Linear equations over finite abelian groups

Bettina Eick

October 28, 2011

1 Introduction

One of the oldest problems in algebra is the solution of a system of linear equations over certain domains. The Gaussian elimination algorithm provides an effective solution over fields. The Smith normal form algorithm yields a method over the integers. Here we consider another variation of this theme.
Let A be a finite (additive) abelian group, let $\alpha_{i,j} \in End(A)$ for $1 \leq i \leq n$ and $1 \leq j \leq m$ and let $b_1, \ldots, b_m \in A$. Then we want to determine the set $\mathcal{S}$ of all $(x_1, \ldots, x_n) \in A^n$ with

$$\begin{array}{rcl} \alpha_{1,1}(x_1) + \ldots + \alpha_{n,1}(x_n) & = & b_1 \\ \vdots \qquad\qquad \vdots \qquad & & \vdots \\ \alpha_{1,m}(x_1) + \ldots + \alpha_{n,m}(x_n) & = & b_m \end{array}$$

As a first step, we combine the endomorphisms $\alpha_{i,j}$ of A to an homomorphism $\alpha : A^n \to A^m$ defined by

$$\alpha(x_1, \ldots, x_n) = (\sum_i \alpha_{i,1}(x_i), \ldots, \sum_i \alpha_{i,m}(x_i)).$$

Our considered problem then translates to determining all elements $x = (x_1, \ldots, x_n)$ of A^n with $\alpha(x) = b$ for $b = (b_1, \ldots, b_m) \in A^m$. In other words, we want to decide whether b is an element of the image of α and, if so, then we want to compute a preimage k of b under α and the kernel K of α. The solutions $\mathcal{S}$ are then given by the coset $k + K$.
In the following we consider two approaches towards this problem. The first uses that A is a quotient of a free abelian group and applies the Smith normal form algorithm. This approach is folklore; In the special case of a homogeneous system of equations a method to solve this is available as 'PcpNullspaceMatMod' in the Polycyclic Package [2]. The second approach uses that A is a direct sum of p-groups and applies Gaussian elimination over finite fields iteratedly. This method extends the special case considered in [1, III.19]. The implementation of this second approach will also be made available in the Polycyclic Package [2].
An alternative approach towards solving the considered problem is given by Hartung [3].

2 Preliminaries

Let $\mathbb{Z}^k$ denote the free abelian group of rank k; that is, the direct sum of k copies of the infinite cyclic group $\mathbb{Z}$. More generally, for an abelian group K and $l \in \mathbb{N}$ we denote with K^l the direct sum of l copies of the abelian group K.
As A is finitely generated abelian, it follows that $A \cong \mathbb{Z}^k/T$ for some $k \in \mathbb{N}$ and for some subgroup T of $\mathbb{Z}^k$. This implies that $A^n \cong \mathbb{Z}^{nk}/T^n$ and $A^m \cong \mathbb{Z}^{mk}/T^m$. Thus we can represent the elements of A^n and A^m by integral vectors. In the following we usually identify the elements of A^n and A^m with integral vectors to shorten notation. Further, we use the representations of A^n and A^m as quotients of free abelian groups to represent the homomorphism α by an $mk \times nk$ integral matrix M.
In this setting, we aim to determine the set $\mathcal{S} \subseteq A^n$ of all solutions $x \in A^n$ solving the integral system of equations

$$Mx \equiv b \bmod T^n.$$

3 The Smith normal form approach

Let $\mathcal{T} \subseteq \mathbb{Z}^{nk}$ denote the solutions of the integral system $xM \equiv b \bmod T^m$. Then the natural homomorphism of abelian groups $\mathbb{Z}^{nk} \to \mathbb{Z}^{nk}/T^n \cong A^n$ induces a surjection $\mathcal{T} \to \mathcal{S}$ with kernel T^n. Hence we can determine $\mathcal{S}$ by computing $\mathcal{T}$. The latter can be achieved as follows.
Let B be an $mk \times mk$ integer matrix whose rows generate T^m and let E denote the $(mk \times nk + mk)$ matrix obtained by concatenating the rows of M and the rows of B. The Smith normal form algorithm allows to determine invertible integer matrices P and Q so that $PDQ = E$ holds for a diagonal matrix D. This yields that

$$yE = b \quad \Leftrightarrow \quad y(PDQ) = b \quad \Leftrightarrow \quad (yP)D = bQ^{-1} =: b'.$$

The solutions y' of the system $y'D = b'$ can be read off readily from the diagonal matrix D. The solutions y of the system $yE = b$ can then be obtained via $y = y'P^{-1}$. The following straightforward lemma exhibits how $\mathcal{T}$ can be determined from these solutions $y \in \mathbb{Z}^{nk+mk}$.

1 Lemma: *Let $y \in \mathbb{Z}^{nk+mk}$ be the concatenation of the vectors $x \in \mathbb{Z}^{nk}$ and $z \in \mathbb{Z}^{mk}$. Then y satisfies $yE = b$ if and only if x satisfies $xM = b \bmod T^m$.*

4 The Gaussian elimination approach

The finite abelian group A is a direct sum of its Sylow subgroups: $A = A_{p_1} \oplus \ldots \oplus A_{p_l}$. The endomorphism α leaves every of the Sylow subgroups invariant. Hence we can solve the considered system for each of the Sylow subgroups and then compose the solution for A from solutions for the Sylow subgroups. We thus assume in the remainder of this section that A is a finite abelian p-group.

We use induction on the exponent of A to solve the considered system. To shorten notation, we write $V = A^n$ and $W = A^m$. Further, we denote $A_i = A/p^iA$ and, similarly, $V_i = V/p^iV$ and $W_i = W/p^iW$. Note that α maps p^iV into p^iW and thus α induces a homomorphism α_i from V_i to W_i via

$$\alpha_i : V_i \to W_i : v + p^iV \mapsto \alpha(v) + p^iW.$$

4.1 The initial step

In the first step of the induction we solve the considered system over A_1. As A_1 is elementary abelian, this reduces to solving the system $xM = b$ over the field with p elements. Thus a single solution k_1 and a basis B_1 for the kernel K_1 of M can be determined with the Gaussian elimination algorithm.

4.2 The induction step

In the induction step we assume that we are given a single solution k_i for the system over A_i and a generating set B_i for the kernel K_i of M over A_i. We wish to determine a single solution k_{i+1} for the system over A_{i+1} and a generatings set B_{i+1} for the kernel K_{i+1} of M over A_{i+1}. We consider the natural epimorphism

$$\nu_i : V_{i+1} \mapsto V_i : v + p^{i+1}V \mapsto v + p^iV$$

with kernel p^iV_{i+1}. Let L_i denote the full preimage of K_i under ν_i. A generating set C_i of L_i can be determined readily from the given generating set B_i and a basis of p^iV_{i+1}. Let $C_i = \{c_1, \ldots, c_r\}$ and consider each c_i as integral vector. Then $c_iM = w_i \in p^iW$. Hence w_i is an integral vector which is divisible by p^i. Let E_i denote the integral matrix whose rows correspond to the vectors w_i/p^i. Further, let $v = k_iM - b$. Then $v \in p^iW$ and thus v can be considered as an integral vector which is divisible by p^i.

2 Lemma: *Let $e_1, \ldots, e_l$ be a generating set for the kernel of E_i over the field with p elements and let u be a solution of the system $xE_i = v/p^i$ over the field with p elements. We consider each e_i and u as integral vectors of length r and denote their coefficients with e_{ij} and u_j, respectively.*

a) *Let $b_i = \sum_{j=1}^r e_{ij}c_j$ for $1 \leq i \leq l$. Then $B_{i+1} = \{b_1, \ldots, b_l\}$ generates K_{i+1}.*

b) *Let $c = \sum_{j=1}^r u_jc_j$. Then $k_{i+1} = k_i - c$ solves $xM = b$ over A_{i+1}.*

Proof: b) This follows directly as $k_{i+1} = k_iM - cM = (v+b) - (uC_iM) = (v+b) - (up^iE_i) \equiv v + b - v = b \bmod p^{i+1}$.

a) First note that $b_jM = e_jC_iM = e_jp^iE_i \equiv 0 \bmod p^{i+1}$. Hence every b_j is contained in K_{i+1}. Vice versa, let $k \in K_{i+1}$. Then $k \in L_i$ and thus $k = \sum_{j=1}^r a_jc_j$. Then

$$\begin{aligned} kM &= (\sum_{j=1}^r a_jc_j)M \\ &= \sum_{j=1}^r a_jw_j \end{aligned}$$

$$
\begin{aligned}
&= (a_1, \ldots, a_r)p^i E_i \\
&= 0 \bmod p^{i+1}
\end{aligned}
$$

if and only if $(a_1, \ldots, a_r)E_i \equiv 0 \bmod p$. Hence $kM = 0 \bmod p^{i+1}$ if and only if $(a_1, \ldots, a_r)$ is an element of the kernel of E_i over the field with p elements. •

4.3 Improvements in special cases

We usually assume that A is given as a direct sum of cyclic groups of increasing order. In this case, bases for p^iV_i and p^iW_i can be read off readily.

References

[1] B. Eick. Spezielle PAG Systeme im Computeralgebra System GAP. Diplomarbeit, RWTH Aachen, 1993.

[2] B. Eick and W. Nickel. *Polycyclic - computing with polycyclic groups*, 2005. A refereed GAP 4 package, see [4].

[3] R. Hartung. Solving linear equations over finitely generated abelian groups. arxiv.org e-Print archive, 2010.

[4] The GAP Group. GAP – *Groups, Algorithms and Programming, Version 4.4*. Available from http://www.gap-system.org, 2005.