

Double Entry Bookkeeping and Error Correction

by

Anil Arya

John Fellingham

Doug Schroeder

Richard Young

Ohio State University

June 1996

The authors wish to thank Jonathan Glover and participants at the Case-Western Reserve University accounting workshop, especially Ted Christensen, Julia Grant, and Gary Previts.

Double Entry Bookkeeping and Error Correction

1. Introduction

The purpose of accounting is often thought of as providing information regarding the value of the firm and its assets -- this view has led us on excursions in capital markets and agency. But in general, we can think of utilizing for valuation purposes any piece of information (accounting or non-accounting).

This paper studies accounting more narrowly. What is unique about the structure of accounting information? We begin with the premise that the distinguishing feature of accounting information is that it is produced via the double entry bookkeeping system.

The beauty of double entry bookkeeping is in some circles well-appreciated. One of the favorite quotes of many accountants is reproduced in Ijiri (1982).

"What advantage does he derive from the system of bookkeeping by double entry! It is among the finest inventions of the human mind." (Johann von Goethe, 1824)

One of the advantages of double entry bookkeeping is its ability to detect errors. If the balance sheet doesn't balance, then at least one error has occurred. Relying only on the balancing property of the balance sheet, though, will not accomplish any of the following.

1. Identify the amount of the error.
2. Identify in which account a single error occurred.
3. Detect the existence of two errors which counterbalance each other.

There are, however, error detecting and correcting capabilities embedded in the structure of double entry bookkeeping which can accomplish all of the preceding goals. The purpose of this paper is to examine the properties of the double entry system using some fundamental results from linear algebra and coding theory.

Surely there are other important features of a double entry system besides its error detecting abilities. For instance, often it is argued that accounting efficiently accumulates and communicates economic information such as income or cost. But this paper purposely offers no economic implications or applications. Rather, the exercise is treated as the examination of intellectual questions surrounding the double entry system of accounting. In this paper we do not apply economics; it's just accounting.

Finally, it has been noted that accounting research has made significant progress on the use of information in various economic settings. But the information remains fairly general and the distinctive characteristics of accounting information are hard to see (Demski 1990). The analysis herein imposes the structure unique to the double entry accounting system and explores its logical consistencies.

A summary of what is accomplished herein follows. We formally establish the link between linear algebra and the double entry bookkeeping system: the system transforms a vector of numerical values corresponding to transaction amounts to a financial statements vector through matrix multiplication. The matrix is called a *generator matrix* (it is used to generate financial statements). This representation of double entry bookkeeping allows us to simultaneously apply *coding theory* and linear algebra's *nullspace* to the analysis of the relationship between transactions and account balances. We show how to construct a *parity check matrix* directly from the generator matrix. Our main results demonstrate how the parity check matrix can be used to detect and correct errors which can not be corrected and may not even be detected by the usual trial balance check. (The trial balance check would be a subset of possible checks performed via the parity check matrix.) The idea is that there are more restrictions on the relations among the transaction which are captured by the generator matrix. These restrictions are then exploited by the parity check matrix, allowing for detection and correction of errors.

The paper is organized as follows. Section 2 establishes the equivalence of a bookkeeping system and matrix representation in linear algebra. In addition, we define double entry bookkeeping. Section 3 introduces the link between the concept of a nullspace in linear algebra and the error correcting abilities of a double entry bookkeeping system. Section 4 presents our main theorem on the detection and correction of up to two errors. Section 5 applies these ideas to an expansive example. In addition, implications for the design of bookkeeping systems are discussed. Section 6 concludes the paper.

2. Generator Matrix

In this section we define some terms which allow us to think carefully about bookkeeping systems. The definitions logically imply that we can use a linear algebra representation of a bookkeeping system.¹

For concreteness, we utilize a simple numerical example which we will refer to throughout the paper. Suppose a firm: (1) sells some stock, (2) uses the proceeds to buy equipment, (3) uses the equipment to generate cash revenue and (4) recognizes depreciation expense. These activities would result in a simple set of financial statements; a balance sheet with two assets (cash and equipment, along with a contra asset) and two equities (capital stock and retained earnings), and an income statement with one revenue and one expense (depreciation). Since retained earnings on the balance sheet can be calculated from the income statement, the financial statements can be viewed as a vector consisting of six elements: cash, equipment, accumulated depreciation, capital stock, sales and depreciation expense.

Suppose stock is sold for \$100, equipment is purchased for \$80, and cash revenue is \$30. Also, let the depreciation expense be \$20. Then the *transactions vector* is defined as: $t = [100 \ 80 \ 30 \ 20]$.

We are now ready to formally define a *bookkeeping system*.

Definition. A *bookkeeping system* is defined as a transformation, g , which takes a (row) vector of transaction amounts, t , and transforms it into a vector of financial statement account balances, a . (We assume opening balances of zero, purely for exposition purposes.³) The bookkeeping transformation has the following two properties.

1. $g(\alpha t) = \alpha g(t)$ for any scalar α .
2. $g(t_1) + g(t_2) = g(t_1 + t_2)$ for any two transaction vectors t_1 and t_2 .

The first property says that if all the transaction amounts are doubled, say, then all account balances must be doubled as well. This can be thought of as relying on a numeraire such as dollars. If the numeraire is changed to half dollars (multiply t by 2), then the account balances on the financial statements must also be multiplied by two.

The second property states that the effect on the account balances is the same even if a transaction is separated into two or more parts. For example, the effect on accounts receivable is the same whether one collection of \$100 occurs, or if the cash collection is divided into two parts: \$40 and \$60. This property is operationalized by the running balances in the T-accounts and by our indifference between making multiple journal entries versus a single composite entry.

The logical equivalence of a bookkeeping transformation and a *generator matrix* is established in Proposition 1.

Proposition 1. For all bookkeeping systems there exists a *generator matrix* G such that $t \cdot G = a$. That is the account vector is obtained by premultiplying the generator matrix by the transaction vector.

¹ We note that there exist a number of linear algebraic developments of bookkeeping systems in the accounting literature. Examples include Mattessich (1964), particularly chapters 3 and 4, and Ijiri (1967, 1975).

² Working with changes in account balances (assuming zero opening balances) is more general than it may appear. To incorporate balances, let account balance at time $i+1$ and i be a_{i+1} and a_i . And let transactions in period i be defined by t_i . Then $a_{i+1} = a_i + t_i \cdot G$. At some point opening balances are zero: a_i

Proof: The bookkeeping system is defined so as to be a linear transformation. The equivalence of a linear transformation and matrix multiplication is a standard result in linear algebra. See, for example, Theorem 8 in Lay (1994, 80), or Strang (1988, 116-121). \square

The generator matrix, G , has the number of rows equal to the number of transactions, m . The number of columns is equal to the number of financial statement accounts, n . G consists of 1's (when the account increases), -1's (when the account decreases), and zeroes (no change). Each row of G effectively records a journal entry. And pre-multiplying G by t produces account balances a .³

Given the definition of a bookkeeping transformation (and a fixed set of accounts) it is logical to impose further structure on the rows of G . In particular, any transaction which affects the same two accounts will be recorded as a single (summary) journal entry, leading to the definition below.

Definition. A *transaction* is a row of G which can not be expressed as a scalar multiple of any row.

The G matrix is thus an efficient way to characterize the *double entry* property, defined below.

Definition. A bookkeeping transformation is *double entry* if every row of G has two non-zero entries, that is, every transaction affects precisely two accounts.

This definition is not as restrictive as it may appear. Composite journal entries, those that affect more than two accounts, can be reproduced by a set of entries that each affect only two accounts. Our representation of a double entry bookkeeping system simplifies the analysis herein, without altering its substance.

³ = 0. So $a_i = a_i + t_i \cdot G = t_i \cdot G$. And $a_i = a_i + t_i \cdot G = (t_i + t_i) \cdot G$. Since all transactions satisfy the linearity property, closing balances will as well.

At this point we continue with our small accounting cycle example to illustrate the development. The double entry system transforms the four element transaction vector into a six dimensional account space. The generator matrix for this example is as follows.

$$G = \begin{array}{c} \begin{array}{cccccc} \text{Cash} & \text{Equip} & \text{Acc} & \text{Cap} & \text{Sales} & \text{Dep} \\ & & \text{Dep} & \text{Stock} & & \text{Exp} \end{array} \\ \left[\begin{array}{cccccc} 1 & 0 & 0 & 1 & 0 & 0 \\ -1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 \end{array} \right] \begin{array}{l} \text{(transaction 1)} \\ \text{(transaction 2)} \\ \text{(transaction 3)} \\ \text{(transaction 4)} \end{array} \end{array}$$

The matrix transformation from transactions amounts to account balances is as follows:
 $t \cdot G = a$, where we recall $t = [100 \ 80 \ 30 \ 20]$.

Premultiplying G by t means we are weighting the rows of G by the transaction amounts. Alternatively, we are taking the inner product of t with each column of G . Each such inner product calculates an account balance, just as we do with a T-account. For example, the calculations in the T-account for cash are accomplished by taking the inner product of t with the first column of G .

$$[100 \ 80 \ 30 \ 20] \cdot [1 \ -1 \ 1 \ 0] = 100 - 80 + 30 + 0 = 50$$

Cash	
100	80
30	
50	

Thus, the calculation of the financial statements is accomplished by matrix multiplication.

$$t \cdot G = [50 \ 80 \ 20 \ 100 \ 30 \ 20]$$

Below they appear in more familiar form.

³ One can see that the types of transactions and the desired accounts affect the construction of G . We shall see that G can be constructed so as to affect the ability to detect and correct errors.

Statement of Financial Position

Cash	50	Capital Stock	100
Equipment	80	Retained Earnings	<u>10</u>
Acc. Dep.	<u>20</u>		
Assets	110	Equities	110

Income Statement

Revenue	30
Depreciation	<u>20</u>
Income	10

It will be quite useful to retain the transaction vector when transforming it into account balances. This is accomplished by concatenating G with an m -dimensional identity matrix, denoted I_m : $t \cdot [I_m | G] = [t | a]$. We refer to $[I_m | G]$ as the *augmented G matrix*, and $[t | a]$ as the *transaction-account vector*. In our example, $[t | a] = [100 \ 80 \ 30 \ 20 \ 50 \ 80 \ 20 \ 100 \ 30 \ 20]$.

3. Constructing the Nullspace of Augmented G

Our purpose is to develop a systematic method of determining whether the stated account balances are consistent with the stated transactions. There are two ways to check for consistency. One way is to see if $[t | a]$ lies in the row space of augmented G . (Recall that if the transactions and account balances are consistent they must satisfy $t \cdot [I_m | G] = [t | a]$.) An indirect but equivalent way is to check if any vector that is orthogonal to the rows of augmented G is also orthogonal to $[t | a]$.⁴ Perhaps surprisingly, the latter approach is generally more useful.

The set of vectors orthogonal to augmented G is itself a subspace, and is called the *nullspace* of augmented G . Since to check $[t | a]$ using the latter method we must compare to this subspace, it is useful to find its basis. We then form a matrix consisting

⁴ See Strang (1988), p. 139.

of the vectors that together form a basis for the nullspace. In Proposition 2 we explain how to construct a basis for the nullspace of augmented G.

Proposition 2. A matrix H whose vectors form a basis for the nullspace of augmented G can be constructed as follows:

$$H = [-G^T \mid I_n].$$

Proof: To demonstrate that H is a basis for the nullspace, we must show that H has the following properties: (1) its rows are linearly independent and (2) it spans the nullspace of augmented G.

The first statement follows from the fact that H contains an identity matrix of dimension n. The second statement follows from a fundamental theorem of linear algebra. See, for example, Strang (1988). In general, the dimension of the nullspace of a matrix is equal to the number of columns less its rank. In augmented G, the number of columns is m+n and its rank is m. (The rank of augmented G is m since it has an identity matrix of size m.) Hence, the dimension of the nullspace of augmented G is equal to m+n-m=n. To show that H spans the nullspace of augmented G we need to simply show that H consists of n linearly independent rows each of which is orthogonal to rows in augmented G. By construction there are n rows in H and, as stated above, they are independent. All that remains is to prove that these n rows are orthogonal to the m rows in augmented G. From the rules of block matrix multiplication, the following statement is true:

$$H \cdot [I_m \mid G]^T = [-G^T \mid I_n] \cdot \begin{bmatrix} I_m \\ G^T \end{bmatrix} = [0], \text{ where } [0] \text{ is an } n \text{ by } m \text{ matrix of zeros. } \quad \square$$

In the coding theory literature, H is referred to as a parity-check matrix. See, for example, Theorem 7.6 in Hill (1986). There is an important corollary to Proposition 2, which we will make use of when we discuss error detecting and correcting in the next section.

Corollary 1. None of the $m+n$ columns of H can be expressed as a scalar multiple of any of its other columns.

Proof: The proof follows directly from these facts:

- (1) The last n columns are the identity matrix.
- (2) The first m columns ($-G^T$) have two entries each, and are not scalar multiples by definition of a transaction. ⊠

For the example the parity check matrix, H , is as follows.

Transactions	Cash	Equip.	Acc. Dep.	Cap. Stk.	Sales	Dep.	
-1 1 -1 0	1	0	0	0	0	0	Cash
0 -1 0 0	0	1	0	0	0	0	Equip.
0 0 0 -1	0	0	1	0	0	0	Acc. Dep.
-1 0 0 0	0	0	0	1	0	0	Cap. Stk.
0 0 -1 0	0	0	0	0	1	0	Sales
0 0 0 -1	0	0	0	0	0	1	Dep.

Note that H is orthogonal to $\begin{bmatrix} \mathbf{t}^T \\ \mathbf{a}^T \end{bmatrix}$, i.e.:

$$H \cdot \begin{bmatrix} \mathbf{t}^T \\ \mathbf{a}^T \end{bmatrix} = \begin{bmatrix} -1 & 1 & -1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & -1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ -1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} 100 \\ 80 \\ 30 \\ 20 \\ 50 \\ 80 \\ 20 \\ 100 \\ 30 \\ 20 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

One way accountants can check for errors is through the use of a trial balance, which we can capture within our framework by a *balancing vector*. The logic behind the trial balance is second nature to accountants. Net assets (assets minus liabilities) are

wealth, and they are logically equal to capital. The trial balance ensures that an increase in net assets is accompanied by either net asset decreases or capital increases in the same amount.⁵

The net assets/capital classification scheme can be imposed within our system (although we do not require it).⁶ The idea of a trial balance check exploits this property of the typical bookkeeping system, and is a special case of a balancing vector, which we describe below in linear algebra terminology.

Definition. A *balancing vector* is a vector b such that $b \cdot a^T = 0$.

Any balancing vector, should it exist, would be only one of many vectors in the nullspace of G , that is, $G \cdot b^T = 0$. Hence, the (nullspace) approach for checking consistency among transactions and account balances can do whatever the trial balance can do, plus more. This is stated formally in Corollary 2.

Corollary 2. The balancing vector augmented by m zeroes, $[0 \mid b]$, resides in the row space of H .

Proof. The account vector a lies in the row space of G . By definition, b is orthogonal to a , which implies it is orthogonal to rows of G . In turn, this implies $[0 \mid b]$ is orthogonal to augmented G , i.e., $[0 \mid b]$ lies in the nullspace of augmented G . From Proposition 2, the rows of H form a basis for the nullspace of augmented G . Hence it follows that $[0 \mid b]$ can be expressed as a linear combination of the rows of H . ⊠

For our example, consider $b = [1 \ 1 \ -1 \ -1 \ -1 \ 1]$. Observe that $b \cdot a^T = 0$, and further $[0 \mid b]$ is a linear combination of all the six rows in H with the weights as given by b . Also note that we have judiciously chosen b -- the positive 1's represent assets and

⁵ The existence of this system has been attributed to an aversion to negative numbers (Ijiri 1982).

⁶ Our definition of G is more general than that usually imposed by accountants. For example, the journal entry increase debt - increase capital would be allowed under our definition of a bookkeeping system. But this would violate an accountant's trial balance.

expenses; the negatives are equities and revenues. Thus, this particular balancing vector produces the same effect as the trial balance: it ensures "debits equal credits."

The balancing vector's place in accounting is important, as it is used to allow separation of wealth (net assets) and capital (Ijiri, 1982). But we do not delve into these issues, as: (1) the balancing vector is not necessary for our main results and (2) we promised not to talk about economics (so we won't).

In general then, if the financial statements have been prepared correctly, $H \cdot [t \mid a]^r = 0$. What if they had not been prepared correctly? We can further exploit properties of H to detect and even correct the errors.

4. Error Detecting and Correcting

In this section, we study the correction and detection of errors. Auditors examine the firm's transactions and accounts balances, and are interested in whether the account balances are correct. One check is to see if the account balances and transactions are consistent, that is, do they satisfy $t \cdot G = a$?

We must be careful about what we mean by an error. Let us denote the actual transactions vector by t . Then we say the account balances are correct if they satisfy $t \cdot G = a$. Let the stated transaction vector and account vectors be denoted by t' and a' . The auditor walks in to the firm and observes t' and a' , and is interested in whether the account balance is correct: is $a' = a$? How can the auditor tell? One way to spot a problem with the financial statements is to see if they are consistent with the transactions as presented - that is, does $t' \cdot G$ equal a' ? If not, the auditor knows errors occurred. Either $t' \neq t$, or $a' \neq a$, or both. Further, if an error occurred, the auditor may be interested in correcting the errors. It turns out that the bookkeeping system (by itself) can be useful in determining why an error occurred, and even sometimes in correcting it.

For now we restrict ourselves to zero, one, or two errors. As mentioned, if there were no errors, $t' \cdot G = a'$. If one error occurs, it is in either the a portion or the t portion of the vector $[t' \mid a']$: either $t' \neq t$ or $a' \neq a$. If two errors occur, two elements of the

observed vector $[t' | a']$ have been written down incorrectly. Either two elements of the account vector are written down incorrectly ($a' \neq a$ and $t' = t$), two elements of the transactions vector have been written down incorrectly ($t' \neq t$ and $a' = a$), or one element of each is incorrect ($t' \neq t$ and $a' \neq a$).

Suppose the transaction-account vector contains a single error. Continuing the numerical example, consider a transaction-account vector which looks like this.

$$[t' | a'] = [100 \ 80 \ 30 \ 20 \ 50 \ 92 \ 20 \ 100 \ 30 \ 20]$$

By comparing to the Statement of Financial Position (correctly) constructed earlier, we see there's an error of 12 in the equipment account. (The sixth entry is $92 = 80 + 12$). Using the trial balance can detect the existence of an error (the balance sheet doesn't balance), but the amount and position of the error can not be determined. Use of the entire nullspace, defined by the parity check matrix H, however, can accomplish both. Thus, the error can be corrected immediately.

The first step is to multiply the transaction-account vector by H. The result, $s = H \cdot \begin{bmatrix} t' \\ a' \end{bmatrix}$, is termed a "syndrome" in the coding jargon.

$$s = H \begin{bmatrix} 100 \\ 80 \\ 30 \\ 20 \\ 50 \\ 80+12 \\ 20 \\ 100 \\ 30 \\ 20 \end{bmatrix} = H \begin{bmatrix} 100 \\ 80 \\ 30 \\ 20 \\ 50 \\ 80 \\ 20 \\ 100 \\ 30 \\ 20 \end{bmatrix} + H \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 12 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} = H \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 12 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 12 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

For this one error case the syndrome, s, is the product of an error amount times a particular column of H. If we can identify which column s is a scalar multiple of, the

position of the error is determined. This is not a complicated task, and for the example can be done by inspection. The syndrome has a single non-zero entry, the second. Only one column of H has only a non-zero second entry. Thus, the column of H in question is the sixth. Correction of the error is accomplished by subtracting 12 from element 6 (equipment) of the transaction-account vector: $92 - 12 = 80$.

Error correction was successful in the example because the syndrome resided in the subspace of only one column of H . This allows us to (easily) identify the transaction or the account in which the error occurred.

The theorem below summarizes this and the rest of our main results on the error detecting and correcting properties of a double entry bookkeeping system.

Theorem. In a double entry bookkeeping system, any single error in the transaction-account vector can be detected, identified, and corrected using the syndrome method. Furthermore, up to two errors in the transaction-account vector can be detected using the syndrome method.

Proof: For the one error case, from Corollary 1 none of the columns in H are scalar multiples of any other, and so the syndrome will always reside in only one column. Hence, the location and magnitude of one error is detectable from the syndrome. Now consider the two error case. We need only show that it is not possible for two errors to yield a zero syndrome. Two errors in transactions will not yield a zero syndrome because no two transaction columns are scalar multiples. One error in transaction and one error in account will not yield a zero syndrome, as transaction columns have two non-zero entries while account columns have only one. Two errors in accounts will not yield a zero syndrome, because account columns are linearly independent. □

⁷ A syndrome is a set of symptoms of an illness. Here we are identifying and curing the illness (correcting the error) by first identifying the syndrome.

Consider the case when there are up to two errors but the balance sheet balances. Below is the stated transaction-account vector.

$$[t \mid a] = [100 \ 80 \ 30 \ 20 \ 40 \ 90 \ 20 \ 100 \ 30 \ 20]$$

It is clear that the balance sheet balances: total assets = total equities = 150. However, the syndrome is not zero:

$$s = H \cdot \begin{bmatrix} t^T \\ a^T \end{bmatrix} = \begin{bmatrix} -10 \\ 10 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}.$$

While the error has been detected, comparison of the syndrome to columns of H narrows down the possible source of the error. In this case there are only two possibilities. First, there may be a single error in the transaction 2 amount, since the syndrome is -10 times the second column of H. That is, the equipment purchase may have been 90, and not 80. Second, there may errors in both the cash and equipment account, since the syndrome is -10 times column 5 plus 10 times column 6. That is, the cash account is understated by 10 and the equipment account is overstated by 10.

It is interesting to compare the syndrome method of error detection and correction with merely checking whether or not the balance sheet balances. As noted in Corollary 2 any balancing vector must reside in the row space of H, hence any information contained in the balancing vector is surely contained in the syndrome.

In the one error case the syndrome will be non-zero and the balance sheet will not balance (if the error is in the financial statements). The balance sheet check can not identify the location or the sign of the error. The syndrome, on the other hand, can do both. In the two error case, it is possible for the balance sheet check to be fooled. Two errors may be counterbalancing and the balance sheet may actually balance. The

syndrome is not so easily fooled. The syndrome can not possibly be zero in the presence of two errors!

It is useful to remark on our assumption that there are two or fewer errors. Suppose that a transaction t' and account balances a' are supplied to the auditor. If it is true that $t' \cdot G = a'$, the bookkeeping system provides no evidence by itself that there is an error. It is possible for three errors to occur and for the syndrome to be zero. One obvious case is when a transaction amount is changed along with the two corresponding financial statement accounts. For example, if a credit sale transaction is overstated by 10, and both sales and accounts receivable are overstated by 10, the bookkeeping system is not helpful, since the erroneous transactions and account vectors, say, t' and a' , satisfy $t' \cdot G = a'$. Thus, three errors may in some cases go undetected. This is of course why internal control systems are monitored by the auditor, in addition to checking transactions against accounts. The accountant must look beyond the bookkeeping system if he wishes to catch strategically placed errors that exceed two in number.

5. Bookkeeping System Design - An example

In this section we consider a slightly more expansive example that allows us to illustrate some complications we could not address in the small example used so far. We can address problems with detecting errors when G consists of rows that are scalar multiples of each other. In addition, we can discuss issues associated with bookkeeping system design. In particular, we show how judiciously increasing the number of accounts can increase the error correcting capability of a double entry bookkeeping system.

We choose as a basis for our example Problem 2-4 in Hartman, Harper, Knoblett and Reckers (1995). There are 12 transactions, i.e., $m=12$.

1. Cash sales \$1500.
2. Purchase equipment \$41 for cash.
3. Purchase merchandise \$2021 on credit.
4. Depreciation \$18.
5. Credit sales \$610.

6. Payment for credit purchases \$2016.
7. Collections from customers \$606.
8. Sale of capital stock \$11.
9. Cost of goods sold \$1992.
10. Cash dividend \$21.
11. Tax expense \$40.
12. Taxes paid \$36.

There are seven financial statement accounts (n=7).

Cash
 Receivables
 Inventory
 Long-term assets (net)
 Current payables
 Long-term Debt
 Shareholders Equity

Based on the 12 transactions and irrespective of the amounts, the 12 by 7 generator matrix, G, is as follows.

$$G = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 1 \\ -1 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & -1 & 0 & 0 & -1 \\ 0 & 1 & 0 & 0 & 0 & 0 & 1 \\ -1 & 0 & 0 & 0 & -1 & 0 & 0 \\ 1 & -1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & -1 & 0 & 0 & 0 & -1 \\ -1 & 0 & 0 & 0 & 0 & 0 & -1 \\ 0 & 0 & 0 & 0 & 1 & 0 & -1 \\ -1 & 0 & 0 & 0 & -1 & 0 & 0 \end{bmatrix}$$

Based on the transactions, the 1 by 12 transaction vector is as follows.

$$t = [1500 \ 41 \ 2021 \ 18 \ 610 \ 2016 \ 606 \ 11 \ 1992 \ 21 \ 40 \ 36]$$

The correct financial statements are as follows.

$$t \cdot G = [3 \ 4 \ 29 \ 23 \ 9 \ 0 \ 50]$$

Unlike the previous example, some of the rows in G are scalar multiples of each other. For example, row 1 is the same as row 8 and is the negative of row 10. The bookkeeping system as designed uses one stockholder's equity account. As such, it does not completely distinguish between increases in cash due to sales of stock and those due to cash sales. In addition, it does not completely distinguish cash outflows due to dividends from these two transactions.

Thus, the Theorem must be modified slightly in the case where G has rows that are scalar multiples of each other. Detecting and correcting errors in transactions becomes problematic. However, errors in financial statement accounts can still be detected and corrected.

Proposition 3. Suppose there are rows in G which are scalar multiples of each other. Then the following are true: (1) any single error in the financial statement account vector can be detected, identified, and corrected using the syndrome, (2) any single error in the transaction vector can be detected, and (3) if there are one or two errors in the transaction-account vector, then any errors in the financial statement account vector can be detected using the syndrome method.

Proof: Consider the case of up to one error. If the syndrome is zero, neither the financial statements nor the transactions have an error. If the syndrome has only one non-zero entry, e in, say, the i th position, the transactions are correct, but there is an error in the i th financial statement account. To correct, subtract e from the account balance. If the syndrome has more than one non-zero entry, because of double entry, the syndrome will necessarily have exactly two non-zero entries, both of the same magnitude. In this case the financial statements are correct, the error is in one of the transactions.

Now consider the case of up to two errors. Suppose there is an error in a financial statement account. There are three possible cases: (1) there is an error only in a financial statement account (i. e., no errors in the transactions), (2) there is an error in a financial

statement account and an error in a transaction amount, and (3) there are errors in two financial statement accounts.

For the first case the syndrome has one non-zero element. In case two the syndrome has either one or three non-zero elements. For case three the syndrome has two non-zero elements. Hence, if there is an error in a financial statement account, the syndrome is necessarily non-zero. The proposition is the contrapositive of this statement.



To demonstrate Proposition 3, consider the following example. One error has been made in a financial statement account (current payables).

$$[t \mid a] = [1500 \ 41 \ 2021 \ 18 \ 610 \ 2016 \ 606 \ 11 \ 1992 \ 21 \ 40 \ 36 \ 3 \ 4 \ 29 \ 23 \ 20 \ 0 \ 50]$$

Detection and correction proceeds as follows. First calculate the syndrome. As an aside, note that the balance sheet test indicates that an error has occurred, but is no help in identifying the location or the sign of the error.

$$s = [-G^T \mid I_7] \cdot \begin{bmatrix} t^T \\ a^T \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 11 \\ 0 \\ 0 \end{bmatrix}$$

The syndrome has only one non-zero element, 11, in the fifth place. Hence, from Proposition 3 the current payables have been overstated by 11. Hence, the correct amount in the current payables account is $20 - 11 = 9$.

Let us explore what happens when there is an error in the transactions. Now, by Proposition 3, we are able to detect but not correct the error. Suppose an error of \$200 occurs in the cash sales transaction amount. Then the transaction-account is as follows.

$[t|a] = [1700 \ 41 \ 2021 \ 18 \ 610 \ 2016 \ 606 \ 11 \ 1992 \ 21 \ 40 \ 36 \ 3 \ 4 \ 29 \ 23 \ 9 \ 0 \ 50]$

The syndrome is:

$$s = [-G^T | I_7] \cdot \begin{bmatrix} t^T \\ a^T \end{bmatrix} = \begin{bmatrix} -200 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ -200 \end{bmatrix}.$$

There are two non-zero entries, so we know the financial statements are correct as submitted. The syndrome also tells us that one of the transactions is off by \$200. The possibilities are that transaction 1 is overstated by \$200, or that transaction 8 is overstated by \$200, or even that transaction 10 is understated by \$200.

The reason the syndrome is unable to distinguish the location of the error is that the columns in H, and the corresponding rows in G for the above transactions, are scalar multiples of one another. One way to remedy the situation is to add two additional accounts. In this case a capital stock account (see transaction 8), and a dividends account (for transaction 10) would do the trick. The new G matrix with the last two columns denoting capital stock and dividends, would look as follows. (In the G matrix below the seventh column is shareholders equity less capital stock and dividends.)

$$G = \begin{array}{c} \left[\begin{array}{cccccc|cc|c} & & & & & & \text{CS} & \text{Div} & \\ 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ -1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & -1 & 0 & 0 & -1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ -1 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 \\ 1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & -1 & 0 & 0 & 0 & -1 & 0 & 0 \\ -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 & 0 & -1 & 0 & 0 \\ -1 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 \end{array} \right] \end{array}$$

In this modified G matrix rows one, eight and ten are no longer scalar multiples of one another. The syndrome is

$$\begin{array}{c} \left[\begin{array}{c} -200 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ -200 \\ 0 \\ 0 \end{array} \right] \end{array}$$

and the only column of H (row of G) within which it resides is the first one, i.e., the first transaction is \$1500. Now both detection and correction are possible.

Attempts to avoid negative numbers has been supplied as a reason for the development of the double entry bookkeeping system (Peters and Emery, 1978). This has resulted in the existence of many accounts (rather than two: net assets and capital). Ijiri (1982) states that negativity avoidance increases control over the data: "It is more effective from the control standpoint to monitor not just net balance but gross amounts of

positive and negative entries separately. Therefore, sales and sales returns are recorded separately, properties and accumulated depreciation are controlled by separate accounts, and netting of treasury stock and capital stock is prohibited." (p. 5)

This is just what we have accomplished by splitting the capital account into capital stock and dividends. Rather than simply subtracting from cash and capital, we subtract from cash but increase another account, dividends. Our analysis shows that controllability is increased in that more errors can be detected. This increase occurs because adding accounts can increase the rank of the generator matrix.

The exercise gives some guidance to the problem of how many and what accounts should be included in the bookkeeping system. Observe that the sum of the last three columns in the above G matrix is the same as the shareholders equity column in the G matrix we presented earlier. To the extent that error detection and correction is an objective, it may be helpful to add additional accounts, so as to usefully disaggregate. Disaggregation helps with error detection and correction, since additional accounts destroy the multiplicative similarity of the rows of G (and columns of H).⁸ In other words, it is desirable to have a situation where the rows are not scalar multiples of each other.

6. Conclusion

An unresolved issue is whether accounting has sufficient intellectual depth to support excursions not grounded in economics. This paper views the double entry bookkeeping system as the accountant's territory, and analyzes its properties using linear algebra and coding theory. The double entry bookkeeping system is modeled as a generator matrix that transforms a transactions vector into a financial statements account vector.

⁸ The concept of accounting aggregation is a pervasive theme in Demski (1994). (See in particular, pages 90-91.)

The Theorem implies that once a generator matrix is specified, single error correction and double error detection in transactions and financial statement accounts can always be achieved by a double entry bookkeeping system. Our analysis has implications for bookkeeping system design, as we show that judicious selection of accounts can enhance its inherent error detection and correction abilities.

Error correcting abilities may become even more important with the increasing amount of accounting information being transmitted electronically.⁹ With this in mind, the advisability of increasing the number of entries (triple-entry bookkeeping?) may be an interesting question to investigate.

⁹ The error correcting ability of the bookkeeping system is similar to error correcting codes used to send photographs from distant points in the solar system. Remarkably clear and impressive images are received even though the messages are subject to significant natural interference.

References

- DEMSKI, J. Accounting Theory. Working paper, University of Florida 1990.
- DEMSKI, J. *Managerial Uses of Accounting Information*. Norwell, MA: Kluwer Academic Publishers, 1994.
- HARTMAN, B., R. HARPER, J. KNOBLETT, AND P. RECKERS. *Intermediate Accounting*. Minneapolis, MN: West Publishing Company, 1995.
- HILL, R. *A First Course in Coding Theory*. New York, NY: Oxford University Press, 1986.
- IJIRI, Y. *The Foundations of Accounting Measurement*. Englewood Cliffs, NJ: Prentice-Hall, 1967.
- IJIRI, Y. *Theory of Accounting Measurement*. Sarasota, FL: AAA, 1975.
- IJIRI, Y. *Triple-Entry Bookkeeping and Income Momentum*. Sarasota, FL: AAA, 1982.
- LAY, D. *Linear Algebra and its Applications*. Reading, MA: Addison-Wesley Publishing Company, 1994.
- MATTESSICH, R. *Accounting and Analytical Methods*. Houston, TX: Scholars Book Company, 1964.
- PETERS, R. AND D. EMERY. The Role of Negative Numbers in the Development of Double Entry Bookkeeping. *Journal of Accounting Research* (Autumn 1978) 424-426.
- STRANG, G. *Linear Algebra and its Applications*. San Diego, CA: Harcourt Brace Jovanovich Publishers, 1988.