(2+2x+x^2)^3

Solve[p==0,x,Modulus→3]
{{x→1},{x→1},{x→1},{x→1},{x→1},{x→1},{x→1},{x→1},{x→1}}

(* Autovalor 1 com multiplicidade 9 *)
n1=MatrixForm[Mod[m-IdentityMatrix[21],3]]
```

$$n1 = \begin{pmatrix} 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 0 & 1 & 2 & 0 & 1 & 2 & 0 & 1 & 2 & 0 & 1 & 2 & 0 & 1 & 2 & 0 & 1 & 2 & 0 \\ 1 & 0 & 2 & 1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 \\ 1 & 1 & 1 & 1 & 2 & 2 & 0 & 0 & 0 & 1 & 1 & 1 & 2 & 2 & 2 & 0 & 0 & 0 & 1 & 1 & 1 \\ 1 & 2 & 0 & 2 & 0 & 0 & 0 & 0 & 0 & 1 & 2 & 0 & 2 & 1 & 0 & 0 & 0 & 0 & 1 & 2 & 0 \\ 1 & 0 & 0 & 2 & 0 & 2 & 0 & 0 & 0 & 1 & 0 & 0 & 2 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 & 0 & 2 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 \\ 1 & 2 & 0 & 0 & 0 & 0 & 0 & 2 & 0 & 1 & 2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 2 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 2 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 2 & 2 & 2 & 2 & 2 & 2 & 2 & 2 & 0 & 0 & 0 \\ 1 & 2 & 0 & 1 & 2 & 0 & 1 & 2 & 0 & 2 & 0 & 0 & 2 & 1 & 0 & 2 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 2 & 0 & 2 & 2 & 0 & 0 & 2 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 2 & 2 & 2 & 0 & 0 & 0 & 2 & 2 & 2 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 2 & 0 & 2 & 1 & 0 & 0 & 0 & 0 & 2 & 1 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 2 & 0 & 0 & 0 & 0 & 0 & 2 & 0 & 0 & 1 & 0 & 2 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 2 & 2 & 2 & 0 & 0 & 0 & 2 & 0 & 0 & 0 & 0 & 0 \\ 1 & 2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 2 & 1 & 0 & 0 & 0 & 0 & 0 & 2 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 2 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 2 & 0 & 0 \\ 1 & 2 & 0 & 1 & 2 & 0 & 1 & 2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 2 & 0 \\ 1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 2 \end{pmatrix}$$

```
MatrixForm[RowReduce[n1,Modulus→3]
```

$$\begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 2 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 2 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 2 & 2 & 2 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 2 & 2 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 2 & 2 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 2 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 2 & 2 & 2 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 2 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 2 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 2 & 2 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 2 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}$$

$$H = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 2 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 2 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 2 & 2 & 2 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 2 & 2 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 2 & 2 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 2 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 2 & 2 & 2 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 2 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 2 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 2 & 2 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 2 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}$$

As tabelas 27, 28, 29 e 30 apresentam os parâmetros dos códigos de Pascal, sobre $GF(p)$, para $p = 5, 7, 11$ e 13, respectivamente.

Tabela 27 – Parâmetros associados aos códigos de Pascal $CP^{(\lambda)}(N_C, k, d)$ sobre $GF(5)$.

Comprimento		Autovalores	Multiplicidade	Dimensão	Taxa	Distância Mínima
N	N_C	λ	m	k	R	d
2	2	4	2	1	50%	2
3	3	1	1	1	33%	3
3	3	4	2	1	33%	2
4	4	2;3	1	1	25%	4
12	12	4	4	3	25%	8
13	12	4	4	3	25%	8
19	13	1	3	3	23,1%	6
21	17	1	4	5	29,4%	6
22	19	1	8	6	31,6%	6
23	21	1	9	7	33%	6
24	24	1	8	8	33%	6
25	25	1	9	9	36%	6
26	24	1	8	8	33,3%	6
27	21	1	9	7	30,4%	6
30	15	1	4	4	26,7%	6

Tabela 28 – Parâmetros associados aos códigos de Pascal $CP^{(\lambda)}(N_C, k, d)$ sobre $GF(7)$.

Comprimento		Autovalores	Multiplicidade	Dimensão	Taxa	Distância Mínima
N	N_C	λ	m	k	R	d
3	3	1;3;5	1	1	33,3%	3;2;3
4	3	3;5	1	1	33,3%	2;3
5	3	1	1	1	33,3%	3
6	5	1	2	2	40%	3
6	4	2;4	1	1	25%	4
7	7	1;2;4	3;2;2	3;2;2	42,8%;28,6%;28,6%	3;2;2
8	8;4	1;2	6;2	6;1	75%;25%	3;4
9	3	1	1	1	33,3%	3
21	21	3;5	5	5	23,8%	6;8
22	22	3;5;6	6	6;6;5	27,3%;27,3%;22,7%	6;8;8
23	23	3;5;6	7;7;6	7;7;6	30,4%;30,4%;26,1%	6;8;8
24	24	3;5;6	8	8;8;7	33,3%;33,3%;29,2%	4;6;5
25	24	3;5;6	8	8;8;7	33,3%;33,3%;29,2%	4;6;5

Tabela 29 – Parâmetros associados aos códigos de Pascal $CP^{(\lambda)}(N_C, k, d)$ sobre $GF(11)$.

Comprimento		Autovalores	Multiplicidade	Dimensão	Taxa	Distância Mínima
N	N_C	λ	m	k	R	d
2	2	5;9	1	1	50%	2
3	3	1;2;6	1	1	33%	4
4	4	10	4	1	25%	4
5	5	10	2	2	40%	4
6	5	10	2	2	40%	4
7	4	10	4	1	25%	4
10	9	1	2	2	22,2%	7
11	11	1	3	3	27,3%	7
12	9	1	2	2	22,2%	7

Tabela 30 – Parâmetros associados aos códigos de Pascal $CP^{(\lambda)}(N_C, k, d)$ sobre $GF(13)$.

Comprimento		Autovalores	Multiplicidade	Dimensão	Taxa	Distância Mínima
N	N_C	λ	m	k	R	d
3	3	1	1	1	33,3%	3
6	6	4;10	2	2	33,3%	4;5
7	6	4;10	2	2	33,3%	4;5
10	7	1	2	2	28,6%	5
11	11	1;3;9	3;2;2	3;2;2	33,3%;22,2%;22,2%	5;7;7
12	11	1;3;9	4;3;3	4;3;3	36,4%;27,3%;27,3%	5;7;7
13	13	1;3;9	5;4;4	5;4;4	38,5%;30,8%;30,8%	5;7;7
14	11	1;3;9	4;3;3	4;3;3	36,4%;27,3%;27,3%	5;7;7
15	9	1;3;9	3;2;2	3;2;2	33,3%;22,2%;22,2%	5;7;7
16	7	1	2	2	28,6%	5

APÊNDICE B - AVALIAÇÃO DAS TRANSFORMADAS DE HAMMING E GOLAY

B.1 - Transformada de Hamming

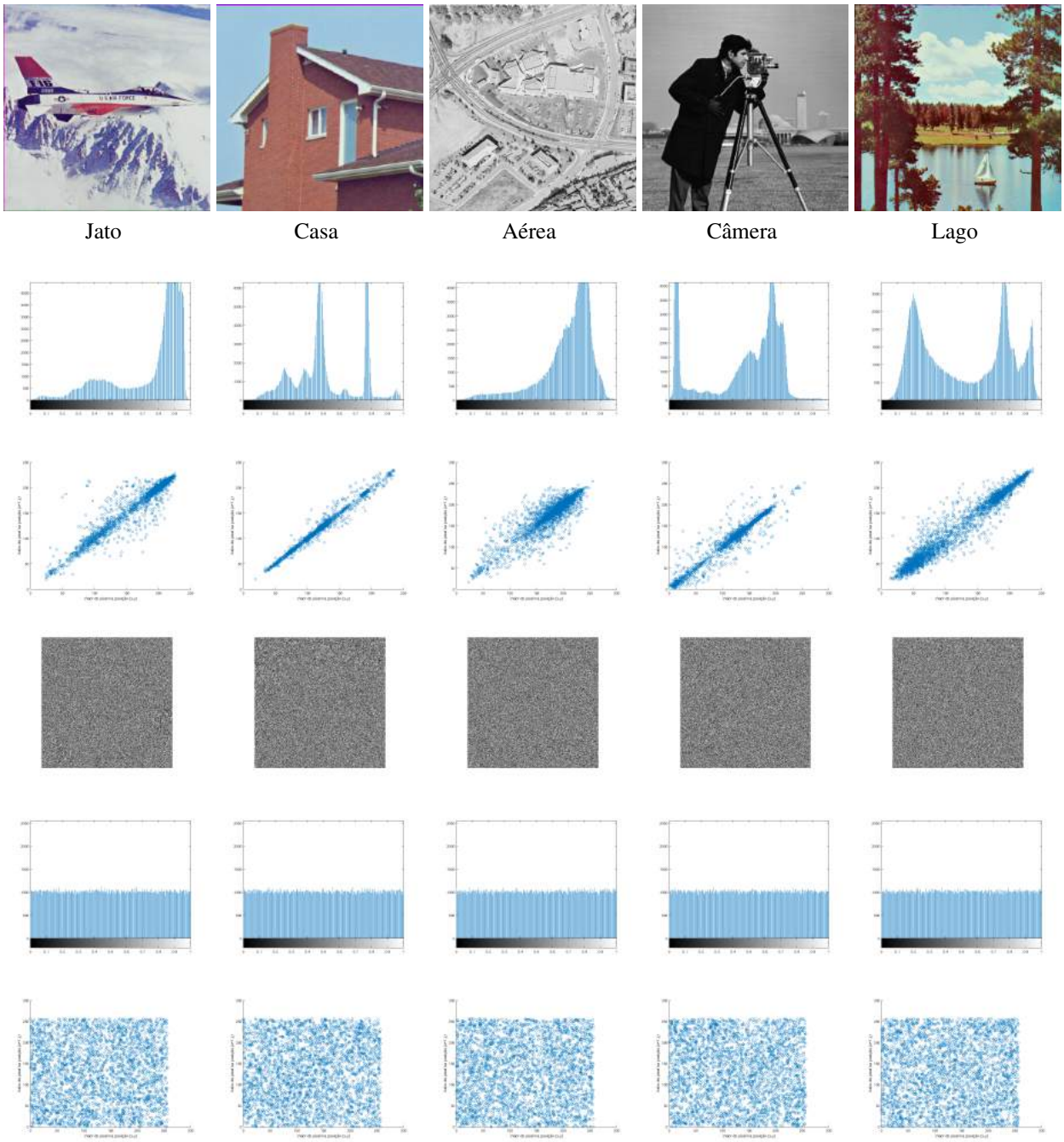


Figura 14 – Imagens utilizadas, seus histogramas, correlações verticais, TNH das imagens, histogramas das transformadas e correlação vertical das transformadas.
Fonte: O autor.

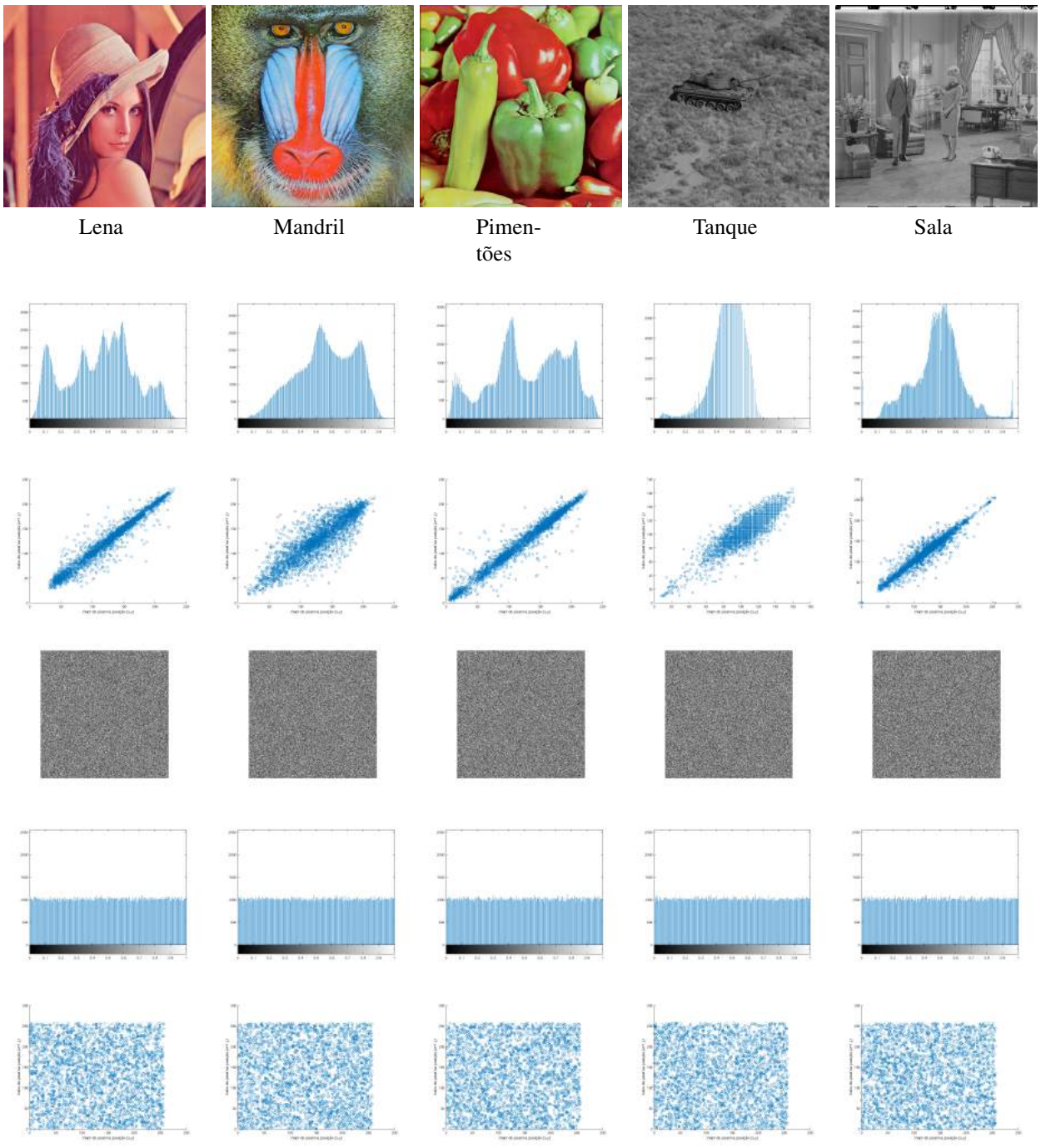


Figura 15 – Imagens utilizadas, seus histogramas, correlações verticais, TNH das imagens, histogramas das transformadas e correlação vertical das transformadas.
Fonte: O autor.

A tabela a seguir mostra os valores obtidos para algumas métricas em função do número de transformadas (NT) calculadas.

Tabela 31 – Métricas NPCR, UACI e χ^2 para a TNH em função do número de transformadas.

Lena					Mandril				Pimentões			
NT	10	20	30	40	10	20	30	40	10	20	30	40
NPCR	41,61	67,80	76,67	88,80	46,00	67,40	80,39	87,50	47,15	69,48	80,56	89,58
UACI	13,98	22,77	25,75	29,83	15,45	22,66	27,02	29,42	15,84	23,35	27,07	30,07
χ^2	262,8	265,1	243,7	210,2	285,3	219,8	236,2	255,7	270,9	236,1	243,2	264,5
Lago					Sala				Jato			
NT	10	20	30	40	10	20	30	40	10	20	30	40
NPCR	45,77	65,81	78,09	87,06	48,15	64,95	77,40	89,71	46,32	64,41	80,41	88,50
UACI	15,38	22,11	26,22	29,23	16,18	21,81	26,01	30,12	15,56	21,64	27,02	29,82
χ^2	255,9	264,2	260,2	241,7	279,0	247,9	286,5	238,4	250,1	272,7	270,3	262,3

B.2 - Transformada de Golay

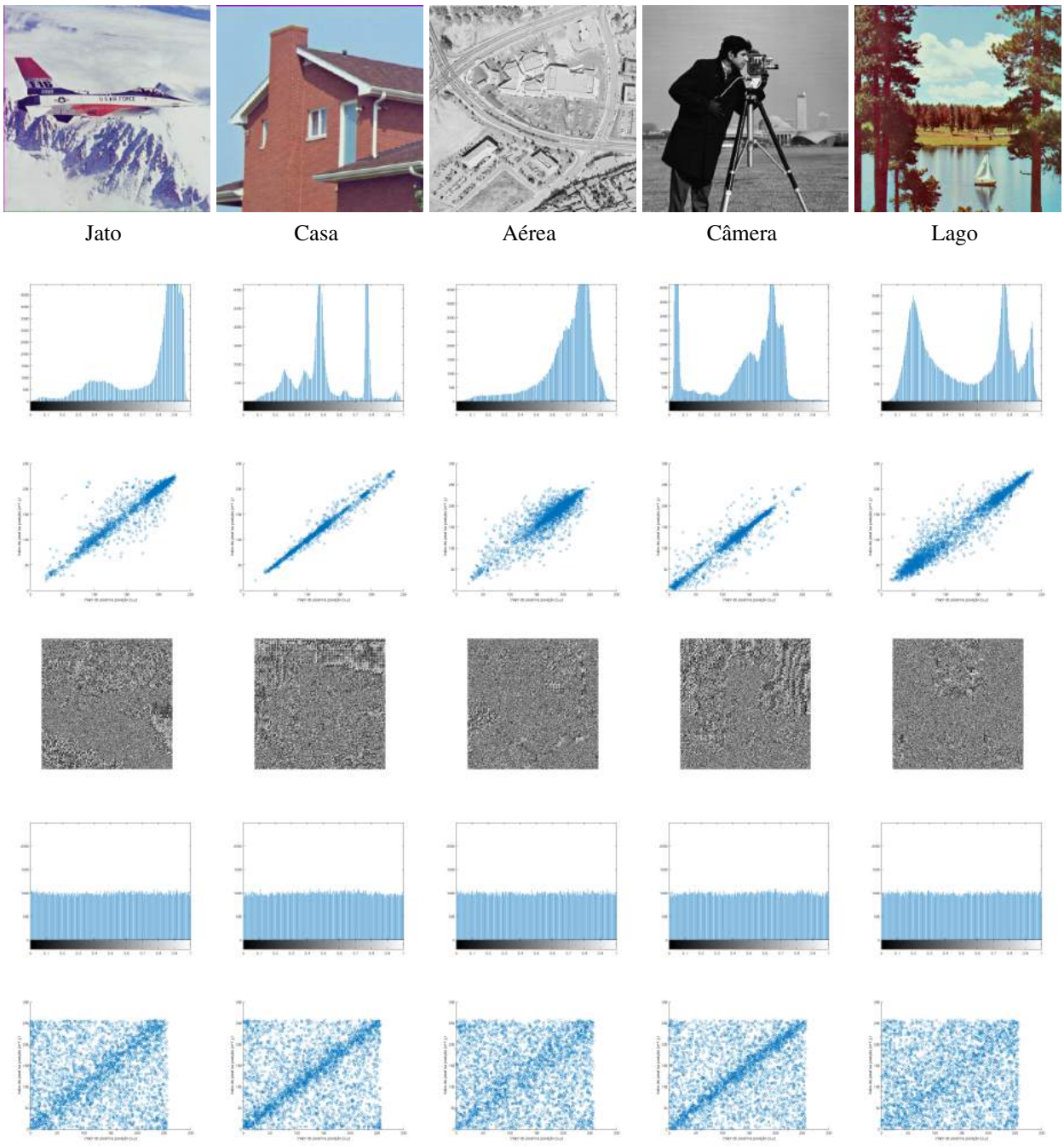


Figura 16 – Imagens utilizadas, seus histogramas, correlações verticais, TNG das imagens, histogramas das transformadas e correlação vertical das transformadas.
Fonte: O autor.

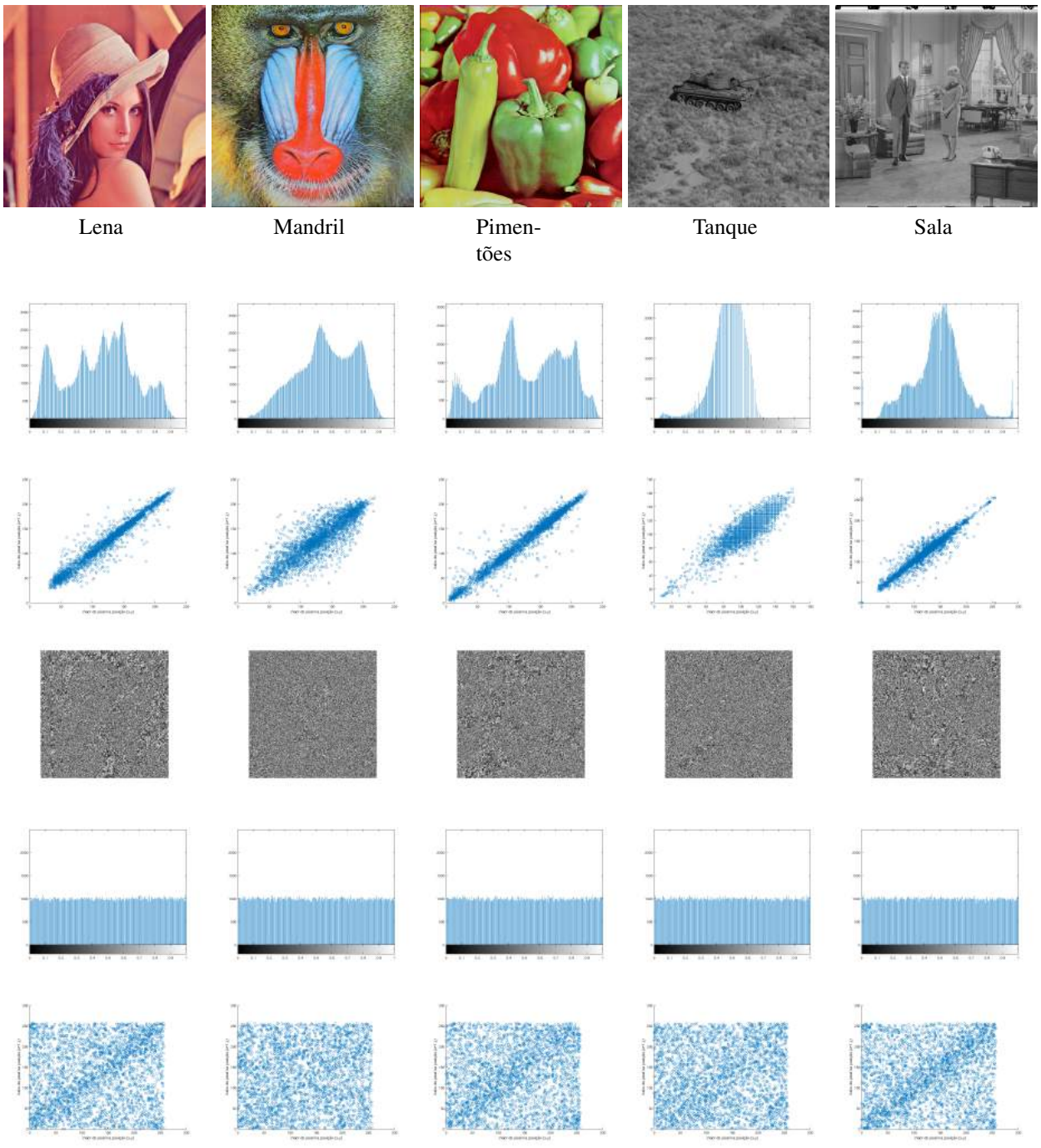


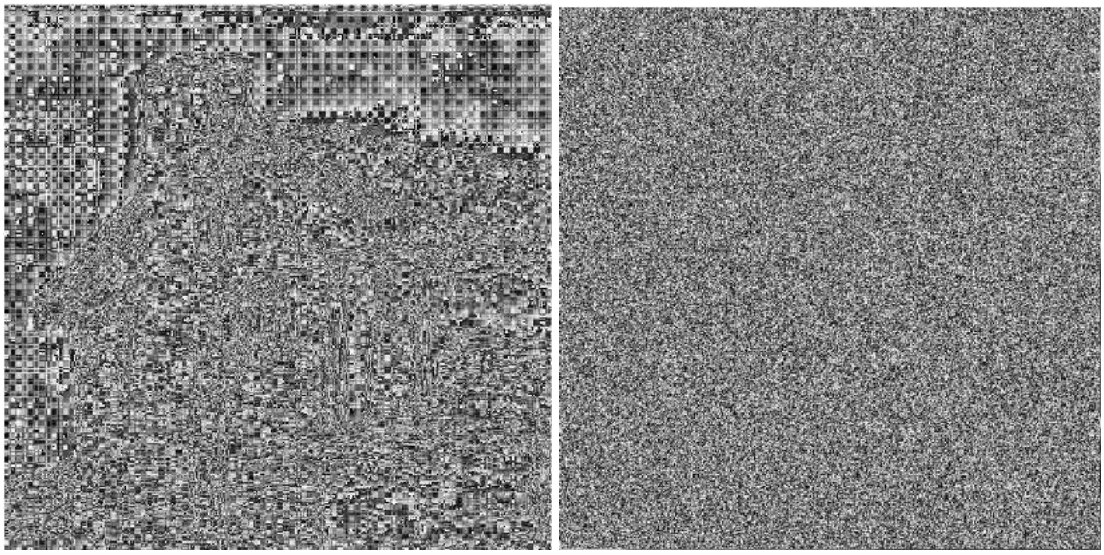
Figura 17 – Imagens utilizadas, seus histogramas, correlações verticais, TNG das imagens, histogramas das transformadas e correlação vertical das transformadas.
Fonte: O autor.

A tabela a seguir mostra os valores obtidos para algumas métricas em função do número de transformadas (NT) calculadas.

Tabela 32 – Métricas NPCR, UACI e χ^2 para a TNG em função do número de transformadas.

Lena					Mandril				Pimentões			
NT	10	20	30	40	10	20	30	40	10	20	30	40
NPCR	67,30	76,28	93,42	96,36	66,39	80,36	91,83	96,00	65,91	80,58	91,55	95,60
UACI	22,61	25,62	31,39	32,38	22,29	26,98	30,85	32,26	22,16	27,08	30,75	32,09
χ^2	305,6	194,0	267,1	220,1	278,9	261,2	263,8	250,7	245,3	279,7	243,7	268,9
Lago					Sala				Jato			
NT	10	20	30	40	10	20	30	40	10	20	30	40
NPCR	63,50	79,44	93,97	96,55	69,00	79,08	90,66	94,58	68,65	75,58	90,42	95,39
UACI	21,35	26,66	31,58	32,44	23,16	26,59	30,46	31,80	23,07	25,38	30,38	32,04
χ^2	279,7	256,2	251,8	217,5	254,5	256,5	240,4	310,1	281,5	257,4	268,7	250,6

A figura a seguir ilustra, no caso da transformada de Golay, o efeito do programa modificado para o cálculo das transformadas com superposição de uma coluna.



(1) **(2)**

Figura 18 – TNG da imagem casa, (1) Programa original e (2) Programa modificado.
Fonte: O autor.

APÊNDICE C - PROCESSAMENTO DE IMAGENS NO MATLAB

Foi usado o aplicativo Matlab ® no processamento das imagens selecionadas para gerar as imagens dos histogramas, as imagens das transformadas, as imagens das correlações, as métricas UACI, NPCR, as medidas de entropia e as correlações horizontal, vertical e diagonal.

C.1 - Programa Principal

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Programa para processar imagens usando a TNP  %
%                                               %
% Entradas:                                     %
% path do arquivo onde se encontra a imagem     %
% p: Característica do corpo                    %
% bs: Dimensão da imagem                        %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

sel = input('Entre com (1) para TNP, (2) para TNH, (3) TNG: ');
p=257;
switch sel
  case 1
    [filename,pathname] = uigetfile('C:\PDI\*.*','Sel. imagem');
    arq=fullfile(pathname, filename);
    bs=8;
    fator=1;
    imagem=zeros(512,512);
    matriz = mod(pascal(bs),p); % Matriz de Pascal
  case 2
    [filename,pathname] = uigetfile('C:\PDI\*.*','Sel. imagem');
    arq=fullfile(pathname, filename)
    bs=8;
    fator = 151;
    imagem=zeros(512,512);
    matriz=[2 1 1 1 1 1 1 0;1 3 3 4 5 6 0 1;2 3 5 5 6 0 1 1;
    3 4 5 0 0 1 2 1;4 5 6 0 2 2 3 1;5 6 0 1 2 4 4 1;
    6 0 1 2 3 4 6 1;0 1 2 3 4 5 6 2];
```

```
  case 3
    [filename,pathname] = uigetfile('C:\PDI\*.*','Sel. imagem');
    arq=fullfile(pathname, filename)
    bs=11;
    fator = 31;
    imagem=zeros(506,506);
    matriz=[2 1 1 2 2 0 1 0 0 0 0; 1 2 2 1 0 2 0 1 0 0 0;
    1 2 2 0 1 2 0 0 1 0 0;1 2 0 2 2 1 0 0 0 1 0;
    1 0 2 2 2 1 0 0 0 0 1; 2 2 0 0 2 0 1 1 0 0 0;
    2 0 2 2 0 2 2 0 1 0 0; 2 0 1 0 1 1 1 1 0 1 0;
    2 1 0 1 0 1 1 0 1 0 1;2 0 0 1 1 1 0 1 1 1 0;
    2 0 2 2 2 0 0 1 0 1 1]
end

image_ler = imread(arq);
%Converter para tons de cinza
imagem_orig = rgb2gray(image_ler);
ent=entropy(imagem_orig);
%Tamanho da imagem: largura e altura
[lin, col] = size(imagem_orig);
imagem(1:lin,1:col)=imagem_orig;
%Imagem Original
figure;
imshow(imagem_orig);
title('Imagem Original');
%Histograma da imagem original
f=figure; %aqui
imhist(mat2gray(imagem_orig),p);
[filename,pathname]=uiputfile('C:\PDI\*.*','Salvar...');
savename=fullfile(pathname,filename);
saveas(f,savename,'png');
title('Histograma Original');
% Transformada da imagem
transformada = Transform(imagem,matriz,lin,col,bs,p);
ent2=entropy(transformada);
f=figure;
imshow(mat2gray(transformada));
[filename,pathname]=uiputfile('C:\PDI\*.*','Salvar...');
savename=fullfile(pathname,filename);
```

```
saveas(f,savename,'bmp');
[r1,r2,r3]=coef_corr(transformada,bs);
switch sel
    case 1
        title('TNP da Imagem Original');
    case 2'
        title('TNH da Imagem Original');
    case 3
        title('TNG da Imagem Original');
end
%Histograma da transformada da imagem
f=figure;
imhist(mat2gray(transformada),p);
[filename,pathname]=uiputfile('C:\PDI\*.*','Salvar...');
savename=fullfile(pathname, filename)
saveas(f,savename,'png');
% Transformada inversa
imagem_rec=inv_Transform(transformada,matriz,lin,col,bs,p,fator);
figure;
imshow(mat2gray(imagem_rec));
title('Imagem Restaurada');
x=NPCR_and_UACI(imagem,transformada,1)
f=figure;
r_xy=AdjancyCorrPixel(imagem);
[filename,pathname] = uiputfile('C:\PDI\*.*','Salvar...');
savename=fullfile(pathname, filename)
saveas(f,savename,'png');
f=figure;
r_xy=AdjancyCorrPixel(transformada);
[filename,pathname] = uiputfile('C:\PDI\*.*','Salvar...');
savename=fullfile(pathname, filename)
saveas(f,savename,'png');
```

Funções Utilizadas pelo Programa Principal

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Programa que calcula a transformada %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

```
function img_tfm = Transform(img, matriz, h, w, bs,p)
% Indices para correr imagem pegando blocos de comp. bs
    for n = 1:bs:h %linha Coloca -6 qdo for TNG
        for m = 1:bs:w %coluna
            block = img(n:n+bs-1, m:m+bs-1);
            % realizando a transformação
            bloco_tfm = mod(round(matriz*block*matriz'),p);
            % realocando o bloco transformado para a imagem
            img_tfm(n:n+bs-1,m:m+bs-1) = bloco_tfm;
        end
    end
end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Função para encontrar a transformada inversa de Pascal %
%                                                         %
% Entradas:                                               %
% tfm_img: TNP da imagem original                         %
% mtx_tfm: Matriz de Pascal modular                       %
% h: altura da imagem                                     %
% w: largura da imagem                                    %
% bs: dimensão da imagem                                  %
% p: característica do corpo                              %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

function img_rec=inv_Transform(transformada,matriz,h,w,bs,p,fator)
inversa=mod(round(inv(matriz)*fator*det(matriz)),p);
% Indices para correr imagem pegando blocos de comprimento bs
    for n = 1:bs:h %Coloca -6 qdo for TNG
        for m = 1:bs:w

            % Definindo o bloco
            block = transformada(n:n+bs-1, m:m+bs-1);
            % realizando a transformação inversa
            bloco_tfm_inv = mod(round(inversa*block*inversa'),p);
            % realocando o bloco transformado para a imagem
            img_rec(n:n+bs-1,m:m+bs-1) = bloco_tfm_inv;
        end
    end
```

```
end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Programa que calcula UACI e NPCR %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

function res=NPCR_and_UACI(img_a,img_b,need_display,m_v)
    %% 1. input_check
    [ height_a, width_a, depth_a ] = size( img_a );
    [ height_b, width_b, depth_b ] = size( img_b );
    if ( ( height_a ~= height_b ) ...
      || (  width_a ~=  width_b ) ...
      || (  depth_a ~=  depth_b ) )
        error( 'input images have to be of same dimensions' );
    end
    class_a = class( img_a );
    class_b = class( img_b );
    if ( ~strcmp( class_a, class_b) )
        error( 'input images have to be of same data type');
    end
    %% 2. measure preparations
    if ( ~exist( 'm_v', 'var') )
        switch  class_a
            case 'uint16'
                m_v = 65535;
            case 'uint8'
                m_v = 255;
            case 'logical'
                m_v = 2;
            otherwise
                m_v = max ( max( img_a(:), img_b(:) ) );
        end
    end
    if ( ~exist( 'need_display', 'var' ) )
        need_display = 1;
    end
    img_a = double( img_a );
    img_b = double( img_b );
    npx = numel( img_a );
```

```
    %% 3. NCPR score and p_value
    res.npcr_score=sum(double(img_a(:)~= img_b(:)))/ npx;
    npcr_mu=(m_v )/(m_v+ 1);
    npcr_var=((m_v)/(m_v+ 1)^2 )/npx;
    res.npcr_pVal=normcdf(res.npcr_score,npcr_mu,sqrt(npcr_var));
    res.npcr_dist=[npcr_mu,npcr_var];
    %% 4. UACI score and p_value
    res.uaci_score=sum(abs(img_a(:)-img_b(:)))/npx/m_v;
    uaci_mu=(m_v+2)/(m_v*3+3);
    uaci_v=((m_v+2)*(m_v^2+2*m_v+3)/18/(m_v+1)^2/ m_v)/npx;
    p_vals=normcdf(res.uaci_score,uaci_mu,sqrt(uaci_v));
    p_vals( p_vals > 0.5 ) = 1 - p_vals( p_vals > 0.5 );
    res.uaci_pVal = 2 * p_vals;
    res.uaci_dist=[uaci_mu,uaci_v];
    %% 5. optional output
    if (need_display)
       format long;
       display(res);
    end
end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Programa correlações vertical, horizontal e diagonal %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

function [r1,r2,r3]=coef_corr(img,bs)

 x = img;
 x = double(x);

for cnt = 1:8
    %x = im(:,:,cnt,1);
    x = double(x);
    P = 32768;
    for k1 = 1:P
        coord(k1,1) = ceil((size(x,1)-1)*rand);
        coord(k1,2) = ceil((size(x,2)-1)*rand);
    end
    for k = 1:P
```

```
        ptos(k) = x(coord(k,1),coord(k,2));
        ptosh(k) = x(coord(k,1)+1,coord(k,2));
        ptosv(k) = x(coord(k,1),coord(k,2)+1);
        ptosd(k) = x(coord(k,1)+1,coord(k,2)+1);
    end

    E = mean(ptos);
    Eh = mean(ptosh);
    Ev = mean(ptosv);
    Ed = mean(ptosd);

    D = mean((ptos-E).^2);
    Dh = mean((ptosh-Eh).^2);
    Dv = mean((ptosv-Ev).^2);
    Dd = mean((ptosd-Ed).^2);

    covh = mean((ptos-E).*(ptosh-Eh));
    covv = mean((ptos-E).*(ptosv-Ev));
    covd = mean((ptos-E).*(ptosd-Ed));

    rxyh(cnt) = covh/sqrt(D*Dh);
    rxyv(cnt) = covv/sqrt(D*Dv);
    rxyd(cnt) = covd/sqrt(D*Dd);

end

r1=rxyh;
r2=rxyv;
r3=rxyd;


%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Programa que desenha a correlação vertical                              %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

function  r_xy=AdjancyCorrPixel( P )
  x1 = double(P(:,1:end-1));
  y1 = double(P(:,2:end));
  randIndex1 = randperm(numel(x1));
```

```
  randIndex1 = randIndex1(1:3000);
  x = x1(randIndex1);
  y = y1(randIndex1);
  r_xy = corrcoef(x,y);
  scatter(x,y);
  xlabel('Valor do pixel na posição (x,y)')
  ylabel('Valor do pixel na posição (x+1,y)')
end


%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Programa que calcula a frequência de ocorrência de cada pixel %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%function ch=FreqRelat(img)
x=im2T;
xx = unique(x);       % temp vector of vals
xx = sort(xx); % sorted input aligns with temp (lowest to highest)
t = zeros(size(xx)); % vector for freqs
% frequency for each value
[lin col]=size(x);
[lin1 col1]=size(xx);
for k = 1:lin1
  for i = 1:lin
    for j=1:col
        if(xx(k)==x(i,j))
            t(k)=t(k)+1;
        end
    end
  end
end
y=0;
v=mean(t(1:end-1));
for i=1:size(t,1)-1;
    y=y+((t(i)-v)^2)/v;
end
ch=y;
```

APÊNDICE D - TAPETES DE PASCAL

D.1 - Explorando a autossimilaridade do triângulo de Pascal

Uma das possíveis aplicações da transformada numérica de Pascal está relacionada às propriedades de autossimilaridade de sua matriz de transformação. Neste sentido, uma possível aplicação consiste no projeto de antenas fractais baseadas nesta matriz de transformação.

Foi desenvolvido um programa em Matlab®, pelo Professor Hélio Magalhães de Oliveira, que permite a geração de tapetes de Pascal, conforme ilustrado no Capítulo 6 desta Tese.

D.2 - Código Fonte do Tapete de Pascal Assimétrico

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Tapetes de Pascal               %
% By Hélio Magalhães de Oliveira %
% Exemplo para Matriz de Pascal  %
% GF(p=5) = Zp; alfa = 2;        %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%Tapete de Pascal Assimétrico

P4=[1 1 1 1; 1 2 3 4; 1 3 1 0; 1 4 0 0]
P4add=P4;
P4add(:,5)=0;
P4add(5,:)=0;
un_X = unique(P4add);
%
%inverse mod 5
%P4inv = modsolve(P4,[],5)
%
Cal1=kron(P4,P4)
P16=mod(Cal1,5)
P16add=P16;
P16add(:,17)=0;
P16add(17,:)=0;
%
cal2=kron(P16,P16)
```

```
P64=mod(cal2,5)
P64add=P64;
P64add(:,512)=0;
P64add(512,:)=0;
%
%
% all carpets [figures]
% fig 1
figure
hold on
pcolor(P4add)
colormap(cool(length(un_X)));
colorbar
% fig 2
figure
hold on
pcolor(P16add)
colormap(cool(length(un_X)));
colorbar
% fig 3
figure
hold on
pcolor(P64add)
colormap(cool(length(un_X)));
colorbar
%end
```

D.3 - Código Fonte do Tapete de Pascal Simétrico

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Tapetes de Pascal                 %
% By Hélio Magalhães de Oliveira    %
% Exemplo para Matriz de Pascal     %
% GF(p=5) = Zp; alfa = 2;           %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%Tapete de Pascal Simétrico
%
% Matriz de Fourier:
```

```
%
% 1  1  1  1
% 1  2  4  3
% 1  4  1  1
% 1  3  4  2
%
%
P7=[1 1 1 1 1 1 1;1 2 3 4 5 6 0; 1 3 6 3 1 0 0; 1 4 3 6 0 0 0;
1 5 1 0 0 0 0;1 6 0 0 0 0 0;1 0 0 0 0 0 0]
P7add=P7;
P7add(:,8)=0; %adiciona uma linha em P4add
P7add(8,:)=0; %adiciona uma coluna em P4add
un_X = unique(P7add); %versao de P4add ordenada e sem repetição
%
%inverse mod 7
%P7inv = modsolve(P7,[],7)
%
Cal1=kron(P7,P7)%Produto de Kronecker de P4 com ele mesmo
P49=mod(Cal1,7) %Reduz módulo 8
P49add=P49;
P49add(:,50)=0; %adiciona uma linha em P49
P49add(50,:)=0; %adciona uma coluna em P49
%
cal2=kron(P7,P49) %Produto de Kronecker entre P4 e P16
P343=mod(cal2,7)   %reduz módulo 8
P343add=P343;
P343add(:,344)=0;   %adiciona uma linha a P64
P343add(344,:)=0;   %adiciona uma coluna a P64
%
cal3=kron(P49,P49) %Produto de Kronecker entre P16 e ele mesmo
P2401=mod(cal3,7)   %reduz módulo 8
P2401add=P2401;
P2401add(:,343)=0;   %adiciona uma linha a P64
P2401add(343,:)=0;   %adiciona uma coluna a P64
%
% all carpets [figures]
% fig 1
figure %Cria objeto gráfico para exibir figura
hold on %Retém o gráfico
```

```
pcolor(P7add)
colormap([1 1 1; 1 1 0; 1 0 1; 1 0 0; 0 1 1]);
colorbar %Mostra a barra de cores
% fig 2
figure
hold on
pcolor(P49add)
colormap([1 1 1; 1 1 0; 1 0 1; 1 0 0; 0 1 1]); %(7,4,1,1,4)
colorbar
% fig 3
figure
hold on
pcolor(P343add)
colormap([1 1 1; 1 1 0; 1 0 1; 1 0 0; 0 1 1]);

colorbar
% fig 4
figure
hold on
pcolor(P2401add)
colormap([1 1 1; 1 1 0; 1 0 1; 1 0 0; 0 1 1]);
colorbar
%end
```