Double-Entry Accounting:
The Mathematical Formulation and Generalization

David Ellerman*

Although double-entry accounting has been used in the business world for 5 centuries, the mathematical formulation of the double entry method is almost completely unknown. In this paper, the usual case of scalars is given first and then generalized to the multi-dimensional case of vectors. The success in maintaining the two-sided accounts, debits and credits, the double-entry principle, and the trial balance in both cases provides strong evidence that the formulation correctly captures the double-entry method in mathematical form.

Introduction

Double-entry bookkeeping illustrates one of the most astonishing examples of intellectual insulation between disciplines, in this case, between accounting and mathematics. Double-entry bookkeeping (DEB) was developed during the fifteenth century and was recorded in 1494 as a system by the Italian mathematician Luca Pacioli [1914]. Double-entry book-keeping has been used for over five centuries in commercial accounting systems. If the mathematical formulation of any field should be well understood, one would think it might be accounting. Remarkably, however, the mathematical formulation of double entry accounting – algebraic operations on ordered pairs of numbers – is largely unknown, particularly in the field of accounting.

The mathematical basis for a precise treatment of DEB was developed in the nineteenth century by William Rowan Hamilton as an abstract mathematical construction using ordered pairs to deal with the complex numbers [Hamilton 1837]. The multiplicative version of this construction is the "group of fractions" which uses ordered pairs of whole numbers (written vertically) to enlarge the system of positive whole numbers to the system of positive fractions containing multiplicative inverses (just reverse the entries in a fraction to get its inverse). The ordered pairs construction that is relevant to conventional DEB is the additive case called the "group of differences." It is used to construct a number system with "additive inverses" by using operations on ordered pairs of positive numbers including zero (unsigned numbers). All that is required to grasp the connection with DEB is to make the identification:

ordered pairs (horizontally written) of numbers in group of differences construction = two-sided T-accounts of DEB (debits on the left side and credits on the right side).

In view of this identification, the group of differences (or fractions in the multiplicative case) will be called the Pacioli group.

In spite of some attention to DEB by mathematicians [e.g., DeMorgan 1869, Cayley 1894, and Kemeny et al. 1962], this connection has not been noted in mathematics (not to mention in

---

* David P. Ellerman retired as visiting scholar at University of California/Riverside after teaching and consulting in a number of fields with two masters degrees (in Philosophy and in Economics) and a doctorate in Mathematics. www.ellerman.org

accounting) with one perhaps solitary exception. In a semi-popular book, D. E. Littlewood noted the connection: The bank associates two totals with each customer's account, the total of moneys credited and the total of mon- eys withdrawn. The net balance is then regarded as the same if, for example, the credit amounts of £102 and the debit £100, as if the credit were £2 and the debit £50. If the debit exceeds the credit the balance is negative.

This model is adopted in the definition of balanced integers. Consider pairs of cardinal numbers \((a, b)\) in which the first number corresponds to the debit, and the second to the credit. [1960, 18]

With this exception, the author has not been able to find a single mathematics book or paper, elementary or advanced, popular or esoteric, which notes that the ordered pairs of the group of differ- ences construction are the \(T\)-accounts used in the business world for about five centuries.

Is the Pacioli group the correct formulation of \(DEB\)? One acid test of a mathema- tical formulation of a theory is the question of whether or not it facilitates the generalization of the theory. Normal bookkeeping does not deal with incommensurable physical quantities; everything is expressed in the common units of money. Is there a generalization of \(DEB\) to deal with multi-dimensional incommensurability with no common measure of value (e.g., multiple "bottom lines" or environmental accounting)?

In the literature on the mathe- matics of accounting there was a proposed "solution" to this question, a system of multi-dimensional physical accounting [see Ijin 1965, 1966, and 1967]. In this system, most of the normal structure of \(DEB\) was lost:

- there was no balance sheet equation,
- there were no equity or proprietorship accounts,
- the temporary or nominal accounts could not be closed, and
- the "trial balance" did not balance.

It is common for certain aspects of a theory to be lost in a generalization of the theory. The accounting community had apparently accepted the failure of all these features of \(DEB\) as the necessary price to be paid to generalize \(DEB\) to incommensurable physical quantities. For example, the systems of "Double-entry multidimensional accounting" previously published in the accounting literature [see also Charnes et al. 1976, or Hase- man and Whinston 1976] had acqui- sed the absence of the balance-sheet equation.

For instance, the convenient idea of an accounting identity is lost since the di- mensional and metric compatibility it as- sumes is no longer present except under special circumstances. [Ijin 1967, 333]

Yet when \(DEB\) is mathematically formulated using the group of differ- ences, then the generalization to vectors of incommensurable physical quantities is immediate and trivial. All of the normal features of \(DEB\) — such as the balance- sheet equation, the equity account, the temporary accounts, and the trial balance — are preserved in the generalization [see Ellerinan 1982, 1983, 1986]. Thus the "accepted" generalized model of \(DEB\) was simply a failed attempt at generalization which had been "received" as a successful generalization that unfortu- nately had to "sacrifice" certain features of \(DEB\).

Due to this remarkable intellectual insulation between mathematics and accounting, the successful mathematical treatment and generalization of double-entry bookkeeping (first published a quarter-century ago) will take many more years to become known and under- stood in accounting.

The Pacioli Group

Multi-dimensional accounting is based on the group of differences or Pacioli group construction starting with non-negative vectors. The usual case of accounting can be identified with the special case using one dimensional vectors or scalars. A vector \(x = (x_1, \ldots, x_n)\) is non-negative if and only if all its components \(x_i\) are non-negative (positive or zero). The ordered pairs of non-nega-

\[\text{We take away the need for fractions to become more used and understood in accounting.}\]

The Pacioli group \(P^+\) consists of the ordered pairs \((x \div y)\) of non-negative \(n\)-dimensional vectors, with the above definition of addition and equality. The Pacioli group \(P^+\) is isomorphic with all of \(R^+\) (the set of all \(n\)-vectors with positive and negative components) under two isomorphisms: the debit isomorphism, which associates \([w \div x]\) with \(w-x\), and the credit isomorphism, which associates \([w \div x]\) with \(x-w\). In order to translate from \(T\)-accounts \([x \div y]\) back and forth to general vectors \(z\), one needs to specify whether to use the debit or credit iso- morphism. This will be done by labeling the \(T\)-account as debit balance or credit balance. Thus if a \(T\)-account \([x \div y]\) is debit balance, the corresponding vector is \(x-y\), and if it is credit balance, then the corresponding vector is \(y-x\).

The Double-Entry Method: Scalar Case

Given an equation \(w = x \div y \div \cdots\), it is not possible to change just one term in the equation and have it still hold.\(^1\)

\(^1\) The double-entry notation was suggested by Pacioli. At the beginning of each entry, we always provide 'per', because first, the debits must be given, and immediately after the credits.
<table>
<thead>
<tr>
<th>Operation on T-accounts</th>
<th>Additive Case</th>
<th>Multiplicative Case</th>
</tr>
</thead>
<tbody>
<tr>
<td>T-accounts add together by adding</td>
<td>$\frac{w}{x} + \frac{y}{z} = \frac{wx + yz}{xz}$</td>
<td>Fractions multiply together by multiplying numerator times numerator and denominator times denominator $\frac{w}{x} \cdot \frac{y}{z} = \frac{wy}{xz}$.</td>
</tr>
<tr>
<td>Identity element for operation</td>
<td>The identity element for addition is the zero T-account $[0, 0]$.</td>
<td>The identity element for multiplication is the unit fraction $\frac{1}{1}$.</td>
</tr>
<tr>
<td>Equality between two T-accounts.</td>
<td>Given two T-accounts $[\frac{w}{x}, \frac{y}{z}]$ and $[\frac{u}{v}, \frac{r}{s}]$, the cross-sums are the two vectors obtained by adding the credit entry in one T-account to the debit entry in the other T-account. The equivalence relation between T-accounts is defined by setting two T-accounts equal if their cross-sums are equal: $[\frac{w}{x}, \frac{y}{z}] = [\frac{u}{v}, \frac{r}{s}]$ if $wz = yz$.</td>
<td>Given two fractions $\frac{w}{x}$ and $\frac{y}{z}$, the cross-multiples are the two integers obtained by multiplying the numerator of one with the denominator of the other. The equivalence relation between fractions is defined by setting two fractions equal if their cross-multiples are equal: $\frac{w}{x} = \frac{y}{z}$ if wz = xy.</td>
</tr>
<tr>
<td>Inverses</td>
<td>The negative or additive inverse of a T-account is obtained by reversing the debit and credit entries: $-\frac{w}{x} = \frac{-x}{w}$.</td>
<td>The multiplicative inverse of a fraction is obtained by reversing the numerator and denominator: $\frac{w}{x}^{-1} = \frac{x}{w}$.</td>
</tr>
<tr>
<td>&quot;Disjointness&quot; of two T-accounts</td>
<td>Given two vectors $x = (x_1, x_2, ..., x_n)$ and $y = (y_1, y_2, ..., y_n)$, let $\max(x,y)$ be the vector with the maximum of $x_i$ and $y_i$ as its ith component, and let $\min(x,y)$ be the vector with the minimum of $x_i$ and $y_i$ as its ith component. Two non-negative vectors $x$ and $y$ are said to be disjoint if $\min(x,y) = 0$.</td>
<td>Given two integers $w$ and $x$, let $\text{lcm}(w,x)$ be the least common multiple of $w$ and $x$, and let $\text{gcd}(w,x)$ be the greatest common divisor of $w$ and $x$ (the largest integer dividing both). Two integers $w$ and $x$ are said to be relatively prime if $\text{gcd}(w,x) = 1$.</td>
</tr>
<tr>
<td>&quot;Reduced form&quot; for a T-account</td>
<td>A T-account $[\frac{x}{y}, \frac{y}{z}]$ is in reduced form if $x$ and $y$ are disjoint.</td>
<td>A fraction $\frac{w}{x}$ is in lowest terms if $w$ and $x$ are relatively prime.</td>
</tr>
<tr>
<td>Unique reduced form representation</td>
<td>Every T-account $[\frac{x}{y}, \frac{y}{z}]$ has a unique reduced representation $[\frac{x-\min(x,y)}{y-\min(x,y)}]$.</td>
<td>Each fraction $\frac{w}{x}$ has a unique representation in lowest terms $\frac{w}{\gcd(w,x)} / \frac{x}{\gcd(w,x)}$.</td>
</tr>
<tr>
<td>Example</td>
<td>Consider the T-account $[\frac{10}{3}, \frac{4}{5}, \frac{2}{3}]$, $[\frac{10}{3}, \frac{4}{5}, \frac{2}{3}]$. The minimum of the debit and credit vectors is $(0, 3, 4)$ so the reduced form representation is $[(2, 0, 4) / (0, 2, 0)]$.</td>
<td>Consider the fraction $\frac{28}{95}$. The greatest common divisor of the numerator and denominator is 7 so the fraction in lowest terms is $\frac{4}{25}$.</td>
</tr>
</tbody>
</table>

Two or more terms must be changed. The fact that two or more terms (or "accounts") must be changed is not the basis for the double-entry method. That mathematical fact is a characteristic of the transaction itself (the changes in the equation), not a characteristic of the method of recording the transaction. The double-entry method is a method of encoding an equation using ordered pairs of T-accounts and using unsigned numbers (non-negative numbers) to record transactions to make changes in the equation. While there is unfortunately considerable confusion about this in the accounting literature, the doublingness of "double-entry" is the two-sidedness of the T-accounts and the mathematical properties that follow (e.g., equal debits and credits in a transaction, and equal debits and credits in the trial balance of the whole set of accounts or ledger).
a complete accounting system to update the balance sheet equation, but would have no two-sided T-accounts, no debits or credits, no double entry principle (equal debits and credits in a transaction), and no trial balance of adding debits and credits.

Unfortunately, the phrase “single entry accounting” is also used to denote simply an incomplete accounting “system” (e.g., no equity account) where there is no equation to be updated. But without such an equation, that is not an alternative “system” at all. The real choice between using the double entry method and the complete single entry method of recording a transaction is the choice between using unsigned (“single-sided”) numbers in two-sided accounts or signed (“two-sided”) numbers in “single-sided” accounts.

Consider an example of a company with the simplified initial balance sheet equation:

\[
\begin{align*}
\text{Assets} &= \text{Liabilities} + \text{Equity} \\
15000 &= 10000 + 5000.
\end{align*}
\]

**Equation 1: Beginning Scalar Balance Sheet**

It is customary in accounting (although not mathematically necessary) to move each term or account to the side of the equation so that it is preceded by a plus sign. A T-account equal to the zero T-account \([0 // 0]\) is called a zero-account. Equations encode as zero-accounts. Each left-hand side (LHS) term \(x\) is encoded as a debit-balance T-account \([x // 0]\) and each right-hand side (RHS) term \(y\) is encoded as a credit-balance T-account \([0 // y]\). These T-accounts then would add up to the zero-account \([0 // 0]\). The balance sheet equation thus encodes as an equation zero-account which, by leaving out the plus signs, becomes the following initial ledger of T-accounts.

\[
\begin{align*}
\text{Assets} &= \text{Liabilities} + \text{Equity} \\
15000 &= 10000 + 5000.
\end{align*}
\]

**Equation 2: Beginning Ledger of T-Accounts**

Consider three transactions in a productive firm.
1. \(\$1200\) of input inventories are used up and charged directly to equity.
2. \(\$1500\) of product is produced, sold, and added directly to equity.
3. \(\$800\) principal payment is made on a loan.

Each transaction is then encoded as a transactional zero-account and added to the appropriate terms of the equational zero-account. For instance, the first transaction subtracts 1200 from Assets and subtracts 1200 from Equity. The Assets account is encoded as a LHS or debit-balance account so the subtracting of a number from it would be encoded as adding the T-account \([1200 // 0]\) to it. Equity is encoded as a RHS or credit-balance term so subtracting 1200 from it would be encoded as adding \([1200 // 0]\) to it. The other transactions are encoded in a similar manner. The initial T-accounts in the ledger add up to the zero account (initial trial balance). Each transaction is encoded as two or more T-accounts that add to the zero-account (double entry principle). Zero added to zero equals zero. Thus adding the transaction zero-accounts to the initial equation zero-account (posting journal to ledger) will yield another equation zero-account (which can be checked by taking another trial balance). Each T-account is then decoded according to how whether it was encoded as debit balance or credit balance to obtain the ending balance sheet equation.

\[
\begin{align*}
\text{Assets} &= \text{Liabilities} + \text{Equity} \\
14500 &= 9200 + 5300.
\end{align*}
\]

**Equation 4: Ending Balance Sheet Equation**

The Double-Entry Method: General Case

The general case of the double-entry method starts with an equation between sums of n-dimensional vectors. Vector equations are first encoded in the Pacioli group constructed from the non-negative n-dimensional vectors. Since the vectors in a T-account must be non-negative, we must first develop a way to separate the positive and negative components of a vector. The positive part of a vector \(x = x^+ - \min(x,0)\) is the maximum of \(x\) and the zero vector [note that \(0\) is used, depending on the context, to refer to the zero scalar or the zero vector]. The negative part of \(x\) is \(x^- = -\min(x,0)\), the negative of the minimum of \(x\) and the zero vector. Both the positive and negative parts of a vector \(x\) are non-negative vectors. Every vector \(x\) has a “Jordan decomposition” \(x = x^+ - x^-\). The two isomorphisms that map vectors to T-accounts of non-negative vectors are the debit isomorphism that maps \(x\) to the T-account \([x^+ // x^-]\) and the credit isomorphism that maps \(x\) to \([x^- // x^+\]).

Given any equation in \(\mathbb{R}_n\), \(w + ... + x + y + ... + z\), each left-hand side (LHS) vector \(x\) is encoded via the debit isomorphism as a debit-balance T-account \([x^+ // x^-]\) and each right-hand side (RHS) vector \(y\) is encoded via the credit isomorphism as a credit-balance T-account \([y // y^-]\). Then the original equation holds if and only if the sum of the encoded
T-accounts is a zero-account:
\[ w + \ldots + x = y + \ldots + z \]
if and only if
\[ \left[ w \leftrightarrow w \right] + \ldots + \left[ x \leftrightarrow x \right] \quad + \left[ y \leftrightarrow y \right] + \ldots + \left[ z \leftrightarrow z \right] \]
Equation 5: Encoding an Equation as an Equation Zero-Account

Given the equation, the sum of the encoded T-accounts is the equation zero-account of the equation. Since only plus signs can appear between the T-accounts in an equational zero-account, the plus signs can be left implicit. The listing of the T-accounts in an equational zero-account (without the plus signs) is the ledger.

Changes in the various terms or "accounts" in the beginning equation are recorded as transactions. Transactions must be recorded as valid algebraic operations which transform equations into equations. Since equations encode as zero-accounts, a valid algebraic operation would transform zero-accounts into zero-accounts. There is only one such operation in the Pacioli group; add on a zero-account. Zero plus zero equals zero. The zero-accounts representing transactions are called transaction zero-accounts. The listing of the transactional zero-accounts is the journal.

A series of valid additive operations on a vector equation can then be presented in the following standard scheme:

Beginning Equation Zero-Account
+ Transaction Zero-Accounts
= Ending Equation Zero-Account

or, in more conventional terminology:

Beginning Ledger
+ Journal
= Ending Ledger

The process of adding the transaction zero-accounts to the initial ledger to obtain the ledger at the end of the accounting period is called posting the journal to the ledger. The fact that a transaction zero-account is equal to \[0 \leftrightarrow 0\] is traditionally expressed as the double-entry principle that transactions are recorded with equal debits and credits. The summing of the debit and credit sides of the braic operations represented in the transactions. The T-accounts in an equational zero-account can be arbitrarily partitioned into two sets: DB (debit balance) and CB (credit balance). T-accounts \[w \leftrightarrow x\] in DB are decoded as \(-w \leftrightarrow x\) on the left side of the equation, and T-accounts \[w \leftrightarrow x\] in CB are decoded as \(-w \leftrightarrow x\) on the right side of the equation. Given a zero-account, this algorithm yields an equation. In an accounting application, the T-accounts in the final equation zero-account would be partitioned into sets DB and CB according to the side of the initial equation from which they were encoded.

Consider the following initial vector equation:
\[(6, -3, 10) \times (-2, 5, -2) = (4, 2, 8)\]

Equation 6: Sample Vector Equation to be Encoded

It encodes as the equation zero-account
\[[(6, 0, 10) \leftrightarrow (0, 3, 0)] + [(0, 5, 0) \leftrightarrow (2, 0, 2)] + [(0, 0, 0) \leftrightarrow (4, 2, 8)]\]

Equation 7: Equation Encoded as a Zero-T-Account

Suppose that the transaction would subtract the vector \((-2, -9, 1)\) from the first vector on the LHS and from the vector on the RHS side of the original equation to obtain the ending equation:
\[(8, 6, 9) \times (-2, 5, -2) = (6, 11, 7)\]

Equation 8: Ending Vector Equation

To perform this operation using the double-entry method, the subtracting of the vector \((-2, -9, 1)\) from the first LHS term is encoded using the credit isomorphism to get \[(2,9,0)\leftrightarrow(0,0,1)\]
which is added to the first LHS or debit balance term in the T-account version of the original equation. In more traditional terminology, we would say that \((-2, -9, 1)\) is "credited" to that debit-balance account. For the subtraction from the RHS term, the vector is encoded using the debit isomorphism to obtain \[(0,0,1)\leftrightarrow(2,9,0)\]
and added to the credit-balance T-account version of the RHS term. That is, \((-2, -9, 1)\) is "debited" to that credit-balance account. This yields another equational zero-account:

Original Equation zero-account:
\[[(6,0,10)\leftrightarrow(0,3,0)] + [(0,5,0)\leftrightarrow(2,0,2)] + [(0,0,0)\leftrightarrow(4,2,8)]\]
+ Transaction zero-account:
\[[(2,9,0)\leftrightarrow(0,0,1)] + [(0,0,1)\leftrightarrow(2,9,0)]\]
Endwing zero-account:
\[[(8,9,10)\leftrightarrow(0,3,1)] + [(0,5,0)\leftrightarrow(2,0,2)] + [(0,0,1)\leftrightarrow(6,11,8)]\]

Equation 9: Beginning Ledger + Journal = Ending Ledger

After a number of such transactions, the ending equation zero-account is then decoded to obtain an equation back in Rs. In this case, let the first two T-accounts be debit-balance and the third one credit-balance (as they were originally encoded). Then the ending equational zero-account decodes as the vector equation:
\[(8, 6, 9) \times (-2, 5, -2) = (6, 11, 7)\]

Equation 10: Decoded Ending Equation

In the scalar case, a T-account will always have a reduced form either as \[d \leftrightarrow 0\] or \[0 \leftrightarrow c\] so that adding \[d \leftrightarrow 0\]
to an account (a term in the equational zero-account) can be described as “debiting d to the account” and similarly for “crediting c to the account.” For vector T-accounts, the reduced form of a T-account does not necessarily have the zero vector on one side or the other. In the case above, the reduced form of the T-account encoding of (2, 0, 0, 0) removes the zero vector to give the “debits” of the reduced form of the T-account. The formula for the “debit” of the reduced form of the T-account is $d = (1, 1, 1, 0)$.

The “credit” of the reduced form of the T-account is $c = (0, 1, 1, 0)$. The “debit isomorphism to” a term, and the “credit” isomorphism to another term.

Concluding Remarks

"Mathematics is the study of analogies between analogies." [Rota 1997, 214] The mathematical analysis of double-entry bookkeeping starts with the analogy between the ordered pairs construction of additive inverses and T-accounts. That allows the concrete procedures of DEB to be reproduced abstractly using the group of differences construction, and then to be extended from ordered pairs of non-negative scalars to ordered pairs of non-negative vectors to obtain the system of n-dimensional DEB. The immediate and straightforward generalization of the ordered pairs treatment to n-dimensional vectors and to fractions with the main structure and principles of DEB preserved – supports the thesis that this treatment captures the mathematical essence of the double-entry method.

This mathematical treatment of DEB serves another purpose – to help clear up the remarkable confusion in the accounting literature about the “doubleness” that is characteristic of DEB. The conventional wisdom is still that the doubleness refers to the fact that each transaction must affect two or more accounts (see any textbook) when that is a characteristic of the transaction itself, not the recording method. Any complete accounting system, such as the double entry method using “single-sided” numbers (no minus signs) with double-sided accounts, or a complete single-entry system using “double-sided” numbers (positive or negative) in single-sided accounts, must change two or more accounts in recording a transaction.

References


