UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL
INSTITUTO DE MATEMÁTICA
PROGRAMA DE PÓS-GRADUAÇÃO EM MATEMÁTICA APLICADA

Fatoração Polinomial Univariada

por

Jonas Szutkoski

Dissertação submetida como requisito parcial
para a obtenção do grau de

Mestre em Matemática Aplicada

Prof. Dr. Vilmar Trevisan
Orientador

Prof. Dr. Luiz Emilio Allem
Coorientador

Porto Alegre, Janeiro de 2014.

CIP - CATALOGAÇÃO NA PUBLICAÇÃO

Szutkoski, Jonas

Fatoração Polinomial Univariada / Jonas Szutkoski.— Porto Alegre: PPGMAp da UFRGS, 2014.

135 p.: il.

Dissertação (mestrado) —Universidade Federal do Rio Grande do Sul, Programa de Pós-Graduação em Matemática Aplicada, Porto Alegre, 2014.
Orientador: Trevisan, Vilmar; Coorientador: Allem, Luiz Emilio

Dissertação: Análise Numérica e Computação Científica
Fatoração polinomial, algoritmos de fatoração

Fatoração Polinomial Univariada

por

Jonas Szutkoski

Dissertação submetida ao Programa de Pós-Graduação em Matemática Aplicada do Instituto de Matemática da Universidade Federal do Rio Grande do Sul, como requisito parcial para a obtenção do grau de

Mestre em Matemática Aplicada

Linha de Pesquisa: Análise Numérica e Computação Científica

Orientador: Prof. Dr. Vilmar Trevisan

Coorientador: Prof. Dr. Luiz Emilio Allem

Banca examinadora:

Prof. Dr. Alveri Alves Sant'Ana
PPGMAT/IM/UFRGS

Prof. Dr. Carlos Hoppen
PPGMAp/IM/UFRGS

Prof. Dr. Severino Collier Coutinho
DCC/IM/UFRJ

Dissertação apresentada
28/02/2014.

Profª Drª Maria Cristina Varrialle
Coordenadora

Sumário

Lista de Figuras

LISTA DE SÍMBOLOS

$\mathbb{Z}/p\mathbb{Z}$	conjunto dos inteiros módulo p
$\mathbb{Z}_p$	conjunto dos inteiros p-ádicos
$lc(f)$	coeficiente líder do polinômio f
$deg(f)$	grau do polinômio$f$
$cont(f)$	conteúdo do polinômio f
$pp(f)$	parte primitiva do polinômio f
$\|f\|$	norma do vetor cujas entradas são os coeficientes do polinômio f
$mdc_K(f,g)$	máximo divisor comum entre os polinômios f e g sobre $K[x]$
$[L:K]$	grau da extensão L de K
$\|S\|$	cardinalidade do conjunto S
$B_{\mathcal{L}}$	base para o reticulado $\mathcal{L}$
$(B_{\mathcal{L}})$	matriz formada pelos vetores de $B_{\mathcal{L}}$
$fer(B_{\mathcal{L}})$	forma escalonada reduzida da matriz $(B_{\mathcal{L}})$
$\lceil a \rfloor$	o inteiro mais próximo do número a, definido como $\lfloor a + 1/2 \rfloor$

RESUMO

Este trabalho trata da fatoração de polinômios em uma indeterminada. A fatoração polinomial é utilizada como uma ferramenta em diversas áreas da matemática, seja para fins aplicados ou puramente teóricos. A teoria de fatoração de polinômios teve seus maiores avanços nas últimas décadas com o desenvolvimento e constante avanço dos computadores.

O objetivo desta dissertação é apresentar um estudo do desenvolvimento desta teoria, começando com os primeiros algoritmos desenvolvidos e terminando com os algoritmos utilizados nos softwares atuais, tais como *Maple*. A maioria destes algoritmos foram implementados pelo autor no software *Maple*, embora de forma simples e sem nos preocuparmos com a eficiência dos mesmos.

ABSTRACT

This work deals with univariate polynomial factorization. Polynomial factorization is used as a tool in several areas of mathematics, for both applied as well as purely theoretical purposes. The theory of polynomial factorization had its major advances in the past few decades, due to the creation and constant development of computers.

The goal of this thesis is to present a study of this theory, starting with the first algorithms developed and closing with the algorithms used in nowadays softwares, such as *Maple*. Most of these algorithms were implemented by the author in *Maple*, although in a simple way and with no worries about efficiency.

AGRADECIMENTOS

Agradeço à minha família, especialmente aos meus pais, por apoiarem todas as decisões que tomei até aqui.

Agradeço também ao professor Vilmar Trevisan, pelo apoio durante a maior parte de minha caminhada através desse mundo maluco da matemática, e ao professor Luiz Emilio Allem, pelos conselhos e discussões durante os vários seminários realizados ao longo dos últimos anos.

Agradeço ao professor Mark van Hoeij, a quem tive o prazer de conhecer durante uma visita a Universidade do Estado da Flórida através do programa *Missão Científica de Curta Duração no Exterior*, e cujos conselhos foram indispensáveis para entender certos detalhes, imperceptíveis aos meus olhos.

Finalmente, agradeço à UFRGS, pelo ensino público e de qualidade e por todo apoio que recebi durante minha graduação e pós-graduação, sem os quais eu não teria chegado onde cheguei.

1 INTRODUÇÃO

A fatoração polinomial está presente em muitas áreas da Matemática Pura e, principalmente, da Matemática Aplicada. Fatoração polinomial é uma área da Matemática ou, mais especificamente, da Computação Algébrica, que teve seus maiores progressos nos últimos 50 ou 60 anos com a invenção e constante desenvolvimento dos computadores. Antes disso, era impensável e até humanamente impossível tentar fatorar um polinômio com, digamos, grau 100 e coeficientes com uma magnitude de 100 dígitos, como mostra a afirmação de Charles Davies (1798-1876), de 1857:

> "When a polynomial is the product of two or more factors, it is desirable to resolve it into its component factors. This may often be done by inspection or by the aid of formulas ."[1]

É claro que Charles Davies não estava dizendo que devemos tentar fatorar um polinômio de grau 100 por inspeção, pois resolver este problema parecia envolver uma quantidade enorme de cálculos e nenhuma utilidade prática. Hoje em dia, tal tarefa é essencial em várias aplicações e pode ser executada em questão de segundos, mostrando como o problema de fatoração polinomial foi bem abordado.

Teoricamente, sempre existe uma fatoração em fatores irredutíveis para qualquer polinômio com coeficientes em um domínio de fatoração única. Este fato já era conhecido por Gauss, no século XIX. Gauss demonstrou que se R é um domínio de fatoração única, então o anel de polinômios sobre esse domínio também o é [2].

[1] "Quando um polinômio é o produto de dois ou mais fatores, é desejável decompô-lo em seus fatores. Isto pode ser feito por inspeção ou pelo auxílio de formulas." Ver [13].

[2] Para uma demonstração, ver [17], Teorema 6.8.

Entretanto, o que queremos são métodos que nos forneçam essa fatoração, ou seja, queremos algoritmos que, em um número finito de passos e, preferencialmente, em um tempo pequeno, nos forneçam essa fatoração.

A história da fatoração polinomial propriamente dita começa com Friedrich Theodor von Schubert, em 1793, quando este descreve o primeiro algoritmo para fatoração de polinômios com coeficientes inteiros. O algoritmo, descrito brevemente na próxima seção, é baseado em fatos simples sobre polinômios. Infelizmente, tal método possui complexidade exponencial no grau do polinômio e tem, portanto, pouca utilidade prática, exceto para polinômios de baixo grau.

Nesta dissertação estudaremos os principais algoritmos de fatoração, desde os primeiros algoritmos eficientes, introduzidos no século passado, até os algoritmos atuais, utilizados nos melhores programas de computação algébrica como, por exemplo, o programa *Maple*.

É importante levar em consideração sobre qual domínio de integridade queremos calcular a fatoração de um polinômio. Por exemplo, sobre o corpo dos números reais, nunca iremos obter um algoritmo que forneça a fatoração exata de um polinômio qualquer, visto que números reais só podem ser aproximados com um certo grau de precisão. Para este caso, existem vários métodos de aproximação das raízes de um polinômio como, por exemplo, o Método da Bissecção e o Método de Newton. Na prática, o problema de fatorar um polinômio geralmente é considerado sobre o anel dos inteiros e extensões finitas de corpos finitos, do corpo dos números racionais e do corpo dos números algébricos.

Começaremos enunciando formalmente nosso problema. Seja R um domínio de fatoração única. Dizemos que $f \in R[x]$, $deg(f) \geq 1$, é *irredutível* se toda vez que tivermos $f = gh$, com $g, h \in R[x]$, então $g \in R$ ou $h \in R$. Ou seja, não

é possível escrever f como um produto de dois polinômios, ambos com graus ≥ 1. Caso contrário, dizemos que f é um polinômio redutível.

Dado um polinômio $f \in R[x]$ de grau n, queremos encontrar uma fatoração para f em fatores irredutíveis. Ou seja, queremos encontrar polinômios irredutíveis $f_1, f_2, \ldots, f_r \in R[x]$ e inteiros positivos $e_1, e_2, \ldots, e_r \in \mathbb{Z}_+$ tais que

$$f = f_1^{e_1} f_2^{e_2} \cdots f_r^{e_r}.$$

O capítulo 2 trata do caso em que $R = \mathbb{F}_p$, ou seja, fatoração sobre corpos finitos. Neste capítulo, estudamos dois tipos de algoritmos: o primeiro deles divide o problema em 3 partes, a saber, fatoração livre de quadrados, fatoração em graus distintos e fatoração de mesmo grau. O segundo tipo são os algoritmos baseados em Álgebra Linear, cujo algoritmo principal é o algoritmo de Berlekamp. Também neste capítulo é apresentado o algoritmo de Niederreiter. Embora seu desenvolvimento pareça muito distinto do algoritmo de Berlekamp, existem diversas ligações entre eles.

O capítulo 3 trata da fatoração de polinômios com coeficientes racionais. Veremos que é suficiente encontrar um algoritmo que fatore polinômios com coeficientes inteiros. Veremos também como podemos encontrar uma fatoração de um polinômio com coeficientes inteiros através de sua fatoração módulo um primo p e uma técnica chamada Levantamento de Hensel. Por fim, estudamos o algoritmo LLL, famoso por ser o primeiro algoritmo para fatoração de polinômios com coeficientes inteiros executado em tempo polinomial.

No capítulo 4 apresentamos o algoritmo de van Hoeij. Embora o algoritmo *LLL* tivesse complexidade polinomial, ele não substituiu os algoritmos anteriores por não ser eficiente na prática. Neste capítulo, veremos como van Hoeij contorna os problemas que levavam à ineficiência do algoritmo LLL, criando um algoritmo que é eficiente tanto na teoria quanto na prática.

Por fim, no capítulo 5, apresentamos uma aplicação da fatoração polinomial. O problema consiste em encontrar todos os subcorpos de uma extensão de corpos finita e separável. Neste capítulo veremos como a fatoração polinomial nos permite resolver esse problema.

É interessante como um problema com uma formulação tão simples pode dar origem a uma teoria tão rica e em constante avanço. A próxima seção apresenta um breve resumo da evolução desta teoria, destacando os fatos que tiveram importância fundamental no desenvolvimento da teoria de fatoração polinomial.

1.1 Um pequeno resumo da história da Fatoração Polinomial Univariada

Em muitos casos, fatorar um polinômio e encontrar suas raízes são a mesma tarefa. O problema de encontrar as raízes de um polinômio já era estudado há muito tempo. Os Babilônios, por volta de $1900 - 1600$ a.C., já possuíam procedimentos numéricos para resolver certos tipos de equações quadráticas e cúbicas. Por exemplo, para encontrar o lado de um quadrado, sabendo que a área menos o lado é 870, eles realizavam os seguintes passos

- Tome a metade de 1, ou seja, $0,5$.
- Eleve esse número ao quadrado, obtendo $0,25$.
- Some esse número à 870.
- O resultado, $870,25$, é o quadrado de $29,5$. A seguir, some $0,5$ à $29,5$, obtendo 30.

O resultado desses passos é a solução para o problema proposto, ou seja, o lado mede 30.

Note que, em notação moderna, o problema equivale a resolver $x^2 - x = 870$, e o que se fez foi calcular

$$x = \sqrt{\left(\frac{b}{2}\right)^2 + c} + \frac{b}{2}$$

para uma solução da equação $x^2 - bx = c$, com b e c positivos.

Como estes problemas surgiam de aplicações práticas, soluções negativas não eram consideradas (somente no século XVI é que os números negativos passam a ser considerados raízes e coeficientes de polinômios!). Por esse motivo, até os tempos modernos, não haviam razões para se resolver uma equação quadrática da forma $x^2 + bx + c = 0$, onde b e c são inteiros positivos, visto que esta equação não possui raízes positivas.

Mais de um milênio se passou até termos um progresso significativo na história da fatoração polinomial. Foi no Renascimento que os próximos avanços significativos ocorreram, quando matemáticos italianos encontraram soluções simbólicas para as equações cúbicas e quárticas por meio de radicais. Os principais nomes relacionados são Scipione del Ferro (1465-1526) e Nicolò Tartaglia (1500-1557), cujos trabalhos foram publicados por Geronimo Cardano (1501-1576) no famoso *Ars Magna*, de 1545.

A partir deste momento, vários outros progressos foram feitos. No século XVI, François Viète (1540-1603) descobriu uma relação entre raízes e coeficientes de um polinômio. No século XVII, Pierre de Fermat (1601-1665), demonstrou seu "Pequeno Teorema". Este teorema, em notação moderna, pode ser enunciado como

$$x^p - x = \prod_{a \in \mathbb{Z}/p\mathbb{Z}} (x - a), \tag{1.1}$$

onde p é um número primo. Ainda no século XVII, Isaac Newton (1642-1727) cria seu método para aproximar raízes reais de um polinômio.

No final do século XVIII, duas ideias foram introduzidas e mais tarde viraram a base dos algoritmos modernos de fatoração sobre corpos finitos. A primeira delas é devida a Adrien Marie Legendre (1752-1833). Legendre desenvolveu um método para encontrar raízes em $\mathbb{Z}/p\mathbb{Z}$ de um polinômio $f \in \mathbb{Z}/p\mathbb{Z}[x]$, para um primo ímpar p. Simbolicamente, Legendre fatorava o polinômio $x^p - x$ como

$$x^p - x = x(x^{(p-1)/2} - 1)(x^{(p-1)/2} + 1).$$

Legendre observou que o máximo divisor comum de f com cada um dos dois últimos fatores divide o conjunto de zeros de f em dois subconjuntos: o dos resíduos quadráticos e os não quadráticos, isto é, os elementos que são o quadrado de algum outro elemento e aqueles que não são. Em vista da equação (1.1), Legendre calculava $g = mdc(f, x^{p-1} - 1)$ para encontrar o produto dos fatores $(x - \alpha)$ com α uma raiz não nula de f. Legendre também observou que se substituirmos x por $x + s$ para qualquer $s \in \mathbb{Z}/p\mathbb{Z}$, continuaremos a ter uma fatoração para $x^p - x$. Assim, Legendre calculava $mdc(g, (x + s)^{(p-1)/2} \pm 1)$ para encontrar as raízes de f, com $s \in \mathbb{Z}/p\mathbb{Z}$ arbitrário. Essa ideia está por trás da maioria dos algoritmos probabilísticos que surgiram décadas depois.

A segunda é devido a Carl Friedrich Gauss (1777-1855). Gauss possui contribuições em quase todas as áreas da matemática. Na área de fatoração simbólica ele é responsável, dentre outras contribuições, pelo cálculo de polinômios primitivos, por descobrir relações entre fatorar polinômios com coeficientes inteiros e racionais e a eliminação Gaussiana para sistemas lineares. Além disso, Gauss é responsável pelo cálculo da parte livre de quadrados de um polinômio e pelo algoritmo que fornece uma decomposição de graus distintos de um polinômio.

Este último algoritmo é resultado da generalização de Gauss para a equação (1.1). Essa generalização diz que

$$x^{p^d} - x = \prod g,$$

onde o produtório é sobre todos os polinômios mônicos e irredutíveis $g \in \mathbb{Z}/p\mathbb{Z}[x]$ cujo grau divide d. Gauss usou este resultado para separar o polinômio em blocos cujos fatores irredutíveis possuem o mesmo grau.

Embora em teoria já fosse possível fatorar um polinômio com coeficientes em um corpo finito, métodos gerais e eficientes para se calcular a fatoração para valores grandes de p só foram desenvolvidos a partir de 1960. Foi a partir desse período que esta teoria teve seus maiores avanços. Um dos principais motivos foi a criação e constante desenvolvimento dos computadores. Outro motivo foi a necessidade prática de se fatorar polinômios cada vez maiores, tanto em grau como na magnitude de seus coeficientes.

Em 1967, E. Berlekamp [6] apresentou um algoritmo baseado em Álgebra Linear para fatorar polinômios sobre corpos finitos. Este algoritmo foi um marco na história da fatoração polinomial por ser um dos primeiros algoritmos determinísticos eficientes para fatoração. A partir de então, vários melhoramentos, além de algoritmos com argumentos probabilísticos, foram publicados, permitindo que cada vez mais polinômios pudessem ser fatorados. Em 1993, H. Niederreiter [33] também apresenta um algoritmo para fatorar polinômios sobre corpos finitos baseado em Álgebra Linear. O algoritmo de Niederreiter, porém, é baseado na equação diferencial

$$y^{(p-1)} + y^p = 0.$$

Embora os sistemas lineares encontrados nos algoritmos de Berlekamp e Niederreiter provenham de desenvolvimentos bem distintos, existem diversas ligações entre eles. Para mais detalhes, veja [15].

Alguns anos após a publicação do algoritmo de Berlekamp, David G. Cantor (1935-2012) e Hans Zassenhaus (1912-1991), seguindo várias ideias encontradas nos trabalhos de Gauss, apresentam outro algoritmo, este probabilístico, para fatorar polinômios sobre corpos finitos [10]. Este algoritmo possui um tempo de execução melhor do que o algoritmo de Berlekamp para valores grandes do primo p. Este fato fez com que este algoritmo fosse usado nos principais programas de computação algébrica até os dias atuais.

Passamos agora para a fatoração de polinômios com coeficientes inteiros. Isaac Newton (1643-1727), em seu livro *Arithmetica Universalis*, de 1707, descreveu um método para encontrar os fatores lineares e quadráticos de um polinômio univariado. Em 1793, o astrônomo Friedrich Theodor von Schubert (1758-1825) mostrou como estender essa técnica para encontrar fatores de qualquer grau. O método de Schubert foi redescoberto independentemente cerca de 90 anos depois por Leopold Kronecker (1823-1891). O método deles pode ser explicado sucintamente como segue. Suponha que queremos fatorar o polinômio $f \in \mathbb{Z}[x]$, de grau n. Esse método baseia-se nos seguintes fatos:

1) Se f pode ser fatorado, então f possui um fator com grau $m \leq \lfloor n/2 \rfloor$.

2) Se $g(x)$ é um fator de $f(x)$, então $g(a)$ divide $f(a)$ para todo inteiro a.

3) Existe um único polinômio de grau no máximo m que interpola $m+1$ pontos.

O algoritmo de fatoração de Schubert-Kronecker pode ser descrito em 4 passos:

1) Escolha $m+1$ pontos $a_0, a_1, \ldots, a_m$.

2) Avalie $f(x)$ em cada um dos pontos $a_i, 0 \leq i \leq m$.

3) Encontre o conjunto F_i dos fatores(positivos e negativos) de cada inteiro $f(a_i)$, para $0 \leq i \leq m$.

4) Para encontrar os fatores de $f(x)$ de grau d, escolha $d+1$ pontos (a_i, b_i), onde $b_i \in F_i$. Interpole esses pontos, obtendo um polinômio $g(x)$. Teste se $g(x)$ divide $f(x)$.

Exemplo 1.1. *Suponha que queremos fatorar o polinômio* $f(x) = x^4 + 4 \in \mathbb{Z}[x]$. *Neste caso,* $m = \lfloor 4/2 \rfloor = 2$. *Assim, escolhemos* $m + 1 = 3$ *pontos* $a_0 = 0$, $a_1 = 1$ *e* $a_2 = -1$. *Avaliando* $f(x)$ *em cada um desses pontos e calculando os fatores primos do resultado obtemos*

$$f(a_0) = f(0) = 4, \quad F_0 = \{\pm 1, \pm 2, \pm 4\}$$

$$f(a_1) = f(1) = 5, \quad F_1 = \{\pm 1, \pm 5\}$$

$$f(a_2) = f(-1) = 5, \quad F_2 = \{\pm 1, \pm 5\}$$

Vamos encontrar os fatores de grau $d = 2$. *Para isso, precisamos escolher* $d + 1 = 3$ *pontos* (a_i, b_i), *onde* $b_i \in F_i$. *Uma escolha seriam os pontos* $(0, 1), (1, -1)$ *e* $(-1, 5)$. *O polinômio interpolador desses pontos é* $g(x) = x^2 - 3x + 1$. *Como* $g(x) \nmid f(x)$, *concluímos que esta não foi uma boa escolha. Já os pontos* $(0, 2), (1, 5)$ *e* $(-1, 1)$ *possuem polinômio interpolador* $g(x) = x^2 + 2x + 2$. *Neste caso,* $g(x)$ *divide* $f(x)$ *e portanto, é um fator de* $f(x)$.

A escolha dos pontos (a_i, b_i) *deve ser tal que* $b_i = g(a_i)$, $1 \leq i \leq d + 1$, *para o mesmo fator* g *de* f. *A primeira escolha de pontos falha pois isso não acontece. Essa condição, de caráter combinatorial, faz com que a complexidade do método seja exponencial, tornando-o ineficiente na prática para valores grandes do grau de* f.

As próximas duas contribuições sobrevivem até os dias de hoje. A primeira delas começa com Kurt Hensel (1861-1941) e os númeors p-ádicos [21]. Em

sua teoria, Hensel demonstra um resultado, hoje conhecido como *Levantamento de Hensel*[3], que permite "levantar" uma fatoração modulo p

$$f \equiv gh \; mod \; p$$

com $f, g, h \in \mathbb{Z}[x]$ e g, h coprimos módulo p, para uma fatoração

$$f \equiv g^* h^* \; mod \; p^k$$

para qualquer $k \geq 2$ e $g^* \equiv g, \; h^* \equiv h \; mod \; p$.

Em 1969, Hans Zassenhaus (1912-1991) publica o artigo [38], no qual ele utiliza o Levantamento de Hensel para obter uma fatoração em $\mathbb{Z}[x]$. Vejamos, brevemente, como este algoritmo funciona. Considere o conjunto $\{f_1, f_2, \ldots, f_r\}$ dos fatores p-ádicos de f, ou seja, sobre os números p-ádicos, $f = f_1 f_2 \cdots f_r$. Se pensarmos nos fatores de f em $\mathbb{Z}/p\mathbb{Z}[x]$ como uma aproximação de ordem 1 dos fatores p-ádicos, podemos usar o Levantamento de Hensel para obter uma fatoração com precisão p^k, para todo $k \geq 2$. Se soubermos os coeficientes de f em $\mathbb{Z}$, podemos obter uma cota L [30] para o tamanho dos coeficientes de qualquer fator g de f em $\mathbb{Z}[x]$. Assim, usando o Levantamento de Hensel para calcular os fatores com precisão $p^k \geq 2L$, podemos verificar se uma dada combinação dos fatores módulo p^k é um fator de f em $\mathbb{Z}[x]$, calculando o módulo simétrico dessa combinação e realizando a divisão em $\mathbb{Z}[x]$. Para obter uma fatoração módulo p, Zassenhaus utiliza o algoritmo de Berlekamp, citado anteriormente. Por esse motivo, esse algoritmo é conhecido como Algoritmo de Berlekamp-Zassenhaus.

Essa parte combinatorial do algoritmo de Berlekamp-Zassenhaus faz com que este algoritmo tenha complexidade exponencial no número de fatores módulo p. Na prática, porém, este algoritmo funciona bem pois a complexidade não é exponencial no grau do polinômio f e sim no número de fatores módulo p, um número que, na maioria das vezes, é menor do que o grau do polinômio f.

[3]do inglês, Hensel Lifting

A segunda contribuição começa com Hermann Minkowski (1864-1909) em seu livro *Geometrie der Zahlen* [31]. Nele Minkowski fornece a base para a teoria conhecida como Geometria dos Números, através do estudo de reticulados. Depois disso, em 1980, Arjen Lenstra (1956-), durante seu doutorado, descobre uma ligação entre vetores minimais em certo reticulado de inteiros e fatores de polinômios em $\mathbb{Z}[x]$. Em 1982, László Lovász (1948-) e os irmãos Arjen e Hendrik Lenstra (1949-) descrevem um algoritmo executado em tempo polinomial para calcular vetores minimais em reticulados de inteiros [28]. Esse algoritmo, comumentemente chamado de LLL, encontrou diversas aplicações, indo além de seu objetivo inicial, que era fatoração de polinômios com coeficientes racionais. Embora esse algoritmo seja executado em tempo polinomial, ele não substituiu o algoritmo de Berlekamp-Zassenhaus, com complexidade exponencial, pois na prática, os reticulados utilizados no algoritmo LLL têm dimensões altas e coeficientes enormes, o que implica em baixa performance computacional. Ou seja, o algoritmo com melhor complexidade teórica não é o mais rápido na prática.

A título de curiosidade, suponha que f tenha grau $n \approx 200$ e cada coeficiente tenha em torno de 200 dígitos. Para as melhores implementações do algoritmo de Berlekamp-Zassenhaus, desde que o número r de fatores módulo p seja $r < 20$, o valor exato de r tem pouco impacto sobre o tempo de CPU que o computador leva para fatorar f. O algoritmo levará em torno de 1 segundo. Por outro lado, o algoritmo LLL, sob as mesmas circunstâncias, leva em torno de 1 dia. Porém, à medida que r ultrapassa o número de 20 fatores, o tempo de CPU começa a crescer exponencialmente para o algoritmo de Berlekamp-Zassenhaus. Para ver isso, se escolhermos p tal que $r = 64$, então o algoritmo LLL continuará levando 1 dia para fatorar, enquanto que o algoritmo de Berlekamp-Zassenhaus levará estimados 100.000 anos!

Note que, se soubéssemos quais subconjuntos dos fatores módulo p fornecem os fatores de f em $\mathbb{Z}[x]$, então o algoritmo de Berlekamp-Zassenhaus levaria apenas 1 segundo, em ambos os casos. Ou seja, a única coisa que reduz de 100.000 anos para 1 segundo o tempo de execução do algoritmo são 64 bits de informação, ou seja, um vetor em $\{0,1\}^{64}$ codificando os fatores de f em $\mathbb{Z}[x]$.

Foi partindo dessa ideia e utilizando um processo para redução de bases de reticulados que, em 2002, Mark van Hoeij [22] descreveu um algoritmo para calcular esses vetores, evitando a parte combinatorial presente no algoritmo de Berlekamp-Zassenhaus, substituindo-a por um espécie de Problema da Mochila[4]. Este novo problema pode ser resolvido utilizando o algoritmo para redução de base de um reticulado. Embora van Hoeij não forneça nenhuma cota para a complexidade do algoritmo, ele mostra que o algoritmo termina. Além disso, o algoritmo tem uma performance melhor comparada com os algoritmos anteriores em todos os testes realizados pelo autor do artigo.

Em 2004, Karim Belabas [3] forneceu, após realizar vários testes, a versão melhor ajustada do algoritmo de van Hoeij. Ainda em 2004, Belabas et al. [4] deram uma cota polinomial para a complexidade de uma versão mais lenta do algoritmo de van Hoeij. Apesar disso, uma cota pior foi encontrada para a versão mais rápida do algoritmo.

Em 2007, van Hoeij e Andrew Novocin [23] deram uma cota assintótica optimal para a versão mais rápida, demonstrando que o algoritmo mais rápido na teoria era também o algoritmo mais rápido na prática.

[4]Knapsack problem

2 FATORANDO POLINÔMIOS SOBRE CORPOS FINITOS

2.1 Introdução

Um dos casos mais importantes de fatoração polinomial é o caso da fatoração sobre corpos finitos. Além de ser interessante por si só, este caso particular é utilizado, por exemplo, para calcular a fatoração de polinômios com coeficientes inteiros. Este caso também tem várias aplicações práticas, por exemplo, em teoria dos códigos, mais especificamente em códigos de redundância cíclica e em códigos BCH, em criptografia de chave pública através de curvas elípticas e em teoria dos números computacional.

Outra importante aplicação dos algoritmos estudados neste capítulo é o cálculo de logaritmos discretos sobre corpos finitos $\mathbb{F}_{p^s}$, onde p é um primo e $s \geq 2$. O cálculo desses logaritmos é muito utilizado em criptografia de chave pública.

Neste capítulo estudaremos dois tipos de algoritmos para fatoração sobre corpos finitos. Um deles utiliza as ideias de Gauss para separar o polinômio em blocos cujos fatores irredutíveis possuem o mesmo grau e, em seguida, fatorar esses blocos. A segunda classe de algoritmos utiliza Álgebra Linear para encontrar os fatores do polinômio. Dentre estes algoritmos, destaca-se o algoritmo de Berlekamp, por ser o primeiro algoritmo determinístico eficiente para fatorar polinômios sobre corpos finitos. Apresentaremos também, a título de curiosidade, um algoritmo proposto por Niederreiter. Este algoritmo é desenvolvido a partir de uma equação diferencial e supera o algoritmo de Berlekamp em alguns casos.

O algoritmo de Berlekamp, desenvolvido em 1967 por E. R. Berlekamp em [6], é um dos primeiros algoritmos eficientes de fatoração sobre corpos finitos.

Este algoritmo utiliza o Teorema Chinês dos Restos para linearizar o problema de fatorar um polinômio. Os fatores do polinômio são obtidos a partir do cálculo do máximo divisor comum entre f e polinômios em um certo conjunto chamado *Conjunto Separador*. Outros algoritmos baseados nessa ideia, mas com outros conjuntos separadores, podem ser encontrados em [9] e [35]. Após o trabalho de Berlekamp surgiram vários melhoramentos para seu algoritmo. Alguns destes melhoramentos podem ser encontrados em [18] e [34].

Outro método, disponível na época, para fatorar polinômios sobre corpos finitos era um método desenvolvido a partir de várias ideias, incluindo algumas do próprio Gauss. Esse método é explicado e melhorado num trabalho de 1981 de D. Cantor e H. Zassenhaus, ver [10]. O algoritmo apresentado neste trabalho, que chamaremos de Algoritmo de Cantor-Zassenhaus, é melhor do que o algoritmo de Berlekamp para valores grandes do primo p.

Enquanto o algoritmo de Berlekamp, em sua forma original, é um algoritmo determinístico, o algoritmo de Cantor-Zassenhaus utiliza argumentos probabilísticos para encontrar os fatores de f. Neste algoritmo, um polinômio é escolhido aleatoriamente e, após algumas transformações neste polinômio, a probabilidade de que ele contenha um fator de f é alta. Calcula-se então o máximo divisor comum deste polinômio com f para encontrar um fator não trivial de f.

Começaremos estudando o algoritmo de Cantor-Zassenhaus. Apesar de ter sido "publicado" após o algoritmo de Berlekamp, este algoritmo utiliza fatos que remontam a Gauss e que também são utilizados no algoritmo de Berlekamp.

Neste trabalho não faremos uma análise minuciosa da complexidade dos algoritmos, apenas enunciaremos seu custo computacional. O custo computacional é calculado a partir do número de operações em $\mathbb{F}_q$. Fatorar um polinômio envolve operações como soma, multiplicação, divisão e o cálculo de mdc's de dois polinômios.

As três primeiras operações podem ser realizadas em $\mathcal{O}(n^2)$ operações em $\mathbb{F}_q$, onde n é o grau do poliômio. por outro lado, o cálculo do máximo divisor comum de dois polinômios pode ser realizado em $\mathcal{O}(n^2 \log q)$ operações no corpo.

2.2 Algoritmos para fatoração em $\mathbb{F}_p[x]$

Em 1981, Cantor e Zassenhaus [10] introduziram um novo algoritmo probabilístico para fatoração sobre corpos finitos, executado em tempo polinomial em n e $\log q$, onde n é o grau de f e q é a cardinalidade do corpo $\mathbb{F}_q$.

Este método é dividido em três passos: Fatoração Livre de Quadrados, Fatoração de Graus Distintos e Fatoração de Mesmo Grau[1]. O primeiro passo tem a finalidade de eliminar fatores irredutíveis com multiplicidade maior do que 1. Aqui tratamos ambos os casos de um corpo finito e de um corpo com característica zero. Assim, a partir desta seção podemos considerar que os polinômios que iremos fatorar não contêm fatores com multiplicidade maior do que um.

O segundo passo separa os fatores irredutíveis de f de acordo com seus graus, criando uma decomposição de f em blocos cujos fatores irredutíveis possuem o mesmo grau. Finalmente, o terceiro passo encontra os fatores irredutíveis desta decomposição de mesmo grau. Segundo [17], página 369, se tomarmos um polinômio aleatoriamente e com grau alto, então o terceiro passo tem uma probabilidade baixa de ser executado várias vezes, sendo que o segundo passo consome a maior parte do tempo computacional utilizado na fatoração.

Ao longo desta seção, p denotará um primo e q denotará uma potência de p. Antes de iniciarmos um estudo de cada um dos três passos, vamos demonstrar

[1] Os dois primeiros passos desse algoritmo já eram conhecidos antes deste trabalho de 1981, sendo o terceiro passo a maior contribuição do trabalho de Cantor e Zassenhaus.

a generalização do Pequeno Teorema de Fermat para corpos finitos, resultado de importância fundamental no nosso estudo.

Teorema 2.1 (Pequeno Teorema de Fermat para corpos finitos)**.** *Seja $\mathbb{F}_q$ um corpo com q elementos, onde q é uma potência de um primo $p > 0$. Para todo $a \in \mathbb{F}_q$, tem-se*

$$a^q = a.$$

Demonstração. Note que $\mathbb{F}_q^\times = \mathbb{F}_q \setminus \{0\}$ é um grupo com $q-1$ elementos. Pelo Teorema de Lagrange (ver [16], Teorema V.3.5), todos os elementos devem satisfazer $a^{q-1} = 1$, ou seja, $a^q = a$, para todo $a \in \mathbb{F}_q^\times$. Ora, como esta última igualdade também vale para $a = 0$, temos

$$a^q = a, \ \forall \ a \in \mathbb{F}_q.$$

□

Corolário 2.2. *Seja $\mathbb{F}_q$ um corpo com q elementos, onde q é uma potência de um primo $p > 0$. Então*

$$x^q - x = \prod_{a \in \mathbb{F}_q} (x - a).$$

Demonstração. Pelo Pequeno Teorema de Fermat, $x = a$ é uma raiz do polinômio $x^q - x$, $\forall \ a \in \mathbb{F}_q$. Assim, $(x-a) \mid (x^q - x)$, $\forall a \in \mathbb{F}_q$. Como $mdc(x-a, x-b) = 1$ para $a \neq b$, segue-se que

$$\prod_{a \in \mathbb{F}_q} (x - a) \mid (x^q - x).$$

Ora, como ambos os polinômios são mônicos e possuem o mesmo grau, vale a igualdade. □

Trataremos agora cada um dos casos do processo de fatoração enunciados acima.

2.2.1 Fatoração Livre de Quadrados

A maioria dos algoritmos de fatoração assumem certas simplificações nos polinômios a serem fatorados. Este passo do processo de fatoração tem o objetivo de reduzir o problema de fatorar um polinômio qualquer ao problema de fatorar um polinômio *livre de quadrados*.

Definição 2.3. *Seja R um domímio de fatoração única e $f \in R[x]$. Dizemos que f é livre de quadrados se toda vez que $g^2 \mid f$, para $g \in R[x]$, então $g \in R$.*

Mais especificamente, se nos é dado um polinômio $f \in \mathbb{F}_q[x]$ cuja fatoração em polinômios irredutíveis é

$$f = f_1^{e_1} f_2^{e_2} \cdots f_r^{e_r},$$

com f_i irredutíveis e e_i inteiros positivos, queremos encontrar $f_1 f_2 \cdots f_r$. O polinômio $f_1 f_2 \cdots f_r$ é chamado de *parte livre de quadrados* de f.

A ideia por trás desse passo é dividir o polinômio f por um polinômio apropriado, tal que o resultado dessa divisão seja a parte livre de quadrados de f.

Definição 2.4. *Seja $f = \sum_{i=0}^{n} a_i x^i \in \mathbb{F}[x]$, onde $\mathbb{F}$ é um corpo finito ou não. Então f', a derivada formal do polinômio f, é definida por*

$$f' = \sum_{i=1}^{n} i a_i x^{i-1}.$$

É possível mostrar que a derivada formal de um polinômio tem as mesmas propriedades básicas da derivada de polinômios definida no Cálculo. Assim, se $f = f_1^{e_1} f_2^{e_2} \cdots f_r^{e_r}$, simples cálculos mostram que

$$f' = \sum_{i=1}^{r} e_i \frac{f}{f_i} f_i'. \tag{2.1}$$

Vamos agora calcular $u = mdc(f, f')$. Note que $f_i^{e_i}|f$ e que $f_i^{e_i-1}|f'$, para todo $i = 1, 2, \ldots, r$. Note também que $f_i^{e_i}$ divide todas as parcelas da soma (2.1) exceto, possivelmente, a i-ésima parcela. Para que $f_i^{e_i}|f'$, é necessário que $f_i|e_i f_i'$.

Se a característica do corpo $\mathbb{F}$ é zero, como $f_i \notin \mathbb{F}$, então $f_i' \neq 0$. Logo $e_i f_i' \neq 0$ e portanto, $f_i \nmid e_i f_i'$. Assim,

$$u = mdc(f, f') = f_1^{e_1-1} f_2^{e_2-1} \cdots f_r^{e_r-1}$$

e podemos calcular a parte livre de quadrados de f simplesmente calculando

$$\frac{f}{u} = \frac{f}{mdc(f, f')} = \frac{f}{f_1^{e_1-1} f_2^{e_2-1} \cdots f_r^{e_r-1}} = f_1 f_2 \cdots f_r.$$

Por outro lado se $\mathbb{F} = \mathbb{F}_q$, onde $q = p^s$ para s natural, então é possível que $e_i f_i' = 0$. Neste caso, $f_i|e_i f_i'$ e portanto, $f_i^{e_i}|mdc(f, f')$. Assim, quando calculamos f/u, perdemos o fator f_i e não podemos calcular a parte livre de quadrados de f simplesmente através de f/u.

Vamos estudar mais a fundo esse caso, ou seja, o caso em que $e_i f_i' = 0$. Primeiro, vamos analisar o que acontece no caso em que $f' = 0$, para um polinômio não constante $f = \sum_{i=0}^{n} a_i x^i \in \mathbb{F}_q[x]$. Como

$$f' = \sum_{i=1}^{n} i a_i x^{i-1} = 0,$$

temos que $p|ia_i$, $\forall\ i = 1, \ldots n$. Como $p \nmid a_i$ para algum i, segue-se que $p|i$. Assim, $i = jp$ para certo j inteiro e podemos escrever

$$f = \sum_{i=0}^{n} a_i x^i = \sum_{i=0}^{n} a_i^q x^i = \sum_{j=0}^{n/p} \left(a_{jp}^{q/p}\right)^p x^{jp} = \sum_{j=0}^{n/p} \left(a_{jp}^{q/p} x^j\right)^p = \left(\sum_{j=0}^{n/p} a_{jp}^{q/p} x^j\right)^p,$$

pois $a^q = a$ e $(f+g)^p = f^p + g^p$ em $\mathbb{F}_q = \mathbb{F}_{p^s}$. Por outro lado, se $f = g^p$, então $f' = pg^{p-1}g' \equiv 0$. Logo, para todo $f \in \mathbb{F}_q[x]$, $f' = 0$ se, e somente se, f é uma p-potência, ou seja, existe $g \in \mathbb{F}_q[x]$ tal que $f = g^p$.

Concluímos deste fato que $f_i' \neq 0$, pois do contrário, teríamos $f_i = g^p$ para algum $g \in \mathbb{F}_q[x]$, contrariando o fato de f_i ser irredutível. Portanto, $e_i f_i' = 0$ se, e somente se, $e_i \equiv 0 \mod p$.

Vamos agora deduzir um método para encontrar a parte livre de quadrados para este caso. Podemos descrever o máximo divisor comum entre f e f' como dois produtórios distintos, da forma

$$mdc(f, f') = \prod_{p \nmid e_i} f_i^{e_i - 1} \prod_{p | e_i} f_i^{e_i}.$$

Seja $u = mdc(f, f')$ e defina $v = f/u$. Assim, podemos ver facilmente que

$$v = \frac{f}{u} = \frac{\prod_{i=1}^{r} f_i^{e_i}}{\prod_{p \nmid e_i} f_i^{e_i - 1} \prod_{p | e_i} f_i^{e_i}} = \prod_{p \nmid e_i} f_i.$$

Logo, v é o produto dos fatores irredutíveis de f cuja multiplicidade não é um múltiplo de p. Precisamos agora encontrar a parte livre de quadrados de $\prod_{p|e_i} f_i^{e_i}$. Inicialmente, vamos demonstrar que

$$\prod_{p | e_i} f_i^{e_i} = \frac{u}{mdc(u, v^n)}, \text{ onde } n = deg(f).$$

Como $e_i \leq n$, para todo $i = 1, \ldots, r$, temos

$$mdc(u, v^n) = mdc\left(\prod_{p \nmid e_i} f_i^{e_i - 1} \prod_{p | e_i} f_i^{e_i}, \prod_{p \nmid e_i} f_i^{n} \right) = \prod_{p \nmid e_i} f_i^{e_i - 1}.$$

Agora, observe que

$$\frac{u}{mdc(u, v^n)} = \frac{\prod_{p \nmid e_i} f_i^{e_i - 1} \prod_{p | e_i} f_i^{e_i}}{\prod_{p \nmid e_i} f_i^{e_i - 1}} = \prod_{p | e_i} f_i^{e_i},$$

como desejado.

Defina $w = \frac{u}{mdc(u, v^n)} = \prod_{p|e_i} f_i^{e_i}$. Então $w' = 0$ e portanto, como vimos acima, w é uma p-potência. Assim, podemos definir

$$\tilde{w} = w^{1/p} = \prod_{p | e_i} f_i^{e_i/p} \in \mathbb{F}_q[x].$$

Aqui, podemos aplicar novamente o raciocínio descrito acima e encontrar o produto dos fatores irredutíveis de $\tilde{w}$ cuja multiplicidade não é um múltiplo de p, isto é, podemos encontrar o produto dos $f_i's$ tais que e_i/p não é um múltiplo de p. Como os fatores irredutíveis têm multiplicidade finita, aplicando este método recursivamente iremos encontrar a parte livre de quadrados de f.

Assim, assumiremos daqui para frente que todos os polinômios que queremos fatorar são livre de quadrados.

Exemplo 2.5. *Considere o polinômio* $f = x^{14}+3x^{13}+2x^{12}+x^{11}+2x^{10}+2x^4+x^3+4x^2+2x+4 \in \mathbb{Z}/5\mathbb{Z}[x]$. *Como não sabemos se* f *possui fatores com multiplicidade múltiplo de* 5, *devemos considerar essa possibilidade. Conforme analisado acima, calculemos*

$$u = mdc(f, f') = x^{10} + 2 \;\; e \;\; v = \frac{f}{u} = x^4 + 3x^3 + 2x^2 + x + 2.$$

Assim, v *é o produto de todos os fatores irredutíveis de* f *cuja multiplicidade não é um múltiplo de* 5. *Vamos agora procurar os fatores irredutíveis cuja multiplicidade é um múltiplo de* 5. *Como visto acima, estes fatores estão em*

$$w = \frac{u}{mdc(u, v^{14})} = x^{10} + 2.$$

Como $w' = 0$, *sabemos que* w *é uma p-potência, ou seja, uma 5-potência. Não é difícil ver que* $w = (x^2 + 2)^5$. *Defina*

$$\tilde{w} = x^2 + 2.$$

Como $\tilde{w}$ *é irredutível em* $\mathbb{Z}/5\mathbb{Z}[x]$, *podemos parar por aqui. Concluimos assim que a parte livre de quadrados de* f *é dada por*

$$(x^4 + 3x^3 + 2x^2 + x + 2) \cdot (x^2 + 2).$$

Se não soubermos que o polinômio $\tilde{w}$ *é irredutível, devemos aplicar todo o raciocínio novamente com* $\tilde{w}$ *no lugar de* f *para obter a parte livre de quadrados de* $\tilde{w}$.

2.2.2 Fatoração de Graus Distintos

De acordo com os resultados da seção anterior, podemos supor que os polinômios que queremos fatorar são livres de quadrados. Esta seção trata da parte do processo que separa os fatores irredutíveis do polinômio f de acordo com seus graus, produzindo diversos blocos $g_1, \ldots, g_k$, onde cada bloco g_i é o produto de todos os fatores de f que possuem o mesmo grau i e $g_k \neq 1$.

Definição 2.6. *A Decomposição em Graus Distintos de um polinômio não constante $f \in \mathbb{F}_q[x]$ é a sequência de polinômios $(g_1, g_2, \ldots, g_k)$, onde g_i é o produto de todos os polinômios mônicos e irredutíveis de grau i que dividem f em $\mathbb{F}_q[x]$. Caso f não possua nenhum fator de grau i, $g_i = 1$. Além disso, $g_k \neq 1$.*

De acordo com esta definição, se $(g_1, g_2, \ldots, g_k)$ é a decomposição em graus distintos de f, então $f = \prod_{i=1}^{k} g_i$.

Exemplo 2.7. *Seja $f = x^6 + x^5 + 2x + 2 \in \mathbb{F}_3[x]$. A fatoração de f em fatores irredutíveis em $\mathbb{F}_3[x]$ é dada por*

$$f = (x^4 + x^3 + x^2 + x + 1) \cdot (x + 2) \cdot (x + 1).$$

Sua decomposição em graus distintos produzirá os blocos g_1, g_2, g_3 e g_4, tais que

$$\begin{aligned} g_1 &= x^2 + 2 = (x + 1) \cdot (x + 2) \\ g_2 &= 1 \\ g_3 &= 1 \\ g_4 &= x^4 + x^3 + x^2 + x + 1 \end{aligned}$$

Apresentaremos agora o teorema que dará origem ao algoritmo para o cálculo da decomposição em graus distintos de um polinômio. Este teorema é, na verdade, uma generalização do Pequeno Teorema de Fermat. Antes de apresentar este teorema, vamos demonstrar um resultado auxiliar.

Proposição 2.8. *Seja L uma extensão do corpo K e sejam $f, g \in K[x] \setminus \{0\}$. Então*

$$mdc_K(f,g) = mdc_L(f,g).$$

Demonstração. Seja $h_1 = mdc_K(f,g) \in K[x]$ e $h_2 = mdc_L(f,g) \in L[x]$. Considerando $h_1 \in L[x]$, é claro que $h_1|h_2$, pois L é uma extensão de K. Por outro lado, sejam $r, s \in K[x]$ tais que $deg(r) < deg(g)$, $deg(s) < deg(f)$ e $h_1 = rf + sg$. Ora, h_2 divide ambos f e g em $L[x]$ logo, h_2 divide $rf + sg$ e portanto, $h_2|h_1$ em $L[x]$. Como ambos h_1 e h_2 são mônicos, segue-se que $h_1 = h_2$. □

Teorema 2.9. *Para todo inteiro $d \geq 1$, $x^{q^d} - x \in \mathbb{F}_q[x]$ é o produto de todos os polinômios mônicos irredutíveis cujo grau divide d.*

Demonstração. Aplicando o Pequeno Teorema de Fermat a $\mathbb{F}_{q^d}$, sabemos que $h = x^{q^d} - x$ é o produto dos fatores $x - a$, para todo $a \in \mathbb{F}_{q^d}$. Isso mostra que h é livre de quadrados em $\mathbb{F}_{q^d}[x]$. Queremos mostrar que h também é livre de quadrados em $\mathbb{F}_q[x]$. Suponha que $g^2 \mid h$ em $\mathbb{F}_q[x]$, para algum polinômio $g \in \mathbb{F}_q[x] \setminus \mathbb{F}_q$. Como h é o produto de fatores lineares, segue-se que existe $a \in \mathbb{F}_{q^d}$ tal que $(x - a) \mid g$ em $\mathbb{F}_{q^d}$. Assim, $(x - a)^2 \mid g^2$ e $g^2 | h$. Mas h é livre de quadrados em $\mathbb{F}_{q^d}[x]$, uma contradição. Assim, h é livre de quadrados em $\mathbb{F}_q[x]$. É suficiente mostrar que, para todo polinômio mônico irredutível $f \in \mathbb{F}_q[x]$ de grau $n < d$, tem-se

$$f \mid (x^{q^d} - x) \Leftrightarrow n|d.$$

Consideremos a extensão de corpos $\mathbb{F}_q \subseteq \mathbb{F}_{q^d}$ e seja $f \in \mathbb{F}_q[x]$, $deg(f) = n < d$, um polinômio mônico irredutível que divide $x^{q^d} - x$. Como $x^{q^d} - x$ é o produto de todos os polinômios do tipo $x - a$, $a \in \mathbb{F}_{q^d}$, existe um subconjunto $A \subseteq \mathbb{F}_{q^d}$ tal que $f = \prod_{a \in A}(x - a)$. Fixe $a \in A$ e consideremos

$$\frac{\mathbb{F}_q[x]}{\langle f \rangle} \cong \mathbb{F}_q(a),$$

onde $\mathbb{F}_q(a)$ é o menor subcorpo de $\mathbb{F}_{q^d}$ contendo a. Este corpo possui q^n elementos e $\mathbb{F}_{q^d}$ é uma extensão de $\mathbb{F}_q(a)$. Ora, sabemos (ver [19], Proposição 4, página 99) que

os graus dessas extensões devem satisfazer

$$[\mathbb{F}_{q^d} : \mathbb{F}_q] = [\mathbb{F}_{q^d} : \mathbb{F}_q(a)] \cdot [\mathbb{F}_q(a) : \mathbb{F}_q],$$

ou seja, $d = [\mathbb{F}_{q^d} : \mathbb{F}_q(a)] \cdot n$ e, portanto, $n \mid d$.

Suponhamos agora que $n \mid d$. Seja $\mathbb{F}_{q^n} \cong \mathbb{F}_q[x]/\langle f \rangle$ e $a^* \in \mathbb{F}_{q^n}$ uma raiz de f. Defina $e = q^{d-n} + q^{d-2n} + \cdots + 1$. Então $q^n - 1$ divide $q^d - 1$, visto que $q^d - 1 = (q^n - 1) \cdot e$. Segue também que $x^{q^n-1} - 1$ divide $x^{q^d-1} - 1$, pois

$$x^{q^d-1} - 1 = (x^{q^n-1} - 1) \cdot (x^{(q^n-1)\cdot(e-1)} + x^{(q^n-1)\cdot(e-2)} + \cdots + 1).$$

Multiplicando por x, obtemos que $x^{q^n} - x$ divide $x^{q^d} - x$. Pelo Pequeno Teorema de Fermat, $a^{q^n} \equiv a \mod q^n$, $\forall\ a \in \mathbb{F}_{q^n}$. Assim, $(x - a^*) \mid x^{q^d} - x$, e portanto, $(x - a^*)$ divide $mdc(f, x^{q^d} - x)$ em $\mathbb{F}_{q^d}[x]$. Logo, $mdc_{\mathbb{F}_{q^d}}(f, x^{q^d} - x) \neq 1$. Conforme a Proposição 2.8, segue-se que $mdc_{\mathbb{F}_q}(f, x^{q^d} - x) \neq 1$. Ora, como f é irredutível em $\mathbb{F}_q[x]$, segue-se que $mdc_{\mathbb{F}_q}(f, x^{q^d} - x) = f$, ou seja, $f \mid x^{q^d} - x$ em $\mathbb{F}_q[x]$. □

Abaixo descrevemos um simples algoritmo, baseado no teorema acima, que calcula a decomposição em graus distintos de um polinômio $f \in \mathbb{F}_q[x]$.

Algoritmo 2.10. *Fatoração de Graus Distintos*

Entrada: *$f \in \mathbb{F}_q[x]$, polinômio mônico e livre de quadrados, com grau n.*

Saída: *$(g_1, g_2, \ldots, g_k)$ a decomposição em graus distintos de f.*

1 $h_0 \leftarrow x$, $f_0 \leftarrow f$, $i \leftarrow 0$

2 ***repetir***

$i \leftarrow i + 1$

$h_i \leftarrow h_{i-1}^q$

$g_i \leftarrow mdc(h_i - x, f_{i-1}), \quad f_i \leftarrow \frac{f_{i-1}}{g_i}$

até $f_i = 1$

3 $k \leftarrow i$

4 **retorna** $(g_1, g_2, \ldots, g_k)$

Teorema 2.11. *O algoritmo* 2.10 *funciona corretamente e realiza* $\mathcal{O}(n^2 \log q)$ *operações no corpo* $\mathbb{F}_q$.

Demonstração. Ver [17], Teorema 14.4. □

2.2.3 Fatoração De Mesmo Grau - Algoritmo de Cantor-Zassenhaus

Para fatorarmos completamente o polinômio $f \in \mathbb{F}_q[x]$, resta apenas fatorar os blocos g_i resultantes do algoritmo da seção anterior. O desenvolvimento que daremos aqui restringe-se a polinômios sobre corpos finitos $\mathbb{F}_q[x]$, onde q é uma potência de um primo ímpar. Uma descrição de como podemos modificar os algoritmos aqui apresentados para tratar do caso em que $q = 2^k$, $k > 0$, pode ser encontrada nos exercícios do Capítulo 14 de [17].

Lema 2.12. *Seja* q *uma potência de um primo ímpar e* $\mathbb{F}_q^\times = \mathbb{F}_q \setminus \{0\}$. *Seja* k *um divisor de* $q-1$ *e* $S = \{b^k : b \in \mathbb{F}_q^\times\}$, *o conjunto das* k*-ésimas potências de* $\mathbb{F}_q^\times$. *Então*

i) S *é um subgrupo de* $\mathbb{F}_q^\times$ *de ordem* $(q-1)/k$.

ii) $S = \{a \in \mathbb{F}_q : a^{(q-1)/k} = 1\}$.

iii) Se $k = 2$, *então* $a^{(q-1)/2} \in \{-1, 1\}$, $\forall\, a \in \mathbb{F}_q^\times$.

Demonstração. Considere o homomorfismo σ_k no grupo multiplicativo de $\mathbb{F}_q$

$$\sigma_k : \mathbb{F}_q^\times \to \mathbb{F}_q^\times, \;\; b \to b^k.$$

Como S é a imagem de σ_k, segue-se que S é um subgrupo de $\mathbb{F}_q^\times$. O núcleo de σ_k é o conjunto $\{a \in \mathbb{F}_q^\times : a^k = 1\}$ das k-ésimas raízes da unidade. Como $\mathbb{F}_q$ é um corpo, o polinômio $x^k - 1$ possui no máximo k raízes em $\mathbb{F}_q$ e portanto,

$$|\ker(\sigma_k)| = |\{a \in \mathbb{F}_q^\times : a^k = 1\}| \leq k. \tag{2.2}$$

Como

$$(b^k)^{(q-1)/k} = b^{q-1} \equiv 1, \; \forall \, b \in \mathbb{F}_q^\times,$$

segue-se que

$$S \subseteq \ker(\sigma_{\frac{q-1}{k}}).$$

Assim,

$$|S| \leq |\ker(\sigma_{\frac{q-1}{k}})| \leq \frac{q-1}{k},$$

de acordo com (2.2). Pelo Teorema Fundamental dos Homomorfismos, segue-se que $\mathbb{F}_q^\times / \ker(\sigma_k) \cong Im(\sigma_k) = S$. Além disso, como $\mathbb{F}_q^\times$ tem ordem finita, segue-se que $|S| = |Im(\sigma_k)| = |\mathbb{F}_q^\times|/|\ker(\sigma_k)|$, ou seja, $|\mathbb{F}_q^\times| = |S| \cdot |\ker(\sigma_k)|$. Assim, temos

$$q - 1 = |\mathbb{F}_q^\times| = |\ker(\sigma_k)| \cdot |Im(\sigma_k)| = |\ker(\sigma_k)| \cdot |S| \leq k\frac{q-1}{k} = q - 1.$$

Logo,

$$|\ker(\sigma_k)| = k, \; |S| = (q-1)/k \;\; e \;\; S = \ker(\sigma_{(q-1)/k}),$$

o que mostra *i*) e *ii*).

Para demonstrar *iii*) note que, de acordo com *i*), para $k = (q-1)/2$, $S = \{b^{(q-1)/2} \; : \; b \in \mathbb{F}_q^\times\}$ é um subgrupo de $\mathbb{F}_q^\times$ de ordem 2. Logo $S = \{1, -1\}$.

□

Lembre que, devido à seção anterior, queremos fatorar um polinômio f de grau n tal que f é o produto de r fatores irredutíveis, $f_1, f_2, \ldots, f_r$, todos com mesmo grau d. Como f_i e f_j são polinômios irredutíveis e distintos, segue-se que $mdc(f_i, f_j) = 1$. Assim, segundo o Teorema Chinês do Restos, a aplicação

$$\sigma : \frac{\mathbb{F}_q[x]}{\langle f \rangle} \to \frac{\mathbb{F}_q[x]}{\langle f_1 \rangle} \times \frac{\mathbb{F}_q[x]}{\langle f_2 \rangle} \times \cdots \times \frac{\mathbb{F}_q[x]}{\langle f_r \rangle},$$

onde $\sigma(g \mod f) = (g \mod f_1, g \mod f_2, \ldots, g \mod f_r)$, é um isomorfismo. Defina as aplicações

$$\sigma_i : \mathbb{F}_q[x]/\langle f \rangle \to \mathbb{F}_q[x]/\langle f_i \rangle,\ \sigma_i(g \mod f) = g \mod f_i.$$

Note que, sendo f_i irredutível e $deg(f_i) = d$, temos $\mathbb{F}_q[x]/\langle f \rangle \cong \mathbb{F}_{q^d}$. Seja $g \in \mathbb{F}_q[x]$. Então $f_i \mid g$ se, e somente se, $g \mod f_i \equiv 0$, ou seja, $\sigma_i(g) = 0$. Assim, se conseguirmos encontrar um polinômio g tal que $\sigma_i(g) = 0$ para alguns valores de i, e $\sigma_i(g) \neq 0$ para outros, teremos encontrado um divisor não trivial de f, a saber, $mdc(g, f)$.

Definição 2.13. *Um polinômio $g \in \mathbb{F}_q[x]$ tal que $\sigma_i(g) = 0$ para alguns valores de i e $\sigma_i(g) \neq 0$ para outros é chamado de polinômio separador*[2] *de f.*

Veremos agora um procedimento probabilístico para encontrar um polinômio separador.

Defina $e = (q^d - 1)/2$. Pelos itens *i)* e *ii)* do Lema 2.12, sabemos que $S = \{b^2 : b \in \mathbb{F}_{q^d}[x]\}$ tem ordem $(q^d - 1)/2$ e que $S = \{b \in \mathbb{F}_{q^d}^{\times} : b^{(q^d-1)/2} = 1\}$. Ou seja, metade dos elementos de $\mathbb{F}_{q^d}^{\times}$, quando elevados a e, são 1. Por outro lado, pelo item *iii)* do mesmo lema, $\forall\ \beta \in \mathbb{F}_{q^d}^{\times}$, $\beta^e \in \{-1, 1\}$, ou seja, a outra metade dos elementos devem satisfazer $\beta^e = -1$.

Assim, se escolhermos um polinômio $g \in \mathbb{F}_q[x]$ com $deg(g) < n$ e $mdc(g, f) = 1$ aleatoriamente, então $\sigma_1(g), \sigma_2(g), \ldots, \sigma_r(g)$ são elementos aleató-

[2] *do Inglês, Splitting Polynomial*

rios de $\mathbb{F}_{q^d}^{\times}$. Além disso, $\epsilon_i := (\sigma_i(g))^e = \sigma_i(g^e)$ é 1 ou -1, ambos os casos com probabilidade $1/2$ de ocorrer. Então $\sigma(g^e - 1) = (\epsilon_1 - 1, \epsilon_2 - 1, \ldots, \epsilon_r - 1)$ e $g^e - 1$ é um polinômio separador de f a menos que $\epsilon_1 = \epsilon_2 = \cdots = \epsilon_r$. Este último caso acontece com probabilidade $2(1/2)^r = 2^{1-r} \leq 1/2$, visto que $r \geq 2$.

Exemplo 2.14. *Seja $f = x^6 + x^4 + x^2 + 1 \in \mathbb{Z}/3\mathbb{Z}[x]$ e suponha que sabemos que f é o produto de 3 fatores quadráticos. Para demonstrar o raciocínio acima, escolhemos "arbitrariamente" um polinômio $g \in \mathbb{Z}/3\mathbb{Z}[x]$, digamos $g = 2x + 1 \in \mathbb{Z}/3\mathbb{Z}[x]$. Neste caso, $r = 3$, $d = 2$ e $e = (3^2 - 1)/2 = 4$. Assim, o polinômio $g^4 - 1 = x^4 + 2x^3 + 2x \in \mathbb{Z}/3\mathbb{Z}[x]$ tem probabilidade $1 - 2(1/2)^r = 3/4$ de ser um polinômio separador para f. De fato, calculando*

$$mdc(f, g^4 - 1) = mdc(f, x^4 + 2x^3 + 2x) = x^2 + x + 2$$

obtemos um fator não trivial de f. Neste caso, como sabemos que todos os fatores irredutíveis de f têm grau 2, concluímos que $x^2 + x + 2$ é um fator irredutível de f.

Se, ao invés de $g = 2x + 1$, tivéssemos escolhido $g = x^2$, então

$$mdc(f, g^4 - 1) = mdc(f, x^8 - 1) = x^6 + x^4 + x^2 + 1,$$

o que não fornece um fator não trivial de f. Assim, $g = x^2$ não é um polinômio separador de f.

O algoritmo a seguir tenta encontrar um polinômio separador para f através das operações descritas acima. Caso ele encontre, este algoritmo retorna um fator de f. Do contrário, o algoritmo retorna uma mensagem de erro. Note que a probabilidade de encontrar um polinômio separador e portanto, um fator de f, é no mínimo $1/2$.

Algoritmo 2.15. *Encontrando um fator de f*

Entrada: *Um polinômio mônico $f \in \mathbb{F}_q[x]$ livre de quadrados, com grau $n > 0$, e um divisor d de n tal que todos os fatores de f possuem grau d.*

Saída: *$g \in \mathbb{F}_q[x]$, um fator mônico de f ou "Erro".*

1 *Escolha $h \in \mathbb{F}_q[x]$ aleatoriamente com $deg(h) < n$.*

2 ***se*** $h \in \mathbb{F}_q$ ***então***

retorna *"Erro"*

3 ***se*** $mdc(f, h) \neq 1$ ***então***

retorna $mdc(f, h)$

4 $g \leftarrow h^{(q^d-1)/2} - 1$

5 ***se*** $mdc(f, g) \neq 1$ *e* $mdc(f, g) \neq f$ ***então***

retorna $mdc(f, g)$

senão

retorna *"Erro"*

Note que $g = h^{(q^d-1)/2} - 1$ é o candidato a ser um polinômio separador para f.

Teorema 2.16. *O algoritmo* 2.15 *funciona corretamente. O algoritmo retorna "Erro" com probabilidade menor do que $2^{1-r} \leq 1/2$, onde $r = n/d \geq 2$ e realiza $\mathcal{O}(n^2 \log q)$ operações no corpo $\mathbb{F}_q$.*

Demonstração. Ver [17] Teorema 14.9. □

Executar o algoritmo k vezes faz com que a probabilidade do algoritmo falhar seja $\leq 2^{-k}$. Assim, com este algoritmo, podemos escrever um algoritmo para calcular a fatoração de f, satisfazendo as condições apresentadas no início desta seção.

Algoritmo 2.17. *Fatoração de Mesmo Grau*

Entrada: *$f \in \mathbb{F}_q[x]$, polinômio mônico e livre de quadrados, com grau $n > 0$, e um divisor d de n tal que todos os fatores irredutíveis de f possuem grau d.*
Saída: *Lista de fatores mônicos e irredutíveis de f.*

1 ***se*** $n = d$ ***então***
retorna f

2 *Chame o Algoritmo* 2.15 *com entradas f e d repetidamente até obter um fator próprio g de f*

3 *Chame este algoritmo recursivamente com entradas g e f/g*

4 ***retorna*** *os resultados das 2 chamadas recursivas do item* 3

Teorema 2.18. *Um polinômio livre de quadrados f com grau $n = rd$, com r fatores irredutíveis de grau d pode ser completamente fatorado com um número esperado de $\mathcal{O}(n^2 \log q)$ operações no corpo $\mathbb{F}_q$.*

Demonstração. Ver [17], Teorema 14.11. □

Calcular a parte livre de quadrados pode ser considerada trivial tanto em teoria quanto na prática. A decomposição em blocos cujos fatores irredutíveis possuem o mesmo grau, descrita no algoritmo 2.10, e o algoritmo que fatora cada um desses blocos, descrito pelo algoritmo 2.17, podem ser executados em tempo polinomial em $n = deg(f)$ e $\log q$. Enquanto que os dois primeiros algoritmos são

determinísticos, o último é probabilístico. Na próxima seção veremos um algoritmo determinístico para fatoração sobre corpos finitos.

2.3 Algoritmos Baseados em Álgebra Linear

Ao invés de executar a fatoração em graus distintos, estes métodos utilizam Álgebra Linear para encontrar os fatores irredutíveis de f. Os primeiros algoritmos modernos para a fatoração de polinômios sobre corpos finitos foram introduzidos por Berlekamp, em [6] e [7], na década de 60. Nestes trabalhos, Berlekamp explora o Teorema Chinês dos Restos para montar um sistema linear, cujas soluções fornecem fatores do polinômio f.

Além do método de Berlekamp, apresentaremos também o método de Niederreiter [33]. Este método constrói um sistema linear a partir de uma equação diferencial. Embora este método pareça muito distinto do método de Berlekamp, existem diversas ligações entre os sistemas lineares encontrados em ambos os casos [15].

2.3.1 Algoritmo de Berlekamp

Seja $f \in \mathbb{F}_q[x]$ um polinômio mônico, livre de quadrados, de grau n e sejam $f_1, f_2, \ldots, f_r \in \mathbb{F}_q[x]$ seus fatores mônicos e irredutíveis. Se denotarmos por $R = \mathbb{F}_q[x]/\langle f \rangle$ e $R_i = \mathbb{F}_q[x]/\langle f_i \rangle$, o Teorema Chinês dos Restos afirma que

$$R \cong R_1 \times R_2 \times \cdots \times R_r,$$

onde o isomorfismo acima é dado por

$$\begin{aligned} \sigma : R &\to R_1 \times R_2 \times \cdots \times R_r \\ g \; mod \; f &\mapsto (g \mod f_1, g \mod f_2, \ldots, g \mod f_r) \end{aligned}$$

Considere a aplicação de Frobenius em R

$$\Phi : R \to R$$
$$g \mod f \mapsto (g \mod f)^q$$

Como estamos trabalhando em $\mathbb{F}_q$, segue-se que $\Phi(g \mod f + h \mod f) = (g \mod f + h \mod f)^q = (g \mod f)^q + (h \mod f)^q = \Phi(g \mod f) + \Phi(h \mod f)$, $\forall g \mod f, h \mod f \in R$. Além disso, se $c \in \mathbb{F}_q$, então $\Phi(c \cdot g \mod f) = (c \cdot g \mod f)^q = c^q \cdot (g \mod f)^q = c \cdot (g \mod f)^q = c \cdot \Phi(g \mod f)$, devido ao Pequeno Teorema de Fermat. Ou seja, a aplicação Φ de Frobenius é $\mathbb{F}_q$-linear. Considere os pontos fixos de Φ, ou seja, o núcleo da aplicação $\Phi - I$, onde I é a aplicação identidade em R. Denote por $\mathcal{B}$ esse conjunto, ou seja

$$\mathcal{B} = \{g \mod f \in R \mid (g \mod f)^q = g \mod f\}.$$

Este conjunto é fequentemente chamado de *Subalgebra de Berlekamp*.

Assim, se $g \in \mathbb{F}_q[x]$, então

$$g \mod f \in \mathcal{B}$$

se, e somente se, $g^q \equiv g \mod f$. Pelo isomorfismo σ acima, isto ocorre se, e somente se,

$$g^q \equiv g \mod f_i,\ i = 1, 2, \ldots, r.$$

Se $g^q \equiv g \mod f_i$, então g é uma raiz do polinômio $x^q - x \in R_i[x] = \mathbb{F}_q[t]/\langle f_i(t)\rangle [x]$. Ora, como $a^q = a$ para todo $a \in \mathbb{F}_q$, segue-se que as únicas raízes de $x^q - x$ em $R_i[x]$ são os elementos de $\mathbb{F}_q$. Logo, $g \mod f_i$ deve ser um elemento de $\mathbb{F}_q$. Resumindo, $g \mod f \in \mathcal{B}$ se, e somente se,

$$g \mod f_i \in \mathbb{F}_q,\ i = 1, 2, \ldots, r.$$

Por outro lado, para constantes $b_1, b_2, \ldots, b_r \in \mathbb{F}_q$, existe, pelo Teorema Chinês dos Restos, um único polinômio $g \in \mathbb{F}_q[x]$ tal que $g \equiv b_i \mod f_i$ ou ainda,

pelo isomorfismo σ acima, $\sigma(g \mod f) = (b_1, b_2, \ldots, b_r)$. Além disso,

$$g^q \equiv b_i^q = b_i \equiv g \mod f_i,\ \forall i = 1, 2, \ldots, r,$$

logo $g^q \equiv g \mod f$ e, portanto, $g \mod f \in \mathcal{B}$.

Em outras palavras, acabamos de demonstrar que

$$\mathcal{B} \cong \mathbb{F}_q \times \mathbb{F}_q \times \cdots \times \mathbb{F}_q = (\mathbb{F}_q)^r, \tag{2.3}$$

ou seja, essa subálgebra, vista como um $\mathbb{F}_q$-subespaço de R, tem dimensão r sobre $\mathbb{F}_q$ e portanto, $\mathcal{B}$ possui q^r elementos.

Observe que, se $g \in \mathbb{F}_q[x]$ é tal que $g \mod f \in \mathcal{B}$, $g \equiv b_i \mod f_i, i = 1, 2, \ldots, r$ e $b_j = 0$, para algum $j \in \{1, 2, \ldots, r\}$, então f_j divide g. Ou seja, $f_j \mid mdc(f, g)$ e portanto $mdc(f, g)$ é um fator não trivial de f. Assim, os polinômios $g \in \mathbb{F}_q[x]$ tais que $g \mod f \in \mathcal{B}$ são ótimos candidatos para fornecerem fatores não triviais de f.

Corolário 2.19. *Seja $f = f_1 f_2 \cdots f_r \in \mathbb{F}_q[x]$, onde os polinômios $f_1, f_2, \ldots, f_r$ são mônicos, irredutíveis e coprimos e seja $g \in \mathbb{F}_q[x]$ tal que $g \mod f \in \mathcal{B}$. Então*

$$f = \prod_{\alpha \in \mathbb{F}_q} mdc(f, g - \alpha). \tag{2.4}$$

Demonstração. Como $g \mod f \in \mathcal{B}$, existem $b_1, b_2, \ldots, b_r \in \mathbb{F}_q$ tais que $g \equiv b_i \mod f_i$. Assim $(g - \alpha) \equiv 0 \mod f_i$ para certo valor de $\alpha \in \mathbb{F}_q$ (a saber, $\alpha = b_i$) e para algum $1 \leq i \leq r$. Logo f_i divide $mdc(f, g - b_i)$. Ou seja, para todo $i = 1, 2, \ldots, r$, existe α tal que f_i divide $mdc(f, g - \alpha)$. Como os polinômios f_i, $1 \leq i \leq r$, são relativamente primos, segue-se que f divide $\prod_{\alpha \in \mathbb{F}_q} mdc(f, g - \alpha)$.

Por outro lado, cada $mdc(f, g - \alpha) | f$. Além disso, $mdc(f, g - \alpha)$ e $mdc(f, g - \alpha')$, para $\alpha \neq \alpha'$, são relativamente primos, pois se $f_i | mdc(f, g - \alpha)$ e $f_i | mdc(f, g - \alpha')$, segue-se que f_i divide $g - \alpha$ e $g - \alpha'$. Então $g - \alpha \equiv 0 \mod f_i$

e $g - \alpha' \equiv 0 \mod f_i$, logo $g - \alpha \equiv g - \alpha' \ mod \ f_i$, mas $g \equiv b_i \mod f_i$. Assim, $b_i - \alpha \equiv b_i - \alpha' \mod f_i$, e como $b_i - \alpha, b_i - \alpha' \in \mathbb{F}_q$, segue-se que $b_i - \alpha = b_i - \alpha'$, ou seja, $\alpha = \alpha'$. Assim, $\prod_{\alpha \in \mathbb{F}_q} mdc(f, g - \alpha)$ divide f. Ora, como ambos os polinômios são mônicos, temos igualdade. □

Note que só teremos uma fatoração não trivial de $f \in \mathbb{F}_q[x]$ se $b_1, b_2, \ldots, b_r$ não forem todos iguais. Por outro lado, se todas as constantes $b_1, b_2, \ldots, b_r$ forem distintas, obteremos todos os fatores irredutíveis de f através do produtório (2.4) acima. Em outras palavras, qualquer polinômio g tal que $1 \leq deg(g) < deg(f)$ e $g \mod f \in \mathcal{B}$ produzirá um fator não trivial de f.

Precisamos, portanto, saber encontrar elementos em $\mathcal{B}$ para poder calcular os mdc's e obter uma fatoração de f. Além disso, não queremos qualquer elemento em $\mathcal{B}$, pois, como vimos acima, existem elementos que nos fornecem apenas uma fatoração trivial.

Vamos resolver este problema usando Álgebra Linear. O $\mathbb{F}_q$-espaço vetorial R possui uma base formada pelos vetores

$$x^{n-1} \mod f, \ \ x^{n-2} \mod f, \ldots, x \mod f, \ \ 1 \mod f.$$

Considere a matriz $Q \in \mathbb{F}_q^{n \times n}$ representando a transformação $\mathbb{F}_q$-linear de Frobenius Φ nesta base. Ou seja, Q é a matriz dada por

$$Q = \begin{pmatrix} q_{0,0} & \cdots & q_{0,n-1} \\ \vdots & & \vdots \\ q_{n-1,0} & \cdots & q_{n-1,n-1} \end{pmatrix}$$

onde as entradas $q_{i,j}$ são obtidas de

$$x^{qk} \equiv q_{k,n-1}x^{n-1} + q_{k,n-2}x^{n-2} + \cdots + q_{k,1}x + q_{k,0} \mod f, \ para \ 0 \leq k \leq n-1. \quad (2.5)$$

Isto é, as entradas da k-ésima linha de Q são os coeficientes do resto da divisão de x^{qk} por f.

Teorema 2.20. *Seja $g \in \mathbb{F}_q[x]$ um polinômio de grau $n-1$. Representando $g = g_{n-1}x^{n-1} + g_{n-1}x^{n-2} + \cdots + g_1x + g_0$ através do vetor linha $v_g = (g_0, g_1, \ldots, g_{n-1})$ temos*

$$g \mod f \in \mathcal{B} \Leftrightarrow (g \mod f)^q = g \mod f \Leftrightarrow v_g = v_gQ. \tag{2.6}$$

Demonstração. De fato, utilizando (2.5)

$$g^q = g(x)^q = g(x^q) = \sum_{k=0}^{n-1} g_k x^{qk} \equiv \sum_{k=0}^{n-1}\sum_{j=0}^{n-1} g_k q_{k,j} x^j = \sum_{j=0}^{n-1}\sum_{k=0}^{n-1} g_k q_{k,j} x^j \mod f$$

Em notação matricial, o último elemento desta sequência de igualdades é apenas v_gQ. Mas como $g^q \equiv g \mod f$, segue-se que $v_g = v_gQ$. Por outro lado, se $v_g = v_gQ$, temos

$$g = \sum_{j=0}^{n-1} g_j x^j = \sum_{j=0}^{n-1}\sum_{k=0}^{n-1} g_k q_{k,j} x^j \equiv \sum_{k=0}^{n-1} g_k x^{qk} = g(x^q) = g^q \mod f$$

e portanto, $g \mod f \in \mathcal{B}$. □

Ou seja, representando polinômios através de vetores, podemos determinar completamente o conjunto $\mathcal{B}$ através do complemento ortogonal do espaço coluna da matriz $Q - I$. De acordo com a equação (2.3), $\mathcal{B}$ possui q^r elementos. Logo, de acordo com (2.6), a equação $v_g = v_gQ$ possui r soluções linearmente independentes.

Considere uma base $v_1, v_2, \ldots, v_r$ do complemento ortogonal do espaço coluna de $Q-I$ e os respectivos polinômios $g_1, g_2, \ldots, g_r \in R$, que formam uma base para $\mathcal{B}$. Como o vetor $v_1 = (1, 0, \ldots, 0)$, correspondente ao polinômio $g = 1 \in \mathbb{F}_q[x]$, é sempre uma solução de

$$v_1 - v_1Q = (0, 0, \ldots, 0),$$

os vetores $v_2, v_3, \ldots, v_r$, cujos polinômios associados são $g_2, g_3, \ldots, g_r$, são tais que $1 < deg(g_i) \leq n-1$ e, portanto, todos eles fornecerão uma fatoração não trivial para f através da equação (2.4).

A seguir apresentamos um algoritmo baseado nestas ideias para fatorar um polinômio em $\mathbb{F}_q[x]$. Primeiramente calculamos os fatores obtidos através de $\prod_{\alpha\in\mathbb{F}_q} mdc(f, g_1 - \alpha)$. Se o número de fatores for menor do que r, então calculamos $\prod_{\alpha\in\mathbb{F}_q} mdc(h, g_2-\alpha)$ para todos os fatores h encontrados no passo anterior. Seguimos com esse raciocínio até obtermos os r fatores de f.

Algoritmo 2.21. *Algoritmo de Berlekamp*

Entrada: *Um polinômio mônico $f \in \mathbb{F}_q[x]$ livre de quadrados de grau n.*

Saída: *O conjunto L dos fatores irredutíveis de f.*

1 ***para*** $i = 0, \ldots, n-1$ ***fazer***

Calcule o resto da divisão de x^{iq} por f

2 *Monte a matriz Q*

3 *Calcule a dimensão r e uma base $v_1, v_2, \ldots, v_r$ do complemento ortogonal do espaço coluna de $Q - I$*

4 ***se*** $r = 1$ ***então***

retorna f

5 *Considere os polinômios $g_1, g_2, \ldots, g_r \in \mathbb{F}_q[x]$ correspondentes aos vetores da base*

6 *Seja L a lista dos mdc's de f e $g_1 - \alpha$, para $\alpha = 0, 1, \ldots, q-1$.*

7 ***enquanto*** $|L| \neq r$ ***fazer***

Escolha o próximo g_i da lista $g_1, g_2, \ldots, g_r$

$L \leftarrow \{mdc(h, g_i - \alpha),$ *para todo* $h \in L$ *e* $\alpha = 0, 1, \ldots q-1\}$

8 ***retorna*** L

Teorema 2.22. *O algoritmo 2.21 encontrará todos os fatores irredutíveis de f.*

Demonstração. De fato, considere dois fatores irredutíveis de f, digamos f_1 e f_2. Queremos demonstrar que os fatores f_1 e f_2 serão separados em algum momento do algoritmo. Como $g_1, g_2, \ldots, g_r \in \mathcal{B}$, segue-se que existem elementos $c_{j1}, c_{j2} \in \mathbb{F}_q,\ j = 1, 2, \ldots, r$, tais que

$$g_j \equiv c_{j1} \mod f_1 \ \ e \ \ g_j \equiv c_{j2} \mod f_2 \text{ para } j = 1, 2, \ldots, r.$$

Suponha, por contradição, que $c_{j1} = c_{j2},\ j = 1, 2, \ldots, r$. De acordo com (2.3), existe $g \in \mathcal{B}$ tal que

$$g \equiv 1 \ mod \ f_1 \ \ e \ \ g \equiv 0 \mod f_2. \tag{2.7}$$

Como $g_1, g_2, \ldots, g_r$ formam uma base de $\mathcal{B}$, segue-se que $g = \alpha_1 g_1 + \alpha_2 g_2 + \cdots + \alpha_r g_r$, para certos $\alpha_1, \alpha_2, \ldots, \alpha_r \in \mathbb{F}_q$. Logo, por (2.7),

$$g = \alpha_1 g_1 + \alpha_2 g_2 + \cdots + \alpha_r g_r \equiv \alpha_1 c_{11} + \alpha_2 c_{21} + \cdots + \alpha_r c_{r1} \equiv 1 \mod f_1$$

e

$$g = \alpha_1 g_1 + \alpha_2 g_2 + \cdots + \alpha_r g_r \equiv \alpha_1 c_{12} + \alpha_2 c_{22} + \cdots + \alpha_r c_{r2} \equiv 0 \mod f_2.$$

Denote $c_1 = \alpha_1 c_{11} + \alpha_2 c_{21} + \cdots + \alpha_r c_{r1}$ e $c_2 = \alpha_1 c_{12} + \alpha_2 c_{22} + \cdots + \alpha_r c_{r2}$. Como $c_1, c_2 \in \mathbb{F}_q$ e, segundo nossa hipótese, $c_1 = c_2$, segue-se que $1 = c_1 = c_2 = 0$, um absurdo. Assim, existe $1 \leq j^* \leq r$ mínimo tal que $c_{j^*1} \neq c_{j^*2}$. Isso significa que quando calcularmos $mdc(f, g_{j^*} - \alpha)$, para $\alpha = 0, 1, \ldots, q-1$, os fatores f_1 e f_2 serão separados. □

Teorema 2.23. *O algoritmo 2.21 fatora um polinômio f de grau n em um número esperado de $\mathcal{O}(n^3 + qrn^2)$ operações em $\mathbb{F}_q$, onde r é o número de fatores de f.*

Demonstração. Ver [26]. □

Quando p é pequeno, o número médio de fatores r é $ln(n)$. Acabaremos esta seção com um exemplo.

Exemplo 2.24. *Queremos fatorar o polinômio* $f = x^5 + x^4 + 2x^3 + 3x^2 + 2 \in \mathbb{Z}/5\mathbb{Z}[x]$. *O primeiro passo é montar a matriz* Q, *que é formada pelos coeficientes de* x^{qk} *mod* f, $0 \leq k < 5$*:*

$$\begin{aligned} x^0 &\equiv 1 \ mod\ f \\ x^5 &\equiv 4x^4 + 3x^3 + 2x^2 + 3 \ mod\ f \\ x^{10} &\equiv x^4 + 2x^2 + 3 \ mod\ f \\ x^{15} &\equiv 3x^4 + x^3 + 4x + 3 \ mod\ f \\ x^{20} &\equiv x^4 + 2x^3 + x^2 + 4x + 2 \ mod\ f \end{aligned}$$

Assim, a matriz Q *é dada por*

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

Em seguida, calculamos uma base para o complemento ortogonal do espaço coluna de $Q - I$. *Esta base é dada pelos vetores*

$$v_1 = (1, 0, 0, 0, 0) \ \ e \ \ v_2 = (0, 1, 2, 3, 1).$$

Estes vetores correspondem aos polinômios

$$g_1 = 1 \ \ e \ \ g_2 = x + 2x^2 + 3x^3 + x^4.$$

O próximo passo é o cálculo dos mdc's. Como $g_1 = 1$ *é uma constante, não obteremos nenhum fator não trivial de* f *com esse polinômio. Vamos agora*

calcular os mdc's com g_2. Temos

$$\begin{aligned} mdc(f, g_2 - 0) &= x^2 + 2 \\ mdc(f, g_2 - 1) &= 1 \\ mdc(f, g_2 - 2) &= 1 \\ mdc(f, g_2 - 3) &= x^3 + x^2 + 1 \\ mdc(f, g_2 - 4) &= 1. \end{aligned}$$

Como sabemos que a dimensão do complemento ortogonal do espaço coluna de $Q - I$ é 2 (e portanto, f tem dois fatores irredutíveis), e acabamos de encontrar 2 fatores de f, segue que estes dois fatores são necessariamente irredutíveis, e f se fatora como

$$f = (x^2 + 2) \cdot (x^3 + x^2 + 1).$$

Note que o Algoritmo de Berlekamp não é executado em tempo polinomial em n e $\log q$, pois sua complexidade é $\mathcal{O}(n^3 + qrn^2)$. O maior problema em aberto nessa área é encontrar um algoritmo determinístico que seja executado em tempo polinomial em n e $\log q$. Uma versão probabilística do algoritmo de Berlekamp pode ser encontrada em [7], cuja complexidade é $\mathcal{O}(n^3 + n \log q)$.

2.3.2 Algoritmo de Niederreiter

O sistema linear a ser resolvido origina-se, neste caso, da equação diferencial $y^{(p-1)} + y^p = 0$. Da mesma forma como no algoritmo de Berlekamp, os fatores do polinômio f são obtidos através do cálculo do máximo divisor comum de f e certos elementos no espaço nulo da matriz encontrada. Uma vantagem do algoritmo de Niederreiter em relação ao algoritmo de Berlekamp é a maior facilidade na hora de montar a matriz. Além disso, o algoritmo de Niederreiter tem uma melhor

performance em relação ao algoritmo de Berlekamp quando estamos trabalhando em $\mathbb{F}_p$, com $p \leq deg(f)$ e p não muito pequeno e em $\mathbb{F}_{2^k}$, para k relativamente pequeno. Algumas demonstrações nesta seção serão omitidas. Para mais detalhes, ver [29] e [33].

O ponto de partida desse método é a equação diferencial no corpo $\mathbb{F}_p(x)$ das funções racionais sobre o corpo $\mathbb{F}_p$

$$y^{(p-1)} + y^p = 0 \tag{2.8}$$

Como $L(y) := y^{(p-1)} + y^p$ é um operador linear, o conjunto das soluções de (2.8) forma um subespaço linear de $\mathbb{F}_p(x)$. Seja $y = h/g \in \mathbb{F}_p(x)$ e defina $deg(y) = deg(h) - deg(g)$. Se $y = h/g$ satisfaz a equação (2.8), então $deg(y) \leq -1$. Isso quer dizer que a solução da equação (2.8) é da forma h/g, com $deg(h) < deg(g)$.

Vamos procurar soluções para a equação (2.8) da forma $y = h/f$, com $f \in \mathbb{F}_p[x]$ fixo e $h \in \mathbb{F}_p[x], deg(h) < n$. O teorema a seguir mostra como isso pode nos ajudar a encontrar fatores do polinômio f.

Teorema 2.25. *Seja $f \in \mathbb{F}_p[x]$ um polinômio mônico, livre de quadrados, grau $n \geq 1$ e fatoração $f = f_1 f_2 \cdots f_r$, com $f_1, f_2, \ldots, f_r$ irredutíveis. Então $y = h/f$ é uma solução de (2.8) se, e somente se, y tem a forma*

$$y = \sum_{i=1}^{r} c_i \frac{f_i'}{f_i}, \quad com\ c_1, c_2, \ldots, c_r \in \mathbb{F}_p. \tag{2.9}$$

Demonstração. A demonstração desse teorema pode ser encontrada em [33] ou em [29]. □

Seja $J(h) := \{1 \leq j \leq r : c_j = 0\}$. Como

$$h = fy = \sum_{i=1}^{r} c_i f_i' \frac{f}{f_i} = \left(\prod_{j \in J(h)} f_j \right) \sum_{i=1}^{r} c_i f_i' \frac{f}{f_i \prod_{j \in J(h)} f_j},$$

temos que

$$mdc(f,h) = \prod_{j \in J(h)} f_j.$$

Assim, se encontrarmos uma solução y na forma (2.9), para a qual algum c_i for nulo, então obteremos um fator de f.

O conjunto dos polinômios h em $\mathbb{F}_p[x]$ tais que h/f é uma solução de (2.8) ainda forma um subespaço vetorial de $\mathbb{F}_p(x)$, e esse espaço pode ser descrito explicitamente, como veremos a seguir.

Resolver a equação (2.8) para y é equivalente a resolver para h a seguinte equação

$$f^p \left(\frac{h}{f}\right)^{(p-1)} = -h^p \tag{2.10}$$

Como $deg(h/f) \leq -1$ (e consequentemente $deg((h/f)^{(p-1)}) \leq -p$), o grau dos polinômios de ambos os lados da equação (2.10) é menor ou igual a $p(n-1)$. Além disso, o lado direito desta equação é um polinômio em x^p (pois estamos trabalhando em $\mathbb{F}_p$). Portanto, (2.10) vale se, e somente se, os coeficientes de x^{ip}, $0 \leq i \leq n-1$, de ambos os lados coincidem. Se $h = \sum_{i=0}^{n-1} a_i x^i$, isso nos fornece um sistema linear $n \times n$ nas incógnitas $a_0, a_1, \ldots, a_{n-1}$.

Seja $M_p(f)$ a matriz dos coeficientes proveniente do lado esquerdo de (2.10) e denote por h o vetor $(a_0, a_1, \ldots, a_{n-1})$. Como $h^p = \sum_{i=0}^{n-1} a_i x^{ip}$, esse sistema pode ser escrito como $M_p(f)h^T = -h^T$, ou ainda

$$(M_p(f) + I_n)\, h^T = 0^T, \quad \text{onde} \quad h = (a_0, a_1, \ldots, a_{n-1}) \in \mathbb{F}_p^n. \tag{2.11}$$

Ou seja, o espaço vetorial das soluções h de (2.10) é o espaço nulo da matriz $M := M_p(f) + I_n$.

Teorema 2.26. *O número de fatores irredutíveis r de f pode ser calculado usando a matriz $M_p(f)$ da seguinte forma*

$$r = n - Posto\left(M_p(f) + I_n\right).$$

Demonstração. Ver [33]. □

Vamos a seguir fazer os cálculos para $p = 2$. Neste caso particular temos várias simplificações, o que permite calcular a matriz $M_p(f)$ explicitamente.

Sobre $\mathbb{F}_2$, a equação (2.10) torna-se

$$h^2 = -h^2 = f^2\left(\frac{h}{f}\right)' = f^2\frac{h'f - hf'}{f^2} = h'f + hf' = (hf)'. \tag{2.12}$$

Se $h = \sum_{i=0}^{n-1} a_i x^i$ e $f = \sum_{i=0}^{n} b_i x^i$, então

$$fh = \sum_{k=0}^{2n-1}\left(\sum_{\substack{i+j=k\\ i,j\geq 0}} a_i b_j\right) x^k.$$

Derivando e levando em consideração que estamos trabalhando em $\mathbb{F}_2$, temos

$$(fh)' = \sum_{\substack{k=1\\ k\ impar}}^{2n-1}\left(\sum_{\substack{i+j=k\\ i,j\geq 0}} a_i b_j\right) x^{k-1}.$$

Fazendo uma substituição de índices $k = 2l + 1$, temos

$$(fh)' = \sum_{l=0}^{n-1}\left(\sum_{\substack{i+j=2l+1\\ i,j\geq 0}} a_i b_j\right) x^{2l}.$$

Por outro lado,

$$h^2 = \sum_{i=0}^{n-1} a_i x^{2i}.$$

Logo, pela equação (2.12), obtemos as seguintes equações

$$\sum_{\substack{i+j=2l+1\\ i,j\geq 0}} a_j b_i = a_l, \quad 0 \leq l \leq n-1.$$

ou, em notação matricial,

$$\begin{pmatrix} b_1 & b_0 & 0 & 0 & \dots & 0 & 0 & 0 \\ b_3 & b_2 & b_1 & b_0 & \dots & 0 & 0 & 0 \\ b_5 & b_4 & b_3 & b_2 & \dots & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & 0 & \dots & b_n & b_{n-1} & b_{n-2} \\ 0 & 0 & 0 & 0 & \dots & 0 & 0 & b_n \end{pmatrix} \cdot \begin{pmatrix} a_0 \\ a_1 \\ a_2 \\ \vdots \\ a_{n-2} \\ a_{n-1} \end{pmatrix} = \begin{pmatrix} a_0 \\ a_1 \\ a_2 \\ \vdots \\ a_{n-2} \\ a_{n-1} \end{pmatrix}.$$

Podemos ver assim que montar a matriz $M_p(f)$ é muito mais simples neste caso do que montar a matriz de Berlekamp. Aqui, conseguimos montar a matriz apenas olhando para os coeficientes do polinômio f, enquanto que para montar a matriz de Berlekamp precisamos calcular o resto da divisão de x^{iq} por f.

Apesar de parecerem métodos muito distintos há várias conexões entre os algoritmos de Berlekamp e Niederreiter. Por exemplo, após aplicar uma certa transformação, o espaço das soluções do sistema de Niederreiter coincide com o espaço das soluções do sistema de Berlekamp. Isso permite explorar os aspectos que cada espaço de soluções tem de melhor e implementar um algoritmo que herdará as vantagens de ambos espaços de soluções. Para maiores detalhes, veja [15].

Terminaremos esta seção com um exemplo ilustrando a simplicidade deste método para o caso $p = 2$.

Exemplo 2.27. *Considere o polinômio* $f = x^5 + x^3 + x^2 + x \in \mathbb{F}_2[x]$. *A matriz* $M_2(f)$, *como descrita acima, é dada por*

$$M_2(f) = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 1 & 1 \\ 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 \end{pmatrix}.$$

O espaço nulo de $M_2(f) + I_5$ *é gerado pelos vetores*

$$v_1 = (0, 1, 0, 0, 0), \;\; v_2 = (1, 0, 1, 1, 0) \;\; e \;\; v_3 = (1, 0, 1, 0, 1),$$

que correspondem aos polinômios

$$g_1 = x, \;\; g_2 = 1 + x^2 + x^3 \;\; e \;\; g_3 = 1 + x^2 + x^4.$$

Vimos que $mdc(f, h) = \prod_{j \in J(h)} f_j$ *é um fator de* f *se* h *satisfaz a equação* (2.10) *ou ainda, por* (2.11)*, se o vetor* h *pertence, neste caso, ao núcleo de* $M_2(f) + I_5$*. Ora, acabamos de demonstrar que esse núcleo é gerado pelos vetores correspondentes aos polinômios* $g_1 = x$*,* $g_2 = 1 + x^2 + x^3$ *e* $g_3 = 1 + x^2 + x^4$*. Assim,* h *fornecerá um fator de* f *se, e somente se,* h *é uma combinação linear de* g_1, g_2 *e* g_3*, com coeficientes em* $\mathbb{F}_2$*. Para este exemplo, os fatores irredutíveis de* f *são dados pelo máximo divisor comum de* f *com os seguintes polinômios:* $h = g_1$*,* $h = g_2$ *e* $h = g_1 + g_2$*. Sabemos que* f *possui* $r = n - Posto(M_2(f) + I_5) = 5 - 2 = 3$ *fatores irredutíveis, de acordo com o Teorema* 2.26*. Além disso,*

$$mdc(f, g_1) = x, \;\; mdc(f, g_2) = 1 + x^2 + x^3 \;\; e \;\; mdc(f, g_1 + g_2) = x + 1$$

são polinômios irredutíveis. Logo, a fatoração de f *em* $\mathbb{F}_2[x]$ *é dada por*

$$f = x(x + 1)(1 + x^2 + x^3).$$

3 FATORANDO POLINÔMIOS SOBRE $\mathbb{Z}$ E $\mathbb{Q}$

3.1 Introdução

Neste capítulo apresentaremos três tipos de algoritmos para fatorar polinômios sobre o anel dos inteiros e sobre o corpo dos números racionais. O primeiro destes algoritmos calcula uma fatoração módulo p, onde p é um primo suficientemente grande. A partir dessa fatoração em $\mathbb{Z}/p\mathbb{Z}[x]$, encontram-se os fatores de f em $\mathbb{Z}[x]$ ou $\mathbb{Q}[x]$. O segundo algoritmo calcula uma fatoração módulo p, onde p é um primo "pequeno" e depois aplica o Levantamento de Hensel para obter uma fatoração módulo p^a, onde a potência p^a é suficientemente grande, ver [38]. Da mesma forma que o algoritmo anterior, essa fatoração módulo p é utilizada para encontrar a fatoração de f em $\mathbb{Z}[x]$ ou $\mathbb{Q}[x]$.

Finalmente, o terceiro algoritmo utiliza a teoria dos reticulados de inteiros, de Hermann Minkowski, e as ideias de Lenstra, Lenstra e Lovász para evitar o problema combinatório encontrado nos algoritmos anteriores, ver [28]. Este último algoritmo, chamado de LLL, tem importância fundamental na história da fatoração polinomial por ser o primeiro algoritmo determinístico a fatorar polinômios com coeficientes inteiros em tempo polinomial.

Como vimos anteriormente, muitos algoritmos consideram simplificações nos polinômios a serem fatorados. Uma destas simplificações é considerar apenas polinômios mônicos. Se o polinômio não for mônico, poderíamos realizar a substituição a seguir. Dado $f = a_n x^n + a_{n-1}x^{n-1} + \cdots + a_1 x + a_0 \in \mathbb{Z}[x]$ tal que $a_n \neq 1$, definimos o seguinte polinômio

$$g(x) = a_n^{n-1} \cdot f\left(\frac{x}{a_n}\right).$$

É fácil ver que $g(x)$ é mônico. Sejam $g_1, g_2, \ldots, g_r$ os fatores de g com multiplicidades $e_1, e_2, \ldots, e_r$, respectivamente. Podemos encontrar os fatores de f realizando uma substituição inversa, ou seja

$$f(x) = \frac{1}{a_n^{n-1}} \cdot g_1^{e_1}(a_n x) \cdot g_2^{e_2}(a_n x) \cdots g_r^{e_r}(a_n x).$$

Por exemplo, considere o polinômio $f(x) = 2x^2 + 3x + 1$. As substituições fornecem o polinômio $g(x) = 2^1 f(x/2) = x^2 + 3x + 2$. Este polinômio se fatora como $g(x) = (x + 2) \cdot (x + 1)$. Realizando a mudança inversa, obtemos $f(x) = \frac{1}{2^1}(2x + 1) \cdot (2x + 2) = (x + 1) \cdot (2x + 1)$.

Assim, bastaria desenvolver um algoritmo que fatorasse apenas polinômios mônicos. O problema dessa substituição é que o polinômio $g(x)$ tem coeficientes muito maiores do que os coeficientes do polinômio original f, especialmente se os coeficientes ou o grau de f forem grandes. Este problema é, na verdade, tão grave que este tipo de substituição não é utilizada na prática.

Antes de iniciarmos nosso estudo propriamente dito, vamos analisar a relação entre fatoração de polinômios em $\mathbb{Z}[x]$ e em $\mathbb{Q}[x]$. De uma forma geral, seja I um domínio de fatoração única e K o corpo quociente de I.

Definição 3.1. *O conteúdo $cont(f)$ de um polinômio $f = a_n x^n + a_{n-1} x^{n-1} + \cdots + a_1 x + a_0 \in I[x]$ é definido como*

$$cont(f) = mdc(a_n, a_{n-1}, \ldots, a_1, a_0).$$

Se $f = c \in I$, então $cont(f) = |c|$. Dizemos que um polinômio $f \in I[x]$ é primitivo se $cont(f) = 1$. Definimos a parte primitiva $pp(f)$ de $f \in I[x]$ por

$$pp(f) = \frac{f}{cont(f)}.$$

Dado um domínio de fatoração única I, um elemento $u \in I$ é chamado de *unidade* ou *elemento inversível*, se u possuir um inverso em I. Dois elementos

$a, b \in I$ são *associados* se $a = ub$, onde $u \in I$ é um elemento inversível. Por exemplo, em $\mathbb{Z}$, os únicos elementos inversíveis são 1 e -1.

Teorema 3.2 (Lema de Gauss)**.** *Se f e g são polinômios primitivos sobre I, então fg também é primitivo.*

Demonstração. Seja $f = \sum_{i=0}^{m} a_i x^i$ e $g = \sum_{i=0}^{n} b_i x^i$. Como f e g são primitivos, para qualquer primo p existem índices minimais j e k tais que p não divide nem a_j e nem b_k. Pelo fato de j e k serem minimais, segue que uma parcela do coeficiente de x^{j+k} em fg não é divisível por p, a saber, $a_j b_k$, enquanto todas as outras parcelas são divisíveis por p. Logo, o coeficiente de x^{j+k} não é divisível por p. Como p é qualquer, segue que fg é primitivo. □

Corolário 3.3. *Se $f \in I[x]$ é primitivo e irredutível em $I[x]$, então f também será irredutível em $K[x]$.*

Demonstração. Suponha que f seja irredutível sobre I mas redutível sobre K, com fatoração $f = f_1 f_2$. Como K é o corpo de frações de I, segue que existem $a \in K$ e $f_1', f_2' \in I[x]$ polinômios primitivos tais que $f = a f_1' f_2'$. Ora, como f_1' e f_2' são primitivos, o Lema de Gauss nos diz que $f_1' f_2'$ também o é. Assim,

$$1 = cont(f) = cont(a f_1' f_2') = a,$$

ou seja, $a = 1$ e portanto $f = f_1' f_2'$ nos fornece uma fatoração não trivial de f em I, um absurdo. □

Assim, para fatorar polinômios com coeficientes racionais, basta encontrar um algoritmo que fatore polinômios com coeficientes inteiros. As próximas seções apresentam 3 algoritmos que resolvem esse problema.

3.2 Fatoração mod p e o Levantamento de Hensel

É possível encontrar uma fatoração para polinômios com coeficientes inteiros sem executar uma fatoração módulo p, para algum primo p. Um exemplo disso é o algoritmo de fatoração de Schubert-Kronecker, descrito brevemente no capítulo introdutório. Porém, a maioria desses métodos são pouco eficientes na prática. Uma melhor abordagem é considerar o polinômio com coeficientes inteiros módulo um primo p. Visto que conhecemos algoritmos eficientes para fatorar polinômios sobre corpos finitos, a ideia é, então, calcular uma fatoração de $f \ mod \ p$ e, de alguma forma, utilizar essa fatoração para encontrar os fatores de f em $\mathbb{Z}[x]$. A figura abaixo ilustra esse esquema. O passo 1 consiste apenas em considerar os coeficientes do polinômio f módulo um primo p. O passo 2 pode ser realizado utilizando os algoritmos do capítulo anterior. Resta apenas encontrarmos um método para realizar o passo 3.

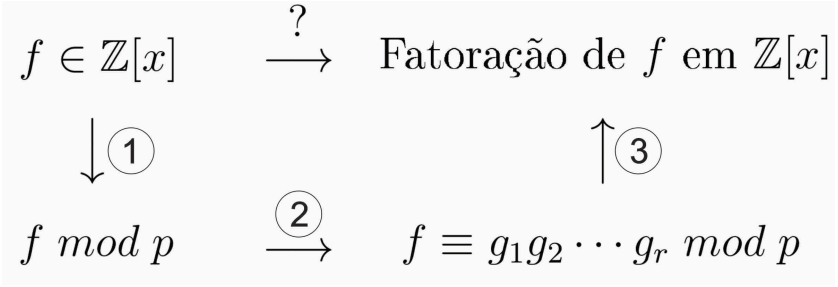

Figura 3.1: Esquema para fatoração em $\mathbb{Z}[x]$.

Nesta seção apresentaremos dois algoritmos para recuperar os fatores de f em $\mathbb{Z}[x]$ utilizando os fatores de $f \ mod \ p$. O primeiro deles consiste em encontrar um primo apropriado p suficientemente grande, enquanto o segundo método consiste em aplicar uma técnica chamada Levantamento de Hensel.

3.2.1 Fatoração módulo um primo grande

Seja $f \in \mathbb{Z}[x]$ livre de quadrados, não necessariamente mônico. Queremos, como sempre, encontrar a fatoração de f em fatores irredutíveis em $\mathbb{Z}[x]$. Como discutido acima, a estratégia é fatorar $f \ mod \ p$, para algum primo p e, de alguma forma, recuperar os fatores de f em $\mathbb{Z}[x]$.

Neste ponto, surgem algumas perguntas. Por exemplo, qual primo devemos tomar?

Para termos algo mais próximo possível de uma correspondência biunívoca entre os fatores de $f \ mod \ p$ e os fatores de f, gostaríamos que o polinômio $f \ mod \ p$ tivesse o mesmo grau que o polinômio f. Assim, p não deve dividir o coeficiente líder de f. Além disso, visto que f é livre de quadrados, gostaríamos que $f \ mod \ p$ também fosse livre de quadrados. O teorema a seguir nos diz quais primos satisfazem essas propriedades.

Definição 3.4. *Dados os polinômios* $f = f_0 + f_1x + \ldots + f_nx^n$ *e* $g = g_0 + g_1x + \ldots + g_mx^m$, *definimos o resultante de* f *e* g, *denotado por* $res(f, g)$, *como o determinante da seguinte matriz*

$$\begin{pmatrix}
f_n & & & & g_m & & & & & \\
f_{n-1} & f_n & & & g_{m-1} & g_m & & & & \\
\vdots & \vdots & \ddots & & \vdots & \vdots & \ddots & & & \\
\vdots & \vdots & & f_n & g_1 & \vdots & & \ddots & & \\
\vdots & \vdots & & f_{n-1} & g_0 & \vdots & & & \ddots & \\
\vdots & \vdots & & \vdots & & g_0 & & & & g_m \\
f_0 & \vdots & & \vdots & & & \ddots & & & \vdots \\
 & f_0 & & \vdots & & & & \ddots & & \vdots \\
 & & \ddots & \vdots & & & & & \ddots & \vdots \\
 & & & f_0 & & & & & & g_0
\end{pmatrix}.$$

Teorema 3.5. *Seja $f \in \mathbb{Z}[x]$ um polinômio mônico e livre de quadrados e $p \in \mathbb{N}$ um primo tal que $p \nmid lc(f)$. Denote o discriminante de f por $disc(f) = res(f, f')$, onde $res(f, f')$ é o resultante dos polinômios f e f'. Então $f \bmod p$ é livre de quadrados se, e somente se, $p \nmid disc(f)$.*

Demonstração. Ver [17], Lema 15.1. □

Assim, para termos certeza de que $f \ mod \ p$ seja livre de quadrados, basta escolher p tal que $p \nmid disc(f)$.

Suponha escolhido o primo p tal que $f \ mod \ p$ é livre de quadrados e a fatoração de f em $\mathbb{Z}[x]$ seja $f = gh$, onde cada um desses polinômios tem coeficientes inteiros. Por outro lado, suponha que, fatorando $f \ mod \ p$, obtemos a fatoração

$$f \ mod \ p = (g \ mod \ p) \cdot (h \ mod \ p).$$

Como podemos recuperar g e h através de $g \ mod \ p$ e $h \ mod \ p$? Para fazer isso iremos utilizar a representação simétrica dos elementos de $\mathbb{Z}/p\mathbb{Z}$, ou seja,

$$\mathbb{Z}/p\mathbb{Z} = \left\{ -\frac{p-1}{2}, -\frac{p-1}{2} + 1, \ldots, -1, 0, 1, \ldots, \frac{p-1}{2} - 1, \frac{p-1}{2} \right\}.$$

A figura abaixo mostra como essa representação é feita.

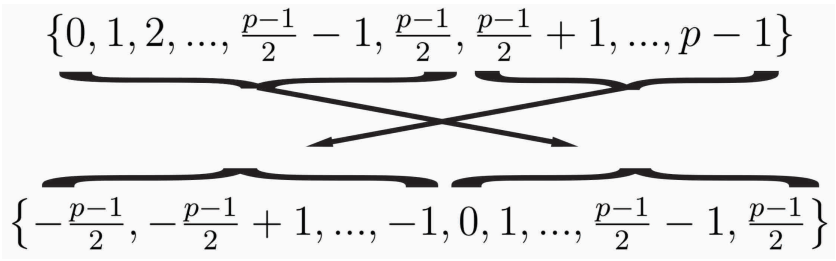

Figura 3.2: Representação simétrica dos elementos de $\mathbb{Z}/p\mathbb{Z}$.

Dessa forma, se os coeficientes de g e h são menores, em módulo, do que $p/2$, então

$$g \ mods \ p = g \ \ e \ \ h \ mods \ p = h,$$

onde *mods* significa que estamos considerando os coeficientes de $f \ mod \ p$ na representação simétrica de $\mathbb{Z}/p\mathbb{Z}$. Dessa forma, podemos recuperar os polinômios g e h através de $g \ mods \ p$ e $h \ mods \ p$.

Por exemplo, considere o polinômio $f = 3x^3 + 5x^2 - 2x + 7$ e $p = 23$. Neste caso, todos os coeficientes de f são menores, em módulo, do que $p/2$. Assim, $f \ mod \ p = 3x^3 + 5x^2 + 21x + 7$, mas

$$f \ mods \ p = 3x^3 + 5x^2 - 2x + 7 = f.$$

A próxima questão refere-se à magnitude do primo p. Para recuperar o fator g de f através de $g \ mod \ p$, precisamos obter um primo p tal que $p/2$ seja maior do que o módulo de qualquer coeficiente de qualquer fator racional de f. O teorema a seguir nos diz quão grande p deve ser.

Definição 3.6. *Seja* $f = a_0 + a_1x + \cdots + a_nx^n \in \mathbb{Z}[x]$. *Definimos as normas* $\| \ \|_2$, $\| \ \|_1$ *e* $\| \ \|_\infty$ *de* f *como*

$$\|f\|_1 = \sum_{i=1}^{n} |a_i|, \quad \|f\|_2 = \left(\sum_{i=1}^{n} |a_i|^2 \right)^{\frac{1}{2}} \quad e \quad \|f\|_\infty = \max_{i=1,\ldots,n} \{|a_i|\}.$$

Teorema 3.7 (Cota de Mignotte)**.** *Sejam* $f, g, h \in \mathbb{Z}[x]$ *tais que* $deg(f) = n \geq 1$, $deg(g) = m$ *e* $deg(h) = k$. *Se* gh *divide* f *em* $\mathbb{Z}[x]$, *então*

i) $\|g\|_\infty \|h\|_\infty \leq \|g\|_2 \|h\|_2 \leq \|g\|_1 \|h\|_1 \leq 2^{m+k} \|f\|_2 \leq (n+1)^{\frac{1}{2}} 2^{m+k} \|f\|_\infty$.

ii) $\|h\|_\infty \leq \|h\|_2 \leq 2^k \|f\|_2 \leq 2^k \|f\|_1$ *e* $\|h\|_\infty \leq \|h\|_2 \leq (n+1)^{\frac{1}{2}} 2^k \|f\|_\infty$.

Demonstração. Para a demonstração do teorema e mais detalhes sobre este resultado, ver [17], seção 6.6, e [30]. □

Em particular, se $h \in \mathbb{Z}[x]$ for um fator de f, então a segunda parte do item *ii*) deste teorema afirma que o maior dos coeficientes de h é limitado, em

módulo, por $B = (n+1)^{\frac{1}{2}} \cdot 2^n \cdot \|f\|_\infty$. Assim, tomando $2B < p < 4B$, podemos recuperar o fator $h \in \mathbb{Z}[x]$ a partir de $h \; mods \; p$. Vejamos um exemplo.

Exemplo 3.8. *Seja* $f = x^3 + 4x - 5$ *e* $p = 167$. *Note que* p *é maior que* $2B$ *o qual, neste caso, é* $2B = 160$. *Fatorando* $f \; mod \; 167$ *obtemos*

$$f \equiv (x + 166) \cdot (x^2 + x + 5) \; mod \; 167.$$

De acordo com nosso raciocínio acima, representando esses fatores com coeficientes em $\{-(p-1)/2, -(p-1)/2+1, \ldots, -1, 0, 1, \ldots, (p-1)/2-1, (p-1)/2\}$, *obteremos os fatores de* f *em* $\mathbb{Z}[x]$. *Esses fatores são* $x-1$ *e* x^2+x+5. *Logo,* f *se fatora como*

$$f = (x - 1) \cdot (x^2 + x + 5).$$

Porém, o problema deste raciocínio é que nem sempre os polinômios $g \; mod \; p$ e $h \; mod \; p$ são irredutíveis. Vejamos o seguinte exemplo.

Exemplo 3.9. *Considere o polinômio* $f = x^4 - 3x^3 + 2x^2 - 9x + 9$. *Como observado acima, os coeficientes de qualquer fator racional de* f *são limitados, em módulo, por*

$$B = (n+1)^{\frac{1}{2}} \cdot 2^n \cdot \|f\|_\infty = \sqrt{5} \cdot 2^4 \cdot 9 \approx 321,9.$$

Escolha p *tal que* $2B < p < 4B$. *Por exemplo,* $p = 647$. *Para este primo* p, *temos a fatoração*

$$f \; mod \; p = (x + 646) \cdot (x + 644) \cdot (x + 549) \cdot (x + 99).$$

Vamos verificar se f *possui fatores lineares em* $\mathbb{Z}[x]$. *Como afirmado acima, precisamos calcular o resto simétrico módulo* p *dos fatores lineares de* $f \; mod \; p$. *Estes fatores são*

$$(x - 1), \; (x - 3), \; (x - 98) \; e \; (x + 99).$$

Todos esses fatores possuem coeficientes limitados pela cota B *porém, nem todos eles são fatores de* f *em* $\mathbb{Z}[x]$. *Os dois últimos fatores* $x-98$ *e* $x+99$ *não são fatores de* f *em* $\mathbb{Z}[x]$. *Assim, os únicos fatores lineares de* f *são* $x - 1$ *e* $x - 3$. *E os fatores*

restantes? Como sobraram dois fatores, e estes não são fatores lineares, eles devem corresponder necessariamente à um fator quadrático de f. De fato,

$$(x - 98) \cdot (x + 99) = x^2 + 648x + 54351$$

que, depois de calcular o resto simétrico de seus coeficientes, fornece o polinômio $x^2 + x + 3$, *que é um fator irredutível de f em $\mathbb{Z}[x]$. Assim,*

$$f = (x - 1) \cdot (x - 3) \cdot (x^2 + x + 3).$$

Ou seja, em geral, precisamos combinar os fatores modulares de f *mod* p para obter os fatores de f em $\mathbb{Z}[x]$. Dessa forma, temos a seguinte situação.

Dado um polinômio $f \in \mathbb{Z}[x]$, podemos usar os algoritmos do capítulo anterior para obter uma fatoração de f *mod* p. Denote por $f_1, f_2, \ldots, f_r$ os fatores irredutíveis de f em $\mathbb{Z}[x]$ e por $g_1, g_2, \ldots, g_k$ os fatores modulares irredutíveis de f *mod* p. Então podemos recuperar qualquer fator $f_i \in \mathbb{Z}[x]$ de f usando $g_1, g_2, \ldots, g_k$ do seguinte modo. Se p é suficientemente grande, então para cada fator $f_i \in \mathbb{Z}[x]$ de f, existe um subconjunto S_i de $\{g_1, g_2, \ldots, g_k\}$ tal que

$$\frac{lc(f)}{lc(f_i)} f_i \equiv lc(f) \prod_{g_j \in S_i} g_j \; mods \; p \tag{3.1}$$

onde esta congruência é, na verdade, uma igualdade em $\mathbb{Z}[x]$. A figura a seguir mostrar como o esquema anterior fica.

Aqui devemos tomar um cuidado extra com os coeficientes quando f não é um polinômio mônico. A maioria dos programas de computação algébrica, tais como *Maple*, quando calculam f *mod* p, transformam esse polinômio em um polinômio mônico, visto que todo elemento em $\mathbb{Z}/p\mathbb{Z}$ possui um inverso. Logo, quando calculamos uma fatoração de f *mod* p, perdemos o coeficiente líder de f. Por causa disso, precisamos guardar o coeficiente líder do polinômio f. Além disso, como não conhecemos o coeficiente líder de cada um dos fatores f_i, $1 \leq i \leq r$, e o

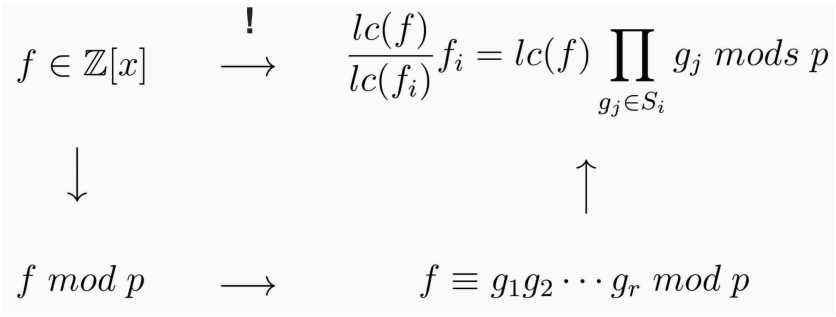

Figura 3.3: Esquema para fatoração em $\mathbb{Z}[x]$ - completo.

produto $\prod_{g_j \in S_i} g_j \ mods \ p$ resulta em um polinômio mônico, multiplicamos ambos os lados da equação (3.1) por $lc(f)$. Dessa forma, resolvemos o problema da perda do coeficiente líder do polinômio f, caso este não seja mônico.

Abaixo, descrevemos o algoritmo baseado nestas ideias para fatorar polinômios com coeficientes inteiros.

Algoritmo 3.10. *Algoritmo para fatoração em $\mathbb{Z}[x]$ - versão primo grande*

Entrada: *$f \in \mathbb{Z}[x]$ polinômio livre de quadrados e de grau n.*

Saída: *$f_1, f_2, \ldots, f_r \in \mathbb{Z}[x]$, os fatores irredutíveis de f.*

1 ***se*** $n = 1$ ***então***

retorna *"f irredutível"*

2 $A \leftarrow \|f\|_\infty, \quad b \leftarrow lc(f), \quad B \leftarrow (n+1)^{1/2} \cdot 2^n \cdot A \cdot b$

3 ***repetir***

escolha um primo p tal que $2B < p < 4B$

até $p \nmid disc(f, f')$

4 $\overline{f} \leftarrow f \ mod \ p$

5 Fatore $\overline{f} \in \mathbb{F}_p[x]$ e denote seus fatores modulares mônicos e irredutíveis por $g_1, g_2, \ldots, g_k$ tais que $f \equiv b g_1 g_2 \cdots g_k \ mod \ p$

6 $T \leftarrow \{1, 2, \ldots, k\}, \quad G \leftarrow \emptyset, \quad s \leftarrow 1, \quad f^* \leftarrow f$

7 **enquanto** $2s \leq |T|$ **fazer**

8 **para** todo subconjunto $S \subseteq T$ tal que $|S| = s$ **fazer**

9 Calcule $g^*, h^* \in \mathbb{Z}[x]$ tais que

$g^* = b \prod_{i \in S} g_i \ mods \ p \quad$ e $\quad h^* = b \prod_{i \notin S} g_i \ mods \ p$

10 **se** $\|g^*\|_1 \cdot \|h^*\|_1 \leq B$ **então**

$T \leftarrow T \setminus S, \quad G \leftarrow G \cup \{pp(g^*)\}$

$f^* \leftarrow pp(h^*), \quad b \leftarrow lc(f^*)$

termine o para e vá para o passo 7.

11 $s \leftarrow s + 1$

12 **retorne** $G \cup \{f^*\}$

O ***para*** no passo 8 do algoritmo 3.10 é realizado para todos os subconjuntos S de T. Essa é a parte que faz com que o algoritmo tenha complexidade exponencial: combinar todos os fatores de $f \ mod \ p$ para encontrar os fatores de f em $\mathbb{Z}[x]$. Na prática, este não é um problema tão grave pois, em geral, o número de fatores $mod \ p$ é menor do que o grau de f. O pior caso acontece quando f é irredutível mas se fatora completamente em fatores lineares em $\mathbb{Z}/p\mathbb{Z}[x]$. Isso significa que iremos combinar $n = deg(f)$ fatores de todas as formas possíveis para, no final, descobrir que f é, na verdade, irredutível.

A condição do passo 10 do algoritmo é válida se, e somente se, $g^*h^* = bf^*$, e esta igualdade é verificada em $\mathbb{Z}[x]$. De fato, se $g^*h^* = bf^*$, então pelo item i) do teorema 3.7, temos

$$\|g^*\|_1 \cdot \|h^*\|_1 \leq (n+1)^{\frac{1}{2}} \cdot 2^{m+k} \cdot \|f\|_\infty \leq (n+1)^{\frac{1}{2}} \cdot 2^n \cdot \|f\|_\infty = B,$$

onde m e k são os graus de g^* e h^*. Reciprocamente, sejam g^* e h^* como no passo 8 do algoritmo. Então

$$g^*h^* \equiv bf^* \; mod \; p. \tag{3.2}$$

Além disso,

$$\|g^*h^*\|_\infty \leq \|g^*h^*\|_1 \leq \|g^*\|_1 \cdot \|h^*\|_1 \leq B \leq p/2$$

Isso implica que ambos os lados da congruência (3.2) tem coeficientes, em módulo, menores do que $p/2$ e portanto, temos a igualdade $g^*h^* = bf^*$ em $\mathbb{Z}[x]$.

Teorema 3.11. *O algoritmo* 3.10 *funciona corretamente. O custo esperado para os passos 3 e 5 do algoritmo é de* $\mathcal{O}(n^3 + \log^3(A))$. *Uma execução dos passos 8 e 9 custa* $\mathcal{O}(n^2 + n\log(A))$ *operações. Os passos 9 e 10 são executados no máximo* 2^{n+1} *vezes.*

Demonstração. Ver [17], Teorema 15.3. □

Assim, o número esperado de operações é, no pior caso, exponencial em $n = deg(f)$. Na prática, porém, o número de fatores módulo p é menor do que o grau de f, o que torna o algoritmo eficiente.

3.2.2 Levantamento de Hensel - Algoritmo de Berlekamp-Zassenhaus

Em 1918, K. Hensel desenvolveu um método que permite calcular uma fatoração de um polinômio módulo p^a, para qualquer natural $a \geq 1$, a partir de sua fatoração módulo p. Essa técnica foi introduzida nos algoritmos modernos de fatoração polinomial por Zassenhaus, em 1969. Para mais detalhes, veja [21] e [38].

Assim, ao invés de executarmos uma fatoração módulo um primo grande p, podemos fatorar f módulo um primo pequeno e depois aplicar o Levantamento de Hensel para encontrar uma fatoração de f módulo uma potência suficientemente grande desse primo p. Na prática, essa segunda abordagem é mais eficiente que apenas fatorar f módulo um primo suficientemente grande.

O método original de Hensel extende a fatoração de f linearmente até a potência p^a, isto é, o método calcula a fatoração módulo $p^2, p^3, \ldots, p^{a-1}$ até chegar na fatoração módulo p^a. O método descrito por Zassenhaus, por outro lado, utiliza uma abordagem "quadrática", isto é, neste método calcula-se a fatoração de f módulo $p^2, p^4, \ldots, p^{2k}$, para certo $k > 0$. Embora existam trabalhos que afirmem que o método linear é mais rápido [32], pode-se mostrar que, em geral, o método quadrático é o mais rápido. Para mais detalhes, ver [1], [36] e [39]. Antes de demonstrarmos o Lema de Hensel, começaremos enunciando alguns resultados auxiliares. As demonstrações destes resultados podem ser encontradas em [37]. Para maiores informações sobre o Lema de Hensel, veja [21].

Teorema 3.12 (Algoritmo Euclidiano Extendido)**.** *Seja K um corpo e sejam $a, b \in K[x]$ tais que $deg(a) \geq deg(b) > 0$. Suponha que a e b não sejam associados, isto é, não existe $c \in K$ tal que $a = cb$. Então existem polinômios $g, s, t \in K[x]$ tais que $g = mdc(a, b)$, $as + bt = g$, $deg(s) < deg(b) - deg(g)$ e $deg(t) < deg(a) - deg(g)$.*

Demonstração. Ver [37], Teorema 3.1.2. □

Corolário 3.13. *Sejam $a, b \in K[x]$ relativamente primos e $c \in K[x]^\times$ tais que $deg(c) < deg(ab)$. Então c pode ser representado unicamente como $c = ua + vb$, onde $deg(u) < deg(b)$ e $deg(v) < deg(a)$.*

Demonstração. Como a e b são relativamente primos, existem $u, v \in K[x]$ tais que $deg(u) < deg(b)$, $deg(v) < deg(a)$ e $1 = ua + vb$. Obviamente, temos

$$c = (cu) \cdot a + (cv) \cdot b.$$

Se os graus de cu e cv não satisfazem as devidas desigualdades, defina $u' = rem(cu, b)$ e $v' = cv + quo(cu, b) \cdot a$, onde $rem(cu, b)$ é o resto da divisão de cu por b e $quo(cu, b)$ é o quociente desta divisão. Dessa forma,

$$u'a + v'b = rem(cu, b) \cdot a + (cv + quo(cu, b) \cdot a) \cdot b =$$

$$= a \cdot \big(quo(cu, b) \cdot b + rem(cu, b)\big) + cvb = a \cdot (cu) + cvb = c \cdot (ua + vb) = c.$$

Pela definição de u', sabemos que $deg(u') < deg(b)$. Como $deg(c) < deg(a) + deg(b)$ e $deg(u'a) < deg(a) + deg(b)$, segue que o mesmo deve ser verdadeiro para $v'b$, ou seja, $deg(v'b) < deg(a) + deg(b)$. Isso implica que $deg(v') < deg(a)$, como queríamos demonstrar. □

A próxima proposição pode ser vista como uma generalização do corolário anterior e será utilizada mais adiante.

Proposição 3.14. *Dados $a_1, a_2, \ldots, a_r \in K[x]$ polinômios relativamente primos e $c \in K[x]$ tal que $deg(c) < deg(a_1) + deg(a_2) + \cdots + deg(a_r)$. Então existem $v_1, v_2, \ldots, v_r \in K[x]$ tais que $deg(v_i) < deg(a_i)$, $i = 1, 2, \ldots, r$, e*

$$c = \sum_{i=1}^{r} v_i \tilde{a}_i, \quad onde \ \ \tilde{a}_i = \prod_{j=1, j\neq i}^{r} a_j.$$

Demonstração. Para $1 < i \leq r$, defina

$$a_i^* = \prod_{j \geq i} a_j.$$

Pelo corolário 3.13, existem $u_1, v_1 \in K[x]$ tais que $deg(u_1) < deg(a_2^*)$, $deg(v_1) < deg(a_1)$ e

$$c = u_1 a_1 + v_1 a_2^* = u_1 a_1 + v_1 \tilde{a}_1. \tag{3.3}$$

Novamente utilizando o corolário 3.13, segue que existem $u_2, v_2 \in K[x]$ tais que $deg(u_2) < deg(a_3^*)$, $deg(v_2) < deg(a_2)$ e

$$u_1 = u_2 a_2 + v_2 a_3^*.$$

Multiplique a equação anterior por a_1 para obter

$$a_1 u_1 = a_1 u_2 a_2 + v_2 \tilde{a}_2. \tag{3.4}$$

Note que, substituindo (3.3) em (3.4), obtemos

$$c = a_1 a_2 u_2 + v_1 \tilde{a}_1 + v_2 \tilde{a}_2.$$

Suponha que temos construídos os polinômios $v_1, v_2, \ldots, v_i$ satisfazendo as condições acima e tais que

$$c = a_1 a_2 \cdots a_i u_i + \sum_{j \leq i} v_j \tilde{a}_j. \tag{3.5}$$

Novamente pelo corolário 3.13, sabemos que existem u_{i+1}, v_{i+1} tais que $deg(u_{i+1}) < deg(a^*_{i+2})$, $deg(v_{i+1}) < deg(a_{i+1})$ e

$$u_i = u_{i+1} a_{i+1} + v_{i+1} a^*_{i+2}.$$

Multiplicando essa última equação por $a_1 a_2 \cdots a_i$ e substituindo em (3.5), temos

$$c = a_1 a_2 \cdots a_{i+1} u_{i+1} + \sum_{j \leq i+1} v_j \tilde{a}_j.$$

Repetimos essa construção até obtermos os vetores $v_1, v_2, \ldots, v_{r-1}$. Neste ponto, temos a expressão

$$c = a_1 a_2 \cdots a_{r-1} u_{r-1} + \sum_{j \leq r-1} v_j \tilde{a}_j.$$

Para finalizar, defina $v_r = u_{r-1}$. Pela definição de u_{r-1}, segue que $deg(u_{r-1}) < deg(a^*_r) = deg(a_r)$. Assim, $deg(v_i) < deg(a_i)$, $i = 1, 2, \ldots, r$ e

$$c = \sum_{i=1}^{r} v_i \tilde{a}_i.$$

□

Não é difícil ver que a demonstração desta proposição nos fornece um algoritmo para calcular os polinômios v_i's. Abaixo apresentamos a "versão linear" do Lema de Hensel.

Teorema 3.15 (Lema de Hensel)**.** *Seja $a \in \mathbb{Z}[x]$ um polinômio primitivo e livre de quadrados. Seja p um primo tal que $p \nmid lc(a)$. Sejam $a_1, a_2, \ldots, a_r \in \mathbb{Z}/p\mathbb{Z}[x]$ tais que $a \equiv a_1 a_2 \cdots a_r \ mod\ p$, $lc(a_1) = lc(a)$ e $lc(a_2) = lc(a_3) = \cdots = lc(a_r) = 1$. Então para todo número natural k existem polinômios $a_1^{(k)}, a_2^{(k)}, \ldots, a_r^{(k)} \in \mathbb{Z}/p^k\mathbb{Z}[x]$ tais que $lc(a_1^{(k)}) = lc(a)$, $lc(a_2^{(k)}) = lc(a_3^{(k)}) = \cdots = lc(a_r^{(k)}) = 1$,*

$$a \equiv a_1^{(k)} a_2^{(k)} \cdots a_r^{(k)} \ mod\ p^k$$

e

$$a_i^{(k)} \equiv a_i \ mod\ p, \quad i = 1, 2, \ldots, r.$$

Demonstração. Provaremos por indução em k. Para $k = 1$, basta tomar $a_i^{(k)} = a_i$. Assim, suponha que a hipótese seja válida para algum $k \geq 1$. Isto é,

$$a \equiv a_1^{(k)} a_2^{(k)} \cdots a_r^{(k)} \ mod\ p^k.$$

Esta igualdade implica que, para algum $\tilde{d} \in \mathbb{Z}/p\mathbb{Z}[x]$, temos

$$a - \prod_{i=1}^{r} a_i^{(k)} \equiv p^k \tilde{d} \ mod\ p^{k+1}.$$

Substituindo o coeficiente líder de $a_1^{(k)}$ por $lc(a)\ mod\ p^{k+1}$, temos, para algum $d \in \mathbb{Z}/p\mathbb{Z}[x]$

$$a - \prod_{i=1}^{r} a_i^{(k)} \equiv p^k d \ mod\ p^{k+1} \tag{3.6}$$

onde $deg(d) < deg(a)$. Queremos encontrar polinômios $b_i \in \mathbb{Z}/p\mathbb{Z}[x]$ tais que $deg(b_i) < deg(a_i)$ e tais que

$$a_i^{(k+1)} = a_i^{(k)} + p^k b_i. \tag{3.7}$$

Usando a expressão (3.7), temos

$$a - \prod_{i=1}^{r} a_i^{(k+1)} \equiv a - \prod_{i=1}^{r} a_i^{(k)} - p^k \Big(\sum_{i=1}^{r} b_i \prod_{j=1, j\neq i}^{r} a_j \Big) \ mod\ p^{k+1} \equiv p^k \Big(d - \sum_{i=1}^{r} b_i \tilde{a}_i \Big) \ mod\ p^{k+1}$$

onde $\tilde{a}_i = \prod_{j=1, j\neq i}^{r} a_j$. Assim, $a_i^{(k+1)},\ i = 1, 2, \ldots, r$ será uma fatoração módulo p^{k+1} se, e somente se,

$$d \equiv \sum_{i=1}^{r} b_i \tilde{a}_i \ mod\ p. \tag{3.8}$$

Ou seja, $\prod_{i=1}^{r} a_i^{(k+1)}$ é uma fatoração módulo p^{k+1} se, e somente se, existem $b_1, b_2, \ldots, b_r \in \mathbb{Z}/p\mathbb{Z}[x]$ tais que (3.8) seja válida. Ora, a existência de uma solução é garantida pela proposição 3.14. □

Esse teorema nos permite escrever um algoritmo para o Levantamento de Hensel.

Algoritmo 3.16. *Levantamento de Hensel*

Entrada: *Polinômio $f \in \mathbb{Z}[x]$ primitivo e livre de quadrados, $K \in \mathbb{N}$ e p primo tal que f mod p é livre de quadrados. Além disso, polinômios $a_1, a_2, \ldots, a_r \in \mathbb{Z}/p\mathbb{Z}[x]$ relativamente primos tais que $f \equiv a_1 a_2 \cdots a_r\ mod\ p$, $lc(a_1) = lc(f)\ mod\ p$ e $lc(a_2) = lc(a_3) = \cdots = lc(a_r) = 1$*

Saída: *$\overline{a}_1, \overline{a}_2, \ldots, \overline{a}_r \in \mathbb{Z}/p^k\mathbb{Z}[x]$ tais que $f \equiv \overline{a}_1 \overline{a}_2 \cdots \overline{a}_r\ mod\ p^K$, $lc(\overline{a}_1) = lc(f)\ mod\ p^K$, $lc(\overline{a}_2) = lc(\overline{a}_3) = \cdots = lc(\overline{a}_r) = 1$ e $\overline{a}_i \equiv a_i\ mod\ p$.*

1 *Encontre polinômios $v_1, v_2, \ldots, v_r \in \mathbb{Z}/p\mathbb{Z}[x]$ tais que $deg(v_i) < deg(a_i)$ e $1 \equiv \sum_{i=1}^{r} v_i \tilde{a}_i\ mod\ p$, onde $\tilde{a}_i = \prod_{j=1, j\neq i}^{r} a_j$.*

2 ***para*** $i = 1, 2, \ldots, r$ ***fazer***

$\overline{a}_i \leftarrow a_i$

3 $k \leftarrow 1$

4 ***enquanto*** $k < K$ ***fazer***

Substitua $lc(\overline{a}_i)$ por $lc(f)\ mod\ p^{k+1}$

$\tilde{d} \leftarrow f - \prod_{i=1}^{r} \overline{a}_i \ mod\ p^{k+1}$

$d \leftarrow \tilde{d}/p^k$

5 **para** $i = 1, 2, \ldots, r$ **fazer**

$b_i \leftarrow rem(dv_i, a_i)$

$\overline{a}_i \leftarrow \overline{a}_i + p^k b_i$

$k \leftarrow k + 1$

6 **retorne** $\overline{a}_1, \overline{a}_2, \ldots, \overline{a}_r$

A definição do polinômio d no passo 4 do algoritmo segue de (3.6), enquanto a definição do polinômio b_i segue da demonstração do corolário 3.13.

Vejamos um exemplo.

Exemplo 3.17. *Considere o polinômio* $f = 6x^7+7x^6+4x^5+x^4+6x^3+7x^2+4x+1 \in \mathbb{Z}[x]$. *Tomando* $p = 5$, $f\ mod\ 5$ *se fatora como*

$$f \equiv (x+3) \cdot (x^2+3) \cdot (x^2+2) \cdot (x^2+4x+2)\ mod\ 5.$$

Sejam $a_1 = x+3$, $a_2 = x^2+3$, $a_3 = x^2+2$ *e* $a_4 = x^2+4x+2$. *Suponha que queremos levantar essa fatoração módulo* $p^7 = 5^7$. *Aplicando o algoritmo acima, obtemos os polinômios*

$$a_1^{(7)} = 6x+3, \quad a_2^{(7)} = x^2+32318, \quad a_3^{(7)} = x^2+45807 \ \ e \ \ a_4^{(7)} = x^2+52084x+26042.$$

Usando um computador, é fácil ver que $f \equiv a_1^{(7)} a_2^{(7)} a_3^{(7)} a_4^{(7)}\ mod\ 5^7$. *Além disso, também é possível verificar que* $a_i^{(7)} \equiv a_i\ mod\ 5$ *para* $i = 1, 2, \ldots, 4$.

Abaixo apresentaremos o segundo algoritmo para fatoração em $\mathbb{Z}[x]$. Este algoritmo, como já discutido, fatora f módulo um primo p, não necessariamente grande, e depois realiza o Levantamento de Hensel até uma potência p^K,

suficientemente grande. A ideia deste algoritmo é essencialmente a mesma do algoritmo apresentado na seção anterior. Se conseguirmos encontrar uma fatoração para f módulo uma potência suficientemente grande do primo p escolhido, então podemos recuperar os fatores de f em $\mathbb{Z}[x]$.

Algoritmo 3.18. *Fatoração com Levantamento de Hensel*

Entrada: *Polinômio $f \in \mathbb{Z}[x]$ primitivo e livre de quadrados*

Saída: *Polinômios $f_1, f_2, \ldots, f_r \in \mathbb{Z}[x]$ tais que $f = f_1 f_2 \cdots f_r$.*

1 *Escolha p tal que $p \nmid lc(f)$ e tal que $f \bmod p$ é livre de quadrados.*

2 *Fatore $f \bmod p$, obtendo os fatores $g_1, g_2, \ldots, g_k \in \mathbb{Z}/p\mathbb{Z}[x]$.*

3 *Normalize os polinômios g_i's, isto é, $lc(g_1) = lc(f) \bmod p$ e*

$lc(g_2) = lc(g_3) = \cdots = lc(g_k) = 1$.

4 $B \leftarrow (n+1)^{1/2} \cdot 2^n \cdot \|f\|_\infty, \quad K \leftarrow \lceil \log_p(2 \cdot |lc(f)| \cdot B) \rceil$

5 *Utilize o algoritmo anterior para encontrar polinômios $\tilde{g}_1, \tilde{g}_2, \ldots, \tilde{g}_k \in \mathbb{Z}/p^k\mathbb{Z}[x]$ satisfazendo $f \equiv \tilde{g}_1 \tilde{g}_2 \cdots \tilde{g}_k \bmod p^K$ e demais propriedades.*

6 $\overline{f} \leftarrow f, \quad C \leftarrow \{2, \ldots, k\}, \quad i \leftarrow 0, \quad m \leftarrow 0$

7 ***enquanto*** $m < |C|$ ***fazer***

$m \leftarrow m + 1$

8 ***para*** *todos* $\{i_1, i_2, \ldots, i_m\} \subseteq C$ ***fazer***

$\tilde{g} \leftarrow lc(\overline{f}) g_{i_1} g_{i_2} \cdots g_{i_m}$ *mods* p

$g \leftarrow pp(\tilde{g})$

9	**se** $g \mid \overline{f}$ **então**
	$i \leftarrow i+1$
	$f_i \leftarrow g, \quad \overline{f} \leftarrow \overline{f}/g, \quad C \leftarrow C \setminus \{i_1, i_2, \ldots, i_m\}$
	$i \leftarrow i+1, \quad f_i \leftarrow \overline{f}$
10	**retorne** $[f_1, f_2, \ldots, f_i]$

Note que, se K é escolhido de acordo com o item 4 do algoritmo, então

$$K \geq \log_p(2 \cdot |lc(f)| \cdot B),$$

ou seja, $p^K \geq 2 \cdot |lc(f)| \cdot B$. Assim, calculando uma fatoração de f módulo p^K, poderemos encontrar os fatores de f em $\mathbb{Z}[x]$ da mesma forma que o algoritmo da seção anterior.

No passo 5 realizamos o levantamento de Hensel da fatoração de $f \ mod \ p$ até uma fatoração de $f \ mod \ p^K$. A partir do passo 5, realizamos a busca pelos fatores de f em $\mathbb{Z}[x]$, combinando os fatores encontrados módulo p^K. No passo 8, tomamos um subconjunto qualquer do conjunto dos fatores módulo p^K e fazemos seu produto. Multiplicamos também pelo termo líder de f. Lembre-se que os coeficientes desse produto são considerados na representação simétrica de $\mathbb{Z}/p\mathbb{Z}$. Note que, assim como o algoritmo 3.10, o passo 8 é responsável pela complexidade exponencial do algoritmo. Na próxima seção veremos o algoritmo LLL, que procura evitar essa parte combinatorial. Finalmente no passo 9, verificamos se o produto encontrado é um fator de f em $\mathbb{Z}[x]$.

Exemplo 3.19. *Considere o polinômio do exemplo* 3.17. *Para este polinômio,*

$$K = \lceil \log_p(2 \cdot |lc(f)| \cdot B) \rceil = \lceil \log_5(2 \cdot 6 \cdot 2534.2) \rceil = \lceil 6.41 \rceil = 7.$$

Como calculado no exemplo 3.17, *a fatoração de* $f \ mod \ 5^7$ *é dada por*

$$f \equiv (6x+3) \cdot (x^2 + 32318) \cdot (x^2 + 45807) \cdot (x^2 + 52084x + 26042) \ mod \ 5^7.$$

Resta agora combinar os fatores encontrados e verificar quais deles são fatores de f *em* $\mathbb{Z}[x]$. *Primeiramente, para* $i = 1$, *multiplicamos cada um dos fatores mod* 5^7 *encontrados pelo termo líder do polinômio* f *e consideremos seus coeficientes na representação simétrica de* $\mathbb{Z}/5^7\mathbb{Z}$. *Observe que o primeiro fator de* f *mod* 5^7 *é considerado apenas no final do processo, depois de encontrados todos os outros fatores. Assim, para* $i = 1$, *os possíveis candidatos a fatores de* f *em* $\mathbb{Z}[x]$ *são*

$$3x^2 + 18829, \;\; 3x^2 - 18829 \;\; e \;\; 3x^2 + 2x + 1,$$

dos quais, apenas o último é um fator de f *em* $\mathbb{Z}[x]$. *O próximo passo é considerar combinações de dois fatores. Além do primeiro fator, restaram apenas os fatores* $x^2 + 32318$ *e* $x^2 + 45807$. *Multiplicando esses dois fatores ao coeficiente líder de* f *e considerando os coeficientes na representação simétrica de* $\mathbb{Z}/5^7\mathbb{Z}$, *obtemos o seguinte polinômio*

$$x^4 + 1,$$

o qual é um fator de f *em* $\mathbb{Z}[x]$. *Neste ponto, resta apenas o primeiro fator que, após multiplicar pelo termo líder do polinômio* f *e considerar seus coeficientes na representação simétrica de* $\mathbb{Z}/5^7\mathbb{Z}$, *nos fornece*

$$2x + 1,$$

que é o último fator do polinômio f. *Assim,*

$$f = (2x + 1) \cdot (3x^2 + 2x + 1) \cdot (x^4 + 1).$$

3.3 O Algoritmo LLL

Como vimos nas seções anteriores, os algoritmos para fatorar polinômios com coeficientes inteiros ou racionais primeiro fatoram os polinômios módulo um primo p e depois combinam esses fatores para encontrar os fatores verdadeiros do polinômio, ou seja, os fatores de f em $\mathbb{Z}[x]$. Essa última parte destes algoritmos de

fatoração tem complexidade exponencial no número de fatores módulo p. Quando o número de fatores é "baixo" (em relação ao grau do polinômio), esses algoritmos funcionam bem na prática. Porém, quando o número de fatores módulo p é próximo do grau do polinômio a ser fatorado, a parte combinatorial domina a complexidade do algoritmo, tornando-o ineficiente.

Nesta seção, apresentaremos um algoritmo para fatoração de polinômios com coeficientes inteiros que evita a parte combinatorial encontrada nos outros algoritmos. Este algoritmo, introduzido em 1982 por Lenstra, Lenstra e Lovász [28] e comumentemente chamado de *Algoritmo LLL*, está baseado no conceito de *lattice* ou *reticulado*, introduzido em 1910 por Hermann Minkowski no livro chamado *Geometrie der Zahlen* [31].

3.3.1 Reticulados

Começaremos estudando um pouco sobre reticulados. Essa teoria foi introduzida em 1910 por Hermann Minkowski e mais tarde possibilitou resolver problemas difíceis da Física Matemática e Teoria dos Números. Essa teoria também permitiu o desenvolvimento de um algoritmo para fatoração de polinômios com coeficientes inteiros em tempo polinomial.

Em poucas palavras, dados n vetores $v_1, v_2, \ldots, v_n \in \mathbb{R}^n$, o reticulado gerado por esses vetores são todas as combinações lineares de $v_1, v_2, \ldots, v_n$ com coeficientes inteiros. Formalmente,

Definição 3.20. *Seja $n \in \mathbb{N}$ e $v_1, v_2, \ldots, v_n \in \mathbb{R}^n$. Então*

$$\mathcal{L} = \Big\{ \sum_{i=1}^{n} r_i v_i \ : \ r_i \in \mathbb{Z} \Big\}$$

é o reticulado ou $\mathbb{Z}-$módulo gerado por $v_1, v_2, \ldots, v_n$.

Se esses vetores são linearmente independentes então eles formam uma base para $\mathcal{L}$. A norma de $\mathcal{L}$ é definida por

$$\|\mathcal{L}\| = |\det(v_{ij})| \in \mathbb{R},$$

onde v_{ij} é a matriz cujas linhas são os vetores $v_1, \ldots, v_n$.

Lema 3.21. *Sejam $\mathcal{N} \subseteq \mathcal{M} \subseteq \mathbb{R}^n$ reticulados gerados por $w_1, w_2, \ldots, w_n$ e $v_1, v_2, \ldots, v_n$, respectivamente, onde $v_i = (v_{i1}, v_{i2}, \ldots, v_{in})$ e $w_j = (w_{j1}, w_{j2}, \ldots, w_{jn})$. Então $\det(w_{ij})$ é um múltiplo inteiro de $\det(v_{ij})$.*

Demonstração. De fato, como $w_i \in \mathcal{M}$, existem $a_{i1}, a_{i2}, \ldots, a_{in} \in \mathbb{Z}$ tais que $w_i = \sum_{j=1}^{n} a_{ij} v_j$, para $i = 1, 2, \ldots, n$. Defina $A = (a_{ij}) \in \mathbb{Z}^{n \times n}$. Assim

$$\det(w_{ij}) = \det \begin{pmatrix} w_1 \\ w_2 \\ \vdots \\ w_n \end{pmatrix} = \det \begin{pmatrix} \sum_{j=1}^{n} a_{1j} v_j \\ \sum_{j=1}^{n} a_{2j} v_j \\ \vdots \\ \sum_{j=1}^{n} a_{nj} v_j \end{pmatrix} =$$

$$= \det \left[\begin{pmatrix} a_{11} & a_{12} & \ldots & a_{1n} \\ a_{21} & a_{22} & \ldots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{n1} & a_{n2} & \ldots & a_{nn} \end{pmatrix} \cdot \begin{pmatrix} v_1 \\ v_2 \\ \vdots \\ v_n \end{pmatrix} \right] = \det(A) \cdot \det(v_{ij}),$$

como $A \in \mathbb{Z}^{n \times n}$ segue que $\det(A) \in \mathbb{Z}$, como queríamos demonstrar.

□

Em particular, se $\mathcal{N} = \mathcal{M}$, concluímos que a definição de norma de um reticulado não depende da base fixada. Como veremos mais adiante, existe uma relação entre fatores de um polinômio e vetores de tamanho minimal em um determinado reticulado. Porém, calcular esses vetores não é uma tarefa fácil. Em

1980, P. van Emde Boas [8] demonstrou que este problema é $\mathcal{NP}$-difícil na norma L_∞, conjecturando que o problema teria a mesma dificuldade na norma L_2. Alguns anos depois, em 1997, Ajtai [2] demonstrou que a conjectura era verdadeira. Além disso, vários outros trabalhos mostraram que mesmo o problema de encontrar um vetor cujo tamanho seja um múltiplo desse vetor de tamanho minimal é um problema difícil. Felizmente, para muitas aplicações, basta encontrar um vetor cujo tamanho seja um certo múltiplo do vetor de tamanho minimal.

Para entender o algoritmo LLL, vamos revisar o processo de ortogonalização de Gram-Schmidt. Dados $v_1, v_2, \ldots, v_n \in \mathbb{R}^n$, pode-se calcular uma base ortogonal $v_1^*, v_2^*, \ldots, v_n^*$ de $\mathbb{R}^n$ da seguinte forma:

$$\begin{aligned} v_1^* &= v_1; \\ v_2^* &= v_2 - \frac{\langle v_2, v_1^* \rangle}{\langle v_1^*, v_1^* \rangle} v_1^*; \\ &\vdots \\ v_n^* &= v_n - \textstyle\sum_{j=1}^{n-1} \frac{\langle v_n, v_j^* \rangle}{\langle v_j^*, v_j^* \rangle} v_j^*; \end{aligned}$$

Se definirmos

$$V = \begin{pmatrix} v_1 \\ v_2 \\ \vdots \\ v_n \end{pmatrix}_{n \times n}, \quad V^* = \begin{pmatrix} v_1^* \\ v_2^* \\ \vdots \\ v_n^* \end{pmatrix}_{n \times n} \quad \text{e} \quad M_{i,j} = \begin{cases} 0, \text{ se } j > i. \\ 1, \text{ se } j = i. \\ \dfrac{\langle v_i, v_j^* \rangle}{\langle v_j^*, v_j^* \rangle}, \text{ se } j < i. \end{cases} \tag{3.9}$$

então $V = MV^*$. Chamaremos esta fórmula de Equação de Gram-Schmidt(EGS). Definindo μ_{ij} como a entrada ij da matriz M, então o i-ésimo vetor do processo de Gram-Schmidt pode ser escrito como

$$v_i^* = v_i - \sum_{j<i} \mu_{ij} v_j^*.$$

Assim, daqui em diante, sempre que falarmos do i-ésimo vetor obtido pelo processo de Gram-Schmidt, estaremos considerando a expressão acima. Além disso, dados vetores $v_1, v_2, \ldots, v_n$, os vetores $v_1^*, v_2^*, \ldots, v_n^*$ sempre denotarão os vetores obtidos pelo

processo de Gram-Schmidt a partir dos vetores $v_1, v_2, \ldots, v_n$. O próximo teorema apresenta algumas propriedades da ortogonalização de Gram-Schmidt.

Teorema 3.22. *Sejam $v_1, v_2, \ldots, v_n \in \mathbb{R}^n$ vetores linearmente independentes e sejam $v_1^*, v_2^*, \ldots, v_n^*$ vetores obtidos pelo processo de Gram-Schmidt. Seja $1 \leq k \leq n$ e U_k o $\mathbb{R}$-espaço vetorial gerado por $v_1, v_2, \ldots, v_k$. Então:*

i) $\langle v_i^*, v_j^* \rangle = 0, \ \forall\, i \neq j$.

ii) Se U_k^* é o $\mathbb{R}$-subespaço gerado por $v_1^*, v_2^*, \ldots, v_k^*$, então $U_k = U_k^*$.

iii) v_k^* é a projeção de v_k no complemento ortogonal de U_{k-1} e portanto, em particular, $\|v_k^*\| \leq \|v_k\|$.

iv) $\det(v_{ij}) = \det(v_{ij}^*)$.

Demonstração. A demonstração desses resultados pode ser encontrada em diversos livros básicos de álgebra linear. □

A seguir, demonstraremos a *Desigualdade de Hadamard*, que será utilizada mais adiante.

Teorema 3.23 (Desigualdade de Hadamard)**.** *Seja $A \in \mathbb{R}^{n \times n}$ uma matriz cujas linhas são os vetores $v_1, v_2, \ldots, v_n \in \mathbb{R}^{1 \times n}$. Então*

$$\det(A) \leq \|v_1\| \|v_2\| \cdots \|v_n\|.$$

Demonstração. Podemos supor que os vetores $v_1, v_2, \ldots, v_n$ são linearmente independentes pois, do contrário, a desigualdade é trivial. Podemos também supor que os vetores $v_1^*, v_2^*, \ldots, v_n^*$, produzidos pelo processo de Gram-Schmidt, possuem norma 1. Dessa forma, devido à ortogonalidade desses vetores, temos que, para todo vetor $v \in \mathbb{R}^n$

$$v = \sum_{i=1}^{n} \langle v, v_i^* \rangle \, v_i^*$$

e consequentemente, $\|v\|^2 = \sum_{i=1}^{n} |\langle v, v_i^* \rangle|^2$. Segundo o item $ii)$ do Teorema 3.22,

$$v_k = \sum_{i=1}^{k} \langle v_k, v_i^* \rangle \, v_i^*. \tag{3.10}$$

Escolha $c_{kl} = \langle v_l, v_k^* \rangle$ para $1 \leq k \leq l$ e $c_{kl} = 0$ para $l < k \leq n$. Dessa forma, de acordo com (3.10), temos $V = CV^*$, onde C é a matriz com entradas c_{ij} definidas acima. Assim,

$$\det(V)^2 = \det(VV^T) = \det(CV^*V^{*T}C^T) = \det(CC^T) = \det(C)^2,$$

visto que V^* é uma matriz ortogonal. Por outro lado, C é uma matriz triangular cujas entradas diagonais são $\langle v_i, v_i^* \rangle$. Assim

$$\det(A)^2 = \prod_{i=1}^{n} |\langle v_i, v_i^* \rangle|^2.$$

Ora, visto que $|\langle v_i, v_i^* \rangle|^2 \leq \sum_{j=1}^{i} |\langle v_j, v_i^* \rangle|^2$, segue que

$$\det(A)^2 \leq \prod_{i=1}^{n} \left(\sum_{j=1}^{i} |\langle v_j, v_i^* \rangle|^2 \right).$$

Porém, cada um desses somatórios é exatamente $\|v_i\|^2$. Assim,

$$\det(A)^2 \leq \prod_{i=1}^{n} \|v_i\|^2.$$

□

O próximo teorema mostra porque o processo de ortogonalização tem um papel importante nessa teoria.

Teorema 3.24. *Seja $\mathcal{L}$ um reticulado gerado pelos vetores linearmente independentes $v_1, v_2, \ldots, v_n$. Sejam $v_1^*, v_2^*, \ldots, v_n^*$ os vetores obtidos pelo processo de Gram-Schmidt. Então para todo $v \in \mathcal{L}$, tem-se*

$$\|v\| \geq \min\{\|v_1^*\|, \|v_2^*\|, \ldots, \|v_n^*\|\}.$$

Demonstração. Como $v \in \mathcal{L}$, existem $a_i \in \mathbb{Z}, i = 1, 2, \ldots, n$, tais que $v = \sum_{i=1}^{n} a_i v_i$. Seja k o maior índice para o qual $a_k \neq 0$. Pelo processo de ortogonalização de Gram-Schmidt,

$$v_i^* = v_i - \sum_{j<i} \mu_{ij} v_j^*$$

ou seja, considerando $\mu_{ii} = 1$, temos

$$v_i = \sum_{j \leq i} \mu_{ij} v_j^*. \tag{3.11}$$

Por outro lado,

$$v = \sum_{i=1}^{n} a_i v_i = \sum_{i=1}^{k} a_i v_i. \tag{3.12}$$

Substituindo cada v_i em (3.12) pela expressão (3.11) e rearranjando os termos, temos um somatório da forma

$$v = a_k v_k^* + \sum_{i<k} \alpha_i v_i^*,$$

para certos $\alpha_i \in \mathbb{R}$. Assim,

$$\|v\|^2 = \langle v, v \rangle = a_k^2 \|v_k^*\|^2 + \Big\langle \sum_{i<k} \alpha_i v_i^*, \sum_{i<k} \alpha_i v_i^* \Big\rangle \geq a_k^2 \|v_k^*\|^2.$$

Visto que $a_k \in \mathbb{Z}$, segue-se que

$$\|v\| \geq |a_k| \|v_k^*\| \geq \|v_k^*\| \geq \min\{\|v_1^*\|, \|v_2^*\|, \ldots, \|v_n^*\|\}.$$

□

Assim, a norma dos vetores calculados pelo processo de ortogonalização de Gram-Schmidt fornecem uma cota inferior para o tamanho dos vetores no reticulado. Ou seja, se queremos encontrar vetores no reticulado com tamanho minimal, bons candidatos seriam os vetores obtidos pelo processo de Gram-Schmidt. O problema, porém, é que estes vetores nem sempre estão no reticulado, pois os coeficientes $\dfrac{\langle v_i, v_j^* \rangle}{\langle v_j^*, v_j^* \rangle}$ nem sempre são números inteiros. Isso motiva a seguinte definição.

Definição 3.25. *Uma base* $\{v_1, v_2, \ldots, v_n\}$ *de um reticulado* $\mathcal{L}$ *é dita reduzida se a base ortogonal* $\{v_1^*, v_2^*, \ldots, v_n^*\}$*, calculada pelo processo de ortogonalização de Gram-Schmidt, satisfaz* $|\mu_{ij}| \leq 1/2$*, para* $1 \leq j < i \leq n$ *e*

$$\|v_i^* + \mu_{ii-1}v_{i-1}^*\|^2 \geq \frac{3}{4}\|v_{i-1}^*\|^2, \ \forall\, 1 < i \leq n.$$

Visto que

$$v_i^* = v_i - \sum_{j=1}^{i-1} \frac{\langle v_i, v_j^* \rangle}{\langle v_j^*, v_j^* \rangle} v_j^* = v_i - \sum_{j=1}^{n} \mu_{ij} v_j^*$$

e que os vetores $v_1^*, v_2^*, \ldots, v_n^*$ são ortogonais, segue-se que

$$\frac{3}{4}\|v_{i-1}^*\|^2 \leq \|v_i^* + \mu_{ii-1}v_{i-1}^*\|^2 = \|v_i^*\|^2 + \mu_{ii-1}^2\|v_{i-1}^*\|^2.$$

Logo,

$$\|v_i^*\|^2 \geq \left(\frac{3}{4} - \mu_{ii-1}^2\right)\|v_{i-1}^*\|^2 \geq \frac{1}{2}\|v_{i-1}^*\|^2.$$

Não é difícil ver que, em geral, temos

$$\|v_i^*\|^2 \geq \frac{1}{2^j}\|v_{i-j}^*\|^2, \ \ \forall\, 1 \leq j < i \leq n. \tag{3.13}$$

Lema 3.26. *Sejam* $v_1, v_2, \ldots, v_n$ *uma base reduzida para o reticulado* $\mathcal{L}$ *e sejam* $v_1^*, v_2^*, \ldots, v_n^*$ *os vetores do processo de ortogonalização de Gram-Schmidt. Então*

$$\|v_i\|^2 \leq 2^{i-1}\|v_i^*\|^2.$$

Demonstração. Como $v_i^* = v_i - \sum_{j=1}^{i-1} \mu_{ij} v_j^*$ e os vetores $v_1^*, v_2^*, \ldots, v_n^*$ são ortogonais, segue-se que

$$\|v_i\|^2 \leq \|v_i^*\|^2 + \sum_{j=1}^{i-1} |\mu_{ij}|^2 \|v_j^*\|^2.$$

Fazendo a mudança de índices $j = i - k$, utilizando a equação (3.13) e o fato de que $|\mu_{ij}| \leq 1/2$, temos

$$\|v_i\|^2 \leq \|v_i^*\|^2 + \sum_{k=1}^{i-1} \frac{1}{4}\|v_{i-k}^*\|^2 \leq \|v_i^*\|^2 + \sum_{k=1}^{i-1} \frac{1}{4} 2^k \|v_i^*\|^2 = \|v_i^*\|^2 (1 + \frac{1}{4}\sum_{k=1}^{i-1} 2^k).$$

Vamos mostrar agora que $1 + \frac{1}{4}\sum_{k=1}^{i-1} 2^k \leq 2^{i-1}$. Usando a fórmula para a soma de uma progressão geométrica, temos

$$1 + \frac{1}{4}\sum_{k=1}^{i-1} 2^k = 1 + \frac{1}{4}\left(\frac{2(1-2^{i-1})}{1-2}\right) = 1 + \frac{1}{4}(2^i - 2) = \frac{1}{2} + 2^{i-2} \leq 2^{i-1},$$

para $1 \leq i \leq n$, como queríamos demonstrar.

□

Proposição 3.27. *Seja $\mathcal{L} \in \mathbb{R}^n$ um reticulado com base reduzida $v_1, v_2, \ldots, v_n$. Então $\|v_1\|^2 \leq 2^{n-1}\|w\|^2$, $\forall\, w \in \mathcal{L}$, $w \neq 0$.*

Demonstração. Sejam $v_1^*, v_2^*, \ldots, v_n^*$ os vetores obtidos pelo processo de Gram-Schmidt. Escrevendo

$$w = \sum_{i=1}^{n} r_i v_i = \sum_{i=1}^{n} r_i' v_i^*, \tag{3.14}$$

para certos $r_i \in \mathbb{Z}$ e $r_i' \in \mathbb{R}$, $i = 1, 2, \ldots, n$. Seja k o maior índice i tal que $r_i \neq 0$. Substituindo recursivamente v_i^* por $v_i - \sum_{j<i} \mu_{ij} v_j^*$, $i = n, n-1, \ldots, 1$ na última expressão da equação (3.14), obteremos outra forma de escrever w como combinação linear de $v_1, v_2, \ldots, v_n$ com coeficientes em $\mathbb{R}$. Note que o coeficiente de v_k nessa nova combinação é r_k'. Ora, como tal combinação é única, segue-se que $r_k' = r_k$. Assim, $r_k' \in \mathbb{Z}$. Logo

$$\|w\|^2 = \sum_{i=1}^{n} |r_i'|^2 \|v_i^*\|^2 \geq |r_k'|^2 \|v_k^*\|^2 \geq \|v_k^*\|^2. \tag{3.15}$$

Por outro lado, pela equação (3.13) com $i = k$ e $j = k-1$, sabemos que

$$\|v_1\|^2 = \|v_1^*\|^2 \leq 2^{k-1}\|v_k^*\|^2 \leq 2^{n-1}\|v_k^*\|^2. \tag{3.16}$$

Logo, combinando equações (3.15) e (3.16), temos

$$\|v_1\|^2 \leq 2^{n-1}\|w\|^2. \tag{3.17}$$

□

Assim, se $v_1, v_2, \ldots, v_n$ é uma base reduzida para $\mathcal{L}$ e $w \in \mathcal{L}$ é um vetor de tamanho minimal, então a desigualdade (3.17) afirma que o tamanho de v_1 é, no máximo, $2^{\frac{n-1}{2}}$ vezes o tamanho de w. Ou seja, podemos não saber calcular o vetor de tamanho minimal, mas, caso saibamos encontrar uma base reduzida para o reticulado, conhecemos um vetor cujo tamanho é no máximo $2^{(n-1)/2}$ vezes o tamanho do vetor de tamanho minimal. Em muitas aplicações, isso será suficiente.

Proposição 3.28. *Seja $\mathcal{L}$ um reticulado com base reduzida $v_1, v_2, \ldots, v_n$ e sejam $w_1, w_2, \ldots, w_t \in \mathcal{L}$ vetores linearmente independentes. Então*

$$\|v_j\|^2 \leq 2^{n-1} \max\{\|w_1\|^2, \|w_2\|^2, \ldots, \|w_t\|^2\},\ j = 1, 2, \ldots, t.$$

Demonstração. Escrevendo $w_j = \sum_{i=1}^{n} r_{ij} v_i$, para certos $r_{ij} \in \mathbb{Z}$. Fixando $j \in \{1, 2, \ldots, t\}$. Seja $i(j)$ o maior inteiro i tal que $r_{ij} \neq 0$. Assim,

$$w_j = \sum_{i=1}^{i(j)} r_{ij} v_i.$$

Como na demonstração da proposição 3.27, temos

$$\|w_j\|^2 \geq |r_{i(j)j}|^2 \|v^*_{i(j)}\|^2 \geq \|v^*_{i(j)}\|^2.$$

Reordenando os w_j de tal forma que $i(1) \leq i(2) \leq \cdots \leq i(t)$. Afirmamos que $j \leq i(j)$, para $j = 1, 2, \ldots, t$. De fato, caso contrário, teríamos que $w_1, w_2, \ldots, w_j \in \mathbb{R}v_1 + \mathbb{R}v_2 + \cdots + \mathbb{R}v_{j-1}$, contradizendo a independência linear de $w_1, w_2, \ldots, w_t$. A partir do lema 3.26 e da equação (3.13), segue-se que $\|v_j\|^2 \leq 2^{i-1} \|v^*_i\|^2$, para todo $1 \leq j \leq i \leq n$. Assim, como $j \leq i(j)$, segue que

$$\|v_j\|^2 \leq 2^{i(j)-1} \|v^*_{i(j)}\|^2 \leq 2^{n-1} \|w_j\|^2, \text{ para } j \text{ arbitrário.}$$

$$\text{Logo, } \|v_j\|^2 \leq 2^{n-1} \max\{\|w_1\|^2, \|w_2\|^2, \ldots, \|w_t\|^2\}. \qquad \square$$

Passamos agora a estudar o Algoritmo de Redução de Base, apresentado em [28], que mostra como calcular uma base reduzida para um reticulado a partir de uma base qualquer.

Como vimos acima, o processo de ortogonalização de Gram-Schmidt produz vetores cujos tamanhos fornecem uma cota inferior para o tamanho dos vetores no reticulado. O único problema, como vimos, é que esses vetores nem sempre estão no reticulado pois os coeficientes μ_{ij} na definição de v_i^* nem sempre são inteiros. O algoritmo que apresentaremos abaixo tenta imitar o processo de ortogonalização de Gram-Schmidt. A diferença é que neste algoritmo, durante o cálculo do que seriam os vetores v_i^* no processo de Gram-Schmidt, são consideradas apenas combinações com coeficientes inteiros, tomando o inteiro mais próximo de μ_{ij}, garantindo assim que o vetor final esteja no reticulado. Além disso, algumas trocas são feitas entre os vetores da base para que o resultado final esteja de acordo com a definição da uma base reduzida.

Algoritmo 3.29. *Algoritmo de Redução de Base*

Entrada: $\{v_1, v_2, \ldots, v_n\}$ *base para o reticulado* $\mathcal{L}$.

Saída: $\{u_1, u_2, \ldots, u_n\}$ *base reduzida para o reticulado* $\mathcal{L}$.

1 ***para*** $i = 1, 2, \ldots, n$ ***fazer***

$u_i \leftarrow v_i$

Calcule U^* *e* $M = (\mu_{ij})$, *conforme (3.9).*

2 ***enquanto*** $i \leq n$ ***fazer***

3 ***para*** $j = i-1, i-2, \ldots, 1$ ***fazer***

$u_i \leftarrow u_i - \lceil \mu_{ij} \rfloor u_j.$

Atualiza a EGS.

4	**se** $i > 1$ **e** $\|u_{i-1}^*\|^2 > 2\|u_i^*\|^2$ **então**
	Troque u_i e u_{i-1} e atualize a EGS.
	$i \leftarrow i - 1$
	senão $i \leftarrow i + 1$.
5	**retorna** $\{u_1, u_2, \ldots, u_n\}$

A matriz U^* no passo 1 do algoritmo é formada pelos vetores da base de Gram-Schmidt de $u_1, u_2, \ldots, u_n$. O passo "Atualiza a EGS" significa que, depois de redefinido o vetor u_i no passo 3 ou da troca dos vetores u_i e u_{i-1} no passo 4, precisamos calcular novamente a base de Gram-Schmidt desses novos vetores, bem como as entradas μ_{ij} da matriz M.

A notação $\lceil \mu_{ij} \rfloor$ representa o inteiro mais próximo de μ_{ij} e é definida como $\lfloor \mu_{ij} + 1/2 \rfloor$. A corretude e a complexidade deste algoritmo podem ser encontradas em [28].

Exemplo 3.30. *Considere o reticulado $\mathcal{L}$ gerado pelos vetores $v_1 = (1.3, 2, 3)$, $v_2 = (2, 3.2, 4)$ e $v_3 = (0, 1, 0) \in \mathbb{R}^3$. Após aplicar o algoritmo acima, obtemos os vetores $u_1 = (-0.1, -0.4, 1)$, $u_2 = (0, 1, 0)$ e $u_3 = (0.8, -0.4, 0)$. Note que $u_1 = 3v_1 - 2v_2$, $u_2 = v_3$ e $u_3 = -4v_1 + 3v_2 - 2v_3$. Assim, os vetores u_1, u_2 e u_3 pertencem, de fato, ao reticulado gerado por v_1, v_2 e v_3. Além disso, $\|u_1\| \approx 1.081$, o que significa que $\|v\| \geq \frac{1.081}{2} \approx 0.5408$, para todo vetor não nulo $v \in \mathcal{L}$.*

3.3.2 Fatores de um Polinômio e Reticulados

Iremos agora explorar a ligação entre fatores de um polinômio e reticulados. Para o restante desta seção, estaremos fazendo as seguintes considerações: Seja p um número primo, k um inteiro positivo, $f \in \mathbb{Z}[x]$ um polinômio livre de

quadrados, primitivo e de grau n. Além disso, seja $h \in \mathbb{Z}/p^k\mathbb{Z}[x]$ um polinômio mônico e de grau l satisfazendo:

1) h divide f em $\mathbb{Z}/p^k\mathbb{Z}[x]$

2) $h \ mod \ p$ é irredutível em $\mathbb{Z}/p\mathbb{Z}[x]$

3) $f \ mod \ p$ é livre de quadrados em $\mathbb{Z}/p\mathbb{Z}[x]$.

Ao longo desta seção veremos como escolher cada uma destas variáveis (p, k, h) de tal forma que, ao final, teremos um algoritmo para calcular um fator irredutível de f em $\mathbb{Z}[x]$.

A escolha do primo p deve ser tal que $p \nmid lc(f)$ e $f \ mod \ p$ seja livre de quadrados. Já vimos que p satisfaz essas condições se, e somente se, $p \nmid res(f, f')$. A escolha do polinômio $h \in \mathbb{Z}/p^k\mathbb{Z}[x]$, na prática, é feita a partir de uma fatoração de $f \ mod \ p$. Escolha um fator irredutível de $f \ mod \ p$, h será o levantamento de Hensel desse fator escolhido até a k-ésima potência do primo p. O número k será escolhido mais adiante.

Veremos agora como encontrar um fator irredutível de f em $\mathbb{Z}[x]$.

Teorema 3.31. *Nas hipóteses descritas acima, existe um único fator irredutível h_0 de f em $\mathbb{Z}[x]$, a menos de sinal, tal que $h \ mod \ p$ divide $h_0 \ mod \ p$ em $\mathbb{Z}/p\mathbb{Z}[x]$. Além disso, se g divide f em $\mathbb{Z}[x]$, então as seguintes afirmações são equivalentes:*

i) $h \ mod \ p$ divide $g \ mod \ p$ em $\mathbb{Z}/p\mathbb{Z}[x]$

ii) h divide $g \ mod \ p^k$ em $\mathbb{Z}/p^k\mathbb{Z}[x]$

iii) h_0 divide g em $\mathbb{Z}[x]$

A figura abaixo mostra um exemplo de polinômio f e a escolha do polinômio h. Cada retângulo representa um fator irredutível do polinômio f em seu

respectivo domínio: $\mathbb{Z}, \mathbb{Z}/p^k\mathbb{Z}$ ou $\mathbb{Z}/p\mathbb{Z}$. Note que fatores irredutíveis em $\mathbb{Z}$ podem ser redutíveis em $\mathbb{Z}/p^k\mathbb{Z}$. O mesmo vale para fatores em $\mathbb{Z}/p^k\mathbb{Z}$ sobre $\mathbb{Z}/p\mathbb{Z}$. Os fatores com contorno escuro formam o polinômio g do Teorema 3.10.

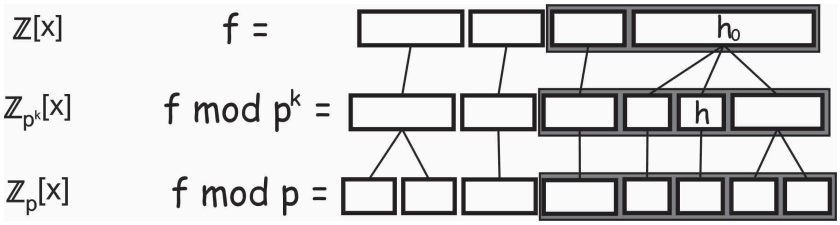

Figura 3.4: Exemplo de polinômio e seus fatores.

Demonstração do Teorema 3.31. A existência de h_0 é óbvia e a unicidade segue do fato de $f \ mod \ p$ ser livre de quadrados em $\mathbb{Z}/p\mathbb{Z}[x]$.

Obviamente, $iii)$ implica $i)$. Supondo $ii)$, segue que $g = hg' + p^k g''$, para certos polinômios g' e g'' em $\mathbb{Z}[x]$. Ora, essa igualdade implica que $g \equiv hg' \ mod \ p$, ou seja, $h \ mod \ p$ divide $g \ mod \ p$ em $\mathbb{Z}/p\mathbb{Z}[x]$.

Vamos agora mostrar que $i) \Rightarrow ii)$ e $i) \Rightarrow iii)$. Suponha $i)$. Como $f \ mod \ p$ é livre de quadrados em $\mathbb{Z}/p\mathbb{Z}[x]$ e $h \ mod \ p$ divide $g \ mod \ p$ em $\mathbb{Z}/p\mathbb{Z}[x]$, segue que $h \ mod \ p$ não divide $(f/g) \ mod \ p$ em $\mathbb{Z}/p\mathbb{Z}[x]$. Logo, $h_0 \ mod \ p$ não divide $(f/g) \ mod \ p$ em $\mathbb{Z}/p\mathbb{Z}[x]$ e consequentemente em $\mathbb{Z}[x]$. Como h_0 divide f em $\mathbb{Z}[x]$, segue que h_0 divide g em $\mathbb{Z}[x]$, demonstrando $iii)$. Iremos agora demonstrar $ii)$. Como $h \ mod \ p$ e $(f/g) \ mod \ p$ são coprimos em $\mathbb{Z}/p\mathbb{Z}[x]$, existem polinômios $r, s \in \mathbb{Z}/p\mathbb{Z}[x]$ tais que $rh + s(f/g) \equiv 1 \ mod \ p$. Pelo Lema de Hensel, existem $r', s' \in \mathbb{Z}[x]$ tais que $r'h + s'(f/g) \equiv 1 \ mod \ p^k$. Ou seja,

$$r'hg + s'f \equiv g \ mod \ p^k.$$

Mas h divide $r'hg \ mod \ p^k$ e h divide $s'f \ mod \ p^k$ em $\mathbb{Z}/p^k\mathbb{Z}[x]$, ou seja, h divide $g \ mod \ p^k$ em $\mathbb{Z}/p^k\mathbb{Z}[x]$.

□

Em particular, tomando $g = h_0 \in \mathbb{Z}[x]$, o item $ii)$ do teorema anterior afirma que h divide $h_0 \ mod \ p^k$. Como estamos interessados nesse fator h_0, e sabemos que h divide $h_0 \ mod \ p^k$, é natural considerar o seguinte conjunto de polinômios.

Para um inteiro $m \geq l$, defina $L_{m,h}$ como o conjunto dos polinômios em $\mathbb{Z}[x]$ de grau no máximo m e que são divisíveis por h em $\mathbb{Z}/p^k\mathbb{Z}[x]$. Esse conjunto $L_{m,h}$ pode ser visto como o conjunto gerado (sobre $\mathbb{Z}$) pelo seguinte conjunto de polinômios

$$\{p^k x^i : \ 0 \leq i < l\} \cup \{hx^j : \ 0 \leq j \leq m - l\}.$$

De fato, se g é um polinômio no reticulado gerado por esse conjunto, então $g = qh + rp^k$, para certos polinômios $r, q \in \mathbb{Z}[x]$, tais que $deg(r) < l$ e $deg(q) \leq m - l$. Ou seja, $deg(g) \leq m$ e $g \equiv qh \ mod \ p^k$. Logo, $g \in L_{m,h}$. Por outro lado, seja $g \in L_{m,h}$. Dividindo g por h em $\mathbb{Z}[x]$, obtemos polinômios $q, r \in \mathbb{Z}[x]$ tais que $g = qh + r$, onde $deg(qh) \leq m$, ou seja, $deg(q) \leq m - l$, e $deg(r) < l$. Como h divide $g \bmod p^k$, segue que $r \equiv 0 \bmod p^k$, ou seja, $r = p^k r'$ e portanto, $g = qh + p^k r'$, com $deg(q) \leq m - l$ e $deg(r') < l$, como queríamos demonstrar.

Note que a escolha do inteiro m está diretamente relacionada com o grau do fator h_0. Como $h \mid h_0 \ mod \ p^k$, segue que $l = deg(h) \leq deg(h_0)$. Além disso, se escolhermos m tal que $deg(h_0) \leq m$, então o fator h_0, que queremos encontrar, está contido no conjunto $L_{m,h}$. Na prática, escolhendo $m = n-1$, teremos a garantia de que o fator $h_0 \in L_{m,h}$.

Utilizando a correspondência natural entre polinômios $a_0 + a_1 x + \cdots + a_m x^m$ em $\mathbb{Z}[x]$, de grau máximo m, e vetores $(a_0, a_1, \ldots, a_m)$ em $\mathbb{Z}^{m+1}$, podemos considerar o conjunto $L_{m,h}$ como um reticulado gerado pelos vetores correspondentes aos polinômios do conjunto acima. Assim, $L_{m,h} \subseteq \mathbb{Z}^{m+1}$. Dessa forma, daqui

em diante trataremos $L_{m,h}$ tanto como conjunto de polinômios quanto reticulado, devendo ficar claro no contexto qual representação estamos usando.

Os próximos resultados nos dizem como podemos encontrar o fator h_0 em $L_{m,h}$.

Teorema 3.32. *Seja $b \in L_{m,h}$ tal que $p^{kl} > \|f\|^m \|b\|^n$, onde h é escolhido de acordo com as hipóteses do início desta seção, $l = grau(h)$ e $n = grau(f)$. Então h_0 divide b em $\mathbb{Z}[x]$. Em particular, $mdc(f, b) \neq 1$.*

Demonstração. Podemos assumir que $b \neq 0$. Seja $s = deg(b)$, $g = mdc(f, b)$ em $\mathbb{Z}[x]$, e $t = deg(g)$. Note que $0 \leq t \leq s \leq m$. Note também que para mostrar que h_0 divide b é suficiente mostrar que h_0 divide g em $\mathbb{Z}[x]$ e que, pela parte *iii)* do teorema anterior, isto é equivalente a mostrar que h divide g em $\mathbb{Z}/p\mathbb{Z}[x]$.

Suponha, por absurdo, que $h \nmid g \ mod\ p$. Como $h \mid f \ mod\ p$, segue que $h \mid f/g \ mod\ p$. Considere o conjunto de polinômios

$$M = \{\lambda f + \mu b \ : \ \lambda, \mu \in \mathbb{Z}[x], deg(\lambda) < s - t, \ e \ deg(\mu) < n - t\}.$$

Considere M' a projeção dos elementos de M nas últimas $n + s - 2t - 1$ entradas, ou seja,

$$M' = \left\{ \sum_{i=t}^{n+s-t-1} a_i x^i \ : \ \sum_{i=0}^{n+s-t-1} a_i x^i \in M \right\}.$$

Além disso, considere o conjunto de polinômios

$$A = \left\{ x^i f \ : \ 0 \leq i < s - t \right\} \cup \left\{ x^j b \ : \ 0 \leq j < n - t \right\}.$$

Note que, considerando M e M' como reticulados, segue que $M \subseteq \mathbb{Z}^{n+s-t}$ e $M' \subseteq \mathbb{Z}^{n+s-2t}$. Vamos mostrar que as projeções do conjunto A são linearmente independentes em M'. De fato, denote por $\pi(\ .\ )$ a projeção em M' e suponha que

$$\lambda_0 \pi(x^0 f) + \lambda_1 \pi(x^1 f) + \cdots + \lambda_{s-t-1} \pi(x^{s-t-1} f) +$$
$$\mu_0 \pi(x^0 b) + \mu_1 \pi(x^1 b) + \cdots + \mu_{n-t-1} \pi(x^{n-t-1} b) = 0,$$

para certas constantes $\lambda_0, \lambda_1, \ldots, \lambda_{s-t-1}, \mu_0, \mu_1, \ldots, \mu_{n-t-1} \in \mathbb{Z}$. Ou seja,

$$\pi(\lambda_0 x^0 f + \lambda_1 x^1 f + \cdots + \lambda_{s-t-1} x^{s-t-1} f + \mu_0 x^0 b + \mu_1 x^1 b + \cdots + \mu_{n-t-1} x^{n-t-1} b) = 0.$$

Defina $\lambda = \lambda_0 x^0 + \lambda_1 x^1 + \cdots + \lambda_{s-t-1} x^{s-t-1}$ e $\mu = \mu_0 x^0 + \mu_1 x^1 + \cdots + \mu_{n-t-1} x^{n-t-1}$. Logo, $\pi(\lambda f + \mu b) = 0$ e portanto, $deg(\lambda f + \mu b) < t$. Suponha que $\lambda f + \mu b \neq 0$. Como $g \mid (\lambda f + \mu b)$, temos que $deg(\lambda f + \mu b) \geq deg(g) = t$, um absurdo. Assim, $\lambda f + \mu b = 0$, e portanto, $\lambda f/g + \mu b/g = 0$, ou ainda, $\lambda f/g = -\mu b/g$. Suponha que $\mu \neq 0$. Como $mdc(f/g, b/g) = 1$, segue que $f/g \mid \mu$, logo $n - t = deg(f/g) \leq deg(\mu)$. Porém, por definição, $deg(\mu) < n - t$, um absurdo. Assim, $\mu = 0$ e consequentemente, $\lambda = 0$. Isso mostra que as projeções de A são linearmente independentes em M'. Como $|A| = n + s - 2t$ e as projeções dos elementos de A geram M' e são linearmente independentes, segue que M' é um reticulado de grau $n+s-2t$, gerado pelos vetores correspondentes aos polinômios do conjunto A.

Assim, pela desigualdade de Hadamard, temos

$$\|M'\| = det(vetores\ da\ base) \leq$$

$$\|\pi(x^0 f)\|\|\pi(x^1 f)\| \cdots \|\pi(x^{s-t-1} f)\|\|\pi(x^0 b)\|\|\pi(x^1 b)\| \cdots \|\pi(x^{n-t-1} b)\|$$

Como a norma da projeção de um vetor é menor ou igual a norma do vetor original e $\|x^i f\| = \|f\|$ e $\|x^j b\| = \|b\|$, para $i = 0, 1, \ldots, s-t-1$ e $j = 0, 1, \ldots, n-t-1$, segue que

$$\|M'\| \leq \|x^0 f\|\ \|x^1 f\|\ \cdots\ \|x^{s-t-1} f\|\ \|x^0 b\|\ \|x^1 b\|\ \cdots\ \|x^{n-t-1} b\| =$$
$$\|f\|\ \|f\|\ \cdots\ \|f\|\ \|b\|\ \cdots\ \|b\| = \|f\|^{s-t}\ \|b\|^{n-t} \leq \|f\|^m\ \|b\|^n < p^{kl}.$$

Vamos agora demonstrar que se $v \in M$ e $deg(v) < t + l$, então $p^k \mid v$. Como estamos supondo que $h \nmid g\ mod\ p$ e h é irredutível em $\mathbb{Z}/p\mathbb{Z}[x]$, segue que existem $\lambda',\ \mu' \in \mathbb{Z}[x]$ tais que $\lambda' h + \mu' g \equiv 1\ mod\ p$, ou ainda,

$$\lambda' h + \mu' g = 1 - pv'$$

em $\mathbb{Z}[x]$, para certo polinômio v'. Seja $r = 1 + pv' + p^2(v')^2 + \cdots + p^{k-1}(v')^{k-1}$. Multiplicando ambos os lados da igualdade acima por r, segue que

$$r\lambda' h + r\mu' g = 1 - p^k(v')^k \equiv 1 \; mod \; p^k.$$

Multiplicando por v/g, obtemos $(v/g)r\lambda' h + vr\mu' \equiv v/g \; mod \; p^k$. Se definirmos $\overline{\lambda} = (v/g)r\lambda'$ e $\overline{\mu} = r\mu'$, temos

$$\overline{\lambda} h + \overline{\mu} v \equiv v/g \; mod \; p^k. \tag{3.18}$$

Como $v \in M$, segue que $v = \lambda f + \mu b$, para certos polinômios $\lambda, \mu \in \mathbb{Z}[x]$. Como $b \in L_{m,h}$, segue que $h \mid b \; mod \; p^k$, pela definição de $L_{m,h}$. Além disso, $h \mid f \; mod \; p^k$ por hipótese, Logo, $h|b \; mod \; p^k$ e $h|f \; mod \; p^k$. Assim $h|(v = \lambda f + \mu b) \; mod \; p^k$ e pela equação (3.18), temos que $h \mid v/g \; mod \; p^k$. Mas h é mônico, $deg(h) = l$, v/g é mônico e $deg(v/g) < t + l - t = l$. Logo, $v/g \equiv 0 \; mod \; p^k$, ou seja, $v \equiv 0 \; mod \; p^k$.

Segundo [11], capítulo I, teorema I.A, podemos escolher uma base $\{b_t, \ldots, b_{s+n-t-1}\}$ para M' tal que $deg(b_j) = j$. Além disso, como provado acima, $p^k \mid b_t, \ldots, b_{t+l-1}$. Assim, quando montamos a matriz dos vetores dessa nova base, teremos uma matriz triangular inferior. Logo,

$$\|M'\| = \det \begin{pmatrix} b_t \\ b_{t+1} \\ \vdots \\ b_{s+n-t-1} \end{pmatrix} = \prod_{j=t}^{n+s-t-1} lc(b_j) \geq (p^k)^l \prod_{j=t+l}^{n+s-t-1} lc(b_j) \geq p^{kl}.$$

Uma contradição. Logo $h \mid g \; mod \; p$ e o resultado segue. □

Assim, se encontrarmos um polinômio $b \in L_{m,h}$ tal que $p^{kl} > \|f\|^m \|b\|^n$, então esse polinômio contém o fator irredutível h_0 de f. Veremos a seguir um resultado que mostra como uma base reduzida para o reticulado $L_{m,h}$ nos ajuda a encontrar esse elemento b. Antes disso, um resultado auxiliar.

Lema 3.33. *Se $g = b_0 + b_1 x + \cdots + b_l x^l \in \mathbb{Z}[x]$ divide $f \in \mathbb{Z}[x]$ então*

i) $|b_i| \leq \binom{l}{i}\|f\|_2,\ 0 \leq i \leq l.$

ii) $\|g\|_2 \leq \binom{2l}{l}^{1/2}\|f\|_2.$

Demonstração. A demonstração do item i) pode ser encontrada em [30]. Supondo o item i), temos

$$\|g\| = \sqrt{\sum_{i=0}^{l} |b_i|^2} \leq \sqrt{\sum_{i=0}^{l} \binom{l}{i}^2 \|f\|^2} = \|f\| \sqrt{\sum_{i=0}^{l} \binom{l}{i}^2}.$$

Utilizando a equação de Vandermonde

$$\sum_{k=0}^{l} \binom{r}{k}\binom{s}{n-k} = \binom{r+s}{n}$$

para $n = r = s = l$ e a igualdade $\binom{l}{l-k} = \binom{l}{k}$, obtemos

$$\sum_{k=0}^{l} \binom{l}{k}\binom{l}{l-k} = \sum_{k=0}^{l} \binom{l}{k}^2 = \binom{2l}{l}.$$

Assim, $\|g\| \leq \|f\|\binom{2l}{l}^{1/2}$, demonstrando o item ii). □

O próximo teorema é o principal resultado por trás do algoritmo de fatoração. Este teorema nos mostra como podemos obter o fator irredutível h_0 através de uma base reduzida para o reticulado $L_{m,h}$.

Teorema 3.34. *Sejam $b_1, b_2, \ldots, b_{m+1}$ uma base reduzida para $L_{m,h}$ e suponha que m satisfaça a desigualdade*

$$p^{kl} > 2^{mn/2}\binom{2m}{m}^{n/2}\|f\|^{m+n}. \tag{3.19}$$

Então

i) $deg(h_0) \leq m$ se, e somente se, $\|b_1\| < \sqrt[n]{p^{kl}/\|f\|^m}$.

ii) Suponha $\|b_1\| < \sqrt[n]{p^{kl}/\|f\|^m}$. *Seja* $t \in \{1, 2, \ldots, m+1\}$ *o maior inteiro tal que* $\|b_t\| < \sqrt[n]{p^{kl}/\|f\|^m}$. *Então* $deg(h_0) = m + 1 - t$ *e* $h_0 = mdc(b_1, b_2, \ldots, b_t)$.

Note que todos os valores na desigualdade (3.19) já estão fixos, exceto o valor de k. Assim, k é escolhido de tal forma que a desigualdade (3.19) é verificada.

Demonstração do Teorema 3.34. Para demonstrar o item i). Suponha que $\|b_1\| < \sqrt[n]{p^{kl}/\|f\|^m}$. Pelo teorema 3.32, $h_0 \mid b_1$. Como $deg(b_1) \leq m$, segue que $deg(h_0) \leq m$.

Reciprocamente, suponha que $deg(h_0) \leq m$. Então $h_0 \in L_{m,h}$ e portanto, pela proposição 3.27, $\|b_1\|^2 \leq 2^m \|h_0\|^2$. Assim, pelo lema 3.33, temos

$$\|b_1\| \leq 2^{m/2}\|h_0\| \leq 2^{m/2}\binom{2m}{m}^{1/2}\|f\|.$$

Logo

$$\|b_1\|^n \|f\|^m \leq 2^{mn/2}\binom{2m}{m}^{n/2}\|f\|^{m+n} < p^{kl},$$

ou seja, $\|b_1\| \leq \sqrt[n]{p^{kl}/\|f\|^m}$.

Para demonstrar ii), seja

$$J = \left\{ j \ : \ 1 \leq j \leq m+1 \ e \ \|b_j\| < \sqrt[n]{p^{kl}/\|f\|^m} \right\}$$

e seja $t = \max\{j \in J\}$. Então, segundo o teorema 3.32, $h_0 \mid b_j, \ \forall \, j \in J$, e portanto, $h_0 \mid h_1 := mdc(b_j, \ j \in J)$. Cada $b_j, \ j \in J$, é divisível por h_1 e $deg(b_j) \leq m$. Logo b_j pertence ao reticulado

$$\mathbb{Z}h_1 + \mathbb{Z}h_1 x + \cdots + \mathbb{Z}h_1 x^{m-deg(h_1)},$$

de ordem $m - deg(h_1) + 1$. Além disso, os vetores b_j são linearmente independentes, pois são vetores da base. Assim

$$|J| \leq m - deg(h_1) + 1. \tag{3.20}$$

Por outro lado, considere o conjunto $\{h_0, h_0x, \ldots, h_0x^{m-deg(h_0)}\}$. Pelo lema 3.33, temos que

$$\|h_0x^i\| = \|h_0\| \leq \binom{2m}{m}^{1/2} \|f\| \text{ para } i \geq 0. \tag{3.21}$$

Além disso, como $deg(h_0x^i) \leq m$, para todo $0 \leq i \leq m - deg(h_0)$, temos que $h_0x^i \in L_{m,h}$. Utilizando a proposição 3.28, segue-se que

$$\|b_j\|^2 \leq \max_{1\leq i\leq m-deg(h_0)+1} \left\{\|b_i\|^2\right\} \leq 2^m \max_{1\leq i\leq m-deg(h_0)+1} \left\{\|h_0x^i\|^2\right\} = 2^m\|h_0\|^2$$

para todo $1 \leq j \leq m - deg(h_0) + 1$. Usando a equação (3.21), segue-se que $\|b_j\| \leq 2^{m/2}\binom{2m}{m}^{1/2}\|f\|$ para $1 \leq j \leq m - deg(h_0) + 1$, ou seja

$$\|b_j\| \leq \sqrt[n]{2^{mn/2}\binom{2m}{m}^{n/2}\|f\|^n} < \sqrt[n]{p^{kl}/\|f\|^m},$$

em vista da equação (3.19).

Assim, $1, 2, \ldots, m - deg(h_0) + 1 \in J$ e portanto,

$$|J| \geq m - deg(h_0) + 1. \tag{3.22}$$

Logo, de acordo com (3.20) e (3.22), segue-se que

$$m - deg(h_0) + 1 \leq |J| \leq m - deg(h_1) + 1$$

e portanto, $deg(h_1) \leq deg(h_0)$. Mas $h_0 \mid h_1$, logo $deg(h_0) = deg(h_1)$. Assim, $|J| = m - deg(h_0) + 1$ e como $1, 2, \ldots, m - deg(h_0) + 1 \in J$, segue-se que

$$J = \{1, 2, \ldots, m - deg(h_0) + 1\}.$$

Por definição,

$$t = \max\{j \in J\} = \max\{1, 2, \ldots, m - deg(h_0) + 1\} = m - deg(h_0) + 1.$$

Donde segue que $deg(h_0) = m - t + 1$.

Como $deg(h_0) = deg(h_1)$, temos $h_1 = ah_0$, para certo $a \in \mathbb{Z}$. Além disso, $deg(h_0) \leq m$ e por i), $h_0 \in L_{m,h}$. Para mostrar que $h_1 = \pm h_0$ (e portanto, $h_0 = mdc(b_j,\ j \in J)$), basta mostrar que h_1 é primitivo. Seja $j \in J$ arbitrário. Como h_0 é primitivo, visto que f é primitivo por hipótese e h_0 é um fator de f, segue-se que $h_0 \mid pp(b_j)$. Logo $h \mid pp(b_j)\ mod\ p$. Como $b_j \in L_{m,h}$, segue-se que $pp(b_j) \in L_{m,h}$, mas b_j é um elemento da base de $L_{m,h}$, ou seja, $b_j = pp(b_j)$. Assim, todo fator de b_j é primitivo, em particular, h_1. □

3.3.3 O Algoritmo LLL

Nesta seção, apresentamos o algoritmo LLL baseado no próprio artigo [28]. Uma descrição do algoritmo, também baseada no artigo original, pode ser encontrada em [37]. Uma terceira descrição do algoritmo pode ser encontrada em [17], embora um pouco diferente da original.

A diferença do algoritmo aqui apresentado para o algoritmo original, apresentado em [28] e [37], é que o algoritmo original é dividido em três subalgoritmos, enquanto aqui, esses subalgoritmos foram englobados em apenas um.

Algoritmo 3.35. *Algoritmo LLL para fatoração*

Entrada: *Polinômio $f \in \mathbb{Z}[x]$, $n = deg(f)$.*
Saída: *Um fator irredutível h_0 de f.*

1 *Escolha o menor primo p que não divide $res(f, f')$.*

2 *Obtenha a fatoração de f em $\mathbb{Z}/p\mathbb{Z}[x]$: $f \equiv lc(f)g_1g_2\cdots g_r\ mod\ p$, com $g_i \in \mathbb{Z}/p\mathbb{Z}[x]$ mônico e irredutível*

3 $m \leftarrow n-1, \quad h \leftarrow g_1, \quad g \leftarrow lc(f)g_2g_3\cdots g_r, \quad l \leftarrow deg(g_1)$

4 **se** $l = n$ **então**

retorne *"f irredutível"*

5 Seja k o menor inteiro tal que $p^{kl} > 2^{mn/2} \cdot \binom{2m}{m}^{n/2} \cdot \|f\|^{m+n}$

6 Use *Levantamento de Hensel* para calcular uma fatoração $f \equiv h'g' \; mod \; p^k$

7 Seja u o maior inteiro tal que $l \leq (n-1)/2^u$

8 **enquanto** $u \geq 0$ **fazer**

9 $m \leftarrow \lfloor (n-1)/2^u \rfloor$

Seja $(b_1, b_2, \ldots, b_{m+1})$ uma base reduzida para $(p^k x^0, \ldots, p^k x^{l-1}, h'x^0, \ldots, h'x^{m-l})$

10 **se** $\|b_1\| \geq \sqrt[n]{p^{kl}/\|f\|^m}$ **então**

$u \leftarrow u - 1$

senão

$T \leftarrow \max\{j \; : \; \|b_j\| < \sqrt[n]{p^{kl}/\|f\|^m}\}$

$h_0 \leftarrow mdc(b_1, b_2, \ldots, b_T)$

retorne h_0

11 **se** $u = 0$ **então**

$h_0 \leftarrow f$

12 **retorne** h_0

A escolha da constante u no passo 7 é feita de acordo com o seguinte raciocínio. Lembre que a constante m está relacionada com o fator h_0 que estamos tentando encontrar. Podemos supor que h_0 é um fator não trivial de f assim, a

única coisa que sabemos é que $l \leq deg(h_0) \leq n-1$. Certamente, fazendo $m = n-1$ no passo 9, a condição do **se** do passo 10 não será atingida, e neste caso, o fator h_0 será calculado. Porém, como não sabemos o grau de h_0, vale a pena tentar usar um m menor do que $n-1$. Poderíamos começar com $m = l$ e, caso a condição do **se** no item 10 fosse satisfeita, incrementar m. Porém, desta forma estaríamos, provavelmente, calculando uma base reduzida várias vezes. Como veremos no exemplo a seguir, o cálculo dessas bases pode ser custoso, tornando o método ineficiente. Para tentar minimizar esse problema, escolhemos u tal que $l \leq (n-1)/2^u$. Assim, não começamos diretamente com $m = n-1$ e também não começamos com $m = l$, incrementando m toda vez que a condição do **se** for satisfeita. Por exemplo, se $l = 10$ e $n-1 = 44$, então $u = 2$. Neste caso, a menor escolha para m é, conforme o passo 9, $m = 11$. Caso $deg(h_0) > 11$, o algoritmo faz $u = 1$, resultando em $m = 22$. Se, mesmo assim $deg(h_0) > 22$, então $u = 0$ e $m = 44$. Neste ponto, teremos certeza de que o algoritmo encontrará o fator h_0.

Teorema 3.36. *O algoritmo* 3.35 *funciona corretamente. O número de operações aritméticas realizadas pelo algoritmo é* $\mathcal{O}(n^6 + n^5 \log(\|f\|))$. *Além disso, os inteiros para os quais essas operações são realizadas possuem tamanho* $\mathcal{O}(n^3 + n^2 \log(\|f\|))$.

Demonstração. Ver [37], Teorema 5.3.10. □

Vejamos um exemplo.

Exemplo 3.37. *Considere o polinômio* $f = x^4 + x^3 + 2x^2 + x + 1$ *e seja* $p = 41$. *Fatorando* f *mod* 41, *obtemos*

$$f \equiv (x+9) \cdot (x+32) \cdot (x^2 + x + 1) \; mod \; 41.$$

Escolhendo $h = x + 32$, *temos que* $k = 5$. *Utilizando o Levantamento de Hensel, obtemos a fatoração*

$$f \equiv (x + 46464143) \cdot (x^3 + 69392059x^2 + 69392059x + 69392058) \; mod \; 41^5.$$

Ou seja, $h' = x + 46464143$ *e* $g' = x^3 + 69392059x^2 + 69392059x + 69392058$. *O próximo passo do algoritmo é escolher* u *conforme o passo* 7 *do algoritmo. É facil ver que* $u = 1$. *Isso implica que, na primeira execução do* ***enquanto*** *no passo* 8, $m = 1$.

Para $m = 1$ *precisamos calcular uma base reduzida para o reticulado formado pelos vetores de*

$$41^5 x^0 \ \ e \ \ h'x^0,$$

ou seja, precisamos calcular uma base reduzida para o reticulado gerado pelos vetores $(41^5, 0)$ *e* $(46464143, 1)$. *Aplicando o algoritmo da seção anterior, temos que uma base reduzida para esse reticulado é dada por*

$$(-10475, -2476) \ \ e \ \ (2476, -10475).$$

Realizando os cálculos, vemos que o primeiro vetor dessa base satisfaz a condição do ***se*** *do passo* 10 *do algoritmo. Isso significa que a escolha de* m *não foi boa, isto é, o fator que estamos procurando possui grau maior do que* $m = 1$.

Seguindo o algoritmo, temos $u = 0$ *e consequentemente,* $m = 3$. *Agora, o reticulado é formado pelos vetores de* $41^5 x^0, h'x^0, h'x^1$ *e* $h'x^2$. *Isto é, precisamos calcular uma base reduzida para o reticulado formado por*

$$(41^5, 0, 0, 0), (46464143, 1, 0, 0), (0, 46464143, 1, 0) \ \ e \ \ (0, 0, 46464143, 1).$$

Novamente aplicando o algoritmo, vemos que uma base reduzida para esse reticulado é dada por

$$(1, 0, 1, 0), (0, 1, 0, 1), (-5238, -1238, 5237, 1238) \ \ e \ \ (1238, -5238, -1238, 5238).$$

Agora, $b_1 = (1, 0, 1, 0)$ *não satisfaz a condição do passo* 10, *isto quer dizer que seremos capazes de calcular o fator de* f. *Seguindo com o algoritmo, os únicos vetores que satisfazem a desigualdade* $\|b_j\| < \sqrt[n]{p^{kl}/\|f\|^m}$ *são* b_1 *e* $b_2 = (0, 1, 0, 1)$.

Assim, $T = 2$ e o fator de f é dado pelo máximo divisor comum de $h_1 = 1 + x^2$, correspondendo ao vetor b_1, e $h_2 = x + x^3$, correspondendo ao vetor b_2.Ou seja, o fator encontrado é $x^2 + 1$.

É claro que, para obter a fatoração completa do polinômio f, basta aplicar o algoritmo para $f/(x^2 + 1) = x^2 + x + 1$. Ao aplicarmos o algoritmo para $x^2 + x + 1$, porém, vemos que este é um polinômio irredutível. Assim, a fatoração de f em $\mathbb{Z}[x]$ é

$$f = (x^2 + 1) \cdot (x^2 + x + 1).$$

Ao analisarmos nossas implementações, realizadas no programa *Maple*, vemos que o algoritmo leva em torno de meio segundo para calcular a fatoração completa de f. Poderíamos dizer que este é um bom algoritmo para fatoração sobre $\mathbb{Z}[x]$ visto que, em teoria, este algoritmo tem complexidade polinomial. Entretanto, o grau do polinômio que usamos no exemplo acima é apenas 4. Consideremos o seguinte exemplo.

Exemplo 3.38. *Considere o polinômio*

$$f = x^{16} + 14x^{15} + 67x^{14} + 134x^{13} + 141x^{12} + 108x^{11} + 121x^{10} + 187x^9 + 221x^8 +$$
$$224x^7 + 182x^6 + 116x^5 + 98x^4 + 103x^3 + 82x^2 + 66x + 27.$$

O primeiro primo que satisfaz todas as condições descritas no algoritmo é $p = 7$ e

$$f \equiv (x^4 + 2x^3 + 6x^2 + 6x + 4) \cdot (x^4 + x^3 + x^2 + 5x + 2) \cdot$$

$$(x^4 + 5x^3 + 3x + 4) \cdot (x^4 + 6x^3 + 2x^2 + 5x + 5) \; mod \; 7.$$

Escolhemos $h = x^4 + 2x^3 + 6x^2 + 6x + 4$. Calculando o valor de k de acordo com o passo 5 do algoritmo, obtemos $k = 56$. Assim, precisamos aplicar o Levantamento de Hensel para obter uma fatoração de f módulo 7^{56}. Embora esse número seja da ordem de 10^{48}, o cálculo do levantamento de Hensel, neste caso, é feito rapidamente.

Porém, quando calculamos a base do reticulado, vemos que as entradas dos vetores também são da ordem de 10^{48}*, e isso faz com que o cálculo da base reduzida seja demorado. Em casos como esse, o método se torna pouco eficiente na prática. O capítulo a seguir mostra como contornar esse problema.*

Na verdade, para este exemplo em particular, precisamos calcular a redução de base de reticulados duas vezes. O tempo necessário para isso corresponde a $99,98\%$ *do tempo total para encontrar um fator do polinômio* f*.*

Depois de aproximadamente 15 *minutos de cálculos, o programa retorna o fator*

$$g = 9 + x + 7x^2 + 4x^3 + 2x^4 + 4x^5 + 9x^6 + 7x^7 + x^8.$$

O fator restante, $f/g = 3+7x+6x^2+4x^3+2x^4+4x^5+9x^6+7x^7+x^8$*, é irredutível, e o algoritmo não demora para acusar isto. Assim, a fatoração de* f *é*

$$f = (3+7x+6x^2+4x^3+2x^4+4x^5+9x^6+7x^7+x^8)\cdot(9+x+7x^2+4x^3+2x^4+4x^5+9x^6+7x^7+x^8).$$

É por esse motivo que este método, embora tenha complexidade polinomial, não substituiu o método de Berlekamp-Zassenhaus, descrito na Seção 3.2.2. O próximo capítulo apresenta o Algoritmo de van Hoeij. Embora este algoritmo também realize redução de bases de reticulados, os vetores do reticulado são outros, contendo informações não dos coeficientes de um fator mas da fatoração em si. Veremos isso com mais detalhes no próximo capítulo.

4 O ALGORITMO DE VAN HOEIJ

4.1 Introdução

Como vimos no capítulo anterior, o algoritmo LLL, embora tenha complexidade polinomial, não é melhor do que o algoritmo de Zassenhaus, cuja complexidade é exponencial, devido ao rápido crescimento das entradas nos vetores das bases dos reticulados.

Neste capítulo veremos o algoritmo de van Hoeij. Este algoritmo, embora também realize reduções de base, considera outras informações para montar o reticulado, fazendo que os vetores da base tenham entradas muito menores do que no caso anterior. Para mais detalhes, veja [22].

No algoritmo LLL, o vetor que estamos procurando é o vetor $v = (v_0, v_1, \ldots, v_n)$ tal que o polinômio $g_v = v_0 + v_1 x + \cdots + v_n x^n$ é um fator de f em $\mathbb{Z}[x]$. Como visto anteriormente, se o fator g_v tiver coeficientes arbitrariamente grandes, então o vetor v também terá e, após alguns cálculos, esses números podem ficar ainda maiores. Esse fato, além de reticulados com dimensão alta, tornam o algoritmo LLL ineficiente.

Van Hoeij, por outro lado, utiliza um vetor diferente em seus cálculos. Seja

$$f = f_1 f_2 \cdots f_k$$

a fatoração em fatores irredutíveis de f sobre o corpo dos inteiros p-ádicos. É possível mostrar que todo fator de f em $\mathbb{Z}[x]$ é o produto de alguns dos fatores f_i. Dessa forma, considere o vetor $\tilde{v} = (v_1, v_2, \ldots, v_k) \in \{0,1\}^k$ tal que o polinômio h definido por $h = \prod_{i=1}^{k} f_i^{v_i}$ é um fator irredutível de f em $\mathbb{Z}[x]$. Encontrar um fator de f em $\mathbb{Z}[x]$ é equivalente a encontrar qualquer um dos vetores v ou $\tilde{v}$. Porém, o vetor $\tilde{v}$

tem duas grandes vantagens sobre o vetor v: a primeira delas é que o vetor $\tilde{v}$ possui entradas em $\{0,1\}$, enquanto que as entradas de v são arbitrariamente grandes. A segunda vantagem é que o tamanho do vetor $\tilde{v}$ é igual ao número de fatores de $f \ mod \ p$, um número que é geralmente menor do que o tamanho do vetor v, igual ao grau do polinômio f.

Lembre que neste trabalho $\mathbb{Z}_p$ denota o conjunto dos inteiros p-ádicos, enquanto que $\mathbb{F}_p$ representa um corpo finito com p elementos e $\mathbb{Z}/p\mathbb{Z}$ representa os conjunto dos inteiros módulo p. Os fatores $f_1, f_2, \ldots, f_k \in \mathbb{Z}_p[x]$ serão chamados de fatores p-ádicos de f, enquanto que fatores em $\mathbb{Z}[x]$ serão chamados de fatores racionais de f. Além disso, os fatores de $f \mod p$ serão chamados de fatores modulares.

Veremos, nas próximas seções, como encontrar os vetores $\tilde{v}$.

4.2 Preliminares

4.2.1 O Traço de um Polinômio

Para encontrar condições lineares sobre os vetores $\tilde{v}$, não podemos utilizar diretamente o vetor $\tilde{v}$. Se $\tilde{v} = (v_1, v_2, \ldots, v_k), \tilde{w} = (w_1, w_2, \ldots, w_k) \in \{0,1\}^k$, então os polinômios $g_{\tilde{v}}$ e $g_{\tilde{w}}$, definidos por $g_{\tilde{v}} = \prod_{i=1}^{k} f_i^{v_i}$ e $g_{\tilde{w}} = \prod_{i=1}^{k} f_i^{w_i}$, não dependem linearmente de $\tilde{v}$ e $\tilde{w}$ pois

$$g_{\tilde{v}+\tilde{w}} = \prod_{i=1}^{k} f_i^{v_i+w_i} = \prod_{i=1}^{k} f_i^{v_i} \prod_{i=1}^{k} f_i^{w_i} = g_{\tilde{v}} g_{\tilde{w}}.$$

Para resolver este problema, introduz-se o conceito de *traço de um polinômio*.

Definição 4.1. *O i-ésimo traço $Tr_i(g)$ do polinômio g é definido como a soma das i-ésimas potências das raízes de g, contadas com multiplicidade, isto é, se $deg(g) = d$*

e $x_1, x_2, \ldots, x_d$ *são as raízes de* g *em seu corpo de decomposição, então*

$$Tr_i(g) = x_1^i + x_2^i + \cdots + x_d^i.$$

É fácil ver que

$$Tr_i(g_1 g_2) = Tr_i(g_1) + Tr_i(g_2), \tag{4.1}$$

para quaisquer polinômios $g_1, g_2 \in \mathbb{Z}_p[x]$. Esta propriedade será responsável pela linearidade que procuramos.

Seja $g \in \mathbb{Z}_p[x]$ um fator de grau d do polinômio f. Procuraremos agora condições sobre $Tr_i(g)$ que nos digam quando $g \in \mathbb{Z}[x]$. Sejam $x_1, x_2, \ldots, x_d$ as raízes de g no fecho algébrico de $\mathbb{Z}_p$. Assim, $g = \prod_{i=1}^{d}(x - x_i)$. Expandindo esse produto, podemos escrever

$$g = x^d + \overline{E}_1 x^{d-1} + \cdots + \overline{E}_d x^0, \tag{4.2}$$

onde $\overline{E}_i = (-1)^i E_i$ e E_i é o i-ésimo polinômio simétrico nas "variáveis" $x_1, x_2, \ldots, x_d$, isto é

$$\begin{aligned}
E_1 &= \sum_{i=1}^{d} x_i = x_1 + x_2 + \cdots + x_d \\
E_2 &= \sum_{1 \le i < j \le d} x_i x_j \\
&\vdots \\
E_i &= \sum_{1 \le j_1 < j_2 < \cdots < j_i \le d} x_{j_1} x_{j_2} \cdots x_{j_i} \\
&\vdots \\
E_d &= x_1 x_2 \cdots x_d.
\end{aligned}$$

Assim, o fator $g \in \mathbb{Z}_p[x]$ é um fator racional de f, isto é, $g \in \mathbb{Z}[x]$ se, e somente se, $E_i \in \mathbb{Z}$, $1 \le i \le d$. Denote por

$$P_i = P_i(x_1, x_2, \ldots, x_d) = x_1^i + x_2^i + \cdots + x_d^i$$

e note que $Tr_i(g) = P_i$. Usando as identidades de Newton

$$P_i = -i\overline{E}_i - \sum_{k=1}^{i-1} P_k \overline{E}_{i-k} \;\; e \;\; i\overline{E}_i = -P_i - \sum_{k=1}^{i-1} P_k \overline{E}_{i-k} \tag{4.3}$$

é possível mostrar que

$$\mathbb{Q}[E_1, E_2, \ldots, E_d] = \mathbb{Q}[P_1, P_2, \ldots, P_d], \tag{4.4}$$

considerando E_i, P_i, $1 \leq i \leq d$, como polinômios nas variáveis $x_1, x_2, \ldots, x_d$ e $\mathbb{Q}[E_1, E_2, \ldots, E_d]$, $\mathbb{Q}[P_1, P_2, \ldots, P_d]$ como $\mathbb{Q}$-álgebras geradas por $E_1, E_2, \ldots, E_d$ e $P_1, P_2, \ldots, P_d$, respectivamente. Para mais detalhes sobre as Identidades de Newton e uma demonstração de (4.4), veja [12], Capítulo 7.

Temos, portanto, o seguinte lema.

Lema 4.2. *Um polinômio mônico $g \in \mathbb{Z}_p[x]$, de grau d, tem coeficientes racionais, isto é, $g \in \mathbb{Q}[x]$, se, e somente se, $Tr_i(g) \in \mathbb{Q}$, $1 \leq i \leq d$.*

Demonstração. De fato, vimos que $g = x^d + \overline{E}_1 x^{d-1} + \cdots + \overline{E}_d x^0$. Assim, $g \in \mathbb{Q}[x]$ se, e somente se, cada E_i é um número racional. Pela equação (4.4), segue-se que cada E_i é gerado, como polinômio nas variáveis $x_1, x_2, \ldots, x_d$ e com coeficientes em $\mathbb{Q}$, por $P_1, P_2, \ldots, P_d$. Assim, $E_i \in \mathbb{Q}$ se, e somente se $P_i \in \mathbb{Q}$, ou seja, se, e somente se $Tr_i(g) \in \mathbb{Q}$, $1 \leq i \leq d$. □

Defina o vetor

$$Tr_{1..d}(g) = \begin{pmatrix} Tr_1(g) \\ Tr_2(g) \\ \vdots \\ Tr_d(g) \end{pmatrix}.$$

Em vista do Lema anterior, temos o seguinte resultado.

Lema 4.3. *Seja $f \in \mathbb{Z}[x]$ um polinômio mônico e de grau n. Seja $d = \lfloor n/2 \rfloor$. Então para qualquer fator mônico $g \in \mathbb{Z}_p[x]$ de f, são equivalentes*

i) $g \in \mathbb{Z}[x]$

ii) $Tr_{1..d}(g) \in \mathbb{Z}^d$

iii) $Tr_{1..d}(g) \in \mathbb{Q}^d$.

Demonstração. É claro que $i) \Rightarrow ii) \Rightarrow iii)$.

Suponha $iii)$. Vamos mostrar que $i)$ vale. Se $deg(g) \leq d$, então $g \in \mathbb{Q}[x]$ pelo Lema 4.2. Assim, temos que $g \in \mathbb{Q}[x]$ é um fator de $f \in \mathbb{Z}[x]$, ou seja, existe $h \in \mathbb{Q}[x]$ mônico tal que $f = gh$. Sejam $a, b \in \mathbb{Z}$ os menores inteiros positivos tais que $ag, bh \in \mathbb{Z}[x]$. segue-se que

$$ab = cont(abf) = cont(abgh) = cont(ag).cont(bh) = 1 \times 1 = 1,$$

onde $cont(p(x))$ é o conteúdo do polinômio $p(x)$, ou seja, o fator comum dos coeficientes de $p(x) \in \mathbb{Z}[x]$. Ora, $ab = 1$ implica que $a = b = 1$ ou $a = b = -1$. Em ambos os casos, $g \in \mathbb{Z}[x]$.

Por outro lado, se $deg(g) > d$, defina $h = f/g$. Assim $deg(h) \leq d$. Além disso, como $f \in \mathbb{Z}[x]$, segue-se que $Tr_{1..d}(f) \in \mathbb{Z}^d$ e, por hipótese, $Tr_{1..d}(g) \in \mathbb{Q}^d$. Assim, $Tr_{1..d}(h) = Tr_{1..d}(f) - Tr_{1..d}(g) \in \mathbb{Q}^d$. Podemos, portanto, aplicar o mesmo raciocínio anterior e concluir que $h \in \mathbb{Z}[x]$. Utilizando um raciocínio análogo ao acima, podemos concluir que $g \in \mathbb{Z}[x]$. □

Temos assim uma condição necessária e suficiente para decidir se o fator $g \in \mathbb{Z}_p[x]$ é um fator com coeficientes inteiros ou não. A seguir, iremos enfraquecer estas condições, multiplicando o vetor $Tr_{1..d}(g)$ por uma matriz com entradas inteiras. Seja $A \in \mathbb{Z}^{s \times d}$ uma matriz com entradas inteiras. Defina

$$T_A(g) = A \cdot Tr_{1..d}(g) = A \cdot \begin{pmatrix} Tr_1(g) \\ Tr_2(g) \\ \vdots \\ Tr_d(g) \end{pmatrix}.$$

Lema 4.4. *Seja $g \in \mathbb{Z}_p[x]$ um fator mônico de $f \in \mathbb{Z}[x]$. Então*

$$g \in \mathbb{Z}[x] \Rightarrow T_A(g) \in \mathbb{Z}^s.$$

Além disso, se a matriz A é tal que o espaço linha de A sobre $\mathbb{Q}$ contém os primeiros $d = \lfloor n/2 \rfloor$ vetores da base canônica, então a recíproca também vale.

Demonstração. Se $g \in \mathbb{Z}[x]$, então $Tr_{1..d}(g) \in \mathbb{Z}^d$ pelo Lema 4.3. Logo, $T_A(g) = A \cdot Tr_{1..d}(g)$ é o produto de uma matriz de inteiros por um vetor de inteiros, ou seja, $T_A(g) \in \mathbb{Z}^s$.

Reciprocamente, suponha que o espaço linha da matriz A contém os $d = \lfloor n/2 \rfloor$ primeiros vetores da base canônica. Seja $e_i, 1 \leq i \leq d$ um desses vetores. segue-se que existem $\alpha_1, \alpha_2, \ldots, \alpha_s \in \mathbb{Q}$ tais que

$$\sum_{i=1}^{s} \alpha_i a_i = e_i, \tag{4.5}$$

onde a_i é a i-ésima linha da matriz A. Além disso,

$$T_A(g) \in \mathbb{Z}^s \Rightarrow A \cdot Tr_{1..d}(g) = A \cdot \begin{pmatrix} Tr_1(g) \\ Tr_2(g) \\ \vdots \\ Tr_d(g) \end{pmatrix} \in \mathbb{Z}^s \Rightarrow a_i \cdot Tr_{1..d}(g) \in \mathbb{Z}, 1 \leq i \leq s. \tag{4.6}$$

Multiplicando (4.5) por $Tr_{1..d}(g)$ obtemos, do lado direito, $Tr_i(g)$. O lado esquerdo resulta em um número que, por (4.6), está em $\mathbb{Q}$. Assim, $Tr_1(g), Tr_2(g), \ldots, Tr_d(g) \in \mathbb{Q}$, ou seja, $Tr_{1..d}(g) \in \mathbb{Q}^d$. Logo, pelo Lema 4.3, segue-se que $g \in \mathbb{Z}[x]$ □

Seja S um subconjunto dos fatores p-ádicos de f e seja $g \in \mathbb{Z}_p[x]$ definido por $g = \prod_{f_i \in S} f_i$. Devido à propriedade aditiva do traço, segue-se que

$$T_A(g) = \sum_{f_i \in S} T_A(f_i).$$

Portanto, uma condição necessária para que g seja um fator racional de f é que a soma dos $T_A(f_i),\ f_i \in S$, tenha entradas inteiras. Além disso, se a matriz A satisfaz a condição do Lema 4.4, então esta condição é também suficiente.

Porém, como podemos decidir isso se, na prática, apenas podemos calcular aproximações dos fatores f_i, $i = 1, 2, \ldots, k$?

4.2.2 Aproximando Fatores p-ádicos

Computacionalmente, podemos calcular apenas aproximações dos fatores $f_i \in \mathbb{Z}_p[x]$ módulo p^a, para $a \in \mathbb{N}$. Estas aproximações, $\mathcal{C}^a(f_i)$, são chamadas de fatores modulares de f. Por exemplo, para $f \in \mathbb{Z}[x]$, podemos ver uma fatoração de $f \ mod\ p$ como uma aproximação de ordem 1 da fatoração $f = f_1 f_2 \ldots f_k$ em $\mathbb{Z}_p[x]$. Utilizando o Levantamento de Hensel, podemos calcular uma fatoração módulo p^a, para $a \in \mathbb{N}$. Da mesma forma, podemos ver esta fatoração como uma aproximação de ordem a da fatoração $f = f_1 f_2 \ldots f_k$ em $\mathbb{Z}_p[x]$.

Todo inteiro p-ádico pode ser representado por potências positivas em p, cujos coeficientes são inteiros não negativos menores do que p. Assim, o mesmo acontece com as entradas de $T_A(g) \in \mathbb{Z}_p^s$. Portanto, se quisermos escrever um algoritmo que utilize esses vetores, precisamos tornar essas séries finitas.

Definição 4.5. *Seja $c \in \mathbb{Z}_p$ um número p-ádico e $a \in \mathbb{Z}$ um inteiro positivo. Defina o resto simétrico $\mathcal{C}^a(c)$ de c módulo p^a como o único inteiro*

$$-p^a/2 < \mathcal{C}^a(c) \leq p^a/2$$

que é congruente a $c \ mod\ p^a$.

Isto é, se

$$c = a_0 + a_1 p + a_2 p^2 + \cdots + a_n p^n + \cdots,$$

onde $c_0, c_1, \ldots, c_n, \ldots \in \{0, 1, \ldots, p-1\}$, então para encontrar $\mathcal{C}^a(c)$, primeiro desconsideramos todas as potências p^α, para $\alpha \geq a$, obtendo,

$$a_0 + a_1 p + a_2 p^2 + \cdots + a_{a-1} p^{a-1}.$$

A seguir, como $a_i \leq p-1$, segue-se que $|a_0 + a_1 p + a_2 p^2 + \cdots + a_{a-1} p^{a-1}| < p^a$. Assim, existe um único inteiro $\mathcal{C}^a(c)$ tal que

$$-\frac{p^a}{2} < \mathcal{C}^a(c) \leq \frac{p^a}{2}$$

e $c \equiv \mathcal{C}^a(c) \; mod \; p^a$.

Dessa forma, podemos tornar o vetor $T_A(g)$ finito, aplicando $\mathcal{C}^a(\ )$ em suas entradas, para certo inteiro a. Além disso, se para algum vetor $v = (v_1, v_2, \ldots, v_k) \in \{0,1\}^k$, o polinômio $g = \prod_{i=1}^{k} f_i^{v_i} \in \mathbb{Z}_p[x]$ é um fator racional de f, então as entradas de $T_A(g)$ são números inteiros e limitados por $\frac{1}{2}p^b$, para certo inteiro b. De acordo com (4.2) e (4.3), as entradas de $T_A(g)$ contêm informações sobre os coeficientes do fator g de f. Como os vetores que estamos procurando são vetores de zeros e uns e que não levam em consideração qualquer informação sobre os coeficientes dos fatores de f, é inútil ter essa informação no reticulado. Por isso, introduzimos a seguinte definição

Definição 4.6. *Seja $c \in \mathbb{Z}_p$ um número p-ádico e sejam $a \geq b \geq 0$ inteiros. Defina $\mathcal{C}_b^a(c)$ como*

$$\mathcal{C}_b^a(c) = \mathcal{C}^{a-b}\left(\frac{c - \mathcal{C}^b(c)}{p^b}\right).$$

Isto é, para encontrar $\mathcal{C}_b^a(c)$, remova as potências p^i para $i \geq a$ e $i < b$. Após, divida por p^b e considere o resto simétrico módulo p^{a-b}.

Assim, além de transformar qualquer número p-ádico, que é uma série infinita em p, em uma expressão finita, $\mathcal{C}_b^a$ tem a função de ignorar informações sobre os coeficientes dos fatores de f, desconsiderando a parte inicial da série.

A seguir veremos quais valores para a e b devemos tomar. Seja B_{rt} uma cota para o módulo das raízes complexas do polinômio f. Então, para qualquer fator racional g de f de grau $d^* \leq d$, temos

$$|Tr_i(g)| = |y_1^i + y_2^i + \cdots + y_{d^*}^i| \leq |y_1|^i + |y_2|^i + \cdots + |y_{d^*}|^i \leq d^* B_{rt}^i \leq dB_{rt}^i,$$

onde $y_1, y_2, \ldots, y_{d^*}$ são as raízes de g. Portanto, conhecendo a matriz A, podemos encontrar cotas para cada uma das entradas do vetor $T_A(g) = A \cdot Tr_{1..d}(g)$.

Seja B_i uma cota para a i-ésima entrada de $T_A(g)$, para qualquer fator racional g de f. Assim, dado um subconjunto S dos fatores p-ádicos irredutíveis de f, e g o produto desses fatores, uma condição necessária para que $g \in \mathbb{Z}[x]$ é que cada entrada de $T_A(g)$ seja menor, em módulo, do que a respectiva cota B_i, calculada acima.

Seja $b = (b_1, b_2, \ldots, b_s)$ uma lista de inteiros positivos tais que $B_i < \frac{1}{2}p^{b_i}$. Definiremos, a seguir, o vetor $T_A^b(g)$, que é responsável por desconsiderar essa informação em $T_A(g)$ sobre os coeficientes do fator g de f.

Definição 4.7. *Seja $g \in \mathbb{Z}_p[x]$ um polinômio mônico. Seja r_i a i-ésima entrada do vetor $T_A(g)$ e sejam $(b_1, b_2, \ldots, b_s)$ como acima. Seja $\overline{r}_i$ o resto simétrico de r_i módulo p^{b_i}, ou seja, $\overline{r}_i = \mathcal{C}^{b_i}(r_i)$. Assim, $r_i - \overline{r}_i$ é divisível por p^{b_i}. Defina o vetor $T_A^b(g) = (u_1, u_2, \ldots, u_s)$, onde*

$$u_i = \frac{r_i - \overline{r}_i}{p^{b_i}}. \tag{4.7}$$

A seguir apresentaremos uma série de propriedades do vetor T_A^b.

Lema 4.8. *Seja $g \in \mathbb{Z}_p[x]$ um fator mônico de $f \in \mathbb{Z}[x]$. Então*

$$g \in \mathbb{Z}[x] \Rightarrow T_A^b(g) = 0.$$

Além disso, se A satisfaz a condição do Lema 4.4, então a recíproca também vale.

Demonstração. Se $g \in \mathbb{Z}[x]$ é um fator racional de f, então a i-ésima entrada de $T_A(g)$ é limitada por $B_i < \frac{1}{2}p^{b_i}$. Assim, $r_i = \overline{r}_i$ e portanto, a i-ésima entrada de $T_A^b(g)$ é zero. Ou seja, $T_A^b(g) = 0$. Reciprocamente, suponha que A satisfaça a condição do Lema 4.4 e que $T_A^b(g) = 0$. Afirmamos que as entradas de $T_A(g)$ são números inteiros. De fato, a i-ésima entrada de $T_A^b(g)$ é zero se, e somente se, $u_i = 0$, ou seja, se e somente se $(r_i - \overline{r}_i)/p^{b_i} = 0$. Assim, $r_i = \overline{r}_i$. Como $\overline{r}_i$ é um inteiro, segue-se que r_i também o é. Ou seja, $T_A(g) \in \mathbb{Z}^s$. Ora, pelo Lema 4.4, segue-se que $g \in \mathbb{Z}[x]$. □

A principal diferença entre $T_A(g)$ e $T_A^b(g)$ é a seguinte. Enquanto $T_A(g)$ nos fornece informações parciais (ou completas, caso A satisfaça a condição do Lema 4.4) sobre os coeficientes do fator g de f, parte dessa informação é descartada na definição de $T_A^b(g)$, pois tudo que é menor do que B_i é arredondado para zero. Por causa disso,

$$T_A^b(f_1 f_2) \neq T_A^b(f_1) + T_A^b(f_2),$$

ou seja, desconsiderando essa informação, deixamos de ter a linearidade de T_A^b. Porém, ainda conseguiremos igualdade introduzindo um erro ϵ, como veremos no lema abaixo.

Lema 4.9. *Seja $S \in \{f_1, f_2, \ldots, f_k\}$ um subconjunto do conjunto dos fatores p-ádicos de $f \in \mathbb{Z}[x]$ e seja $g \in \mathbb{Z}_p[x]$ o produto deles. Então*

$$T_A^b(g) = \epsilon + \sum_{f_i \in S} T_A^b(f_i),$$

onde $\epsilon = (\epsilon_1, \epsilon_2, \ldots, \epsilon_s)^T \in \mathbb{Z}^s$ e $\mid \epsilon_i \mid \leq \frac{|S|}{2}$.

Demonstração. Antes de demonstrar o caso geral, vamos supor que $g = f_1 f_2$. Então $T_A(g) = T_A(f_1) + T_A(f_2)$. Considere a i-ésima entrada desses vetores e denote-as, respectivamente, por r_g, r_{f_1} e r_{f_2}. Assim, a i-ésima entrada dessa equação vetorial fica

$$r_g = r_{f_1} + r_{f_2}. \tag{4.8}$$

Vamos agora chegar na i-ésima entrada do vetor $T_A^b(g)$, dada por (4.7). Divida toda a equação (4.8) por p^{b_i}. Após, some e subtraia $\overline{r}_*/p^{b_i}$ para cada uma das parcelas de (4.8). Temos, portanto

$$\frac{r_g - \overline{r}_g}{p^{b_i}} + \frac{\overline{r}_g}{p^{b_i}} = \frac{r_{f_1} - \overline{r}_{f_1}}{p^{b_i}} + \frac{\overline{r}_{f_1}}{p^{b_i}} + \frac{r_{f_2} - \overline{r}_{f_2}}{p^{b_i}} + \frac{\overline{r}_{f_2}}{p^{b_i}}.$$

Note que as parcelas $\frac{r_* - \overline{r}_*}{p^{b_i}}$ são as i-ésimas entradas de $T_A^b(*)$, ou seja, o erro, neste caso, é

$$\epsilon_i = -\frac{\overline{r}_g}{p^{b_i}} + \frac{\overline{r}_{f_1}}{p^{b_i}} + \frac{\overline{r}_{f_2}}{p^{b_i}}.$$

No caso geral, seja $S \subseteq \{f_1, f_2, \ldots, f_k\}$ e $g = \prod_{f_i \in S} f_i$. Neste caso, o erro é composto por $\mid S \mid +1$ parcelas do tipo $\frac{\overline{r}_*}{p^{b_i}}$(uma para cada elemento de S mais uma para o polinômio g). Além disso, note que

$$\left| \frac{\overline{r}_*}{p^{b_i}} \right| = \frac{|\overline{r}_*|}{p^{b_i}} < \frac{p^{b_i}/2}{p^{b_i}} = \frac{1}{2}.$$

Assim, o valor absoluto do erro ϵ_i não ultrapassa $\frac{|S|+1}{2}$, caso $p \neq 2$. Além disso, como todos os $\overline{r}_*$ são inteiros, $\epsilon_i \in \mathbb{Q}$.

Por outro lado, ϵ_i é a diferença das i-ésimas entradas de $T_A(g)$ e $T_A^b(g)$, ou seja, é um inteiro p-ádico. Logo, $\epsilon_i \in \mathbb{Z}_p \cap \mathbb{Q}$ e o único denominador possível é uma potência de p. Segundo [20], Proposição 3.3.4, segue-se que

$$\mathbb{Z}_p \cap \mathbb{Q} = \{a/b \in \mathbb{Q} \ : \ p \nmid b\}.$$

Ora, a única maneira de ϵ_i pertencer a esta interseção é sendo um inteiro, ou seja, $\epsilon_i \in \mathbb{Z}$. Como ϵ_i é um inteiro e $|\epsilon_i| < \frac{|S|+1}{2}$, segue-se que $|\epsilon_i| \leq \frac{|S|}{2}$.

Se $p = 2$, podemos ter $\frac{\overline{r}_{f_i}}{p^{b_i}} \leq 1/2$ porém, $|\epsilon_i| = \frac{|S|+1}{2}$ não ocorre. Caso contrário, deveríamos ter $\frac{\overline{r}_*}{p^{b_i}} = 1/2$ para todos os $f_i \in S$ e g. Logo

$$\epsilon_i = \sum_{f_i \in S} \frac{|\overline{r}_{f_i}|}{p^{b_i}} - \frac{|r_g|}{p^{b_i}} = |S|\frac{1}{2} - \frac{1}{2} \leq \frac{|S|}{2}.$$

□

Lema 4.10. *Seja $S \subseteq \{f_1, f_2, \ldots, f_k\}$ um subconjunto do conjunto de fatores p-ádicos de $f \in \mathbb{Z}[x]$ e seja $g \in \mathbb{Z}_p[x]$ o produto deles. Se $g \in \mathbb{Z}[x]$, então*

$$\epsilon + \sum_{f_i \in S} T_A^b(f_i) = 0 \tag{4.9}$$

para algum vetor $\epsilon \in \mathbb{Z}^s$ com entradas limitadas, em valor absoluto, por $|S|/2$. Além disso, se A satisfaz a condição do Lema 4.4, então a recíproca também vale.

Demonstração. Se $g \in \mathbb{Z}[x]$, então $T_A^b(g) = 0$ pelo Lema 4.8. Pelo Lema 4.9, segue a equação acima.

Reciprocamente, suponha que a equação acima seja válida e que a matriz A satisfaça a condição do Lema 4.4. Pelo Lema 4.9, existe $\epsilon' \in \mathbb{Z}^s$ tal que

$$T_A^b(g) = \epsilon' + \sum_{f_i \in S} T_A^b(f_i).$$

Logo, $T_A^b(g) = \epsilon' - \epsilon \in \mathbb{Z}^s$. Seja u_i a i-ésima entrada de $T_A^b(g)$. Por definição, $u_i = \frac{r_i - \overline{r}_i}{p^{b_i}}$, onde r_i é a i-ésima entrada de $T_A(g)$. Assim, para cada i, existe um inteiro $n_i \in \mathbb{Z}$ tal que

$$\frac{r_i - \overline{r}_i}{p^{b_i}} = n_i,$$

ou ainda, $r_i = \overline{r}_i + n_i p^{b_i} \in \mathbb{Z}$. Logo, segue também que $T_A(g) = A \cdot Tr_{1..d}(g) \in \mathbb{Z}^s$. Como $A \in \mathbb{Z}^{s \times d}$, segue-se que $Tr_{1..d}(g) \in \mathbb{Q}^d$ e, pelo Lema 4.3, segue-se que $g \in \mathbb{Z}[x]$. □

Iremos agora tratar da finitude das expressões dos números p-ádicos. Escolha inteiros a_i, $1 \leq i \leq s$, tais que $a_i > b_i$. Seja $\overline{c}_{j,i}$ a i-ésima entrada de $T_A(f_j)$ e $\tilde{c}_{j,i}$ a i-ésima entrada de $T_A^b(f_j)$. Defina

$$c_{j,i} = \mathcal{C}_{b_i}^{a_i}(\overline{c}_{j,i}) = \mathcal{C}^{a_i - b_i}(\tilde{c}_{j,i}) \tag{4.10}$$

e seja $C_j \in \mathbb{Z}^s$ definido como

$$C_j = (c_{j,1}, c_{j,2}, \ldots, c_{j,s}). \tag{4.11}$$

Em outras palavras, a i-ésima entrada de C_j é uma aproximação da i-ésima entrada de $T_A^b(f_j)$, com precisão $a_i - b_i$. Assim, para podermos calcular o vetor $T_A^b(f_j)$, precisamos levantar uma fatoração de $f \ mod \ p$ até p^a, onde $a > a_i$, $1 \leq i \leq s$.

Vamos agora reformular o Lema 4.10. Sejam $e_1, e_2, \ldots, e_s$ a base canônica de $\mathbb{Z}^s$.

Teorema 4.11 (The Factorization Knapsack Problem)**.** *Seja $f \in \mathbb{Z}[x]$ um polinômio mônico e livre de quadrados e sejam $f_1, f_2, \ldots, f_k$ seus fatores p-ádicos irredutíveis. Seja $A \in \mathbb{Z}^{s \times d}$ uma matriz $s \times d$ com entradas inteiras. Para todo $S \subseteq \{f_1, f_2, \ldots, f_k\}$, se o produto g dos elementos de S é um fator racional de f, então*

$$\sum_{i=1}^{s} \left(\epsilon_i + \gamma_i p^{a_i - b_i}\right) e_i + \sum_{i=1}^{k} v_i C_i = 0, \tag{4.12}$$

para inteiros ϵ_i, γ_i com valores absolutos no máximo $|S|/2$. Além disso, a_i e b_i são escolhidos conforme descrito desta seção e

$$v_i = \begin{cases} 1, & \text{se } f_i \in S \\ 0, & \text{em caso contrário.} \end{cases}$$

Além disso, se A satisfaz a condição do Lema 4.4, então a recíproca também é verdadeira, para a_i suficientemente grande.

Demonstração. Pelo Lema 4.10, existe $\epsilon \in \mathbb{Z}^s$ tal que

$$\epsilon + \sum_{f_j \in S} T_A^b(f_j) = 0.$$

A i-ésima entrada dessa equação vetorial é dada por $\epsilon_i + \sum_{f_l \in S} (T_A^b(f_l))_i = 0$. Ora, a i-ésima entrada de $T_A^b(f_l)$ é $\tilde{c}_{l,i}$. Assim

$$\epsilon_i + \sum_{f_l \in S} \tilde{c}_{l,i} = 0. \tag{4.13}$$

Pela definição, $c_{j,i} = \mathcal{C}^{a_i - b_i}(\tilde{c}_{j,i}) = \tilde{c}_{j,i} - \tilde{\gamma}_i p^{a_i - b_i}$, para certo $\tilde{\gamma}_i \in \mathbb{Z}_p$. Ou seja,

$$\tilde{c}_{j,i} = c_{j,i} + \tilde{\gamma}_i p^{a_i - b_i}, \ \tilde{\gamma}_i \in \mathbb{Z}_p. \tag{4.14}$$

Substituindo (4.14) em (4.13), temos

$$\epsilon_i + \sum_{f_l \in S} (c_{l,i} + \tilde{\gamma}_i p^{a_i - b_i}) = 0.$$

Note que, como $\epsilon_i, c_{l,i} \in \mathbb{Z}$, segue-se que $\tilde{\gamma}_i \in \mathbb{Z}, 1 \leq i \leq s$. Removendo do somatório o segundo termo (que não depende de l), temos

$$\epsilon_i + |S|\tilde{\gamma}_i p^{a_i - b_i} + \sum_{f_l \in S} c_{l,i} = 0.$$

Denote por γ_i o número $|S|\tilde{\gamma}_i$. Pela definição de v_i, podemos escrever

$$(\epsilon_i + \gamma_i p^{a_i - b_i}) + \sum_{l=1}^{k} v_l c_{l,i} = 0, \;\; 1 \leq i \leq s.$$

Ou ainda, em notação matricial,

$$\sum_{i=1}^{s} (\epsilon_i + \gamma_i p^{a_i - b_i}) e_i + \sum_{i=1}^{s} \left(\sum_{l=1}^{k} v_l c_{l,i} \right) e_i = 0.$$

Mudando a ordem dos somatórios, obtemos

$$\sum_{i=1}^{s} (\epsilon_i + \gamma_i p^{a_i - b_i}) e_i + \sum_{l=1}^{k} \left(v_l \sum_{i=1}^{s} c_{l,i} e_i \right) = 0.$$

Ora, $\sum_{i=1}^{s} c_{l,i} e_i$ é o vetor C_l. Assim, obtemos

$$\sum_{i=1}^{s} (\epsilon_i + \gamma_i p^{a_i - b_i}) e_i + \sum_{l=1}^{k} v_l C_l = 0.$$

Reciprocamente, para a_i suficientemente grande, então a equação (4.12) converge, na norma p-ádica, para a equação (4.9). Como A satisfaz a condição do Lema 4.4, segue, pelo Lema 4.10, que g é um fator racional de f.

□

Note que as incógnitas na equação (4.12) são os números $\epsilon_i, \gamma_i,\ 1 \leq i \leq s$, e $v_j,\ 1 \leq j \leq k$. Assim, se A satisfaz a condição do Lema 4.4 e

$$Sol = (\gamma_1, \ldots, \gamma_s, \epsilon_1, \ldots, \epsilon_s, v_1, \ldots, v_k) \tag{4.15}$$

é uma solução da equação (4.12), então o polinômio dado por

$$g = \prod_{i=1}^{k} f_i^{v_i}$$

é um fator racional de f.

Veremos, na próxima seção, como poderemos utilizar essa equação para encontrar esse vetor $v = (v_1, v_2, \ldots, v_k)$, que fornece um fator racional para f.

4.3 Criando o Reticulado

Seja $f \in \mathbb{Z}[x]$ e sejam $f_1, f_2, \ldots, f_k$ os fatores irredutíveis p-ádicos de f. Seja W o conjunto de todos os vetores $v = (v_1, v_2, \ldots, v_k) \in \{0,1\}^k$ tais que o polinômio g_v, definido por $g_v = \prod_{i=1}^{k} f_i^{v_i}$, está definido sobre $\mathbb{Z}[x]$, ou seja,

$$W = \{v = (v_1, v_2, \ldots, v_k) \in \{0,1\}^k \ :\ g_v = \prod_{i=1}^{k} f_i^{v_i} \in \mathbb{Z}[x]\}. \tag{4.16}$$

Assim, W é o conjunto dos vetores que representam todos os fatores de f em $\mathbb{Z}[x]$. Note que, se $g_1, g_2, \ldots, g_r$ são os fatores irredutíveis de f em $\mathbb{Z}[x]$, então os vetores $w_1, w_2, \ldots, w_r \in \{0,1\}^k$, tais que $w_i = (w_{i1}, w_{i2}, \ldots, w_{ik})$ e

$$g_i = \prod_{j=1}^{k} f_j^{w_{ij}},$$

formam uma base para W. De fato, se um fator g de f é dado por, digamos, $g_i g_j$, então o vetor de 0's e 1's correspondente desse fator é dado pela soma dos vetores w_i e w_j acima. Além disso, a matriz composta por esses vetores está na forma

escalonada reduzida. Como as entradas dos vetores $w_1, w_2, \ldots, w_r$ são 0's e 1's e se a entrada j de um vetor w_i é um, então as j-ésimas entradas de todos os outros vetores são 0. Dessa forma, podemos reordenar as linhas na matriz para obter uma matriz em sua forma escalonada reduzida.

Vejamos um exemplo. Seja $f \in \mathbb{Z}[x]$ e suponha que $f = f_1 f_2 f_3 f_4 f_5$ em $\mathbb{Z}_p[x]$ e que os fatores irredutíveis de f em $\mathbb{Z}[x]$ sejam dados por

$$g_1 = f_1 f_3 f_4, \;\; g_2 = f_2 \;\; e \;\; g_3 = f_5.$$

Então os vetores w_1, w_2 e w_3 são dados por

$$w_1 = (1,0,1,1,0), \;\; w_2 = (0,1,0,0,0) \;\; e \;\; w_3 = (0,0,0,0,1),$$

e a matriz formada por esses vetores é

$$\begin{pmatrix} 1 & 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{pmatrix},$$

que está na forma escalonada reduzida.

Defina

$$B = \{v = (v_1, v_2, \ldots, v_k) \in W \;:\; g_v = \prod_{i=1}^{k} f_i^{v_i} \text{ é irredutível em } \mathbb{Z}[x]\}.$$

Dessa forma, B é o conjunto dos vetores que representam todos os fatores irredutíveis de f em $\mathbb{Z}[x]$. Note que, devido ao fato de $\mathbb{Z}[x]$ ser um domínio de fatoração única, os vetores de W são somas dos vetores de B.

Usaremos as seguintes notações: Se $\mathcal{L} \subseteq \mathbb{Z}^n$ é um reticulado, então $B_{\mathcal{L}}$ denotará uma base para $\mathcal{L}$. Além disso, a matriz cujas linhas são os vetores dessa base será denotada por $(B_{\mathcal{L}})$ e a forma escalonada reduzida dessa matriz será denotada por $fer(B_{\mathcal{L}})$.

Assim, encontrando uma base B_W para W, também teremos uma base para B. De fato, como a forma escalonada reduzida é única, segue-se que $fer(B_W)$ é a matriz cujos vetores são uma base para B.

Lema 4.12. *Seja W o reticulado gerado pelo conjunto de vetores* (4.16). *Seja $\mathcal{L} \subseteq \mathbb{Z}^n$ um reticulado tal que $W \subseteq \mathcal{L} \subseteq \mathbb{Z}^n$. Seja $R = fer(B_{\mathcal{L}})$. Então $W = \mathcal{L}$ se, e somente se, as seguintes condições valem:*

i) Cada coluna de R contém exatamente um 1. *Todas as demais entradas são* 0.

ii) Se $(v_1, v_2, \ldots, v_k)$ é uma linha de R, então $\prod_{i=1}^{k} f_i^{v_i} \in \mathbb{Z}[x]$.

Demonstração. Se $W = \mathcal{L}$, então o espaço gerado pelas linhas de (B_W) e $(B_{\mathcal{L}})$ é o mesmo. Pela unicidade da forma escalonada reduzida, segue-se que $fer(B_W) = fer(B_{\mathcal{L}}) = R$. Ora, $fer(B_W)$ é formada pelos vetores w_i definidos acima. Assim, cada coluna de R contém precisamente um 1 e todas as outras entradas são zero. Além disso, como cada linha de R é um vetor w_i, o item ii) segue da definição de w_i.

Por outro lado, suponha que i) e ii) sejam verdadeiras. Como $W \subseteq \mathcal{L}$, segue-se que $w_1, w_2, \ldots, w_r$ devem ser combinações lineares das linhas de R. Lembre que os vetores w_i representam os fatores irredutíveis de f em $\mathbb{Z}[x]$. Assim, dizer que w_i é uma combinação das linhas de R (que também representam fatores de f em $\mathbb{Z}[x]$ por ii)) significa dizer que g_i pode ser escrito como um produto desses fatores. Ora, como g_i é irredutível em $\mathbb{Z}[x]$, segue-se que as linhas de R devem ser os próprios vetores w_i. Assim, $W = \mathcal{L}$. □

Assim, encontrando uma base para W, podemos calcular todos os fatores racionais de $f \in \mathbb{Z}[x]$. A ideia para encontrar essa base é a seguinte. Inicialmente, tomamos $\mathcal{L} = \mathbb{Z}^k$. Assim, $W \subseteq \mathcal{L}$. Se $W \neq \mathcal{L}$, a ideia é construir um reticulado

$\mathcal{L}'$ tal que $W \subseteq \mathcal{L}' \subseteq \mathcal{L}$ e usar o Lema 4.12 para verificar se $W = \mathcal{L}'$. Se $W \neq \mathcal{L}'$, substituímos $\mathcal{L}$ por $\mathcal{L}'$ e repetimos o processo, até obtermos $W = \mathcal{L}$. Uma vez feito isso, temos uma base para W e, consequentemente, uma base para B.

A próxima seção mostra como podemos calcular esse novo reticulado $\mathcal{L}'$. Antes disso, iremos mostrar um resultado simples mas importante que ajudará a entender o algoritmo. O algoritmo de van Hoeij está baseado em duas ideias fundamentais:

i) Criar um reticulado que contenha todas as soluções da equação (4.12). E que as soluções tenham algo de "diferente" dos vetores que não são uma solução de (4.12).

Como já falamos anteriormente, cada solução da equação (4.12) nos fornece um fator irredutível de f em $\mathbb{Z}[x]$. Além disso, os vetores que representam uma solução terão sua norma limitada por uma certa constante M, como veremos na próxima seção.

ii) Encontrar um modo de separar os vetores que são solução da equação (4.12) daqueles que não são.

Quanto ao segundo item, temos o seguinte resultado.

Lema 4.13. *Seja $\mathcal{L} \subseteq \mathbb{Z}^n$ um reticulado e $v \in \mathcal{L}$ tal que $\|v\| \leq M$, para alguma constante M. Sejam $b_1, b_2, \ldots, b_n$ uma base para $\mathcal{L}$ e $b_1^*, b_2^*, \ldots, b_n^*$ a base calculada pelo processo de ortogonalização de Gram-Schmidt. Se $\|b_n^*\| > M$ então $v \in Span_{\mathbb{Z}}\{b_1, b_2, \ldots, b_{n-1}\}$.*

Demonstração. Como $v \in \mathcal{L}$ e $b_1^*, b_2^*, \ldots, b_n^*$ também geram $\mathcal{L}$ (porém sobre $\mathbb{R}$), existem $\alpha_1, \alpha_2, \ldots, \alpha_n \in \mathbb{R}$ tais que

$$v = \sum_{i=1}^{n} \alpha_i b_i^*.$$

Seja I o maior inteiro tal que $\alpha_I \neq 0$. Se $I < n$, como $Span_{\mathbb{Z}}\{b_1^*, b_2^*, \ldots, b_I^*\} = Span_{\mathbb{Z}}\{b_1, b_2, \ldots, b_I\}$ (ver teorema 3.7-*ii*)) segue-se que $v \in Span_{\mathbb{Z}}\{b_1, b_2, \ldots, b_{n-1}\}$.

Suponha, por absurdo, que $I = n$. Aplicando o mesmo raciocínio empregado na Proposição 3.3.1, temos que $\alpha_n \in \mathbb{Z}$. Assim

$$\|v\| = \|\alpha_1 b_1^* + \alpha_2 b_2^* + \cdots + \alpha_n b_n^*\| = |\alpha_1|\|b_1^*\| + |\alpha_2|\|b_2^*\| + \cdots + |\alpha_n|\|b_n^*\| \geq |\alpha_n|\|b_n^*\|,$$

devido à ortogonalidade dos vetores b_i^*. Como $\alpha_n \neq 0$, por hipótese, e $\alpha_n \in \mathbb{Z}$, segue-se que

$$\|v\| \geq |\alpha_n|\|b_n^*\| \geq \|b_n^*\| > M.$$

Ou seja, $\|v\| > M$, uma contradição. Assim, $\alpha_n = 0$ e $v \in Span_{\mathbb{Z}}\{b_1^*, b_2^*, \ldots, b_{n-1}^*\}$. Novamente pela propriedade *ii*) do teorema 3.7, segue-se que

$$v \in Span_{\mathbb{Z}}\{b_1, b_2, \ldots, b_{n-1}\}.$$

□

De uma forma geral, pode-se provar que se $\mathcal{L} \subseteq \mathbb{R}^n$ é um reticulado gerado por $b_1, b_2, \ldots, b_n$ e existe $1 \leq k^* < n$ tal que $\|b_k^*\| > M$, para todo $k^* \leq k \leq n$, então para todo $v \in \mathcal{L}$ é tal que $\|v\| \leq M$, temos $v \in Span_{\mathbb{Z}}\{b_1, b_2, \ldots, b_{k^*-1}\}$.

Assim, se encontrarmos uma constante M tal que toda solução da equação (4.12) é limitada por M, então isto nos permitirá encontrar reticulados menores que ainda contenham todas as soluções da equação (4.12).

4.4 O Algoritmo de van Hoeij

Escolha uma matriz $A \in \mathbb{Z}^{s \times d}$ e inteiros a_i, b_i, $1 \leq i \leq s$, tais que

$$a_i > b_i > \log_p(2B_i),$$

onde B_i é uma cota para a i-ésima entrada de $T_A(g)$, para qualquer fator racional g de f. Considere $\mathcal{L} = \mathbb{Z}^k$. Assim, $W \subseteq \mathcal{L}$. Se $W \neq \mathcal{L}$, mostraremos, nesta seção, como calcular um reticulado $\mathcal{L}'$ tal que $W \subseteq \mathcal{L}' \subseteq \mathcal{L}$.

Para fazer isso, iremos construir um reticulado maior Λ, cuja projeção nas k primeiras coordenadas seja uma base para $\mathcal{L}$. Sejam $e_1, e_2, \ldots, e_s$ a base canônica para $\mathbb{Z}^s$. O vetor nulo $(0, 0, \ldots, 0) \in \mathbb{Z}^k$ será denotado por $0^k \in \mathbb{Z}^k$ e a notação $(v, w) \in \mathbb{Z}^{k+s}$ representa a concatenação dos vetores $v \in \mathbb{Z}^k$ e $w \in \mathbb{Z}^s$. O reticulado $\Lambda \subseteq \mathbb{Z}^{k+s}$ é definido pela base $B_\Lambda = B_C \cup B_{p^*}$, onde

$$B_{p^*} = \{(0^k, p^{a_i - b_i} e_i) \ : \ 1 \leq i \leq s\} \quad e \quad B_C = \{(Cv, vm) \ : \ v \in B_{\mathcal{L}}\}, \tag{4.17}$$

onde C é uma constante definida a seguir e m é uma matriz definida por

$$m = \begin{pmatrix} \mathcal{C}_1 \\ \mathcal{C}_2 \\ \vdots \\ \mathcal{C}_k \end{pmatrix} = \begin{pmatrix} c_{11} & c_{12} & \ldots & c_{1s} \\ c_{21} & c_{22} & \ldots & c_{2s} \\ \vdots & \vdots & \ddots & \vdots \\ c_{k1} & c_{k2} & \ldots & c_{ks} \end{pmatrix}, \tag{4.18}$$

onde C_i é o vetor linha definido em (4.11).

Inicialmente, quando $\mathcal{L} = \mathbb{Z}^k$, a base B_Λ de Λ é dada pela matriz

$$(B_\Lambda) = \begin{pmatrix} C & 0 & \ldots & 0 & c_{11} & \ldots & c_{1s} \\ 0 & C & \ldots & 0 & c_{21} & \ldots & c_{2s} \\ \vdots & \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \ldots & C & c_{k1} & \ldots & c_{ks} \\ 0 & 0 & \ldots & 0 & p^{a_1 - b_1} & \ldots & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \ldots & 0 & 0 & \ldots & p^{a_s - b_s} \end{pmatrix}.$$

O número l de elementos de B_Λ é igual a s adicionado do número de elementos de $B_{\mathcal{L}}$. Seja

$$M = \sqrt{C^2 k + s\frac{k^2}{4}}, \tag{4.19}$$

onde C é um inteiro positivo escolhido de tal forma que $C^2k \approx sK^2/4$. Lembre que uma solução da equação (4.12) é um vetor Sol da forma (4.15). Defina o vetor

$$v_{Sol} = (Cv_1, Cv_2, \ldots, Cv_k, -\epsilon_1, -\epsilon_2, \ldots, -\epsilon_s), \tag{4.20}$$

para toda solução Sol de (4.12). Note que não usamos as variáveis γ_i, $1 \leq i \leq s$. Note também que $v_{Sol} \in \Lambda$, para cada solução Sol de (4.12), pois temos

$$v_{Sol} = (v_1, v_2, \ldots, v_k, \gamma_1, \gamma_2, \ldots, \gamma_s)(B_\Lambda).$$

Isto quer dizer que v_{Sol} é uma combinação das linhas da matriz (B_Λ), com coeficientes $v_1, v_2, \ldots, v_k, \gamma_1, \gamma_2, \ldots, \gamma_s$. Além disso, note que

$$\|v_{Sol}\|^2 = (Cv_1)^2 + (Cv_2)^2 + \cdots + (Cv_k)^2 + \epsilon_1^2 + \epsilon_2^2 + \cdots + \epsilon_s^2.$$

Lembre que v_i é zero ou um e que o número de v_i's não nulos é $|S| \leq k$, onde $S = \{f_i \ : \ v_i = 1, \ 1 \leq i \leq k\}$. Além disso, $|\epsilon_i| \leq |S|/2$. Assim, temos

$$\|v_{Sol}\|^2 \leq C^2|S| + s\frac{|S|^2}{4} \leq C^2k + s\frac{k^2}{4} = M^2, \tag{4.21}$$

e portanto, todo vetor v_{Sol} em Λ correspondendo à uma solução Sol de (4.12) possui norma limitada por M.

Podemos usar o algoritmo de redução de base para encontrar uma base reduzida $V_1, V_2, \ldots, V_l$ para Λ. Sejam $V_1^*, V_2^*, \ldots, V_l^*$ a base calculada pelo processo de Gram-Schmidt. Seja $r \leq l$ o menor inteiro tal que $\|V_i^*\| > M$, para todo $r \leq i \leq l$. Defina Λ' como o reticulado gerado pelos vetores $\{V_i \ : \ i < r\}$ e defina $\mathcal{L}'$ como a projeção de $\frac{1}{C}\Lambda'$ nas primeiras k coordenadas.

Lema 4.14. *O reticulado $\mathcal{L}'$ definido acima é tal que $W \subseteq \mathcal{L}' \subseteq \mathcal{L}$.*

Demonstração. Como Λ' é definido com apenas alguns vetores da base de Λ, segue-se que $\mathcal{L}' \subseteq \mathcal{L}$. Além disso, segundo o Lema 4.13, se $\|V_i^*\| > M$, para todo $k \geq r$, então todos os vetores com norma menor ou igual à M estão em $\Lambda' = Span_{\mathbb{Z}}\{V_i \ : \ i < r\}$.

Ou seja, se Sol é uma solução de (4.12) e v_{Sol} o vetor em Λ associado a essa solução, então $\|v_{Sol}\| \leq M$ e portanto, $v_{Sol} \in \Lambda'$. Assim, se as entradas $v_1, v_2, \ldots, v_k$ de Sol correspondem ao fator irredutível $g = \prod_{i=1}^{k} f_i^{v_i}$ de f em $\mathbb{Z}[x]$, então $(v_1, v_2, \ldots, v_k)$ é dado por $1/C$ vezes a projeção de v_S nas k primeiras coordenadas. Como $v_{Sol} \in \Lambda'$, segue-se que $(v_1, v_2, \ldots, v_k) \in \mathcal{L}'$. Ou seja, $W \subseteq \mathcal{L}'$. □

Se $r < dim(\mathcal{L})$, então o algoritmo progride, pois

$$dim(\mathcal{L}') \leq r < dim(\mathcal{L}).$$

O cálculo de uma base reduzida para Λ é necessário pois, sem ele, certamente teríamos $r \geq dim(\mathcal{L})$. Todas essas ideias podem ser reunidas no seguinte algoritmo.

Algoritmo 4.15. *Algoritmo de van Hoeij*

Entrada: *Um polinômio mônico e livre de quadrados $f \in \mathbb{Z}[x]$ de grau n.*

Saída: *A fatoração de f em fatores irredutíveis.*

1 *Calcule uma fatoração para f mod p e denote seus fatores por $f_1, f_2, \ldots, f_k$*

2 $a \leftarrow \lceil \log_p(2^n \cdot \sqrt{n+1} \cdot \|f\|_\infty) \rceil$

3 *Utilize o Levantamento de Hensel para obter uma fatoração de f mod p^a. Denote seus fatores por $f_1^{(a)}, f_2^{(a)}, \ldots, f_k^{(a)}$*

4 *Seja $B_L = \{e_1, e_2, \ldots, e_k\}$ a base canônica para $\mathbb{Z}^k$*

5 *Escolha uma matriz $A \in \mathbb{Z}^{s \times d}$, para certo inteiro $s \geq 1$ e $d = \lfloor n/2 \rfloor$*

6 Calcule uma cota B_i para a i-ésima entrada de $T_A(g)$, para todo fator racional g de f

7 Escolha inteiros $a > a_i > b_i > \log_p(2B_i)$, $1 \leq i \leq s$

8 Calcule a base para Λ dada por (4.17)

9 Utilize o algoritmo de Redução de Base para encontrar uma base reduzida $v_1, v_2, \ldots, v_l$ para Λ

10 Calcule uma base ortogonal $v_1^*, v_2^*, \ldots, v_l^*$ para $v_1, v_2, \ldots, v_l$, utilizando o processo de ortogonalização de Gram-Schmidt

11 Seja $M = \sqrt{C^2 k + s\frac{k^2}{4}}$ e seja r o menor inteiro tal que $\|v_i^*\| > M$, para todo $i > r$

12 Defina $\Lambda' = Span_{\mathbb{Z}}(v_1, v_2, \ldots, v_r)$ e $\mathcal{L}'$ é gerado pela projeção dos vetores de $\frac{1}{C}\Lambda'$ nas k primeiras coordenadas

13a Seja $R = fer(B_{\mathcal{L}'})$. Verifique se R satisfaz a condição $i)$ do Lema 4.12.

13b Para cada linha $l_j = (l_{j_1}, l_{j_2}, \ldots, l_{j_k})$ de R, $1 \leq j \leq r$, verifique se o polinômio g_j, definido por $g_j = \prod_{i=1}^{k} \left(f_i^{(a)}\right)^{l_{j_i}} \ mods\ p^a$, é um divisor de f em $\mathbb{Z}[x]$

14 Caso ambos os passos $13a$ e $13b$ sejam satisfeitos, então $f = g_1 g_2 \cdots g_r$ é a fatoração de f em $\mathbb{Z}[x]$ em fatores irredutíveis

15 **retorna** $g_1, g_2, \ldots, g_r$.

É possível demonstrar que este algoritmo termina. Para uma demonstração deste fato, veja [22]. Para uma demonstração sobre a complexidade do algoritmo, veja [4].

O primo p é escolhido tal que $f \ mod \ p$ é livre de quadrados. Também é possível escolher p de forma a reduzir o número de fatores de $f \ mod \ p$. A constante a do passo 2 é a cota de Mignotte do Teorema 3.3. Caso a dimensão r do novo reticulado $\mathcal{L}'$, calculado no passo 12, não seja menor do que a dimensão de $\mathcal{L}$, então podemos usar valores maiores para a diferença $a_i - b_i$. Caso o passo $13a$ ou $13b$ não se verifique, é preciso utilizar uma nova matriz A no passo 5.

Existem várias estratégias para a escolha da matriz A no passo 5. Uma delas é escolher $d = s$, para algum $s > 1$, e tomar A como a matriz identidade $s \times s$. Uma segunda estratégia é utilizar uma matriz cujas entradas são inteiros aleatórios. Na segunda estratégia, vários Tr_i's são combinados numa única entrada de $T_A(g)$, fazendo com que o reticulado tenha mais informação sobre os fatores de f. Se tomamos um inteiro pequeno $s > 1$ e $d = \lfloor n/2 \rfloor$ e tomamos uma matriz A $s \times d$ com entradas aleatórias, então é bastante provável que a condição do Lema 4.4 seja satisfeita. A vantagem desta segunda estratégia é que o número de linhas (ou o valor de s) que é necessário para se encontrar uma fatoração de f é menor pois vários Tr_i's foram combinados em uma única linha de A.

A diferença $a_i - b_i$ também influencia no resultado final e no tempo de execução do algoritmo, sendo que quanto maior essa diferença, maior é o tempo necessário para o cálculo da Base Reduzida de Λ.

Assim, é possível ver que precisamos ajustar vários detalhes para ter um balanço entre garantia de um resultado e baixo tempo computacional. Para uma discussão sobre a relação entre o número de Tr_i's utilizados e o número de cálculos de uma base reduzida, veja [23]. Para a versão melhor ajustada do algoritmo de van Hoeij, veja [5].

Vejamos um exemplo.

Exemplo 4.16. *Considere o polinômio* $f = x^4+x^3+2x^2+x+1 \in \mathbb{Z}[x]$. *O primeiro* p *tal que* f *mod* p *é livre de quadrados é* $p = 5$. *Uma fatoração para* f *mod* 5 *é dada por*

$$f \ mod \ 5 = (x+3)(x+2)(x^2+x+1).$$

Denote por f_1, f_2, f_3 *os fatores p-ádicos de* f *e* $f_1^{(1)} = x+3$, $f_2^{(1)} = x+2$ *e* $f_3^{(1)} = x^2+x+1$ *uma aproximação de grau* 1 *desses fatores. A constante* a, *do passo* 2, *é dada por*

$$a = \left\lceil \log_p(2^4 \cdot \sqrt{4+1} \cdot \|f\|_\infty) \right\rceil = 32.$$

Realizando o levantamento de Hensel, obtemos os fatores

$$f_1^{(a)} = x + 8032510391590452844818, \quad f_2^{(a)} = x + 15250553973796510045807 \ \ e$$

$$f_3^{(a)} = x^2 + x + 1.$$

A matriz que escolhemos aqui é a matriz identidade $d \times d$, *onde* $d = k = 3$. *Passamos agora a calcular cotas para as entradas de* $T_A(g)$, *para todo fator racional* g *de* f. *Como* A *é a matriz identidade,* $T_A(g) = Tr_{1..d}(g)$. *Como deduzido anteriormente, se* B_{rt} *é uma cota para as raízes complexas de* f, *então* dB_{rt}^i *é uma cota para a* i*-ésima entrada de* $T_{1..d}(g)$. *Aqui, utilizamos uma cota baseada no Teorema de Rouché. Essa cota é dada por* $B_{rt} = 1 + \|f\|_\infty$. *Assim, as cotas* B_i*'s são dadas por*

$$B_1 = 3 \cdot (1+2)^1 = 9, \quad B_2 = 3 \cdot (1+2)^2 = 27 \ \ e \ \ B_3 = 3 \cdot (1+2)^3 = 81.$$

O próximo passo é escolher os valores de a_i *e* b_i, $1 \leq i \leq s$. *Sabemos que esses valores devem satisfazer*

$$a > a_i > b_i > \log_p(2B_i).$$

Aqui, tomamos $a_1 = a_2 = a_3 = a - 1$ *e* $b_i = \left\lceil \log_p(2 \cdot B_i) \right\rceil + 1$. *Assim,*

$$a_1 = a_2 = a_3 = 31 \ \ e \ \ b_1 = 3, \ b_2 = 4 \ \ e \ \ b_3 = 5.$$

O próximo passo é calcular uma base para o reticulado Λ. *Para fazer isso, precisamos calcular a matriz* m *dada em* (4.18). *De acordo com* (4.10) *e* (4.11), *para calcular essa matriz precisamos calcular* $\mathcal{C}^{a_i-b_i}(\tilde{c}_{j,i})$, *onde* $\tilde{c}_{j,i}$ *é a* i*-ésima entrada de* $T_A^b(f_j)$. *Como o traço depende dos coeficientes de* f_i, *que não sabemos quem são, e como iremos trabalhar com as aproximações* $T_A^b(f_i)$, *podemos utilizar* $f_i^{(a)}$ *no lugar de* f_i. *Assim, utilizando as identidades de Newton, podemos calcular os vetores* $Tr_{1..d}(f_j^{(a)})$, $1 \leq j \leq k$. *Esses vetores são dados por*[1]

$$Tr_{1..d}(f_1^{(a)}) = \begin{pmatrix} -8.032510392 \cdot 10^{21} \\ 6.452122319 \cdot 10^{43} \\ -5.182673958 \cdot 10^{65} \end{pmatrix}$$

$$Tr_{1..d}(f_2^{(a)}) = \begin{pmatrix} -1.525055397 \cdot 10^{22} \\ 2.325793965 \cdot 10^{44} \\ -3.546964640 \cdot 10^{66} \end{pmatrix}$$

e

$$Tr_{1..d}(f_3^{(a)}) = \begin{pmatrix} -1 \\ -1 \\ 2 \end{pmatrix}$$

A seguir, calculamos os vetores $T_A^b(f_1^{(a)}), T_A^b(f_2^{(a)})$ *e* $T_A^b(f_3^{(a)})$. *Esses vetores são dados por*

$$T_A^b(f_1^{(a)}) = \begin{pmatrix} -9 \\ 0 \\ 1160 \end{pmatrix}, T_A^b(f_2^{(a)}) = \begin{pmatrix} 9 \\ 0 \\ -1160 \end{pmatrix} \text{ e } T_A^b(f_3^{(a)}) = \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix}.$$

Note que $T_A^b(f_3^{(a)})$ *é o vetor nulo. De acordo com o Lema* 4.8, *já sabemos que o fator* $f_3 = x^2 + x + 1$ *é um fator de* f *em* $\mathbb{Z}[x]$. *Porém, continuaremos com o algoritmo.*

[1] *Esses números foram escritos em notação científica para melhor visualização.*

Calculados os vetores $T_A^b(f_j^{(a)})$, *podemos montar a matriz* m, *que é dada por*

$$m = \begin{pmatrix} -9 & 0 & 1160 \\ 9 & 0 & -1160 \\ 0 & 0 & 0 \end{pmatrix}.$$

A seguir, calculamos uma base para Λ. *A matriz cujas linhas são os vetores dessa base é dada por*

$$(B_\Lambda) = \begin{pmatrix} 2 & 0 & 0 & -9 & 0 & 1160 \\ 0 & 2 & 0 & 9 & 0 & -1160 \\ 0 & 0 & 2 & 0 & 0 & 0 \\ 0 & 0 & 0 & 3.725290298 \cdot 10^{19} & 0 & 0 \\ 0 & 0 & 0 & 0 & 7.450580597 \cdot 10^{18} & 0 \\ 0 & 0 & 0 & 0 & 0 & 1.490116119 \cdot 10^{18} \end{pmatrix}.$$

O próximo passo é calcular uma base reduzida para Λ. *Essa base é dada pelos vetores*

$$(0,0,2,0,0,0), (2,2,0,0,0,0), (2,0,0,-9,0,1160),$$

$$(-1.284503630 \cdot 10^{15}, 1.284503630 \cdot 10^{15}, 0, 1.156053267 \cdot 10^{16}, 0, 9.190844940 \cdot 10^{13}),$$

$$(4.040013066 \cdot 10^{18}, -4.040013066 \cdot 10^{18}, 0, 8.927853895 \cdot 10^{17}, 0, -3.875657505 \cdot 10^{13})$$

$$e \;\; (0,0,0,0,7.450580597 \cdot 10^{18}, 0).$$

Em seguida, calculamos uma base ortogonal pelo processo de Gram-Schmidt. Essa base é dada por

$$(0,0,2,0,0,0), (2,2,0,0,0,0), (1,-1,0,-9,0,1160),$$

$$(-1.284503630 \cdot 10^{15}, 1.284503630 \cdot 10^{15}, 0, 1.156053267 \cdot 10^{16}, 0, 9.190844940 \cdot 10^{13}),$$

$$(4.039471408 \cdot 10^{18}, -4.039471408 \cdot 10^{18}, 0, 8.976603129 \cdot 10^{17}, 0, 0)$$

$$e \;\; (0,0,0,0,7.450580597 \cdot 10^{18}, 0).$$

O valor da constante M, dada pela equação (4.19), é $\approx 4,33$ e os únicos vetores da base ortogonal cuja norma é menor do que $4,33$ são $(0,0,2,0,0,0,0)$ e $(2,2,0,0,0,0,0)$. Portanto, o reticulado Λ' é o reticulado gerado por $(0,0,2,0,0,0,0)$ e $(2,2,0,0,0,0,0)$ e o reticulado $\mathcal{L}'$ é gerado por $(0,0,1)$ e $(1,1,0)$, ou seja, pela projeção dos vetores de $\frac{1}{C}\Lambda'$ nas $k=3$ primeiras coordenadas.

Passamos agora a verificar as condições do Lema 4.12. Neste caso,

$$R = fer(B_{\mathcal{L}'}) = \begin{pmatrix} 1 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}.$$

Claramente vemos que R satisfaz a condição i) do lema, visto que cada coluna de R contém apenas um 1 e as demais entradas são zero. Para verificar a condição ii), precisamos verificar se os polinômios definidos pelas linhas da matriz R são fatores de f. Esses polinômios são

$$g_1 = f_1^1 f_2^1 f_3^0 \equiv (x + 8032510391590452844818)(x + 15250553973796510045807) \equiv$$

$$x^2 + 1 \; mods \; p^a$$

e

$$g_2 = f_1^0 f_2^0 f_3^1 \equiv x^2 + x + 1 \equiv x^2 + x + 1 \; mods \; p^a.$$

Como ambos polinômios são fatores de f em $\mathbb{Z}[x]$, segue-se que a fatoração de f é dada por

$$f = (x^2 + 1)(x^2 + x + 1).$$

Como discutido anteriormente, o Algoritmo LLL (seção 3.3) não substituiu o Algoritmo de Berlekamp-Zassenhaus (seção 3.2.2), mesmo tendo uma complexidade melhor. Isso se deve ao fato de que, na prática, o algoritmo LLL utiliza vetores com entradas de ordem astronômicas e isso faz com que o cálculo da base reduzida do reticulado seja demorado. Embora as constantes encontradas no exemplo acima também são grandes, o Algoritmo de van Hoeij, por outro lado, utiliza

vetores diferentes para encontrar os fatores do polinômio f, resolvendo o problema muito mais rápido. Esse fato fez com que o Algoritmo de van Hoeij substituísse o Algoritmo de Berlekamp-Zassenhaus (e variações), considerado por mais de 40 anos como o melhor algoritmo para fatoração de polinômios com coeficientes inteiros.

5 APLICAÇÃO: GERANDO SUBCORPOS

5.1 Introdução

Neste quinto e último capítulo iremos apresentar uma aplicação da fatoração de polinômios. É difícil encontrar uma aplicação direta da fatoração polinomial. Na maioria dos casos, fatoração polinomial é usada como uma ferramenta para se obter algo além dos fatores do polinômio.

A seguir, mostraremos como podemos utilizar a fatoração de um polinômio para encontrar todos os subcorpos de uma extensão finita e separável. Isto é, se K é uma extensão finita e separável de k, queremos encontrar todos os subcorpos contidos em K e que contêm k. Para mais detalhes, veja [24, 25].

5.2 Teorema Principal

Antes de mais nada, começaremos recordando alguns conceitos.

Definição 5.1. *Seja $k \subseteq K$ uma extensão de corpos e $\alpha \in K$. O polinômio minimal de α sobre k é definido como o polinômio mônico $p(x) \in k[x]$ com menor grau tal que $p(\alpha) = 0$.*

Definição 5.2. *Seja K um corpo. Dizemos que um polinômio $f \in K[x]$ é separável se f possui todas as raízes distintas em seu corpo de decomposição. Caso contrário, f é dito inseparável. Além disso, um elemento algébrico $\alpha \in K$ é dito separável se seu polinômio minimal em $K[x]$ é separável.*

Definição 5.3. *Seja $k \subseteq K$ uma extensão finita de corpos. Dizemos que $k \subseteq K$ é uma extensão separável se todo elemento em K é separável sobre k, isto é, se o polinômio minimal de qualquer elemento em K sobre k é separável.*

A maioria das extensões de corpos vistas em um curso de Matemática são separáveis, como afirma o teorema abaixo.

Teorema 5.4. *Seja K um corpo e $f \in K[x]$ um polinômio irredutível. Então f é separável se, e somente se, sua derivada é não nula. Em particular, se K tem característica zero, então todo polinômio irredutível é separável. Por outro lado, se K tem característica p, então um polinômio irredutível é separável se, e somente se, ele não é um polinômio em x^p.*

Demonstração. Para uma demonstração deste resultado e mais detalhes sobre extensões separáveis, veja [14] e [27]. □

Assim, polinômios irredutíveis e inseparáveis só existem em característica p. O primeiro exemplo de uma extensão finita e não separável é dado pelo corpo das funções racionais sobre um corpo de característica p. Veja o exemplo abaixo.

Exemplo 5.5. *Seja p um número primo e $K = \mathbb{F}_p(u)$ o corpo das funções racionais em u sobre o corpo finito $\mathbb{F}_p$. Seja $L = K(\alpha)$ uma extensão de K, onde α é uma raiz de $x^p - u$ em alguma extensão finita de K. O polinômio $x^p - u \in K[x]$ é irredutível pelo critério de Eisenstein utilizando o elemento primo u. Assim, $x^p - u$ é o polinômio minimal de α sobre K. Vamos mostrar agora que $x^p - u$ não é separável. Como α é uma raiz desse polinômio, segue que $\alpha^p - u = 0$, logo*

$$x^p - u = x^p - \alpha^p = (x - \alpha)^p,$$

visto que K possui característica p. Assim, $x^p - u$ não possui todas as raízes distintas em seu corpo de decomposição e é, portanto, um polinômio não separável. Ou seja, o elemento α não é separável e consequentemente, a extensão $K \subseteq L$ não é separável.

Ao longo deste capítulo consideraremos a seguinte situação: Seja $k \subseteq K$ uma extensão finita e separável de corpos e fixe $\alpha \in K$ um elemento primitivo de

K sobre k, isto é, $K = k(\alpha)$. Além disso, seja $f \in k[x]$ o polinômio minimal de α sobre k.

Seja $f = f_1 f_2 \cdots f_r$ a fatoração de $f \in k[x]$ sobre K, onde cada $f_i \in K[x]$ é irredutível e mônico. Como f é o polinômio minimal de $\alpha \in K$, podemos supor que $f_1 = x - \alpha$. Defina o corpo $K_i = K[x]/\langle f_i(x)\rangle$, $1 \leq i \leq r$. Como $K = k(\alpha)$, podemos denotar cada elemento de K como $g(\alpha)$, onde $g(x) \in k[x]$ possui grau menor que $n = deg(f)$. Defina a aplicação

$$\phi_i : K \to K_i, \quad g(\alpha) \mapsto g(x) \; mod \; f_i$$

e os conjuntos

$$L_i = \ker(\phi_i - id) = \{g(\alpha) \in K : g(x) \equiv g(\alpha) \; mod \; f_i\}, \;\; 1 \leq i \leq r, \tag{5.1}$$

onde $id : K \hookrightarrow K_i$ é a imersão canônica. Cada L_i é um subcorpo de K. Além disso, como mostra o teorema abaixo, todo subcorpo de K que contém k é a interseção de alguns dos L_i's.

Teorema 5.6. *Seja L um corpo tal que $k \subseteq L \subseteq K$. Então*

$$L = \bigcap_{i \in I} L_i,$$

para algum conjunto $I \subseteq \{1, 2, \ldots, r\}$.

Demonstração. Seja f_L o polinômio minimal de α sobre L. Como L é uma extensão de k, o polinômio f_L divide f. Além disso, como $f = f_1 f_2 \cdots f_r$ em $K[x]$, f_L é o produto de alguns dos f_i's, ou seja, existe $I \subseteq \{1, 2, \ldots, r\}$ tal que $f_L = \prod_{i \in I} f_i$. Vamos mostrar as seguintes igualdades

$$L = \{g(\alpha) \in K : g(x) \equiv g(\alpha) \; mod \; f_L\} = \bigcap_{i \in I} L_i.$$

Vamos demonstrar a primeira igualdade. Seja $g(\alpha) \in L$. Queremos mostrar que $g(x) \equiv g(\alpha) \; mod \; f_L$.

Defina $h(x) = g(x) - g(\alpha) \in L[x]$. Então $h(\alpha) = 0$ e portanto, $(x - \alpha)|h(x)$ em $K[x]$. O conjunto dos polinômios $p \in L[x]$ tais que $p(\alpha) = 0$ é dado por $\langle f_L \rangle$, visto que f_L é irredutível. Assim, $h(x) \in \langle f_L \rangle$ e, portanto, $h(x) \equiv 0 \; mod \; f_L$, ou seja, $g(x) \equiv g(\alpha) \; mod \; f_L$. Por outro lado, suponha que $g(\alpha) \in \{g(\alpha) \in K : g(x) \equiv g(\alpha) \; mod \; f_L\} \subseteq K$. Assim, $g(\alpha) \equiv g(x) \; mod \; f_L$. Além disso, como $g(x) \in k[x]$, a divisão de $g(x)$ por f_L produzirá coeficientes em L e portanto, $g(x) \; mod \; f_L \in L[x]$. Logo, $g(\alpha) \equiv g(x) \; mod \; f_L \in L[x]$. Assim,

$$g(\alpha) \in K \cap L[x].$$

Como $L \subseteq K$, segue que $g(\alpha) \in L$, e a primeira igualdade está provada.

A segunda igualdade é verificada pelo Teorema Chinês dos Restos. Suponha que $g(\alpha) \in L$. Então $g(x) \equiv g(\alpha) \; mod \; f_L$ se, e somente se, $g(x) \equiv g(\alpha) \; mod \; f_i$, $\forall \; i \in I$, ou seja, se, e somente se, $g(\alpha) \in L_i$, $\forall \; i \in I$. □

Assim, para encontrarmos todos os subcorpos de K que contêm k, basta encontrar os subcorpos L_i, $1 \leq i \leq r$, e realizar todas as interseções entre eles.

Definição 5.7. *Os subcorpos $L_1, L_2, \ldots, L_r$ construídos acima são chamados de Subcorpos Principais da extensão $k \subseteq K$. Um conjunto S de subcorpos de K que contém k é um Conjunto Gerador de $k \subseteq K$, se todo subcorpo de K e que contém k pode ser escrito como uma interseção de elementos em S.*

Na próxima seção apresentamos um algoritmo que realiza essas interseções.

5.3 O Algoritmo

A seguir apresentaremos um algoritmo para o cálculo dos subcorpos de uma extensão $k \subseteq K$. De acordo com o teorema 5.6, basta calcular todas as

interseções dos subcorpos principais. Estes subcorpos são encontrados através da fatoração do polinômio minimal $f \in k[x]$ do elemento α na extensão K.

O algoritmo funciona da seguinte forma: na primeira etapa, calculamos a interseção de K com cada um dos L_i's. Como cada $L_i \subseteq K$, estas interseções são os próprios subcorpos L_i's. A seguir, para cada $1 \leq i < r$, calculamos a interseção de L_i e L_{i+1}. Poderíamos seguir desta maneira até calcular todas as interseções possíveis. Porém, desta forma, estaríamos possivelmente calculando interseções que já foram calculadas em etapas anteriores. Por isso, a cada etapa do algoritmo, verificamos se o subcorpo já foi calculado ou não.

Para fazer esta verificação, a cada subcorpo L de K e que contém k, associamos o vetor $e = (e_1, e_2, \ldots, e_r) \in \{0,1\}^r$, tal que $e_i = 1$ se, e somente se, $L \subseteq L_i$. Assim, temos o seguinte algoritmo.

Algoritmo 5.8. *Subcorpos*

Entrada: *Um conjunto gerador $S = \{L_1, L_2, \ldots, L_r\}$ para $k \subseteq K$.*

Saída: *Todos os subcorpos de K e que contêm k.*

1 *Seja $e = (e_1, e_2, \ldots, e_r)$, onde $e_i = 1$ se $L_i = K$ e $e_i = 0$ caso contrário.*

2 *$ListaSubcorpos \leftarrow \{K\}$.*

3 *Chame o algoritmo $PróximoSubcorpo(S, K, e, 0)$.*

4 ***retorna*** *$ListaSubcorpos$.*

O algoritmo PróximoSubcorpo não retorna valor algum. Sua finalidade é adicionar novos subcorpos à variável *ListaSubcorpos*, que é tratada como uma variável global. A entrada deste algoritmo consiste no conjunto gerador S de $k \subseteq K$,

um subcorpo L, seu vetor de zeros e uns associado e o menor inteiro $0 \leq s \leq r$ tal que $L = \bigcap\{L_i \ / \ 1 \leq i \leq s \ e \ e_i = 1\}$.

Algoritmo 5.9. *PróximoSubcorpo*

Entrada: *Um conjunto gerador $S = \{L_1, L_2, \ldots, L_r\}$ para $k \subseteq K$, um subcorpo $L \subseteq K$ e seu vetor associado e, e o inteiro $0 \leq s \leq r$ tal que $L = \bigcap\{L_i \ : \ 1 \leq i \leq s \ e \ e_i = 1\}$.*

1 ***para*** *i tal que $s < i \leq r$ e $e_i = 0$* ***faça***

2 $M \leftarrow L \cap L_i$

3 *Seja $\tilde{e}$ o vetor associado ao subcorpo M.*

4 ***se*** *$\tilde{e}_j \leq e_j$, para $1 \leq j < i$* ***então***

$ListaSubcorpos \leftarrow ListaSubcorpos \bigcup \{M\}$.

Chame PróximoSubcorpo$(S, M, \tilde{e}, i)$.

Neste algoritmo, subcorpos que são isomorfos mas não são idênticos são considerados diferentes.

Teorema 5.10. *Dado um conjunto gerador S de $k \subseteq K$ com r elementos, o algoritmo Subcorpos retorna todos os subcorpos calculando no máximo rm interseções e realiza no máximo r^2m testes para verificar se um corpo está contido em outro, onde m é o número de subcorpos de K que contêm k.*

Demonstração. Seja m o número de subcorpos de K que contêm k. Como S é um conjunto gerador, todos esses subcorpos são interseções dos elementos de S. Assim, a condição do passo 4 do algoritmo 5.9 vale se, e somente se, o subcorpo $M = L \cap L_i$ ainda não foi calculado. Desse modo, cada subcorpo é adicionado à

lista *ListaSubcorpos* exatamente uma vez e o número de chamadas do algoritmo *PróximoSubcorpo* é exatamente m. Para cada uma destas chamadas, o número de i's tais que $e_i = 0$ e $s \leq i \leq r$ é limitado por r logo, o número total de interseções calculadas no passo 2 do algoritmo *PróximoSubcorpo* é limitado por rm. Para cada uma destas interseções, o passo 3 realiza um teste para verificar se cada um dos $L_i \in S$ é um subconjunto de M, e o número máximo destes testes, para cada M, é limitado por r. Assim, o número total destes testes é limitado por $(rm)r = r^2m$. □

A figura a seguir mostra como as interseções são feitas. A cada etapa, o item 4 do algoritmo *PróximoSubcorpo* verifica se o subcorpo já foi calculado, de modo que cada subcorpo é calculado apenas uma vez. O símbolo % representa o corpo imediatamente acima.

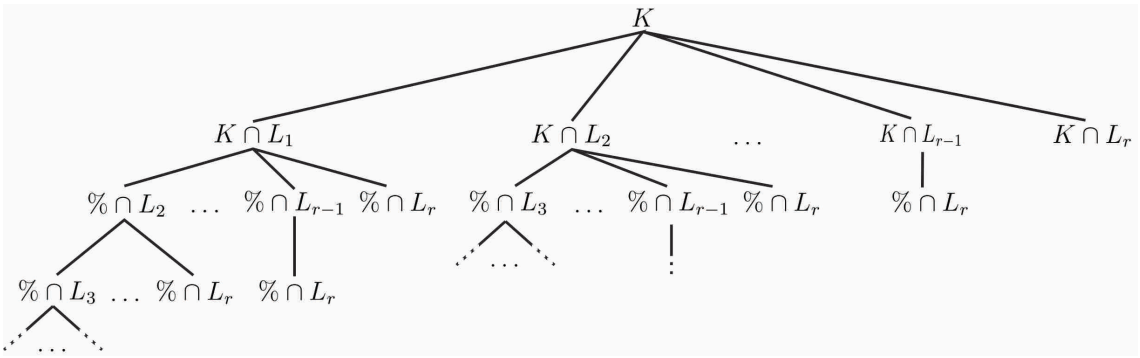

Figura 5.1: Esquema das interseções dos subcorpos principais.

Vejamos um exemplo.

Exemplo 5.11. *Considere a extensão $K = \mathbb{Q}(\alpha)$ do corpo dos números racionais, onde α é uma raiz de $f = x^8 - 5 \in \mathbb{Q}[x]$. Vamos calcular todos os subcorpos de $\mathbb{Q}(\alpha)$ que contêm $\mathbb{Q}$.*

Visto que f é irredutível sobre $\mathbb{Q}$, o polinômio minimal de α sobre $\mathbb{Q}$ é f. Em $\mathbb{Q}(\alpha)[x]$, f fatora-se como

$$f = (x - \alpha)(x + \alpha)(x^4 + \alpha^4)(x^2 + \alpha^2).$$

Os polinômios $f_1 = x - \alpha$, $f_2 = x + \alpha$, $f_3 = x^4 + \alpha^4$ *e* $f_4 = x^2 + \alpha^2$ *fornecem os subcorpos principais* L_1, L_2, L_3 *e* L_4. *Estes subcorpos podem ser calculados da seguinte forma: cada subcorpo de* $\mathbb{Q}(\alpha)$ *pode ser visto como um subespaço vetorial do mesmo, cuja base é dada, neste caso, por* $1 \mod f, x \mod f, x^2 \mod f, \ldots, x^7 \mod f$. *Assim, a condição* $g(\alpha) \in L_i$ *se, e somente se,* $g(x) \equiv g(\alpha)$ *mod* f_i, *nos fornece equações para calcular o subcorpo* L_i.

Por exemplo, seja $g = a_0 + a_1 x + a_2 x^2 + \cdots + a_7 x^7 \mod f$ *um elemento genérico de* $\mathbb{Q}(\alpha)$, *escrito na base* $1 \mod f, x \mod f, x^2 \mod f, \ldots, x^7 \mod f$ *de* $\mathbb{Q}(\alpha)$. *Queremos encontrar condições sobre os coeficientes* a_i *para que* $g(\alpha) \in L_4$, *por exemplo. De acordo com* (5.1), $g(\alpha) \in L_4$ *se, e somente se,* $g(x) \equiv g(\alpha)$ *mod* f_4, *ou seja,* $g(x)$ *mod* $f_4 = g(\alpha)$. *Realizando os cálculos, obtemos a seguinte igualdade*

$$(a_1 - a_3\alpha^2 + a_5\alpha^4 - a_7\alpha^6)x - (a_2 - a_4\alpha^2 + a_6\alpha^4)\alpha^2 =$$

$$a_1\alpha + a_2\alpha^2 + a_3\alpha^3 + a_4\alpha^4 + a_5\alpha^5 + a_6\alpha^6 + a_7\alpha^7.$$

Resolvendo para os a_i*'s, a solução para esta equação é dada por*

$$a_1 = a_2 = a_3 = a_5 = a_6 = a_7 = 0, \quad a_0 = a_0 \ \ e \ \ a_4 = a_4.$$

Ou seja, uma base para L_4, *visto como subespaço vetorial de* $\mathbb{Q}(\alpha)$, *é dada por* $\{1, x^4\}$.

Depois de calculados os subcorpos L_i*'s, devemos realizar as interseções entre eles. Primeiramente,* $i = 1$ *e* $e = (0, 0, 0, 1)$. *Calculamos a interseção de* K *com* L_1. *Esta interseção é o próprio subcorpo* L_1, *que coincide com o corpo dos números racionais pela escolha do polinômio* f_1. *Seu vetor associado é* $\tilde{e} = (1, 1, 1, 1)$. *Como o vetor* $\tilde{e}$ *satisfaz a condição do passo* 4 *do algoritmo PróximoSubcorpo, o subcorpo* L_1 *é adicionado à lista de subcorpos. A seguir, uma nova chamada ao algoritmo PróximoSubcorpo é realizada, com entradas* S, L_1, *o vetor* $e = (1, 1, 1, 1)$, *associado ao subcorpo* L_1, *e* $s = 1$. *Como não existe nenhum* $i > s = 1$ *satisfazendo*

o passo 1 *do algoritmo, isto é,* $e_i = 0$*, partimos para a interseção com o próximo subcorpo principal.*

Agora, $s = 2$ *e* $e = (0, 0, 0, 1)$*. Calculamos a interseção de* K *com* L_2*, fornecendo o subcorpo* L_2*, cujo vetor associado é* $\tilde{e} = (0, 1, 1, 1)$*. Novamente, o vetor* $\tilde{e}$ *satisfaz a condição do passo* 4 *e o subcorpo* L_2 *é adicionado à lista de subcorpos. Uma nova chamado ao algoritmo PróximoSubcorpo é feita, com entradas* S*,* L_2*, o vetor* $e = (0, 1, 1, 1)$*, associado ao subcorpo* L_2*, e* $s = 2$*. Novamente, como não existe nenhum* $i > s = 2$ *satisfazendo o passo* 1 *do algoritmo, a interseção com o próximo subcorpo principal é calculada.*

As interseções com os subcorpos L_3 *e* L_4 *ocorrem da mesma forma.*

Assim, os subcorpos de $\mathbb{Q}(\alpha)$ *que contém* $\mathbb{Q}$*, vistos como subespaços vetoriais de* $\mathbb{Q}(\alpha)$ *cuja base é* $1, x, x^2, \ldots, x^7$*, são gerados pelas bases* $\{1\}$*,* $\{1, x^4\}$*,* $\{1, x^2, x^4, x^6\}$ *e* $\{1, x, x^2, x^3, x^4, x^5, x^6, x^7\}$*. O subcorpo gerado por* $\{1\}$ *corresponde ao corpo base* $\mathbb{Q}$ *e o subcorpo gerado por* $\{1, x, x^2, x^3, x^4, x^5, x^6, x^7\}$ *corresponde ao corpo* $\mathbb{Q}(\alpha)$*.*

Este capítulo mostrou uma das várias aplicações da fatoração polinomial. Embora a fatoração polinomial seja usada apenas como ferramenta, ela é essencial em diversas aplicações que encontramos na matemática. Assim, fica clara a importância de estudar tal assunto e encontrar algoritmos cada vez melhores para fatorar polinômios.

CONCLUSÃO E TRABALHOS FUTUROS

Fatoração polinomial é, sem sombra de dúvidas, uma ferramenta essencial para muitas aplicações. Neste trabalho, vimos como um problema de fácil proposição pode gerar uma teoria rica em detalhes. Foi interessante ver também como teoremas são transformados em algoritmos, como é o caso do Teorema 2.2, que permitiu o desenvolvimento do Algoritmo 2.1.

Outra parte interessante do trabalho foi a implementação dos algoritmos. Além de aprofundar o entendimento do algoritmo, a implementação destes algoritmos mostrou que, embora em teoria o algoritmo funcione, na prática devemos tomar certos cuidados extras, sem os quais não seria possível obter os resultados desejados, como é o caso da fatoração de polinômios não mônicos com coeficientes inteiros. Também notamos que é na prática, ou seja, através das implementações dos algoritmos e do cálculo de exemplos, que percebemos detalhes como, por exemplo, o motivo pelo qual o algoritmo LLL não é eficiente. Isso se dá, entre outros motivos, pelo tamanho dos números envolvidos nos cálculos, detalhe que não pode ser visto com facilidade apenas com a teoria.

Quanto aos trabalhos futuros, existem várias coisas que podem ser feitas. Uma delas seria um estudo mais aprofundado do Algoritmo de van Hoeij e como realizar todas as escolhas feitas ao longo do algoritmo para termos um algoritmo mais eficiente.

Assim como o Algoritmo de Berlekamp 2.4 e o Algoritmo de Niederreiter possuem diversas interseções, e essas interseções podem ser utilizadas para se buscar um algoritmo melhor ainda, um outro tópico para trabalhos futuros é estudar as relações que existem entre os algoritmos LLL e o algoritmo de van Hoeij.

Além disso, durante a visita ao professor Mark van Hoeij, na Universidade do Estado da Flórida, foram discutidas novas maneiras de realizar as interseções dos subcorpos principais, representadas na figura 5.1. Podemos representar cada subcorpo pela partição de um determinado conjunto. Dessa forma, a interseção de dois subcorpos é dada pela partição que é um refinamento de ambas as partições de cada um dos subcorpos. Em teoria, calcular o refinamento de partições é muito mais rápido do que calcular as interseções através de Álgebra Linear (modo como essas interseções são feitas no Capítulo 5), além da necessidade de realizar cálculos no próprio corpo. Assim, comprovar essas ideias é outro tópico de pesquisas futuras.

Referências Bibliográficas

[1] J. A. Abbott, "On the Factorization of Polynomials over Algebraic Number Fields", PhD Thesis, School of Mathematical Sciences, University of Bath, Thecnical Report, 1989.

[2] M. Ajtai, "The Shortest Vector Problem in L_2 is NP-hard for Randomized Reductions, STOC'98 Proceedings of the thirtieth annual ACM Symposium on Theory of Computing, pp. 10-19, New York, 1998.

[3] K. Belabas, "A relative van Hoeij Algorithm over Number Fields", J. Symbolic Computation, 37, pp. 641-668, 2004.

[4] K. Belabas, M. van Hoeij, J. Klüners, A. Steel, "Factoring Polynomials over Global Fields", Journal de Théorie des Nombres de Bordeaux, vol. 21, issue 1, pp. 15-39, 2009.

[5] K. Belabas, G. Hanrot, P. Zimmermann, "Tuning and Generalizing van Hoeij's Algorithm", relatório de pesquisa, INRIA, 4124, 2001.

[6] E. R. Berlekamp, "Factoring Polynomials over Finite Fields", Bell System Tech. J., vol. 46, pp. 1853-1859, 1967.

[7] E. R. Berlekamp, "Factoring Polynomials over Large Finite Fields", Math. Comput., 24, pp. 713-735, 1970.

[8] P. van Emde Boas, "Another NP-Complete Partition Problem and the Complexity of Computing Short vectors in Lattice", Tech. Report, Dept. of Mathematics, Univ. of Amsterdam, 1980.

[9] P. Camion, "Improving an Algorithm for Factoring Polynomials over a Finite Field and Constructing large irreducible polynomials", IEEE Transactions of Information Theory, vol. 29, no. 3, 1983.

[10] D. Cantor, H. Zassenhauss, "A new Algorithm for Factoring Polynomials over Finite Fields", Math. Comp., 36, pp. 587-592, 1981.

[11] J. W. S. Cassels, *An Introduction to the Geometry of Numbers*, Springer Verlag, 1971.

[12] D. Cox, J. Little, D. O'Shea, *Ideals, Varieties and Algorithms: An Introduction to Computational Algebraic Geometry and Commutative Algebra*, 2nd ed., Springer Verlag, 1996.

[13] C. Davies, *Elements of Algebra*, published by A. S. Barnes & Co., New York, 1857.

[14] S. Dummit, R. Foote, *Abstract Algebra*, 3rd ed., Wiley, Nova Yorque, 2004.

[15] P. Fleischmann, "Connections between the algorithms of Berlekamp and Niederreiter for factoring polynomials over F_q", Linear Algebra and its Applications, 192, pp. 101-108, 1993.

[16] A. Garcia, Y. Lequain, *Elementos de Algebra*, 5 ed., Projeto Euclides, Rio de Janeiro, IMPA, 2008.

[17] J. von zur Gathen, J. Gerhard, *Modern Computer Algebra*, 2nd. ed. Cambridge University Press, 2003.

[18] J. von zur Gathen, "Factoring Polynomials and Primitive Elements for special primes", Theoretical Computer Science, vol. 52, pp. 77-89, 1987.

[19] A. Gonçalves, *Introdução à Álgebra*, 5 ed., Projeto Euclides, Rio de Janeiro, IMPA, 2008.

[20] F. Q. Gouvêa, *p-adic Numbers: An Introduction*, 2nd, ed. Springer Verlag, 2003.

[21] K. Hensel, "Eine Neue Theorie der algebraischen Zahlen", Mathematische Zeitschrift, Teubner, Berlin, pp. 433-452, 1918.

[22] M. van Hoeij, "Factoring Polynomials and the Knapsack Problem", J. Number Theory, vol. 95, issue 2, pp. 167-189, 2002.

[23] M. van Hoeij, A. Novocin, "Complexity Results for Factoring Univariate Polynomials over the Rationals", preprint, 2007. Disponível em *http://www.math.fsu.edu/~hoeij/papers/2007/paper6.pdf*.

[24] M. van Hoeij, J. Klüners, A. Novocin, "Generating Subfields", Journal of Symbolic Computation, vol. 52, pp. 17-34, 2013.

[25] J. Klüners, "On Computing Subfields - A detailed description of the algorithm", Journal de Théorie des Nombres de Bourdeaux, 10, pp. 243-271, 1998.

[26] D. E. Knuth, *The Art of Computer Programming*, vol. 2:*Seminumerical Algorithms*, 2nd ed., Addison-Wesley, Reading, Mass., USA, 1981.

[27] S. Lang, *Algebra*, 3 ed. revisada, Springer-Verlag, Nova Yorque, 2002.

[28] A. K. Lenstra, H. W. Lenstra, L. Lovász, "Factoring Polynomials with Rational Coefficients", Math. Ann. 261, pp. 515-534, 1982.

[29] D. Looser, "Niederreiter's Factorization Algorithm for Polynomials over Finite Fields, 2008.

[30] M. Mignotte, "An Inequality about Factors of Polynomials", Mathematics of Computation, vol. 28, no. 128, pp. 1153-1157, 1974.

[31] H. Minkowski, *Geometrie der Zahlen*, B. G. Teubner, Berlin, 1910.

[32] A. Miola, D. Yun, "Computational Aspects of Hensel-type univariate polynomial greatest common divisor algorithms", Proc. of Eurosam'74, Royal

Institut of Technology, Stockholm, Sweden, 1974: SIGSAM Bulletin vol. 8, no. 3, pp. 46-54, 1974.

[33] H. Niederreiter, "A New Efficient Factorization Algorithm for Polynomials over small Finite Fields", Appl. Alg. Eng. Commun. Comp., 4, pp. 81-87, 1993.

[34] M. Rabin, "Probabilistic Algorithms in Finite Fields", SIAM J. Comput., vol. 9, no. 2, pp. 273-280, 1980.

[35] V. Shoup, "On the Deterministic Complexity of Factoring Polynomials over Finite Fields", Information Processing Letters 33, pp. 261-267, 1990.

[36] P. S. Wang, "Parallel p-adic Constructions in the Univariate Polynomial Factoring Algorithm", Proceedings, MACSYMA Users' Conference, Cambridge, MA. MIT, pp. 310-318, 1979.

[37] F. Winkler, *Polynomial Algorithms in Computer Algebra*, Texts and Monographs in Symbolic Computation, Springer Verlag, 1996.

[38] H. Zassenhaus, "On Hensel Factorization, I", J. Number Theory, vol. 1, pp. 291-311, 1969.

[39] H. Zassenhaus, "A Remark on the Hensel Factorization Method", Math. Comp., vol. 32, no. 141, pp. 287-292, 1974.