Soving $AX = B$ using the Hermite normal form

By Keith Matthews

October 26, 2011

[1]

1 Introduction

We consider the problem of solving $AX = B$, where A, X, B are integer matrices of size $m \times n, n \times 1, m \times 1$, respectively, with A nonzero.

One classical method, described in M. Newman's book ([2, page 36]) uses the Smith Normal Form of A.

Another approach, based on applying the modified LLL algorithm (MLLL) to $\left[\frac{A^t}{B^t}\right]$, is given in M. Pohst's book [4, pages 23–24].

There are other methods designed to avoid coefficient explosion, such as that of Chou–Collins [1].

In this note we present another method, presumably well–known, which finds the Hermite normal form of $G = \left[\begin{array}{c|c} A^t & 0 \\ \hline B^t & 1 \end{array}\right]$.

Let $H = \mathbf{HNF}(A^t) = \left[\frac{C}{0}\right]$, where C consists of nonzero rows.

If a solution of $AX = B$ exists, then

$$\mathbf{HNF}(G) = \left[\begin{array}{c|c} C & 0 \\ \hline 0 & 1 \\ \hline 0 & 0 \end{array}\right].$$

[1] Previous version October 8, 1998

Conversely if

$$\mathbf{HNF}(G) = \left[\begin{array}{c|c} D & 0 \\ \hline 0 & 1 \\ \hline 0 & 0 \end{array}\right],$$

and P is a unimodular matrix such that $PG = \mathbf{HNF}(G)$, then P has the form

$$P = \left[\begin{array}{c|c} Q_1 & 0 \\ \hline -Y^t & 1 \\ \hline R & 0 \end{array}\right]$$

(if the nullspace $N(A)$ (which consists of the integer vectors X such that $AX = 0$), is trivial, R will be absent) and from the equation $PG = \mathbf{HNF}(G)$, we deduce that $AY = B$.

If the LLL–based Hermite normal form algorithm of Havas, Majewski, Matthews [3] is used and the nullspace $N(A) = \{X \in \mathbb{Z}^n | AX = 0\}$ is non–trivial, the corresponding unimodular tranformation matrix P, tends to have small entries. In particular, Y is likely to have small entries and the columns of R^t are likely to form a basis for $N(A)$ having small entries.

For matrices A of large dimension, it would be advisable to initially employ a faster Hermite normal form algorithm such as that of Kannan–Bachem [5, pages 349–357], to decide if the system is soluble and also if $\operatorname{rank} A = n$, as the Gram–Schmidt basis component of the LLL–based algorithm is time–consuming to compute for large matrices.

2 An example of Pohst

Here

$$G = \left[\begin{array}{c|c} A^t & 0 \\ \hline B^t & 1 \end{array}\right] = \left[\begin{array}{rrrrrrr} -8 & 1 & -7 & -9 & -2 & -1 & 0 \\ 5 & -2 & 3 & -3 & 1 & 1 & 0 \\ 7 & 0 & 6 & 4 & -5 & -8 & 0 \\ -7 & -10 & 5 & 9 & -4 & 4 & 0 \\ 3 & -4 & 1 & -2 & 3 & -8 & 0 \\ -7 & 3 & 2 & 6 & 7 & -1 & 0 \\ 4 & 8 & 5 & 1 & -8 & -9 & 0 \\ 9 & 5 & 0 & -10 & -8 & 8 & 0 \\ -6 & 2 & -6 & -9 & -5 & 6 & 0 \\ 3 & -1 & -1 & -7 & 9 & 8 & 1 \end{array}\right].$$

Applying the LLL–based Hermite normal form algorithm to G, with pa-

rameter $\alpha = 1$, gives the unimodular tranformation matrix

$$P = \begin{bmatrix} -3 & -8 & -4 & 2 & 8 & 2 & 1 & 5 & 0 & 0 \\ -15 & -31 & -47 & 13 & 37 & -4 & 22 & 18 & -11 & 0 \\ 24 & -17 & 7 & 6 & 2 & 11 & -10 & 23 & -29 & 0 \\ 45 & -98 & -90 & 44 & 69 & 11 & 26 & 94 & -118 & 0 \\ -65 & 94 & 73 & -43 & -55 & -16 & -15 & -99 & 133 & 0 \\ -43 & 10 & -2 & -8 & 11 & -6 & 8 & -25 & 53 & 0 \\ -38 & -25 & 6 & -1 & 34 & 12 & -3 & 3 & 44 & 1 \\ 102 & -214 & -137 & 86 & 141 & 47 & 23 & 207 & -233 & 0 \\ -86 & -51 & -18 & 2 & 79 & 15 & 10 & 5 & 85 & 0 \\ 16 & -20 & 54 & -3 & 1 & 30 & -36 & 22 & 3 & 0 \end{bmatrix}$$

and

$$PG = \mathbf{HNF}(G) = \left[\frac{I_7}{0}\right].$$

From row 7 of P, we read off the short solution

$$X = [38, 25, -6, 1, -34, -12, 3, -3, -44]^t$$

of $AX = B$. This is in fact the shortest solution. Also the last three rows of P constitute a short basis for $N(A)$:

$$[102, -214, -137, 86, 141, 47, 23, 207, -233]^t,$$
$$[86, 51, 18, -2, -79, -15, -10, -5, -85]^t,$$
$$[16, -20, 54, -3, 1, 30, -36, 22, 3]^t.$$

3 Remarks

1. The algorithm is implemented, along with that of Kannan–Bachem, in the author's number theory calculator program CALC at

   `http://www.numbertheory.org/calc/krm_calc.html`.

   There is a slower BCMATH version at

   `http://www.numbertheory.org/php/axb.html`.

2. This note arose in answer to a query of J.S. Silverman in 1998.

References

[1] Tsu–Wu J. Chou, G.E. Collins,, *Algorithms for the solution of linear diophantine equations*, Siam J. Computing (1982) 687–708.

[2] M. Newman, *Integral Matrices*, Academic Press, 1972

[3] George Havas, B.S. Majewski, K.R. Matthews, *Extended gcd and Hermite normal form algorithms via lattice reduction*, Experimental Mathematics 7 No. 2 (1998) 125–136.

[4] M. Pohst, *Computational Algebraic Number Theory*, DMV Seminar Band 21, Birkhäuser, 1994.

[5] C.C. Sims, *Computing with finitely presented groups*, Cambridge University Press, 1994.

KEITH MATTHEWS
DEPARTMENT OF MATHEMATICS
UNIVERSITY OF QUEENSLAND
BRISBANE
AUSTRALIA 4072
E–mail: keithmatt@gmail.com