MASARYKOVA UNIVERZITA
PŘÍRODOVĚDECKÁ FAKULTA
ÚSTAV MATEMATIKY A STATISTIKY

Bakalářská práce

BRNO 2015 JAN PLHÁK

MASARYKOVA UNIVERZITA
PŘÍRODOVĚDECKÁ FAKULTA
ÚSTAV MATEMATIKY A STATISTIKY

Název práce na titulní list

Bakalářská práce

Jan Plhák

Vedoucí práce: Bc. Lukáš Vokřínek, PhD. **Brno 2015**

Bibliografický záznam

Autor:	Jan Plhák Přírodovědecká fakulta, Masarykova univerzita Ústav matematiky a statistiky
Název práce:	Název práce
Studijní program:	Matematika
Studijní obor:	Obecná matematika
Vedoucí práce:	Bc. Lukáš Vokřínek, PhD.
Akademický rok:	2014/2015
Počet stran:	?? + ??
Klíčová slova:	Klíčové slovo; Klíčové slovo; Klíčové slovo; Klíčové slovo; Klíčové slovo; Klíčové slovo; Klíčové slovo; Klíčové slovo

Bibliographic Entry

Author:	Jan Plhák Faculty of Science, Masaryk University Department of Mathematics and Statistics
Title of Thesis:	Title of Thesis
Degree Programme:	Mathematics
Field of Study:	Mathematics
Supervisor:	Bc. Lukáš Vokřínek, PhD.
Academic Year:	2014/2015
Number of Pages:	?? + ??
Keywords:	Keyword; Keyword; Keyword; Keyword; Keyword; Keyword; Keyword; Keyword; Keyword

Abstrakt

V této bakalářské/diplomové/rigorózní práci se věnujeme ...

Abstract

In this thesis we study ...

Místo tohoto listu vložte kopii oficiálního (podepsaného) zadání práce.

Poděkování

Na tomto místě bych chtěl(-a) poděkovat ...

Prohlášení

Prohlašuji, že jsem svoji bakalářskou/diplomovou/rigorózní práci vypracoval(-a) samostatně s využitím informačních zdrojů, které jsou v práci citovány.

Brno xx. měsíce 20xx . . . .

Jan Plhák

Obsah

Úvod .... **viii**

Přehled použitého značení .... **ix**

Kapitola 1. Smithův normální tvar **1**

Kapitola 2. Triangularizace celočíselných matic **5**

2.1 GCD redukce 7

2.2 Sloupcová redukce 9

2.3 RST algoritmus 12

Kapitola 3. Výpočet Smithova normálního tvaru trojúhelníkových matic **16**

3.1 Hermitův normální tvar 16

3.2 Sloupcová eliminace 17

Závěr **19**

Příloha **20**

Seznam použité literatury **21**

Úvod

Cílem této práce je seznámit čtenáře s efektivním algoritmem pro výpočet Smithova normálního tvaru celočíselných matic.

Přehled použitého značení

Pro snažší orientaci v textu zde čtenáři předkládáme přehled základního značení, které se v celé práci vyskytuje.

$\mathbb{C}$ množina všech komplexních čísel
$\mathbb{R}$ množina všech reálných čísel
$\mathbb{Z}$ množina všech celých čísel
$\mathbb{N}$ množina všech přirozených čísel
$\mathbb{C}$ množina všech komplexních čísel
$\mathbb{R}$ množina všech reálných čísel
$\mathbb{Z}$ množina všech celých čísel
$\mathbb{N}$ množina všech přirozených čísel
$\mathbb{C}$ množina všech komplexních čísel
$\mathbb{R}$ množina všech reálných čísel
$\mathbb{Z}$ množina všech celých čísel
$\mathbb{N}$ množina všech přirozených čísel
$\mathbb{C}$ množina všech komplexních čísel
$\mathbb{R}$ množina všech reálných čísel
$\mathbb{Z}$ množina všech celých čísel
$\mathbb{N}$ množina všech přirozených čísel
$\mathbb{C}$ množina všech komplexních čísel
$\mathbb{R}$ množina všech reálných čísel
$\mathbb{Z}$ množina všech celých čísel
$\mathbb{N}$ množina všech přirozených čísel
$\mathbb{C}$ množina všech komplexních čísel
$\mathbb{R}$ množina všech reálných čísel
$\mathbb{Z}$ množina všech celých čísel
$\mathbb{N}$ množina všech přirozených čísel
$\mathbb{C}$ množina všech komplexních čísel
$\mathbb{R}$ množina všech reálných čísel
$\mathbb{Z}$ množina všech celých čísel
$\mathbb{N}$ množina všech přirozených čísel

$\mathbb{C}$	množina všech komplexních čísel
$\mathbb{R}$	množina všech reálných čísel
$\mathbb{Z}$	množina všech celých čísel
$\mathbb{N}$	množina všech přirozených čísel
$\mathbb{C}$	množina všech komplexních čísel
$\mathbb{R}$	množina všech reálných čísel
$\mathbb{Z}$	množina všech celých čísel
$\mathbb{N}$	množina všech přirozených čísel
$\mathbb{C}$	množina všech komplexních čísel
$\mathbb{R}$	množina všech reálných čísel
$\mathbb{Z}$	množina všech celých čísel
$\mathbb{N}$	množina všech přirozených čísel
$\mathbb{C}$	množina všech komplexních čísel
$\mathbb{R}$	množina všech reálných čísel
$\mathbb{Z}$	množina všech celých čísel
$\mathbb{N}$	množina všech přirozených čísel
$\mathbb{C}$	množina všech komplexních čísel
$\mathbb{R}$	množina všech reálných čísel
$\mathbb{Z}$	množina všech celých čísel
$\mathbb{N}$	množina všech přirozených čísel
$\mathbb{C}$	množina všech komplexních čísel
$\mathbb{R}$	množina všech reálných čísel
$\mathbb{Z}$	množina všech celých čísel
$\mathbb{N}$	množina všech přirozených čísel
$\mathbb{C}$	množina všech komplexních čísel
$\mathbb{R}$	množina všech reálných čísel
$\mathbb{Z}$	množina všech celých čísel
$\mathbb{N}$	množina všech přirozených čísel
$\mathbb{C}$	množina všech komplexních čísel
$\mathbb{R}$	množina všech reálných čísel
$\mathbb{Z}$	množina všech celých čísel
$\mathbb{N}$	množina všech přirozených čísel
$\mathbb{C}$	množina všech komplexních čísel
$\mathbb{R}$	množina všech reálných čísel
$\mathbb{Z}$	množina všech celých čísel
$\mathbb{N}$	množina všech přirozených čísel
$\mathbb{C}$	množina všech komplexních čísel
$\mathbb{R}$	množina všech reálných čísel

$\mathbb{Z}$	množina všech celých čísel
$\mathbb{N}$	množina všech přirozených čísel
$\mathbb{C}$	množina všech komplexních čísel
$\mathbb{R}$	množina všech reálných čísel
$\mathbb{Z}$	množina všech celých čísel
$\mathbb{N}$	množina všech přirozených čísel
$\mathbb{C}$	množina všech komplexních čísel
$\mathbb{R}$	množina všech reálných čísel
$\mathbb{Z}$	množina všech celých čísel
$\mathbb{N}$	množina všech přirozených čísel
$\mathbb{C}$	množina všech komplexních čísel
$\mathbb{R}$	množina všech reálných čísel
$\mathbb{Z}$	množina všech celých čísel
$\mathbb{N}$	množina všech přirozených čísel
$\mathbb{C}$	množina všech komplexních čísel
$\mathbb{R}$	množina všech reálných čísel
$\mathbb{Z}$	množina všech celých čísel
$\mathbb{N}$	množina všech přirozených čísel
$\mathbb{C}$	množina všech komplexních čísel
$\mathbb{R}$	množina všech reálných čísel
$\mathbb{Z}$	množina všech celých čísel
$\mathbb{N}$	množina všech přirozených čísel

Kapitola 1

Smithův normální tvar

V této kapitole se budeme zbývat definicí Smithova normálního tvaru (budeme značit SNF) celočíselných matic $Mat_{n\times m}\mathbb{Z}$, dokážeme jeho existenci pro libovolnou $A \in Mat_{n\times m}\mathbb{Z}$ a konečně uvedeme souvislost mezi SNFa konečně generovanými komutativními grupami.

Definice 1.1. Řekneme že matice A $\in Mat_{n\times m}\mathbb{Z}$ je ve Smithově normálním tvaru jestliže

$$A = \begin{pmatrix} q_1 & 0 & \cdots & \cdots & \cdots & \cdots & 0 \\ 0 & q_2 & \ddots & & & & \vdots \\ \vdots & \ddots & \ddots & \ddots & & & \vdots \\ \vdots & & \ddots & q_k & \ddots & & \vdots \\ \vdots & & & \ddots & 0 & \ddots & \vdots \\ \vdots & & & & \ddots & \ddots & 0 \\ 0 & \cdots & \cdots & \cdots & \cdots & 0 & 0 \end{pmatrix}$$

a platí $q_i|q_{i+1}$ kde $i \in \{1,\dots,k-1\}$. Čísla q_i pak nazýváme *invariantními faktory*.

Věta 1.2 (O Smithově normálním tvaru)**.** *Pro libovolnou celočíselnou matici $B \in Mat_{n\times m}\mathbb{Z}$ existují invertibilní matice $P,Q \in Mat_{n\times m}\mathbb{Z}$ a matice A ve Smithově normálním tvaru takové, že platí*

$$B = P\cdot A\cdot Q$$

Smithův normální tvar je jednoznačný až na znaménka invariantních faktorů.

Než se pustíme do samotného důkazu této věty, je dobré si uvědomit, jak vlastně vypadají invertibilní celočíselné matice. To popisuje následující lemma.

Lemma 1.3. *Buď $A \in Mat_{n\times m}\mathbb{Z}$. Pak je A invertiblní, právě tehdy když je čtvercová a* $\det(A) = \pm 1$.

Důkaz. Buď $A \in Mat_{n\times m}\mathbb{Z}$ invertibilní. Existuje tedy matice $A^{-1} \in Mat_{n\times m}\mathbb{Z}$ taková, že $AA^{-1} = E$. Pak je ovšem A^{-1} inverzí pro A také nad $\mathbb{Q}$. Proto A musí být čtvercová, neboť každá invertiblní matice nad $\mathbb{Q}$ je čtvercová a má nenulový determinant. Navíc platí

$$\det(A)\cdot\det(A^{-1}) = \det(AA^{-1}) = \det(E) = 1$$

a protože determinant celočíselné matice je z definice determinantu také celočíselný, musí platit $det(A) = det(A^{-1}) = \pm 1$ neboť v okruhu $\mathbb{Z}$ máme pouze dvě jednotky a to právě ± 1.

Buď naopak $A \in Mat_{n\times m}\mathbb{Z}$ čtvercová s determinantem ± 1. Pak inverzní matici A^{-1} můžeme spočítat z algebraických doplňků jako

$$A^{-1} = \tfrac{1}{\det(A)} \cdot A_{\mathrm{adj}} = \pm A_{adj}$$

nicméně prvky matice A_{adj} - algebraické doplňky - se vypočítají ze subdeterminantů (minorů) matice A a musí být proto celočíselné. Matice A^{-1} je tedy celočíselná. □

Z tohoto lemmatu tedy plyne, že pokud chceme celočíselnou matici B převést do SNFpomocí invertibilních matic, musíme tak činit pouze prostřednictvím matic majících determinant ± 1. Nyní tedy můžeme přikročit k důkazu samotné věty o SNF.

Důkaz. (Věty o Smithově normálním tvaru). Nejprve dokážeme existenci SNF. Pro tento účel budeme potřebovat Euklidův algoritmus. Ten funguje následujícím způsobem.

Pro libovolná $a, b \in \mathbb{Z}$ taková, že $|a| > |b|$ vydělíme číslo a číslem b se zbytkem. Tedy $a = qb + c$. Pak ovšem platí, že $gcd(a,b) = gcd(b,c)$ neboť

$$gcd(a,b) = d \Rightarrow d|(a-qb) \Rightarrow d|c \Rightarrow d|gcd(b,c)$$

a naopak

$$gcd(b,c) = e \Rightarrow e|(qb+c) \Rightarrow e|a \Rightarrow e|gcd(a,b).$$

Takto můžeme postupovat rekurzivně a po konečném počtu kroků bude $c = 0$ a b příslušné danému kroku bude právě hledaný největší společný dělitel. Poznamenejme, že užití Euklidova algoritmu je z výpočetního hlediska výhodně, neboť má logaritmickou složitost.

Dále protože výsledné transformační matice P, Q musí být invertibilní nad $\mathbb{Z}$, plyne z předchozího lemmatu, že jejich determinant musí být roven ± 1. Evidentně tedy nemůžeme násobit řádek či sloupec matice jiným číslem než ± 1. Můžeme však prohodit libovolné dva řádky, protože to lze realizovat pomocí transformační matice,

$$\begin{pmatrix} 1 & & & & & & \\ & \ddots & & & & & \\ & & 0 & & 1 & & \\ & & & \ddots & & & \\ & & 1 & & 0 & & \\ & & & & & \ddots & \\ & & & & & & 0 \end{pmatrix}$$

která má evidentně determinant roven -1. Analogicky můžeme prohazovat prohazovat libovolné dva sloupce. A konečně pomocí transformační matice

$$\begin{pmatrix} 1 & & & & & & \\ & \ddots & & & & & \\ & & 1 & & & & \\ & & \vdots & \ddots & & & \\ & & m & \dots & 1 & & \\ & & & & & \ddots & \\ & & & & & & 1 \end{pmatrix}$$

můžeme k libovolnému řádku přičíst m-násobek jiného řádku.

Nyní budeme postupovat následujícím způsobem. Na pozici $(1,1)$ přesuneme libovolný nenulový prvek matice B (Pokud $B = 0$, pak je již ve SNFa žádné operace provádět nemusíme). Pak postupně pro každý prvek pod a napravo od prvku b_1^1 aplikujeme Euklidův algoritmus (konkrétně jeho implementaci pomocí řádkových a sloupcových operací, která potřebuje pouze operace násobení řádku/sloupce číslem -1, přičítání násobku řádku/sloupce k jinému a prohazování dvou řádků/sloupců), čímž na pozici $(1,1)$ vyrobíme největší společný prvků v prvním sloupci a řádku. Tyto prvky můžeme tedy snadno vyeliminovat, čímž získáme matici ve tvaru

$$B = \begin{pmatrix} b_1^1 & 0 & \dots & 0 \\ 0 & * & \dots & * \\ \vdots & \vdots & & \vdots \\ 0 & * & \dots & * \end{pmatrix}.$$

Pokud nyní existuje nějaký prvek b_j^i, který ještě není dělitelný b_1^1, můžeme přičíst j-tý sloupec k prvnímu sloupci a opět vyrobit na pozici $(1,1)$ prvek b'^1_1 takový, že $b'^1_1|b_1^1$ a zároveň $b'^1_1|b_j^i$, který jej již dělit bude. Poznamenejme, že tento prvek bude nutně menší než původní b_1^1, díky čemuž náš algoritmus skončí po konečném počtu kroků.

Celkem máme algoritmus, který převede matici B do výše uvedeného tvaru a navíc $b_1^1|b_j^i$. Označme takto vzniklou matici C a nechť $q_k = c_1^1$. Nyní můžeme postupovat indukcí a aplikovat tento algoritmus na submatici, která vznikne vynecháním prvního sloupce a řádku matice C. Neboť q_k dělil všechny prvky matice C, bude dělit i prvek v levém horním rohu submatice (označme jej q_{k+1}) po aplikaci výše uvedeného algoritmu. Dostáváme, že $q_k|q_{k+1}$, což jsme měli dokázat.

Zbývá dokázat jednoznačnost. Označme

$$gcd_{i\times i}(A) = gcd\{det(X)|X \text{ je submatice } A \text{ tvaru } i\times i\}$$

Prvně ukážeme, že platí rovnost

$$q_1\dots q_i = gcd_{i\times i}(A)$$

kde A je matice ve SNF. Pokud submatice X obsahuje k-tý řádek, ale neobsahuje k-tý sloupec matice A, bude její determinant evidentně nulový, neboť A je diagonální a X tak bude obsahovat nulový řádek. Stačí tedy uvažovat submatice jejichž diagonála leží na hlavní diagonále matice A. To znamená, že platí

$$gcd_{i\times i}(A) = gcd\{q_{k_1}\dots q_{k_i}|1\le k_1 < \dots < k_i \le r\}.$$

Navíc A je ve SNF, proto $q_i | q_{i+1}$ z čehož plyne

$$gcd\{q_{k_1} \dots q_{k_i} | 1 \leq k_1 < \dots < k_i \leq r\} = q_1 \dots q_i,$$

což jsme chtěli dokázat.

Konečně ukážeme, že největší společný dělitel subdeterminantů je invariantní vzhledem k elementárním řádkovým operacím (invariance vzhledem k sloupcovým operacím pak plyne ze symetrie).

Invariance vzhledem k násobení řádku číslem -1 a vzhledem k prohození řádků je zřejmá, neboť tyto operace maximálně změní znaménko některých subdeterminantů. To ovšem nemá žádný vliv na výsledného největšího společného dělitele. Pro přičítání násobku řádku je situace ovšem poněkud složitější. Každý nový subdeterminant je pak celočíselnou kombinací subdeterminantů předchozí matice. Z toho plyne, že

$$gcd_{i \times i}(A) | gcd_{i \times i}(A').$$

Jak jsme ale ukázali dříve, operace přičtení řádku je invertibilní. Můžeme tedy celý proces zopakovat opačným směrem a stejnou argumentací dostáváme

$$gcd_{i \times i}(A') | gcd_{i \times i}(A).$$

Největší společný dělitel subdeterminantů se tedy nezmění.

Předpokládejme nyní, že SNFnení jednoznačný a existují matice A, C a P, Q, T, U takové, že platí $B = P \cdot A \cdot Q = T \cdot C \cdot U$, kde A, C jsou různé a ve SNFa P, Q, T, U jsou celočíselné invertibilní matice. Pak násobení invertibilními maticemi P, Q, T, U odpovídá postupnému provádění elemntárních řádkových a sloupcových úprav, o kterých jsme ovšem dokázali, že nemění největšího společného dělitele subdeterminantů. To speciálně znamená, že hlavní minory matic A, C ve Smithově normálním tvaru jsou si rovny a proto i invariatní faktory musí být stejně. To je spor s předpokladem. Smithův normální tvar je tedy jednoznačný.

□

Kapitola 2

Triangularizace celočíselných matic

V této kapitole se budeme zabývat popisem algoritmu pro výpočet redukovaného schodovitého tvaru celočíselných matic. Tento algoritmus představil Arne Storjohann v článku nazvaném „*A fast+practial+deterministic algorithm for triangularizing integer matrices*" [6]. Definujme nejdříve tvar matice, jehož vytvoření bude našim cílem.

Definice 2.1. Řekneme že matice $A \in Mat_{n\times m}\mathbb{Z}$ je v redukovaném schodovitém tvaru (RST) jestliže splňuje následující podmínky:

(c1) Buď r hodnost matice A. Pak prvních r řádků je nenulových.

(c2) Pro každé $1 \leq i \leq r$ buď $A[i, j_i]$ první nenulový prvek v i-tém řádku. Pak $j_1 < j_2 < \cdots < j_r$.

(c3) Pro každé $1 \leq i \leq r$ platí $A[i, j_i] > 0$.

(c4) Pro každé $1 \leq k < i \leq r$ platí $A[i, j_i] > A[k, j_i] \geq 0$.

Poznámka 2.2. Poznamenejme, že první a druhá podmínka nám zaručují schodovitý tvar matice A. Tento však zjevně není jednoznačný. Proto je nutné přidat ještě podmínky (*c3*) a (*c4*). (*c3*) zajišťuje, že členy nad pivoty budou kladné a (*c4*) říká, že prvky nad pivoty budou pivoty omezeny. Tyto podmínky pak určují tvar matice A jednoznačně vzhledem k elementárním operacím.

Příklad 2.3. Pro ilustraci uvádíme následující matici v RST:

$$\begin{pmatrix} 2 & 33 & 6 & 0 & 39 & 73 \\ 0 & 0 & 24 & 0 & 444 & 8 \\ 0 & 0 & 0 & 1 & 22 & 23 \\ 0 & 0 & 0 & 0 & 0 & 75 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}$$

Pro další práci budeme potřebovat malé lemma, které budeme využívat například při přičítání násobku řádku k jinému řádku. Tento výsledek uvedl v roce 1992 E. Bach [8]. Nejprve však dokažme následující tvrzení:

Lemma 2.4. *Mějme kladná celá čísla $g > 1$ a N taková, že $g|N$. Pak existuje algoritmus, který najde celá čísla X, Y taková, že $N = XY$, $\gcd(X,Y) = 1$ a $\gcd(g,Y) = 1$.*

Důkaz. Nechť $\tilde{X} = g$ a $\tilde{Y} = \frac{N}{g}$. Pak můžeme vypočítat $h = \gcd(\tilde{X}, \tilde{Y})$ a zvolit $X = \tilde{X}h$, $Y = \frac{\tilde{Y}}{h}$. Z toho plyne, že $\gcd(X,Y) = 1$. Avšak $1 = \gcd(X,Y) = \gcd(gh,Y)$ implikuje, že také $\gcd(g,Y) = 1$. A konečně počítejme: $XY = \tilde{X}h\frac{\tilde{Y}}{h} = gh\frac{N}{gh} = N$. □

A nyní již k samotnému Bachovu lemmatu.

Lemma 2.5 (Bach). *Mějme celá čísla a, b a N, pro která platí $N > 0$ a zároveň $\gcd(a,b,N) = 1$. Pak existuje deterministický algoritmus, který pro čísla a, b a N vypočte celé číslo $0 < c < N$ takové, že $\gcd(a + cb, N) = 1$.*

Důkaz. Hledaný algoritmus můžeme zapsat následujícím způsobem: Nyní ukážeme, že

```
begin
    if gcd(a,N) = 1 then
        c ← 0;
    else if gcd(a+b,N) = 1 then
        c ← 1;
    else
        g ← gcd(a,N);
        (X,Y) ← čísla z předchozího lemmatu 2.4 pro proměnné g a N;
        c ← číslo 0 < c < N takové, že c ≡ 1 mod X a zároveň c ≡ 0 mod Y;
    end
end
```

takto definovaný algoritmus je skutečně korektní. V prvních dvou případech, kdy za c volíme 0 a 1, jsou podmínky kladené na koeficient c evidentně splněny. Zaměřme tedy naši pozornost na třetí možnost. Z $\gcd(a,N) \neq 1$ plyne, že g ve svém rozkladu obsahuje alespoň jedno prvočíslo dělící N, je tedy korektní aplikovat lemma 2.4 na čísla N a g. Výsledná čísla X, Y pak budou nesoudělná a pro výpočet vhodného c můžeme využít Čínskou zbytkovou větu.

Díky tomu můžeme odvodit $\gcd(a + cb, X) = \gcd(a + b, X) = 1$. První rovnost plyne z požadavků, které klademe na c. Dokážeme ještě druhou rovnost. Předpokládejme, že $\gcd(a + b, X) = d > 1$. Pak ale $d = \gcd(a + b, X) = \gcd(gk + b, gh)$, pro vhodná k, h. Z Bezoutovy rovnosti ovšem dostáváme celá čísla i, j taková, že můžeme psát

$$\begin{aligned} d &= \gcd(a+b, X) \\ &= \gcd(gk+b, gh) \\ &= i(gk+b) + jgh \\ &= (ik+jk)g + ib \end{aligned}$$

Druhá pak z toho, že X bude obsahovat všechny společné dělitele N a a a navíc z předpokladů máme $\gcd(a,b,N) = 1$. Podobně platí $\gcd(a + cb, Y) = \gcd(a,Y) = 1$, protože lemma 2.4 zaručuje $\gcd(g,Y) = 1$, $N = XY$ a zvolili jsme $g = \gcd(a,N)$.

A konečně z ekvivalence $\gcd(a + cb, N) = 1 \iff \gcd(a + cb, X) = 1 \wedge \gcd(a + cb, Y) = 1$ dostáváme naše tvrzení. □

Důsledek 2.6. *Uvedený algoritmus můžeme snadno rozšířit na případy, kdy hledáme $0 < c < N$ takové, že $\gcd(a+cb, N) = d$, kde $d = \gcd(a,b,N)$.*

Důkaz. Pomocí algoritmu z lemmatu 2.5 můžeme najít řešení úlohy $\gcd(\frac{a}{d} + c\frac{b}{d}, \frac{N}{d}) = 1$. Takto získané c pak jistě splní naše požadavky, neboť pronásobením předchozí rovnosti číslem d dostáváme $d = d\gcd(\frac{a}{d} + c\frac{b}{d}, \frac{N}{d}) = \gcd(d(\frac{a}{d} + c\frac{b}{d}), d\frac{N}{d}) = \gcd(a+cb, N)$. □

V následujících podkapitolách nejdříve popíšeme několik klíčových procedur, které budou upravovat vstupní matici A pomocí unimodulárních (mající determinant roven ± 1, tedy invertibilních) matic. Tyto procedury postupně propojíme a v poslední podkapitole pak obdržíme samotný algoritmus pro výpočet RST.

2.1 GCD redukce

Jak jsme viděli již v důkazu věty o Smithově normálním tvaru, častou operací, kterou s maticí při převodu do SNFprovádíme, je eliminace všech prvků nacházejících se pod nějakým námi zvoleným pivotem. Takováto eliminace je poměrně náročná, neboť pro každý prvek musíme vytvářet největší společný dělitel s pivotem. Bylo by proto výhodné, kdybychom mohli nějakým způsobem upravit prvky ve sloupci tak, že největší společný dělitel nějakých dvou prvků daného sloupce bude zároveň největším společným dělitelem všech prvků daného sloupce. A přesně to je obsahem následující věty.

Věta 2.7 (GCD redukce)**.** *Nechť $B \in Mat_{n\times m}\mathbb{Z}$ je matice $(k+2)\times k$ a $rank(B) = 2$, kterou můžeme zapsat jako*

$$B = \begin{pmatrix} N & \bar{N} \\ a_0 & \bar{a}_0 \\ b_1 & \bar{b}_1 \\ \vdots & \vdots \\ b_k & \bar{b}_k \end{pmatrix},$$

kde N je kladné. Pak existuje deterministický algoritmus, který pro matici B vypočte unimodulární matici

$$C = \begin{pmatrix} 1 & & & & \\ & 1 & c_1 & \cdots & c_k \\ & & 1 & & \\ & & & \ddots & \\ & & & & 1 \end{pmatrix}$$

takovou, že bude platit

$$CB = \begin{pmatrix} N & \bar{N} \\ a_k & \bar{a}_k \\ b_1 & \bar{b}_1 \\ \vdots & \vdots \\ b_k & \bar{b}_k \end{pmatrix} \qquad \text{kde} \qquad \begin{aligned} a_k &= a_0 + c_1 b_1 + \cdots + c_k b_k \\ \bar{a}_k &= \bar{a}_0 + c_1 \bar{b}_1 + \cdots + c_k \bar{b}_k \end{aligned}$$

a navíc CB bude splňovat následující podmínky:

(c1) hlavní submatice $\begin{pmatrix} N & \bar{N} \\ a_k & \bar{a}_k \end{pmatrix}$ *je regulární a*

(c2) $gcd(N, a_k) = gcd(N, a_0, b_1, b_2, \ldots, b_k)$.

Důkaz. Bez újmy na obecnosti můžeme předpokládat, že $k > 0$. Pokud by k bylo nulové, můžeme zřejmě za C zvolit identitu, které splní naše požadavky. Dále můžeme předpokládat, že hlavní submatice je regulární a tedy platí $N\bar{a}_0 - \bar{N}a_0 \neq 0$. Pokud by tomu tak nebylo, přičteme k druhému řádku nějaký řádek $2 < s \leq k+2$, pro který platí $N\bar{b}_s - \bar{N}b_s \neq 0$. Takový řádek jistě bude existovat, neboť matice B má plnou hodnost. Výsledná matice pak bude mít hlavní submatici regulární. Pro takto upravenou matici můžeme spočítat hledané koeficienty c_i a konečně ke koeficientu c_s přičteme 1, což bude přesně odpovídat onomu přičtení s-tého řádku, které jsme provedli na začátku.

Nyní ukážeme, jak iterativně vypočítat c_l pro $l = 1, \ldots, k$. Označme mezivýsledky našeho výpočtu následujícím způsobem:

$$\begin{aligned} a_l &= a_0 + c_1 b_1 + \cdots + c_l b_l \\ \bar{a}_l &= \bar{a}_0 + c_1 \bar{b}_1 + \cdots + c_l \bar{b}_l \end{aligned} \tag{2.1}$$

Po provedení kroku $l-1$ a na začátku kroku l jsou vypočítány koeficienty $c_1, \ldots, c_{l-1}$ a jsou splněny podmínky

(1) $gcd(N, a_i) = gcd(N, a_0, b_1, b_2, \ldots, b_i)$

(2) $N\bar{a}_i - \bar{N}a_i \neq 0$

pro $i = l-1$. Poznamenejme, že pro $i = 0$ jsou podmínky (1) a (2) splněny triviálně. Teď musíme provést indukční krok - najít vhodné c_l takové, že budou splněny podmínky (1) a (2) pro $i = l$.

Nechť $g = gcd(a_{l-1}, b_l)$. Pak můžeme dělením se zbytkem najít celá čísla q_1, q_2 a $0 \leq \tilde{a}_{l-1}, \tilde{b}_l < N$ taková, že platí

$$\begin{aligned} a_{l-1}/g &= q_1 N + \tilde{a}_{l-1} \\ b_l/g &= q_2 N + \tilde{b}_l \end{aligned} \tag{2.2}$$

Čísla $\tilde{a}_{l-1}$ a $\tilde{b}_l$ jsou nesoudělná (snadno plyne z Bezoutovy rovnosti). Pomocí algoritmu, který jsme použili v důkazu lemmatu 2.5, můžeme najít nejmenší kladné číslo t takové, že bude platit

$$gcd(\tilde{a}_{l-1} + t\tilde{b}_l, N) = 1 \tag{2.3}$$

a volbou $c_l \leftarrow t$ zajistíme splnění podmínky (1). Skutečně:

$$\begin{aligned} gcd(a_l,N) &= gcd(a_{l-1}+tb_l,N) \\ &= gcd(g(q_1N+\tilde{a}_{l-1})+tg(q_2N+\tilde{b}_l),N) \\ &= gcd(g(\tilde{a}_{l-1}+t\tilde{b}_l)+g(q_1+tq_2)N,N) \\ &= gcd(g(\tilde{a}_{l-1}+t\tilde{b}_l),N) \\ &= gcd(g,N) \\ &= gcd(a_{l-1},b_l,N) \\ &= gcd(N,a_0,b_1,b_2,\ldots,b_l) \end{aligned}$$

přičemž poslední rovnost plyne z indukčního předpokladu.

Nakonec musíme zajistit splnění i druhé podmínky (2). Buď l index aktuálního kroku a předpokládejme, že platí

$$\begin{vmatrix} N & \bar{N} \\ a_{l-1}+xb_l & \bar{a}_{l-1}+x\bar{b}_l \end{vmatrix} = 0 \tag{2.4}$$

pak ovšem z indukčního předpokladu plyne, že $N\bar{b}_l - \bar{N}b_l \neq 0$. To implikuje, že prvek x je určen jednoznačně a můžeme jej vyjádřit jako

$$x = -\frac{N\bar{a}_{l-1} - \bar{N}a_{l-1}}{N\bar{b}_l - \bar{N}b_l} \tag{2.5}$$

Poznamenejme, že z indukčního předpokladu také plyne, že $x \neq 0$.

Pokud nám tedy v kroku 2.3 výjde c_l různé od x, je vše v pořádku. Pokud ovšem $c_l = t = x$, nebyla by podmínka (2) splněna. To ale můžeme snadno napravit. Předpokládejme tedy, že $0 < x = t$. Nechť $\bar{t}$ je nejmenší nezáporné číslo, pro které platí $gcd(\tilde{a}_{l-1}+\bar{t}(-\tilde{b}_l),N) = 1$. Volbou $c_l \leftarrow -\bar{t}$ zajistíme splnění podmínky (2), protože $c_l = -\bar{t} \leq 0 < x$. Platnost podmínky (1) pro takovouto volbu c_l se pak dokáže zcela analogicky, jako jsme to již provedli výše pro $c_l = t$. □

2.2 Sloupcová redukce

V této části si ukážeme, jak využít výsledků předcházející věty 2.7 k eliminaci prvků ve sloupečku. Mějme tedy jako v předchozím $n \times 2$ vstupní matici B, kteroužto můžeme zapsat následujícím způsobem:

$$B = \begin{pmatrix} N & \bar{N} \\ a_0 & \bar{a}_0 \\ b_1 & \bar{b}_1 \\ \vdots & \vdots \\ b_k & \bar{b}_k \end{pmatrix}, \tag{2.6}$$

kde $k \geq 0$, $N > 0$ a trailing $(k+2) \times 2$ submatrix má plnou hodnost (nejsme si vědomi českého ekvivalentu pro výraz trailing (sub)matrix, a budeme jej proto v následujícím textu používát v nezměněné původní podobě).

Našim cílem bude nalézt $n \times n$ unimodulární matice

$$C = \begin{pmatrix} 1 & & & & & & \\ & \ddots & & & & & \\ & & 1 & * & \cdots & * & \\ & & & 1 & & & \\ & & & & \ddots & & \\ & & & & & 1 & \end{pmatrix} \quad \text{a} \quad Q = \begin{pmatrix} 1 & & & * & * & & & \\ & \ddots & & \vdots & \vdots & & & \\ & & 1 & * & * & & & \\ & & & * & * & & & \\ & & & * & * & & & \\ & & & * & * & 1 & & \\ & & & \vdots & \vdots & & \ddots & \\ & & & * & * & & & 1 \end{pmatrix}, \tag{2.7}$$

které budou reprezentovat příslušné invertibilní operace takové, že součin matic QCB můžeme psát jako

$$QCB = \begin{pmatrix} * & * \\ \vdots & \vdots \\ * & * \\ t_1 & * \\ & t_2 \\ & * \\ & \vdots \\ & * \end{pmatrix} \tag{2.8}$$

a budou splněny podmínky následující věty.

Věta 2.8 (Sloupcová redukce)**.** *Mějme matici $B \in Mat_{n\times 2}\mathbb{Z}$, kterou můžeme zapsat jako v 2.6 s tím, že $k \geq 0$, $N > 0$ a trailing $(k+2) \times 2$ submatrix má plnou hodnost. Pak existuje algoritmus* ***ColumnReduction***(B,k)*, který na vstupu vezme B a k, a jako výstup vrátí $n \times n$ matice C a Q, které lze vyjádřit jako v 2.7. Navíc bude platit, že součin QCB lze psát jako 2.8 a bude splňovat následující podmínky:*

(c1) $t_1 > 0$ *a* $t_2 > 0$,

(c2) prvky nad t_1 v prvním sloupci jsou nezáporné a shora omezené číslem $t_1 - 1$,

(c3) prvky nad a pod t_2 ve druhém sloupci jsou nezáporné a shora omezené číslem $t_2 - 1$.

Důkaz. Nejprve aplikujeme algoritmus věty 2.7 o GCD redukci na submatici matice B tvořenou posledními $k+2$ řádky. Tím získáme transformační $(k+2) \times (k+2)$ matici C', kterou když vhodně vložíme do jednotkové matice $n \times n$, získáme hledanou matici C, která bude splňovat naše požadavky. Konkrétně:

$$C = \begin{pmatrix} 1 & & & \\ & \ddots & & \\ & & 1 & \\ & & & C' \end{pmatrix}.$$

Aplikací matice C na vstupní matici B dostáváme

$$CB = \begin{pmatrix} * & * \\ \vdots & \vdots \\ * & * \\ N & \bar{N} \\ a_k & \bar{a}_k \\ b_1 & \bar{b}_1 \\ \vdots & \vdots \\ b_k & \bar{b}_k \end{pmatrix}$$

s tím, že $gcd(N, a_k) = gcd(N, a_k, b_1, b_2, \ldots, b_k)$ a navíc submatice

$$\begin{pmatrix} N & \bar{N} \\ a_k & \bar{a}_k \end{pmatrix}$$

bude regulární.

Aplikací rozšířeného Euklidova algoritmu na dvojici (N, a_k) obdržíme uspořádanou trojici (t_1, m_1, m_2) takovou, že $m_1 N + m_2 a_k = t_1 = gcd(N, a_k)$. Nyní můžeme vytvořit matici

$$U = \begin{pmatrix} m_1 & m_2 \\ -sa_k/t_1 & sN/t_1 \end{pmatrix},$$

kde $s \in \{1, -1\}$ je zvoleno tak, aby $t_2 = (-sa_k/t_1)\bar{N} + (sN/t_1)\bar{a}_k$ bylo kladné. Matice U je unimodulární, neboť

$$\det U = \begin{vmatrix} m_1 & m_2 \\ -sa_k/t_1 & sN/t_1 \end{vmatrix} = \frac{s(m_1 N + m_2 a_k)}{t_1} = \frac{st_1}{t_1} = \pm 1.$$

A konečně můžeme zkonstruovat matici

$$Q = \begin{pmatrix} 1 & & & & & & \\ & \ddots & & & & & \\ & & 1 & & & & \\ & & & U & & & \\ & & & & 1 & & \\ & & & & & \ddots & \\ & & & & & & 1 \end{pmatrix},$$

která, jak plyne z předchozího, bude také unimodulární. Aplikací Q na matici CB dostáváme

$$QCB = \begin{pmatrix} * & * \\ \vdots & \vdots \\ * & * \\ t_1 & * \\ & t_2 \\ b_1 & \bar{b}_1 \\ \vdots & \vdots \\ b_k & \bar{b}_k \end{pmatrix}$$

a platí $t_1 \mid b_i$ kde $i = 1, \dots, k$. Můžeme tedy snadno vyeliminovat prvky pod t_1. Následně snadno provedeme redukci prvků nad t_1 a konečně i prvků nad a pod t_2. To lze provést pomocí elementárních řádkových operací, které můžeme odpovídajícím způsobem zapsat do matice Q. Takto upravená matice Q již bude splňovat podmínky věty a důkaz je tak hotov.

□

2.3 RST algoritmus

V následujícím textu využijeme výše popsanou proceduru Sloupcové redukce k vytvoření RST algoritmu, který na vstupu bere $n' \times m'$ vstupní matici A' a vrací její redukovaný schodovitý tvar včetně transformační matice. Nejdříve však musíme definovat pojem rank profile (nejsme si vědomi existence vhodného českého ekvivalentu, proto budeme používat původní anglický výraz).

Definice 2.9. Buď A matice $n \times m$ a nechť r značí hodnost matice A. Nechť G je reprezentace matice A ve schodovitém tvaru. Pod pojmem **rank profile** pak rozumíme uspořádanou r-tici $(j_1, \dots, j_r)$, kde j_i je sloupcový index prvního nenulového prvku v i-tém řádku matice G.

Abychom se vyhnuli ošetřování množství speciálních případů (například matice mající hodnost 0 a podobně), budeme namísto matice A' uvažovat matici

$$A = \begin{pmatrix} 1 & & \\ & A' & \\ & & 1 \end{pmatrix}. \tag{2.9}$$

Poznamenejme, že takováto matice bude mít rank profile ve tvaru $(1, j_2, \dots, j_{r-1}, m)$, kde $r \geq 2$ je hodnost matice A a $n \times m$ jsou její rozměry.

Nyní budeme definovat RST algoritmus. Pro názornost nejdříve uvedeme variantu, která potřebuje dopředu znát rank profile. Ten je možné spočítat například Gausovou eliminační metodou. Poznamenejme, že Gausova eliminace spadá do složitostní třídy $\mathcal{O}(n^3)$, což je zanedbatelné vzhledem k celkové složitosti našeho algoritmu. Přesto však později uvedeme také jednoduchou modifikaci RST algoritmu, která již rank profile nevyžaduje. Nyní přistupme k samotné definici.

Algoritmus: Výpočet redukovaného schodovitého tvaru
Data: Celočíselná $n \times m$ matice A mající rank profile $(j_1, \dots, j_r)$, kterou lze zapsat jako v 2.9.
Result: Matice Q, C a T splňující $QCA = T$, kde Q a C jsou unimodulární a T má prvních $m-1$ sloupců v redukovaném schodovitém tvaru.

```
begin
    Q^(0) ← I_n;
    C^(0) ← I_n;
    T^(0) ← A;
    for k ← 1 to r − 1 do
        B_k ← n × 2 matice (col(T^(k−1), j_k) | col(T^(k−1), j_{k+1}));
        (Q̃, C̃) ← ColumnReduction(B_k, n − k − 1);
        Q^(k) ← Q̃C̃Q^(k−1)C̃^(−1);
        C^(k) ← C̃C^(k−1);
        T^(k) ← Q̃C̃T^(k−1);
    end
    return (Q^(r−1), C^(r−1), T^(r−1))
end
```

Abychom dokázali, že výše uvedený algoritmus je skutečně korektní, ukážeme nejdříve, že matice $Q^{(i)}, C^{(i)}, T^{(i)}$, které dostáváme v průběhu výpočtu, můžeme zapsat následujícím způsobem:

$$\overset{Q^{(i)}}{\left(\begin{array}{c|c|c} I_1 & & \\ \hline & * & \\ \hline & * & I_{n-i-1} \end{array}\right)} \overset{C^{(i)}}{\left(\begin{array}{c|c|c} I_1 & & \\ \hline & * & * \\ \hline & & I_{n-i-1} \end{array}\right)} A = \overset{T^{(i)}}{\left(\begin{array}{c|c|c} I_1 & & \\ \hline & R_i & * \\ \hline & & * \end{array}\right)} \tag{2.10}$$

pro $i = 0, 1, \dots, r-1$, kde R_i je matice v redukovaném schodovitém tvaru.

Lemma 2.10. *Pro všechna $i = 0, 1, \dots, r-1$ můžeme matice $Q^{(i)}, C^{(i)}$ a $T^{(i)}$ psát jako v 2.10, kde R_i je $(i-1) \times (j_{i+1} - 2)$ matice v redukovaném schodovitém tvaru. Navíc bude platit:*

(c1) $Q^{(i)}$ a $C^{(i)}$ jsou unimodulární a

(c2) prvek $T^{(i)}[i+1, j_{i+1}]$, který budeme značit N_{i+1}, bude nenulový a navíc bude platit $0 \le T^{(i)}[l, j_{i+1}] < N_{i+1}$ pro $l = 1, \dots, i, i+1, \dots, m$.

Důkaz. Důkaz provedeme indukcí. V inicálním stavu algoritmu $i = 0$ jsou požadavky lemmatu triviálně splněny, neboť $Q^{(0)} = C^{(0)} = I_n$. Předpokládejme tedy, že lemma platí pro $i = k - 1 < r$, pro nějaké k kladné. Dokážeme, že pak lemma platí také pro $i = k$.

$n \times 2$ submatici B_k, tvořenou sloupci j_k a j_{k+1} matice $T^{(k-1)}$, můžeme psát jako

$$B_k = \begin{pmatrix} * & * \\ \vdots & \vdots \\ * & * \\ N_k & \bar{N}_k \\ a_a & \bar{a}_0 \\ b_1 & \bar{b}_1 \\ \vdots & \vdots \\ b'_k & \bar{b}'_k \end{pmatrix},$$

kde $k' = n - k - 1$. Z indukčního předpokladu plyne, že submatice B_k má následující dvě vlastnosti. (1) $N_k > 0$, což plyne z podmínky c2 a toho, že $N_k = T^{(k-1)}[k, j_k]$. A také (2) submatice složená z posledních $k' + 2$ řádků matice B_k bude mít plnou hodnost. To dostaneme z toho, že B_k je složená ze sloupců j_k a j_{k+1} matice $T^{(k-1)}$ a navíc všechny prvky nalevo od j_k-tého sloupce v řádcích $k, k+1, \dots, m$ matice $T^{(k-1)}$ jsou nulové (skutečně, jedná se o prvky, které se nachází pod blokem R_{k-1} z vyjádření 2.10).

Vlastnosti (1) a (2) nám zaručují, že submatice B_k je validním vstupem pro algoritmus **ColumnReduction**. Z něj pak získáme matice $\tilde{Q}$ a $\tilde{C}$, které budou unimodulární. Navíc díky jejich specifické struktuře, kterou nám garantuje věta 2.8, budou mít matice $Q^{(k)} \leftarrow \tilde{Q}\tilde{C}Q^{(k-1)}\tilde{C}^{-1}$ a $C^{(k)} \leftarrow \tilde{C}C^{(k-1)}$ strukturu zachycenou v rovnosti 2.10.

Nyní ukážeme, že rovnost 2.10 je splněna pro $i = k$ na konci k-tého cyklu našeho algoritmu. To znamená ověřit rovnost $T^{(k)} = Q^{(k)}C^{(k)}A$. Poznamenejme, že matici $T^{(k)}$ vypočítáme jakožto $T^{(k)} \leftarrow \tilde{Q}\tilde{C}T^{(k-1)}$ a navíc z indukčního předpokladu vím, že platí $T^{(k-1)} = Q^{(k-1)}C^{(k-1)}A$. To všechno nám dohromady dává následující rovnosti:

$$\begin{aligned} T^{(k)} &= \tilde{Q}\tilde{C}T^{(k-1)} \\ &= \tilde{Q}\tilde{C}(Q^{(k-1)}C^{(k-1)}A) \\ &= \tilde{Q}\tilde{C}(Q^{(k-1)}(\tilde{C}^{-1}\tilde{C})C^{(k-1)}A) \\ &= (\tilde{Q}\tilde{C}Q^{(k-1)}\tilde{C}^{-1})(\tilde{C}C^{(k-1)})A \\ &= Q^{(k)}C^{(k)}A \end{aligned}$$

a rovnost 2.10 je pro $i = k$ skutečně splněna.

Nakonec uveďme, že díky struktuře součinu matic $\tilde{Q}\tilde{C}B_k$, kterou nám garantuje věta 2.8, a díky tomu, že submatice tvořená řádky $k, k+1, \dots, m$ a sloupci $j_k, j_k + 1, \dots, j_{k+1} - 1$ má hodnost 1 (plyne z definice rank profile a indukčního předpokladu), bude možné matici $T^{(k)} = \tilde{Q}\tilde{C}T^{(k-1)}$ schématicky zapsat jako v rovnosti 2.10 a navíc bude splňovat požadavky našeho lemmatu. □

RST algoritmus 2 bere na vstupu libovolnou matici A' o rozměrech $n \times m$, která všam musí byt vložena do $(n+2) \times (m+2)$ matice A tak, jak je znázorňu je 2.9. Nyní ukážeme, že toto není nikterak omezující, protože výstupní matice ($Q^{(r-1)}$, $C^{(r-1)}$, $T^{(r-1)}$) mohou být zachyceny následujícím schématem:

$$\overset{Q^{(r-1)}}{\left(\begin{array}{c|c|c} I_1 & & \\ \hline & Q' & \\ \hline & * & I_1 \end{array}\right)}\overset{C^{(r-1)}}{\left(\begin{array}{c|c|c} I_1 & & \\ \hline & C' & * \\ \hline & & I_1 \end{array}\right)}\overset{A}{\left(\begin{array}{c|c|c} I_1 & & \\ \hline & A' & \\ \hline & & I_1 \end{array}\right)} = \overset{T^{(r-1)}}{\left(\begin{array}{c|c|c} I_1 & & \\ \hline & R_{r-1} & * \\ \hline & & N_r \end{array}\right)} \tag{2.11}$$

Věta 2.11. *Algoritmus pro výpočet redukovaného schodovitého tvaru je korektní. Výstupní matice ($Q^{(r-1)}$, $C^{(r-1)}$, $T^{(r-1)}$) mají tvar, který odpovídá rovnosti 2.11. Speciálně platí, že Q' a C' jsou unimodulární a splňují $Q'C'A' = R_{r-1}$, kde R_{r-1} je redukovaný stupňovitý tvar matice A'.*

Důkaz. Platnost rovnosti 2.11 snadno plyne přímo z lemmatu 2.10 a rovnosti 2.10 pro $i = r-1$. Stejně tak lze snadno odvodit, že Q' a C' jsou unimodulární a splňují $Q'C'A' = R_{r-1}$. □

Kapitola 3

Výpočet Smithova normálního tvaru trojúhelníkových matic

V této sekci navážeme na předchozí kapitolu a uvedeme co to je Hermitův normální tvar a jaký je jeho vztah k redukovanému schodovitému tvaru. Těchto výsledků pak využijeme pro vytvoření algoritmu, který vypočítá Smithův normální tvar právě z Hermitova normálního tvaru. Při popisu algoritmů a výsledků v této kapitole budeme vycházet zejména z článku „*Computing Hermite and Smith normal forms of triangular integer matrices*" [7], který v roce 1998 publikoval Arne Storjohann.

3.1 Hermitův normální tvar

Definice 3.1. Nechť $H \in Mat_{n\times n}\mathbb{Z}$ je matice mající plnou hodnost, kterou lze zapsat následujícím způsobem:

$$H = \begin{pmatrix} h_1 & h_{12} & \dots & h_{1n} \\ & h_2 & \dots & h_{2n} \\ & & \ddots & \vdots \\ & & & h_n \end{pmatrix}. \tag{3.1}$$

Pak H je v Hermitově normálním tvaru, právě když splňuje následující podmínky:

(c1) Pro každé $j \in \{1,\dots,n\}$ platí $h_j > 0$ a zároveň

(c2) $0 \leq h_{ij} < h_j$ pro všechna $1 \leq i < j \leq n$.

Poznámka 3.2. Matice splňující požadavky předchozí definice jsou tedy horní trojúhelníkové a regulární. Navíc pro ně platí, že prvky nad diagonálou jsou nezáporné a shora omezené prvkem na diagonále, který musí být kladný.

V článku [7] autor dále rozvádí, jakým způsobem je možné vypočítat Hermitův normální tvar z horní trojúhelníkové matice, jejíž prvky jsou omezeny determinantem. Tím se však v tomto textu nemusíme zabývat, neboť lze snadno nahlédnout, že máme-li matici v redukovaném schodovitém tvaru a vhodným způsobem přeskupíme její sloupce, dostaneme matici jejíž hlavní čtvercová submatice bude splňovat podmínky Hermitova normálního tvaru. Toto schéma detailněji popíšeme na konci této kapitoly a navíc uvedeme algoritmus pro výpočet SNF čtvercové matice v Hermitově normálním tvaru.

3.2 Sloupcová eliminace

Buď T $k \times m$ matice mající hodnost k, jejíž prvních $k-1$ sloupců je ve Smithově normálním tvaru. Matici T můžeme schématicky zapsat následujícím způsobem:

$$T = \left(\begin{array}{ccccc|ccc} a_1 & & & & t_1 & * & \dots & * \\ & a_2 & & & t_2 & \vdots & & \vdots \\ & & \ddots & & \vdots & \vdots & & \vdots \\ & & & a_{k-1} & t_{k-1} & \vdots & & \vdots \\ & & & & t_k & * & \dots & * \end{array}\right). \tag{3.2}$$

Cílem této sekce bude dokázat následující tvrzení.

Věta 3.3 (Sloupcová eliminace)**.** *Nechť T je matice, kterou lze zapsat stejně jako v rovnosti 3.2 a navíc splňuje následující podmínky:*

(c1) hlavní $k \times k$ submatice má plnou hodnost,

(c2) prvních $k-1$ sloupců je ve Smithově normálním tvaru,

(c3) prvky t_i, $i \in \{1,\dots,k-1\}$, jsou redukovány modulo t_k.

Pak existuje deterministický algoritmus, který pomocí ekvivalentních řádkových a sloupcových operací převede hlavní $k \times k$ submatici T do Smithova normálního tvaru.

Pro lepší čitelnost rozdělíme důkaz této věty do několika lemmat, které na konci této kapitoly spojíme v kompletní důkaz.

Lemma 3.4. *Buď T matice splňující stejné podmínky jako v předpokladech věty 3.3 a navíc nechť $k > 1$. Pak existuje deterministický algoritmus, který převede matici T na ekvivalentní matici splňující stejné podmínky, ale navíc bude platit*

(c4) $\gcd(a_i,t_i) = \gcd(a_i,t_i,t_{i+1},\dots,t_k)$ *pro* $1 \le i \le k-1$.

Důkaz. Poznamenejme, že prvek $T_{i,j}$ se bude vztahovat k aktuálnímu stavu transformované matice v průběhu algoritmu. Ostatní proměnné (jako například t_i) budou odpovídat původním hodnotám, které jsme si zafixovali na začátku algoritmu.

Algoritmus bude pracovat iterativně vzhledem k proměnné r, která bude značit aktuálně zpracovávaný řádek. Je zřejmé, že pro $r = k$, bude platit

$$\gcd(T_{r,r},T_{r,k}) = \gcd(T_{r,r},T_{r,k},T_{r+1,k},\dots,T_{k,k}). \tag{3.3}$$

Můžeme tedy předpokládat, že pro nějaké i, $1 \le i < k$, splňuje matice T rovnost 3.3 pro všechna $r = k,k-1,\dots,i+1$. Nyní ukážeme, jakým způsobem aplikovat ekvivalentní řádkové a sloupcové operace na matici T tak, aby výsledná matice splňovala podmínky (c1)-(c4) a platila rovnost 3.3 pro $r = k, k-1, \ldots, i$.

Buď $0 \leq c < a_i$ řešením rovnosti $\gcd(t_i + ct_{i+1}, a_i) = \gcd(t_i, t_{i+1}, a_i)$. Takovéto c můžeme získat z důsledku 2.6. Přičtením c-násobku řádku $i+1$ k i-tému řádku $\mathrm{row}(T, i) \mathrel{+}= \mathrm{row}(T, i+1)$ získáme matici T' ve tvaru

$$T' = \left(\begin{array}{cccccc|ccc}
a_1 & & & & & t_1 & * & \ldots & * \\
& \ddots & & & & \vdots & \vdots & & \vdots \\
& & a_i & ca_{i+1} & & t_i + ct_{i+1} & \vdots & & \vdots \\
& & & a_{i+1} & & t_{i+1} & \vdots & & \vdots \\
& & & & \ddots & \vdots & \vdots & & \vdots \\
& & & & a_{k-1} & t_{k-1} & \vdots & & \vdots \\
& & & & & t_k & * & \ldots & *
\end{array}\right).$$

□

Závěr

Příloha

Sem můžete přidat přílohu. Pokud chcete “přílohy”, tak upravte definici záhlaví v souboru sci.muni.thesis.sty, viz řádek 644.

Seznam použité literatury

[1] S. J. Monaquel a K. M. Schmidt, *On M-functions and operator theory for non-self-adjoint discrete Hamiltonian systems*, v „Special Issue: 65th birthday of Prof. Desmond Evans“, J.Comput. Appl. Math. **208** (2007), č. 1, 82–101.

[2] M. Murata, *Positive solutions and large time behaviors of Schrödinger semigroups, Simon's problem*, J. Funct. Anal. **56** (1984), č. 3, 300–310.

[3] J. Qi a S. Chen, *Strong limit-point classification of singular Hamiltonian expressions*, Proc. Amer. Math. Soc. **132** (2004), č. 6, 1667–1674 (elektronicky).

[4] Z. Pospíšil, *An inverse problem for matrix trigonometric and hyperbolic functions on measure chains*, v „Colloquium on Differential and Difference Equations — CDDE 2002“ (Brno, 2002), Folia Fac. Sci. Natur. Univ. Masaryk. Brun. Math. **13**, str. 205–211, Masarykova univerzita, Brno, 2003.

[5] R. Šimon Hilscher a P. Zemánek, *Friedrichs extension of operators defined by linear Hamiltonian systems on unbounded interval*, v „Equadiff 12“, Proceedings of the Conference on Differential Equations and their Applications (Brno, 2009), J. Diblík, O. Došlý, P. Drábek a E. Feistauer, editoři, Math. Bohem. **135** (2010), č. 2, 209–222.

[6] A. Storjohann, *A Fast+Practical+Deterministic Algorithm for Triangularizing Integer Matrices*, Cosi, Springer-Verlag, Zurich, 1996.

[7] A. Storjohann, *Computing Hermite and Smith normal forms of triangular integer matrices*, Cosi, Springer-Verlag, Zurich, 1998.

[8] E. Bach, *Linear algebra modulo N, unpublished manuscript*, Prosinec 1992.