Exact Orthogonalization of Integer Matrices

Thomas R Bewley, *UC San Diego* and *USAF Academy*

January 25, 2025

Abstract

Define an "integer matrix" as a matrix with integer elements. This note develops an algorithm, dubbed Integer Gram-Schmidt (IGS), that for any small integer matrix $A_{m,n}$ of rank r develops the exact matrix decomposition $A = QD^{-1}R$, where $\{Q, D, R\}$ themselves are also integer matrices, and where the r columns of Q orthogonally span the column space of A, D is diagonal, and R is in row echelon form; IGS also generates an integer matrix L, the $m-r$ columns of which exactly span the left nullspace of A. Applying IGS to A^T, of course, generates corresponding exact integer orthogonal bases for the row space and nullspace of A. IGS is a natural modification of the Modified Gram-Schmidt (MGS) procedure, with at each step each subsequent column of Q scaled by the inverse of its greatest common denominator (GCD), rather than its 2-norm, in order to minimize the magnitude of the integers in each column of Q.

1 Main result

Given any real matrix $A_{m,n}$ of rank r, the Modified Gram Schmidt (MGS) algorithm (§5.2.8 of [1]) computes a real matrix decomposition $A = QR$, where $\{Q, R\}$ are real matrices, the r columns of $Q_{m,r}$ orthogonally span the column space of A, $Q^TQ = I$, and $R_{r,m}$ is in row echelon form.

Given any *integer* matrix $A_{m,n}$ of rank r, we can instead write an exact *integer* matrix decomposition $A = QD^{-1}R$, where $\{Q, D, R\}$ are integer matrices, the r columns of $Q_{m,r}$ orthogonally span the column space of A, $D_{r,r} = Q^TQ$ is diagonal, and $R_{r,m}$ is in row echelon form; we can also form an integer matrix $L_{m,m-r}$, the $m-r$ columns of which orthogonally[1] span the left nullspace of A. As just one example with $m = 5$, $n = 3$, and $r = 3$, we may write such a decomposition as

$$A = \begin{pmatrix} -3 & 3 & 1 \\ 4 & 1 & -3 \\ 4 & -2 & 1 \\ -2 & -2 & 2 \\ -2 & 2 & -3 \end{pmatrix}, \quad Q = \begin{pmatrix} -3 & 108 & 654 \\ 4 & 101 & -202 \\ 4 & -46 & 305 \\ -2 & -124 & -100 \\ -2 & 72 & -675 \end{pmatrix}, \quad L = \begin{pmatrix} 234 & 0 \\ 218 & 10 \\ 275 & -11 \\ 410 & 7 \\ 225 & -9 \end{pmatrix},$$

$$D = \begin{pmatrix} 49 & 0 & 0 \\ 0 & 44541 & 0 \\ 0 & 0 & 1027170 \end{pmatrix}, \quad R = \begin{pmatrix} 49 & -13 & -9 \\ 0 & 909 & -705 \\ 0 & 0 & 3390 \end{pmatrix}.$$

When such a decomposition is computed correctly, Q^TQ and L^TL are diagonal, Q^TL and A^TL are zero, and of course $A = QD^{-1}R$. The Integer Gram Schmidt (IGS) scheme developed in this paper, which is provided in executable Matlab syntax in Algorithm 1, generates such decompositions exactly using integer (int64) arithmetic only. As seen in Algorithm 1, IGS consists of eight main steps:

1.) initialize $Q = A$,
2.) orthogonalize the columns of Q,
3.) strip out the resulting of zero columns of Q,
4.) initialize $L = I_{m\times m}$,
5.) orthogonalize the columns of L against the columns of Q,
6.) orthogonalize the columns of L,
7.) strip out the resulting of zero columns of L, and
8.) generate $R = Q^TA$ and $D = Q^TQ$.

[1] For the purpose of this paper, "orthogonality" of a set of integer vectors implies solely that the dot product between any two different vectors in the set is zero; said vectors are not (as is often customary) also assumed to be of unit length.

Some observations related to the above steps:

(2a) Rather than normalizing the columns of Q, which would convert them to real vectors, we instead divide each column of Q by its GCD, thus minimizing the magnitudes of its integer entries.

(2b) The i'th column of Q, denoted here $\mathbf{q}^i$, is not normalized, so $f_i = \mathbf{q}^i \cdot \mathbf{q}^i \neq 1$. Thus, rather than updating $\mathbf{q}^j \leftarrow \mathbf{q}^j - (\mathbf{q}^i \cdot \mathbf{q}^j)\,\mathbf{q}^i$ as in MGS, we effectively scale the RHS of this update by f_i, instead performing the update $\mathbf{q}^j \leftarrow f_i\,\mathbf{q}^j - (\mathbf{q}^i \cdot \mathbf{q}^j)\,\mathbf{q}^i$, which keeps the entries of $\mathbf{q}^j$ as integers after the update, while projecting the vector $\mathbf{q}^j$ in the same direction as does the standard MGS update.

(8) Rather than computing R and D while orthogonalizing the columns of Q, it is simplest to just compute them after the fact, leveraging the identity $Q^T(A = QD^{-1}R) \;\Rightarrow (Q^TA) = (Q^TQ)D^{-1}R$, where by orthogonality (Q^TQ) is diagonal, and is thus set equal to D.

The rest of the steps of the Integer Gram-Schmidt algorithm are self explanatory.

Algorithm 1: The Integer Gram-Schmidt (IGS) algorithm, in executable Matlab syntax.

```
function [Q,D,R,r,L] = IGS(A)
% Copyright 2025 by Thomas Bewley, published under BSD 3-Clause License.
[m,n]=size(A); Q=int64(A); % Convert to integers (all math below done on integers!)
for i=1:n                  % orthogonalize the columns of Q
  Q(:,i)=Q(:,i)/gcd_vec(Q(:,i)); f(i)=dot_product(Q(:,i),Q(:,i));
  if f(i)>0, for j=i+1:n;
    Q(:,j)=f(i)*Q(:,j)-Q(:,i)*dot_product(Q(:,i),Q(:,j));
  end, end
end
index=[1:n]; for i=1:n     % strip out the zero columns of Q
  if f(i)==0, l=length(index);
    for j=1:l, if index(j)==i
      index=index([1:j-1,j+1:l]); break
    end, end
  end
end, Q=Q(:,index); f=f(index); r=length(index);
L=int64(eye(m)); for j=1:r % orthogonalize columns of L against Q
  for i=1:m
    L(:,i)=f(j)*L(:,i)-Q(:,j)*dot_product(Q(:,j),L(:,i));
    L(:,i)=L(:,i)/gcd_vec(L(:,i));
  end
end
for j=1:m                  % orthogonalize the columns of L
  h(j)=dot_product(L(:,j),L(:,j));
  for i=j+1:m
    L(:,i)=h(j)*L(:,i)-L(:,j)*dot_product(L(:,j),L(:,i));
    L(:,i)=L(:,i)/gcd_vec(L(:,i));
  end
end
index=[1:m]; for i=1:m     % strip out the zero columns of L
  if dot_product(L(:,i),L(:,i))==0, l=length(index);
    for j=1:l, if index(j)==i
      index=index([1:j-1,j+1:l]); break
    end, end
  end
end, L=L(:,index);
Q=double(Q); L=double(L); % convert back to double (Matlab default)
R=Q'*A; D=Q'*Q;           % generate R and D
end % function IGS
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
function [p]=dot_product(u,v)
p=0; for i=1:length(u), p=p+u(i)*v(i); end
end
function [g]=gcd_vec(u)
g=gcd(u(1),u(2)); for i=3:length(u), g=gcd(g,u(i)); end
end
```

2 Discussion

An exact integer $QD^{-1}R$ decomposition may be rewritten a few different ways:

- as $A = Q_1 R$ where $Q_1 = QD^{-1}$ is Q with its columns scaled by the diagonal elements of D^{-1};
- as $A = Q\,R_1$ where $R_1 = D^{-1}R$ is R with its rows scaled by the diagonal elements of D^{-1};
- as $A = Q_2 R_2$ where $Q_2 = QD^{-1/2}$ and $R_2 = D^{-1/2}R$.

The rational expressions for Q_1 and R_1 above are, of course, also exact, as are the expressions for Q_2 and R_2, before the (real) divisions and square roots are calculated. Note that the third form above, with $Q_2^T Q_2 = I$, is equivalent a standard (real) QR decomposition of A.

Note also that an integer $QD^{-1}R$ decomposition of an integer matrix A can sometimes be formed by taking a standard (real) QR decomposition of A, then attempting to express the (real) elements of Q as rational expressions. This approach, however, is highly susceptible to error due to the finite-precision arithmetic involved. IGS, on the other hand, is based on integer arithmetic only, and is thus not susceptible such errors.

The magnitudes of the elements of an integer $QD^{-1}R$ decomposition grow rapidly as m and n and the magnitude of the integer elements of A grow. Variable precision integer arithmetic can easily be implemented to overcome this, starting with 64-bit integers and increasing to 128-bit integers, etc, as the need arises in the IGS computations.

Integer and rational matrices of the type discussed above play a valuable role in dynamical systems theory and crystallography [2, 3], and are also valuable in the pedagogical setting when introducing orthogonal bases of the four fundamental subspaces [4] of a matrix A leveraging simple examples with integer elements. There has been significant previous work in the generation of rational orthogonal matrices (see, e.g., [5], and the references contained therein). To the best of our knowledge, the integer $QD^{-1}R$ decomposition, and the natural modification of the Modified Gram-Schmidt (MGS) procedure identified herein which exactly generates it, had not previous been discovered, and may well be useful in such settings.

References

[1] Golub, GH, & van Loan, CF (1996) *Matrix Computations*, Third Edition. Johns Hopkins.

[2] Kaczorek, T (2007) *Polynomial and Rational Matrices: Applications in Dynamical Systems Theory.* Springer, London.

[3] Rodríguez-Andrade, MA, Aragón-González, G, Aragón, JL, Gómez-Rodríguez, A (2011) Coincidence lattices in the hyperbolic plane *Acta Crystallogr. A* **67**, 35-44.

[4] Strang, G (2006) *Linear Algebra and Its Applications*, 4th edition. Cengage.

[5] Rodríguez-Andrade, MA, Aragón-González, G, Aragón, JL (2016) The generation of all rational orthogonal matrices in Rp,q, Linear Algebra and its Applications **496**, 101-113.