Fast Algorithms for Linear Algebra Modulo N*

Arne Storjohann and Thom Mulders

Institute of Scientific Computing
ETH Zurich, Switzerland
{storjoha,mulders}@inf.ethz.ch

Abstract. Many linear algebra problems over the ring $\mathbb{Z}_N$ of integers modulo N can be solved by transforming via elementary row operations an $n \times m$ input matrix A to Howell form H. The nonzero rows of H give a canonical set of generators for the submodule of $(\mathbb{Z}_N)^m$ generated by the rows of A. In this paper we present an algorithm to recover H together with an invertible transformation matrix P which satisfies $PA = H$. The cost of the algorithm is $O(nm^{\omega-1})$ operations with integers bounded in magnitude by N. This leads directly to fast algorithms for tasks involving $\mathbb{Z}_N$–modules, including an $O(nm^{\omega-1})$ algorithm for computing the general solution over $\mathbb{Z}_N$ of the system of linear equations $xA = b$, where $b \in (\mathbb{Z}_N)^m$.

1 Introduction

The reduction of a matrix A over a field to reduced row echelon form H is a central topic in elementary linear algebra. The nonzero rows of H give a canonical basis for the row span $\mathrm{S}(A)$ of A, that is, the set of all linear combinations of rows of A. Being able to compute H quickly leads directly to fast algorithms for problems such as: testing whether two matrices have the same row span; testing whether a vector belongs to the row span of a matrix; computing the nullspace of a matrix; solving systems of linear equations.

This paper gives fast algorithms for solving similar linear algebra problems over the ring $\mathbb{Z}_N$ of integers modulo N. If N is prime, then $\mathbb{Z}_N$ is a field and $\mathrm{S}(A)$ is a vector space. Otherwise, $\mathrm{S}(A)$ forms a $\mathbb{Z}_N$-module but not a vector space. Computing over $\mathbb{Z}_N$ is a more subtle problem than computing over a field because of the existence of zero divisors. A classical result states that over a field two matrices in echelon form with the same row span will have the same number of nonzero rows — the rank. Over $\mathbb{Z}_N$ this is not the case. For example, the matrices

$$A = \begin{bmatrix} 4 & 1 & 0 \\ 0 & 0 & 5 \\ 0 & 0 & 0 \end{bmatrix} \quad \text{and} \quad B = \begin{bmatrix} 8 & 5 & 5 \\ 0 & 9 & 8 \\ 0 & 0 & 10 \end{bmatrix} \quad \text{over } \mathbb{Z}_{12}$$

* This work has been supported by grants from the Swiss Federal Office for Education and Science in conjunction with partial support by ESPRIT LTR Project no. 20244 — ALCOM–IT.

G. Bilardi et al. (Eds.): ESA'98, LNCS 1461, pp. 139–150, 1998.

have the same row span but not the same number of nonzero rows. Moreover, considered as matrices over $\mathbb{Z}$, both A and B are in Hermite normal form — a canonical form for row spans over $\mathbb{Z}$ (see [5]). A canonical echelon form for row spans in the module $(\mathbb{Z}_N)^m$ is defined by Howell in [4]. Two matrices A and B over $\mathbb{Z}_N$ will have $\mathrm{S}(A) = \mathrm{S}(B)$ if and only if their Howell forms coincide. In the previous example, the Howell form of A and B is

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

In addition to being an echelon form, the Howell form has some additional properties which make it particularly well suited to solving tasks involving $\mathbb{Z}_N$-modules.

We give an asymptotically fast algorithm for computing the Howell form of an $A \in \mathbb{Z}_N^{n \times m}$ with $n \geq m$. The algorithm requires $O(nm^{\omega-1})$ operations with integers bounded in magnitude by N to compute the following: the Howell form H of A; an invertible transformation matrix $P \in (\mathbb{Z}_N)^{n \times n}$ with $PA = H$; a kernel $Y \in (\mathbb{Z}_N)^{n \times n}$ of A. Here, ω is the exponent for matrix multiplication over rings: two $n \times n$ matrices over a ring can be multiplied in $O(n^\omega)$ ring operations. Standard matrix multiplication has $\omega = 3$ whereas the current record in [3] allows $\omega < 2.376$. In this paper we assume that $\omega > 2$.

An important feature of our algorithm is that we return a representation for the $n \times n$ transforming matrix P which allows matrix-vector products involving P to be computed in $O(nm)$ instead of $O(n^2)$ operations when P is dense. Let $A, B \in (\mathbb{Z}_N)^{n \times m}$ be given. We get $O(nm^{\omega-1})$ algorithms for: determining if $\mathrm{S}(A) = \mathrm{S}(B)$, and, if so, returning transformation matrices $P, P^{-1} \in (\mathbb{Z}_N)^{n \times n}$ such that $PA = B$ and $A = P^{-1}B$; computing a canonical spanning set for the union or intersection of the modules generated by the rows of A and B; determining inconsistency or computing a general solution to a system $xA = b$ of linear equations.

The rest of this paper is organised as follows. In Sect. 2 we define some *basic operations* from $\mathbb{Z}_N$ and bound their cost in terms of bit operations. In Sect. 3 we recall the definition and properties of the Howell form and give an iterative algorithm for its computation. In Sect. 4 we give an asymptotically fast algorithm to compute the Howell form of a matrix. Finally, in Sect. 5 we show how to apply the algorithm of the previous section to solving a variety of linear algebra problems over $\mathbb{Z}_N$.

We will frequently write matrices using a block decomposition. An unlabeled block has all entries zero and the generic label $*$ indicates a block with possibly nonzero entries. For convenience, we allow the row and/or column dimension of a matrix or block to be zero.

2 Basic Operations

In this section we define certain basic operations from $\mathbb{Z}_N$ that we will use in the rest of this paper to describe algorithms. Our main complexity results will

be given in terms of numbers of basic operations. We also bound in this section the bit complexity of these basic operations. We will represent elements from the ring $\mathbb{Z}_N$ as integers from the set $S = \{0, 1, \ldots, N-1\}$ and henceforth identify elements from $\mathbb{Z}_N$ with elements from S. For $a, b \in$ S the *basic operations* are:

b1) Basic arithmetic operations: return $c \in S$ such that $c = a + b$, $c = a - b$, $c = ab$ or $c = -a$ in $\mathbb{Z}_N$;
b2) Quo(a, b): when $b \neq 0$, return $q \in S$ such that $a - qb = r$ with $0 \leq r < b$;
b3) Gcd(a, b): return $\gcd(a, b)$;
b4) Div(a, b): when $\gcd(b, N)|a$, return a $c \in S$ such that $bc = a$ in $\mathbb{Z}_N$;
b5) Gcdex(a, b): return $g, s, t, u, v \in S$ such that $g = \gcd(a, b) = sa+tb$, $ua+vb = 0$ and $sv - tu = 1$ in $\mathbb{Z}_N$;
b6) Ann(a): return $c \in S$ such that $c = N/\gcd(a, N)$ in $\mathbb{Z}_N$. Then c generates the ideal of all elements which annihilate a in $\mathbb{Z}_N$;
b7) Stab(a, b): return $c \in S$ such that $\gcd(a + cb, N) = \gcd(a, b, N)$. Then $a + cb$ generates the ideal generated by a and b in $\mathbb{Z}_N$;
b8) Unit(a): when $a \neq 0$, return $c \in S$ such that $\gcd(c, N) = 1$ and $ca = \gcd(a, N)$ in $\mathbb{Z}_N$;

Let $M(t)$ be a bound on the number of bit operations required to multiply two $\lceil t \rceil$-bit integers. Standard arithmetic has $M(t) \in O(t^2)$ while the current record is $M(t) \in O(t \log t \log\log t)$ (see [6]).

¿From [1] we know that we can perform operations b1, b2, b3, b4, b5 and b6 in $O(M(\log N)\log\log N)$ bit operations. We now show that operations b7 and b8 can be performed in the same time. First we give algorithm Split, that for integers D and a will split off the factors in D which are common to a. In this algorithm all operations are over $\mathbb{Z}$.

Algorithm Split
Input: $D, a \in \mathbb{Z}$ such that $D > 0$ and $0 \leq a < D$
Output: $M \in \mathbb{Z}$ such that $M > 0$, $M|D$ and for all prime numbers p we have:

$$\begin{cases} p|M \Rightarrow p \not| a \\ p|D/M \Rightarrow p|a \end{cases}$$

for i **to** $\lceil \log\log D \rceil$ **do**
    $a := a^2 \bmod D$
od;
$D/\gcd(a, D)$

Theorem 1. *The previous algorithm is correct. The cost of the algorithm is* $O(M(\log D)\log\log D)$ *bit operations.*

Proof. Let p be a prime number and define $\mathrm{ord}_p(n) = \max\{i \mid p^i | n\}$ for $n \in \mathbb{Z}$. When $a = 0$ it is clear that the algorithm is correct, so assume that $a \neq 0$. Define $a_0 = a$ and $a_{i+1} = a_i^2 \bmod D$. Then for all i, $\mathrm{ord}_p(a_i) \geq \min(\mathrm{ord}_p(D), 2^i \mathrm{ord}_p(a))$. Since $\mathrm{ord}_p(D) \leq \log D$ the correctness of the algorithm now follows.

The complexity of the algorithm follows easily. □

Lemma 1. *We can perform operations b7 and b8 in $O(M(\log N)\log\log N)$ bit operations.*

Proof. To compute $\mathrm{Stab}(a, b)$ first compute $g = \gcd(a, b, N)$. Let $c \in S$ such that $c \equiv \mathrm{Split}(N/g, a/g) \pmod N$. It is easy to see that $\gcd(a/g + cb/g, N/g) = 1$. Return c.

To compute $\mathrm{Unit}(a)$ first compute $s, g \in \mathrm{S}$ such that $g = \gcd(a, N)$ and $sa \equiv g \pmod N$. Then $\gcd(s, N/g) = 1$. When $g = 1$, return s. Otherwise, compute $d = \mathrm{Stab}(s, N/g)$. Then $\gcd(s + dN/g, N) = 1$ and $(s + dN/g)a \equiv g \pmod N$. Return $c \in S$ such that $c \equiv s + dN/g \pmod N$. □

In the sequel we also need to perform the extended stabilization operation: Given $a_0, a_1, \ldots, a_n \in S$, compute $c_1, \ldots, c_n \in S$ such that $\gcd(a_0 + c_1a_1 + \cdots + c_na_n, N) = \gcd(a_0, a_1, \ldots, a_n, N)$.

Lemma 2. *The extended stabilization operation requires $O(n)$ basic operations.*

Proof. To compute $c_1, \ldots, c_n$ we can apply the stabilization operation in sequence on $(a_0, a_1), (a_0 + c_1a_1, a_2), \ldots, (a_0 + c_1a_1 + \cdots + c_{n-1}a_{n-1}, a_n)$. □

3 Spans in the Module $(\mathbb{Z}_N)^m$ and the Howell Form

In this section we recall the definition of the Howell form of a matrix which was introduced in [4]. Let $R = \mathbb{Z}_N$ and $A \in R^{n\times m}$ be given. The row span $\mathrm{S}(A)$ of A is the R–submodule of R^m generated by the rows of A, that is, the set of all R-linear combinations of rows of A. The nonzero rows of the Howell form of A give a canonical set of generators for $\mathrm{S}(A)$. For two matrices A and B we have $\mathrm{S}(A) = \mathrm{S}(B)$ if and only if A and B have the same Howell form.

If $U \in R^{n\times n}$ is invertible over R, then $\mathrm{S}(UA) = \mathrm{S}(A)$. Recall that a square matrix U is invertible precisely when U has determinant a unit from R. The matrix U corresponds to a sequence of elementary row operations: interchanging two rows; multiplying a row by a unit; adding a multiple of one row to a different row. Any matrix A over R can be transformed using only elementary row operations to a matrix H that satisfies the following conditions:

e1) Let r be the number of nonzero rows of H. Then the first r rows of H are nonzero.
e2) For $1 \le i \le r$ let $H[i, j_i]$ be the first nonzero entry in row i. Then $j_1 < j_2 < \ldots < j_r$.

A matrix H having properties e1 and e2 is said to be in row echelon form. For example, the first matrix in (1) is in row echelon form. Further elementary row operations can be applied so that H satisfies:

e3) $H[i, j_i] \mid N$ for $1 \le i \le r$.
e4) For $1 \le k < i \le r$, $0 \le H[k, j_i] < H[i, j_i]$.

A matrix H which satisfies e1, e2, e3 and e4 is said to be in reduced row echelon form. For example, in (1) the first matrix can be transformed to the second matrix which is in reduced row echelon form. Finally, a matrix is said to be in Howell form if it is in reduced row echelon form and also satisfies:

e5) (Howell property) When $a \in \mathrm{S}(H)$ has zeros as its first $(j_i - 1)$ components then $a \in \mathrm{S}(H[i \ldots m])$, where $H[i \ldots m]$ denotes the matrix comprised of the ith through mth row of H.

For example, in (1) the third matrix is the Howell form of the first two matrices.

$$\begin{bmatrix} 4 & 1 & 10 \\ 0 & 0 & 5 \\ 0 & 0 & 0 \end{bmatrix}, \quad \begin{bmatrix} 4 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{bmatrix}, \quad \begin{bmatrix} 4 & 1 & 0 \\ 0 & 3 & 0 \\ 0 & 0 & 1 \end{bmatrix} \text{ over } \mathbb{Z}_{12} \ . \tag{1}$$

Theorem 2. *For any $A \in R^{n \times m}$ there exists a unique $H \in R^{m \times m}$ that is in Howell form and such that $\mathrm{S}(H) = \mathrm{S}(A)$.*

Proof. See [4].

The matrix H from Theorem 2 is called the Howell form of A. The theorem shows that the Howell form of matrices can be used to determine whether two matrices have the same row span.

A matrix H which satisfies e1, e2 and e5 is said to be in weak Howell form. We say that H is a (weak) Howell basis for A if H has no zero rows, H is in (weak) Howell form and $\mathrm{S}(H) = \mathrm{S}(A)$.

Definition 1. *Let $A \in R^{n \times m}$. We define the kernel* $\ker(A)$ *of A as $\{r \in R^n \mid rA = 0\}$. The kernel of A is an R-submodule of R^n. $K \in R^{n \times n}$ is called a kernel for A if $\mathrm{S}(K) = \ker(A)$.*

The next two lemmas, for which we omit the proofs, give two key properties of the Howell form.

Lemma 3. *Let $H = \left[\begin{array}{c|c} H_1 & F \\ \hline & H_2 \end{array}\right]$, K_i a kernel for H_i $(i = 1, 2)$ and $K_1 F = S H_2$. Then $K = \left[\begin{array}{c|c} K_1 & -S \\ \hline & K_2 \end{array}\right]$ is a kernel for H.*

Lemma 4. *Let $H = \left[\begin{array}{c|c} H_1 & F \\ \hline & H_2 \end{array}\right]$ such that H_1 and H_2 are in weak Howell form and H_1 contains no zero rows. Let K be a kernel for H_1 and suppose that $\mathrm{S}(KF) \subseteq \mathrm{S}(H_2)$. Then H is in weak Howell form.*

Next we will give an algorithm to compute the Howell form of a matrix. The algorithm is in the same spirit as Howell's constructive proof for the existence of the form (see [4]). Following [2], we first triangularize the input matrix and then keep the work matrix in triangular form. This saves space by avoiding the

need to augment the work matrix with m extra rows as in [4]. The correctness of the algorithm follows from Lemma 4.

Algorithm Howell
Input: $A \in R^{n \times m}$
Output: The Howell form of A
if $n < m$ **then**
    Augment A with zero rows to make it square;
    $n := m$
fi;
Comment Put A in upper triangular form
for j **from** 1 **to** m **do**
    for i **from** $j+1$ **to** n **do**
        $(g, s, t, u, v) := \text{Gcdex}(A[j, j], A[i, j])$;
        $\begin{bmatrix} A[j,*] \\ A[i,*] \end{bmatrix} := \begin{bmatrix} s & t \\ u & v \end{bmatrix} \begin{bmatrix} A[j,*] \\ A[i,*] \end{bmatrix}$
    od
od;
Comment Put A in Howell form
Augment A with one zero row;
for j **from** 1 **to** m **do**
    if $A[j, j] \neq 0$ **then**
        $A[j,*] := \text{Unit}(A[j, j])A[j,*]$;
        for i **from** 1 **to** $j-1$ **do**
            $A[i,*] := A[i,*] - \text{Quo}(A[i, j], A[j, j])A[j,*]$
        od;
        $A[n+1,*] := \text{Ann}(A[j, j])A[j,*]$
    else
        $A[n+1,*] := A[j,*]$
    fi;
    for i **from** $j+1$ **to** m **do**
        $(g, s, t, u, v) := \text{Gcdex}(A[i, i], A[n+1, i])$;
        $\begin{bmatrix} A[i,*] \\ A[n+1,*] \end{bmatrix} := \begin{bmatrix} s & t \\ u & v \end{bmatrix} \begin{bmatrix} A[i,*] \\ A[n+1,*] \end{bmatrix}$
    od
od;
Move all nonzero rows to the top of A;
Output first m rows of A

A straightforward count proves the following theorem.

Theorem 3. *Algorithm Howell requires* $O(m^2 \max(n, m))$ *basic operations.*

4 Asymptotically Fast Computation of the Howell Form

All matrices will be over the ring $R = \mathbb{Z}_N$ and complexity results are given in terms of basic operations. Let $M(a, b, c)$ be a bound on the number of ba-

sic operations required to multiply an $a \times b$ by a $b \times c$ matrix over R. Using $M(n,n,n) \in O(n^\omega)$ together with a block decomposition we have

$$M(a,b,c) \in \begin{cases} O(abc^{\omega-2}) & \text{if } c = \min(a,b,c) \\ O(acb^{\omega-2}) & \text{if } b = \min(a,b,c) \\ O(bca^{\omega-2}) & \text{if } a = \min(a,b,c) \end{cases} \tag{2}$$

Definition 2. *Let $A \in R^{n\times m}$ and k be such that $0 \le k \le n$ and $n-k \ge r$ where r is the number of nonzero rows in the Howell form of A. An* index k transform *of A is a 6-tuple of matrices (Q,U,C,H,K,S) which satisfy and can be written using a conformal block decomposition as*

$$\underset{Q}{\begin{bmatrix} I & & \\ & I & \\ & * & I \end{bmatrix}} \underset{U}{\begin{bmatrix} I & & \\ & * & \\ & & I \end{bmatrix}} \underset{C}{\begin{bmatrix} I & & \\ * & I & * \\ & & I \end{bmatrix}} \underset{A}{\begin{bmatrix} \bar{A} \\ * \\ * \end{bmatrix}} = \underset{T}{\begin{bmatrix} \bar{A} \\ H \\ \end{bmatrix}} \tag{3}$$

with U invertible, $\bar{A}$ the first k rows of A, H a weak Howell basis for A, K a kernel for H and S such that $\bar{A} = SH$.

Remark 1. When (Q,U,C,H,K,S) is an index k transform of A, then a kernel for T is given by

$$W = \begin{bmatrix} I & -S & \\ & K & \\ & & I \end{bmatrix}. \tag{4}$$

A kernel for A is given by $WQUC$.

Note that if $k = 0$ then T is a weak Howell form of A. Later we will show how to transform a weak Howell form to a Howell form. First we bound the cost of computing an index k transform. Define $T(n,m,r)$ to be a bound on the number of basic operations required to compute an index k transform for an $A \in R^{n\times m}$ which has a Howell basis with r rows. Our result is the following.

Theorem 4. *$T(n,m,0) \in O(nm)$. If $r > 0$ then $T(n,m,r) \in O(nmr^{\omega-2})$.*

Before proving Theorem 4 we give some intermediate results. The algorithm supporting the theorem is recursive and the following technical result will be used to bound the cost of the merge step. Let I_s denote the $s \times s$ identity matrix.

Lemma 5. *If*

$$Q_1 = \begin{bmatrix} I_k & & & \\ & I_{r_1} & & \\ & \bar{q}_1 & I_{r_2} & \\ & q_1 & & I \end{bmatrix}, \quad U_1 = \begin{bmatrix} I_k & & & \\ & u_1 & & \\ & & I_{r_2} & \\ & & & I \end{bmatrix}, \quad C_1 = \begin{bmatrix} I_k & & & \\ c_1 & I_{r_1} & \bar{d}_1 & d_1 \\ & & I_{r_2} & \\ & & & I \end{bmatrix},$$

$$Q_2 = \left[\begin{array}{c|c|c|c} I_k & & & \\ \hline & I_{r_1} & & \\ \hline & & I_{r_2} & \\ \hline & & q_2 & I \end{array}\right],\ U_2 = \left[\begin{array}{c|c|c|c} I_k & & & \\ \hline & I_{r_1} & & \\ \hline & & u_2 & \\ \hline & & & I \end{array}\right],\ \bar{C}_2 = \left[\begin{array}{c|c|c|c} I_k & & & \\ \hline & I_{r_1} & & \\ \hline c_2 & \bar{c}_2 & I_{r_2} & d_2 \\ \hline & & & I \end{array}\right],$$

are all in $R^{n\times n}$ and the block decomposition is conformal, then

$$Q_2U_2\bar{C}_2Q_1U_1C_1 = \overbrace{\left[\begin{array}{c|c|c|c} I_k & & & \\ \hline & I_{r_1} & & \\ \hline & & I_{r_2} & \\ \hline & q_1 & q_2 & I \end{array}\right]}^{Q} \overbrace{\left[\begin{array}{c|c|c|c} I_k & & & \\ \hline & u_1 & u_{12} & \\ \hline & u_{21} & u_{22} & \\ \hline & & & I \end{array}\right]}^{U} \overbrace{\left[\begin{array}{c|c|c|c} I_k & & & \\ \hline c_{11} & I_{r_1} & & c_{12} \\ \hline c_2 & & I_{r_2} & d_2 \\ \hline & & & I \end{array}\right]}^{C} \quad (5)$$

where

$$\begin{aligned} c_{11} &= c_1 - \bar{d}_1c_2 \\ c_{12} &= d_1 - \bar{d}_1d_2 \\ u_{21} &= u_2(\bar{q}_1 + d_2q_1 + \bar{c}_2)u_1 \\ u_{12} &= u_1\bar{d}_1 \\ u_{22} &= u_2 + u_2(\bar{q}_1 + d_2q_1 + \bar{c}_2)u_1\bar{d}_1 \\ &= u_2 + u_{21}\bar{d}_1 \end{aligned}$$

Moreover, the computation of Q, U and C requires at most $O(n(r_1 + r_2)^{\omega-1})$ basic operations.

Proof. The first part of the lemma follows from a straightforward computation. The cost of computing Q, U and C is $O(M(a, b, c))$ where at least two of the dimensions a, b and c equal r_1 or r_2 and the third dimension is bounded by n. The second part of the lemma now follows from (2). □

Lemma 6. *There exists an absolute constant c, such that if $m = m_1 + m_2$, then*

$$T(n, m, r) \le T(n, m_1, r_1) + T(n, m_2, r_2) + cnm_2r^{\omega-2}$$

for some $r_1, r_2 \ge 0$ with $r = r_1 + r_2$.

Proof. Let $A \in R^{n\times m}$ and k be such that $0 \le k \le n$ and $n - k \ge r$ where r is the number of rows in a Howell basis of A. We will compute an index k transform (Q, U, C, H, K, S) for A by merging transforms for matrices A_1 and A_2 of column dimension m_1 and m_2 respectively. The result will follow if we bound by $O(nm_2r^{\omega-2})$ basic operations the cost of constructing A_1 and A_2 and merging their transforms. Choose A_1 to be the first m_1 columns of A. Compute an index k transform $(Q_1, U_1, C_1, H_1, K_1, S_1)$ for A_1 at a cost bounded by $T(n, m_1, r_1)$ basic operations where r_1 is the row dimension of H_1. We now have

$$W_1Q_1U_1C_1 \overbrace{\left[\begin{array}{c|c} \bar{A}_1 & E \\ \hline * & * \\ \hline * & * \end{array}\right]}^{A} = \overbrace{\left[\begin{array}{c|c|c} I & -S_1 & \\ \hline & K_1 & \\ \hline & & I \end{array}\right]}^{W_1} \left[\begin{array}{c|c} \bar{A}_1 & E \\ \hline H_1 & F \\ \hline & * \end{array}\right] = \left[\begin{array}{c|c} & E - S_1F \\ \hline & K_1F \\ \hline & * \end{array}\right] \quad (6)$$

where E and F are new labels. Note that the submatrix of A comprised of blocks $\bar{A}_1$ and E is $\bar{A}$ of (3). Choose A_2 as the last m_2 columns of the matrix on the right hand side of (6). Note that the last $n-k-r_1$ rows of A_2 is the unmodified trailing $(n-k-r_1) \times m_2$ block of A. Because of the special structure of the matrices involved in the multiplications, we can recover A_2 and F in $O(nm_2r_1^{\omega-2})$ basic operations. In particular, note that S_1, K_1 and every subblock of Q_1, U_1 and C_1 which is neither the identity nor the zero block has column and/or row dimension equal to r_1. Compute an index $k+r_1$ transform $(Q_2, U_2, C_2, H_2, K_2, S_2)$ for A_2 at a cost bounded by $T(n, m_2, r_2)$ basic operations where r_2 is the row dimension of H_2. Now we show how to recover an index k transform (Q, U, C, H, K, S) for A. Define

$$H = \left[\begin{array}{c|c} H_1 & F \\ \hline & H_2 \end{array}\right], \quad K = \left[\begin{array}{c|c} K_1 & -S_{22} \\ \hline & K_2 \end{array}\right] \text{ and } S = \left[\begin{array}{c|c} S_1 & S_{21} \end{array}\right] \text{ where } S_2 = \left[\begin{array}{c} S_{21} \\ \hline S_{22} \end{array}\right].$$

That K is a kernel for H follows from Lemma 3. That H is in weak Howell form follows from Lemma 4. That $SH = \bar{A}$ follows from direct computation. We now show how to recover Q, U and C such that $QUCA = T$ where T is as in (3). At a cost of $O(nr^{\omega-1})$ basic operations compute

$$\bar{C}_2 = \left[\begin{array}{c|c|c} I & & \\ \hline \bar{a} & I & b \\ \hline & & I \end{array}\right] \text{ where } C_2 = \left[\begin{array}{c|c|c} I & & \\ \hline a & I & b \\ \hline & & I \end{array}\right] \text{ and } \bar{a} = a \left[\begin{array}{c|c} I & -S_2 \\ \hline & K_1 \end{array}\right].$$

A straightforward multiplication verifies that $Q_2U_2\bar{C}_2Q_1U_1C_1A = T$. The matrices Q, U and C of the index k transform for A can now be recovered in $O(nr^{\omega-1})$ basic operations using Lemma 5. That H is a weak Howell basis for A follows from the fact that H is a weak Howell basis for T. This also shows that $r_1 + r_2 = r$. □

Lemma 7. $T(n, 1, r) \in O(n)$.

Proof. Let $A \in R^{n\times 1}$, $0 \le k \le n$ and $n - k \ge r$ where r is 0 if A is the zero matrix and 1 otherwise. If $r = 0$ then H is 0×1, K is 0×0, S is $k \times 0$ and we can choose $Q = U = C = I_n$. Assume henceforth that $r = 1$. Using the extended stabilization operation from Lemma 2, recover a C of the correct shape such that $\gcd((CA)_{k+1,1}, N)$ is the gcd of all entries in A and N. Set $H = [\,a\,]$, where $a = (CA)_{k+1,1}$. Set $K = [\,\mathrm{Ann}(a)\,]$. Set $Q = I_n$ except with $Q_{i,k+1} = -\mathrm{Div}(A[i,1], a)$ for $k+2 \le i \le n$. Set S to be the $k \times 1$ matrix with $S[i,1] = \mathrm{Div}(A[i,1], a)$ for $1 \le i \le k$. Set $U = I_n$. □

Lemma 8. *For $\alpha, x, y \in \mathbb{R}$ such that $0 < \alpha < 1$ and $x, y > 0$ we have $x^\alpha + y^\alpha \le 2^{1-\alpha}(x+y)^\alpha$.*

Proof. The function $z \mapsto \frac{1+z^\alpha}{(1+z)^\alpha}$ has for $z > 0$ an absolute maximum at $z = 1$ with value $2^{1-\alpha}$. By substituting y/x for z into this function, the lemma follows easily. □

We now return to the proof of Theorem 4.

Proof. (of Theorem 4) Let $A \in R^{n \times m}$. By augmenting A with at most $m-1$ zero columns we may assume without loss of generality that m is a power of two. This shows that it will be sufficient to prove the theorem for the special case when m is a power of 2. Let c be the absolute constant of Lemma 6. By Lemma 7 we have that $T(n, 1, r) \in O(n)$. Choose an absolute constant e such that $T(n, 1, r) \leq en$. From Lemma 6 it now follows easily that $T(n, m, 0) \leq enm$. Now choose an absolute constant d such that $c/2 \leq d(1 - 2^{2-\omega})$ and $d/2 \geq e/2 + c/2$. We claim that for $r > 0$

$$T(n, m, r) \leq dnmr^{\omega-2} \ . \tag{7}$$

We will prove (7) by induction on $\log_2 m$. Note that (7) is true for $m = 1$. Assume that (7) is true for some power of two m. Then

$$T(n, 2m, r) \leq T(n, m, r_1) + T(n, m, r_2) + cnmr^{\omega-2}$$

for some $r_1, r_2 \geq 0$ with $r_1 + r_2 = r$. When r_1 and r_2 are both nonzero we get the following:

$$\begin{aligned} T(n, 2m, r) &\leq dnm(r_1^{\omega-2} + r_2^{\omega-2}) + cnmr^{\omega-2} \\ &\leq 2nmr^{\omega-2}(2^{2-\omega}d + c/2) \quad \text{(Lemma 8)} \\ &\leq dn(2m)r^{\omega-2}. \end{aligned}$$

When $r_1 = 0$ we get the following:

$$\begin{aligned} T(n, 2m, r) &\leq enm + dnmr^{\omega-2} + cnmr^{\omega-2} \\ &\leq 2nmr^{\omega-2}(e/2 + d/2 + c/2) \\ &\leq dn(2m)r^{\omega-2}. \end{aligned}$$

The case $r_2 = 0$ is similar. □

Now we will show how we can transform a matrix in weak Howell form to Howell form.

Corollary 1. *Let (Q, U, C, H, K) be an index 0 transform for an $A \in R^{n \times m}$. An index 0 transform (Q', U', C, H', K') for A which has H' in Howell form can be computed in $O(nr^{\omega-1})$ basic operations where r is the number of rows in H.*

Proof. If $r = 0$ then nothing needs to be done. Assume $r > 0$. Recover an invertible and diagonal $D \in R^{r \times r}$ such that DH satisfies property e3. This requires r basic operations of type b7. Let T be the submatrix of DH comprised of columns $[j_1, j_2, \ldots, j_r]$ where j_i is as in property e2 of the Howell form. Recover a unit upper triangular matrices R such that RDH satisfies property e4. Next recover R^{-1}. Both R and R^{-1} can be recovered in $O(r^\omega)$ basic operations using the algorithm in [7, Theorem 3]. Set $Q' = \operatorname{diag}(RD, I_{n-r})\, Q \operatorname{diag}(D^{-1}R^{-1}, I_{n-r})$, $U' = \operatorname{diag}(RD, I_{n-r})\, U$ and $K' = KD^{-1}R^{-1}$. Correctness follows easily and the cost follows from (2). □

Since the Howell form of an invertible matrix is the identity matrix we can obtain the inverse of an invertible matrix U as the transforming matrix from U to its Howell form. We get the following corollary:

Corollary 2. *The inverse of an invertible matrix in $R^{n\times n}$ can be computed in $O(n^\omega)$ basic operations.*

5 Some Applications

Let $R = \mathbb{Z}_N$ and $A_i \in R^{n_i \times m}$ be given for $i = 1, 2$. By augmenting with 0-rows, we may assume without loss of generality that $n = n_1 \geq n_2 \geq m$. Let A denote a copy of A_1 and let r be the number of rows in a Howell basis for A. Note that $r \leq m$ and in the worst case $r = m$. For brevity we give space and time bounds in terms of n and m only.

We propose solutions to some basic tasks involving modules. A worst case running time bound of $O(nm^{\omega-1})$ basic operations for all the tasks follows from (2), Theorem 4 and Corollaries 1 and 2. For some tasks we state in addition the cost under the assumption that some quantities have been pre-computed. The correctness of the proposed solutions are left as an exercise.

Task 1: Kernel computation [Find a kernel $Y \in R^{n\times n}$ for A.]
Compute an index 0 transform (Q, U, C, H, K) for A. By Remark 1 we can choose

$$Y = \overset{WQU}{\left[\begin{array}{c|c} * & \\ \hline * & I_{n-r} \end{array}\right]} \overset{C}{\left[\begin{array}{c|c} I_r & * \\ \hline & I_{n-r} \end{array}\right]}. \tag{8}$$

Return the decomposition for Y as the product of the two matrices shown in (8). This has two advantages. First, both WQU and C will have only $O(nr)$ nonzero entries; their product may have $O(n^2)$ nonzero entries. Second, premultiplying a vector by C and then by WQU costs only $O(nr)$ basic operations.

Task 2: Equality of spans. [Determine if $\mathrm{S}(A_1) = \mathrm{S}(A_2)$.]
By augmenting A_2 with 0-rows, we may assume without loss of generality that $n_1 = n_2$. Compute an index 0 transform $(Q_i, U_i, C_i, H_i, K_i)$ for A_i which has H_i in Howell form $(i = 1, 2)$. Then $\mathrm{S}(A_1) = \mathrm{S}(A_2)$ if and only if $H_1 = H_2$. A transformation matrix P such that $A_1 = PA_2$ and $P^{-1}A_1 = A_2$ is given by $P = (Q_1U_1C_1)^{-1}Q_2U_2C_2$. A straightforward multiplication will verify that

$$P = \overset{(2I-C_1)}{\left[\begin{array}{c|c} I_r & * \\ \hline & I_{n-r} \end{array}\right]} \overset{((Q_2-Q_1)U_1+I)}{\left[\begin{array}{c|c} I_r & \\ \hline * & I_{n-r} \end{array}\right]} \overset{U_1^{-1}U_2}{\left[\begin{array}{c|c} * & \\ \hline & I_{n-r} \end{array}\right]} \overset{C_2}{\left[\begin{array}{c|c} I_r & * \\ \hline & I_{n-r} \end{array}\right]}. \tag{9}$$

As was the case for the kernel, return the decomposition for P as the product of the four matrices shown in (9). In particular, if $X_1 \in R^{k\times n}$, then an X_2 such that $X_1A_1 = X_2A_2$ can be recovered as $X_2 \leftarrow X_1P$ in $O(nmk^{\omega-2})$ basic operations for $k \leq m$ and $O(nkm^{\omega-2})$ basic operations for $k > m$.

Task 3: Union of modules. [Find the Howell basis for $\mathrm{S}(A_1) \cup \mathrm{S}(A_2)$.]
Return the Howell basis for $[\,A_1^T \,|A_2^T\,]^T$. Here, X^T denotes the transpose of X.

Task 4: Intersection of modules. [Find the Howell basis for $\mathrm{S}(A_1) \cap \mathrm{S}(A_2)$.]
Compute a kernel Y for $[\,A_1^T \,|\, A_2^T\,]^T$ as in (8). Return the Howell basis for $Y[\,A_1^T \,|\, 0\,]^T$.

Task 5: Testing containment. [Determine whether or not $b \in \mathrm{S}(A)$.]
Compute an index 0 transform (Q, U, C, H, K) for A. Recover a row vector y such that

$$\left[\begin{array}{c|c} 1 & y \\ \hline & I \end{array}\right] \left[\begin{array}{c|c} 1 & b \\ \hline & H \end{array}\right] = \left[\begin{array}{c|c} 1 & b' \\ \hline & H \end{array}\right]$$

with the right hand side in Howell form. Then $b \in \mathrm{S}(A)$ if and only if $b' = 0$. If $b \in \mathrm{S}(A)$, then $xA = b$ where $x \leftarrow [\,y \,|\, 0\,]UC$. Assuming that the index 0 transform for A is precomputed, testing containment and recovering x requires $O(m^2)$ and $O(nm)$ basic operations respectively.

Task 6: Solving systems of linear equations. [Find a general solution to $xA = b$.]
Determine containment of b in $\mathrm{S}(A)$ using Task 5. If $b \notin \mathrm{S}(A)$ then return "no solution exists". If $b \in \mathrm{S}(A)$, find a y such that $yA = b$ using Task 5. Return (y, Y) where Y is a kernel for A as in (8). Every solution x to $xA = b$ can be expressed as y plus some linear combination of the rows of Y.

References

1. A. V. Aho, J. E. Hopcroft, and J. D. Ullman. *The Design and Analysis of Computer Algorithms.* Addison-Wesley, 1974.
2. E. Bach. Linear algebra modulo N. Unpublished manuscript., December 1992.
3. D. Coppersmith and S. Winograd. Matrix multiplication via arithmetic progressions. *Journal of Symbolic Computation*, 9:251–280, 1990.
4. J. A. Howell. Spans in the module $(\mathbb{Z}_m)^s$. *Linear and Multilinear Algebra*, 19:67—77, 1986.
5. M. Newman. *Integral Matrices.* Academic Press, 1972.
6. A. Schönhage and V. Strassen. Schnelle Multiplikation grosser Zahlen. *Computing*, 7:281–292, 1971.
7. A. Storjohann and G. Labahn. Asymptotically fast computation of Hermite normal forms of integer matrices. In Y. N. Lakshman, editor, *Proc. Int'l. Symp. on Symbolic and Algebraic Computation: ISSAC '96*, pages 259–266. ACM Press, 1996.