The detection of double errors in ISBN- and ISSN-like codes

EGGHE, Leo & ROUSSEAU, Ronald (2001) The detection of double errors in ISBN- and ISSN-like codes. In: Mathematical and Computer Modelling, 33(8-9). p. 943-955.

DOI: 10.1016/S0895-7177(00)00291-0
Handle: http://hdl.handle.net/1942/794

On the detection of double errors in ISBN and ISSN-like codes

Leo Egghe
LUC, Universitaire Campus, B-3590 Diepenbeek, Belgium
and UIA, IBW, Universiteitsplein 1, B-2610 Wilrijk, Belgium

and

Ronald Rousseau
UIA, IBW, Universiteitsplein 1, B-2610 Wilrijk, Belgium
and KHBO, Zeedijk 101, B-8400 Oostende, Belgium

Abstract

Coding of the ISBN and ISSN is studied, and possible alternatives, not all equivalent with the official ones, are formulated. A minimum requirement for a useful code is that all single errors as well as all permutations of two symbols must be detectable. The strength of alternative codes, in particular with respect to the detection of double errors is investigated. We give a complete description of the method based on division by 11 (the official one). In this case we are also able to describe the power of the method with respect to the detection of three or four errors. We show that, using division by 11, all coding methods detect the same percentage of errors. In case larger prime numbers are used as divisor numerical experiments were performed showing that different methods have different detection capabilities for double errors.

Acknowledgement. The authors are indebted to Brendan Rousseau (Fontys Hogeschool, Eindhoven, The Netherlands) for help in programming error detection calculations. We also thank Dr. J.-M. Debois for informing us about reference [1].

Introduction

The ISBN (International Standard Book Number) and ISSN (International Standard Serial Number) are well-known codes helping the information professional to identify books or serials [2]. They consist respectively of 10 and 8 digits of which the last one is a check digit (0,1,...,9 or X). Although their origins are different: publishers and booksellers introduced the ISBN, while the ISSN was created by UNISIST (a UNESCO project) they both are ISO standards [1], [3] and serve a similar purpose, namely attaching a unique number to every publication. Here the term 'publication' should be understood in its full diversity: e.g. hard and soft covers editions receive different ISBNs. Moreover, an additional digit, a so-called check digit, is added ensuring that certain typing (or printing) errors can be detected. Preventing mistakes of this kind has economic implications: in this way wrong deliveries – and hence extra administrative costs - are prevented (or their numbers are at least considerably reduced).

The ISBN consists of four parts, usually separated by a hyphen. The first part is a regional code (e.g. 0 for the English region), the second part is a publisher code (e.g. 444 indicates an Elsevier book), the third part uniquely determines the work and the last part is a check digit. Classically, the calculation of the check digit for an ISBN is performed as follows: multiply the first number by 10, the second one by 9, and so on, until finally the ninth number is multiplied by 2. Then these numbers are added and divided by 11. The check digit is then eleven minus the remainder (after division by 11). Note that '10' is written as X. In case the rest is 0 the check digit is also 0 (which, by the way, is equal to 11 mod 11, i.e. the rest of 11 after a division by 11).

It was shown by Egghe [4] (based on a note by McMurdo [5] and a letter by MacCormack [6]) that this coding method is equivalent to the following one. Multiply the first number by 1, the second one by 2, and so on until the ninth one is multiplied by 9. Add these numbers and divide the sum by 11. The check digit is then equal to the rest (where again 10 is written as X). Note that this formulation, which can also by found in [7], is slightly easier than the official one.

We will now rewrite these algorithms as mathematical formulae. Let $X = (x_1,\ldots,x_9)$ denote the ISBN without the check digit. Then the first method boils down to calculating

$$\sum_{k=1}^{9}(11-k)x_k \qquad (1)$$

taking the rest after a division by 11 and, if this rest is not zero, using 11 minus this rest as a check digit. The rest of a number, say n, after division by another number, say d, is denoted as n mod d. So the check digit, following the official method is:

$$(11-(\sum_{k=1}^{9}(11-k)x_k)\bmod 11)\bmod 11 \qquad (2)$$

The second method calculates the simpler form:

$$\left(\sum_{k=1}^{9}k\,x_k\right)\bmod 11 \qquad (3)$$

As an example the reader may verify that both methods yield 4 as the check digit for 056603515.

It is well known that this check digit detects all single errors, as well as all permutations of two different symbols. We will prove this result in a more general setting (see next section).

The official method to calculate the check digit of the ISSN is very similar to that for the ISBN: multiply the first number by 8, the second one by 7 and so on until the last one (the seventh) is multiplied by 2. Then all these numbers are added and divided by 11. Again the check digit is 11 minus the rest after a division by 11, where 10 is written as X, and if the rest was 0, the check digit is again 0. Also here a simpler method can be used. Mathematically, the official method of calculating a check digit for an ISSN boils down to the calculation of:

$$\left(11-(\sum_{k=1}^{7}(11-(k+2))\,x_k)\bmod 11\right)\bmod 11 \qquad (4)$$

where $X = (x_1,\ldots,x_7)$ denotes an ISSN without check digit. The alternative method attains the same result by performing the simpler calculation (5) (a proof is given in Appendix A):

$$\left(\sum_{k=1}^{7}(k+2)\,x_k\right)\bmod 11 \qquad (5)$$

So, a first advice we want to give is that it is probably preferable to teach formulae (3) and (5) in LIS-schools, because they are simpler to explain than the official versions.

It is now not surprising that we consider, in analogy of (3), the expression

$$\left(\sum_{k=1}^{7}k\,x_k\right)\bmod 11 \qquad (6)$$

and wonder if it also yields a good (perhaps even better) method of determining a check digit for an ISSN. Although (6) is also capable of detecting all single errors and all permutations of different symbols (see next section) it is not equivalent to (4) or (5). Indeed, the ISSN of the Journal of Information Science is 01655515, while (6) would give 3 as check digit.

More generally, one could consider formulae such as

$$\left(\sum_{k=1}^{7} \alpha_k \, x_k \right) \bmod 11 \qquad (7)$$

with $\alpha_k \in \mathbb{N}_0$, and with 9 instead of 7 for an ISBN. Even more generally, one could study

$$\left(\sum_{k=1}^{7} \alpha_k \, x_k \right) \bmod d \qquad (8)$$

In the next section we will study (7) and give a proof that this method is (almost) always capable of detecting single errors and permutations of two different symbols. This result will be valid under fairly general conditions, including the cases mentioned earlier, and for other divisors than 11.

In the third section we will consider the natural question: which of these methods is best? Indeed, since we have so many, non-equivalent methods of calculating a check digit, all capable of detecting single errors and the permutation of two different symbols, which method detects the most double errors? Note that no method can find all double errors if we include the check digit. Indeed, a single error in one of the first seven symbols can always be 'corrected' by a compensating error in the check digit. This kind of double error can never be detected. Yet, many double errors in the

first seven symbols can never be detected by one single method (examples follow) although different methods may fail to detect other double errors. Hence, it is interesting to investigate this more deeply. This is done in the third section with 11 as divisor. Here we prove that, although different methods generally detect different double errors, the same number of double errors stays undetected. It is indeed shown that exactly 10% of all double errors stay undetected. Some consequences for the case of three or four mistakes are drawn.

Section four investigates the same issues but now in the case of other divisors. We introduce the important case of division by 13 (here one can use 0,1,..., 9, X, Y, Z as check digits). We show, by computer calculations, that now different methods may perform differently. This contrasts sharply with the standard case of division by 11. Extensive computer calculations give hints for the best methods. This is important in case one wants to have a higher detection rate. These investigations may also be useful when in time the number of digits in the ISBN or the ISSN must increase, as for instance when using DOIs (Digital Object Identifiers) [8].

The paper closes with a fifth section in which we study high values of d (the divisor) in order to find bounds, i.e. minimum values for the number of undetected double errors. We end this section by proposing a number of open questions for further research.

Notation: the symbol □ denotes the end of a proof.

General multilinear framework for detecting single errors and permutations of two different symbols

In this section $x_1,\ldots,x_n$ denote, not necessarily different, numbers taken from the set $\{0,1,\ldots,9\}$. The symbols α_k denote different numbers from the set $\{1,\ldots,p-2\}$, where p is a given prime number larger than or equal to 11.The check digit y for the code $X = (x_1,\ldots,x_n)$ is then obtained as follows:

$$y=\left(\sum_{k=1}^{n}\alpha_k\, x_k\right)\bmod p \qquad (9)$$

We will further represent y by one symbol: a one-digit number or a letter. This means that an n-digit code is finally represented as an (n+1)-digit sequence.

We can say that this (n+1)-digit code is acceptable if

$$\left(\sum_{k=1}^{n+1}\alpha_k\, x_k\right)\bmod p = 0 \qquad (10)$$

with $\alpha_{n+1} = -1$.

We prove the following result for the procedure outlined above.

Theorem 1

Let n be a natural number, $x_i \in \{0,1,\ldots,9\}$ for $i = 1,\ldots,n$ and let p be a prime number, larger than or equal to 11. Assume further that $\alpha_i \in \{1,\ldots,p-2\}$, $i = 1,\ldots,n$. If all α_i are different, then the check digit x_{n+1} for the code characterised by $X = (x_1,\ldots,x_n)$, and calculated as

$$y=\left(\sum_{k=1}^{n}\alpha_k\, x_k\right)\bmod p \qquad (9)$$

is capable of detecting single errors as well as permutations of different symbols.

Proof: Given in Appendix B

Note

By the term 'single error' we mean a replacement of x_i, i=1,...,n+1 by another digit from the set {0,1,...,9}. We assume this also in the proof of the theorem (Appendix B). Yet, x_{n+1} can also be X. However if the check digit must be X and it is not, this is immediately detected by the algorithm. Also, if the check digit is X and it is interchanged with another digit (that can never be X) this is also immediately detected as X can not occur among the first n symbols.

Examples

1. Classical ISBN

   Here n = 9, $\alpha_i = i$, for i = 1,...9 (in the alternative formulation), $\alpha_{10} = -1$, p = 11.

2. Classical ISSN

   Here n = 7, $\alpha_i = i+2$, for i = 1,...7 (in the alternative formulation), $\alpha_8 = -1$, p = 11.

3. Non-equivalent variant for ISSN

   Take n = 7, $\alpha_i = i$, for i = 1,...7, $\alpha_8 = -1$, p = 11, cf.(6)

4. ISBN-like code using p = 13

   Take n = 9, $\alpha_i = i$, for i = 1,...9, $\alpha_{10} = -1$, p = 13.

   The check digit for 056603515 becomes 2 here (it was 4 for the official method).

   As the remainder can now be 11 or 12, we will use Y if $x_{10} = 11$ and Z if $x_{10} = 12$.

   In this way we obtain the acceptable code: 056623515Z

5. ISSN-like code using p = 13

   Take n = 7, $\alpha_i = i+2$, for i = 1,...7, $\alpha_8 = -1$, p = 13.

The check digit for the Journal of Information Science would now be 5 (by coincidence the same as the official one). So, let us take another journal, say the Journal of Documentation. Its official ISSN is 00220418. This alternative would yield 0022041Y (the remainder is 11).

6. ISSN-like code using 13, another non-equivalent variant

   Here we take $n = 7$, $\alpha_i = i$, for $i = 1,...7$, $\alpha_8 = -1$, $p = 13$.

   The check digit for the Journal of Information Science would now be Z (the remainder is 12).

Note that from p larger than 11 on, the numbers α_i can be larger than 10. Note further that we do not require the α_i to be monotone (increasing or decreasing). Theorem 1 stays valid as long as the α_i are different, not equal to zero and smaller than p -1. This leads to an enormous potential for ISBN and ISSN-like codes. It would now be interesting if we could detect those codes that detect the least amount of double errors. The next section is devoted to this for $p = 11$. The general case is studied in the fourth section.

Detecting double errors: the case p = 11

As in section two we assume that the algorithm verifies if

$$\left(\sum_{k=1}^{n+1} \alpha_k x_k \right) \bmod p = 0 \qquad (10)$$

here with p = 11. The α_i, i = 1,...,n are different numbers taken from the set {1,...,9}, $\alpha_{n+1} = -1$. Note that this includes the official ISBN and ISSN algorithms.

Suppose we have a double error, say for i and j, $i \neq j$, $i,j \in \{1,...,n+1\}$. So, instead of (10) we check if

$$\left(\sum_{\substack{k=1 \\ k \neq i,j}}^{n+1} \alpha_k x_k + \alpha_i y_i + \alpha_j y_j \right) \bmod 11 = 0 \qquad (11)$$

where $x_i \neq y_i$ and $x_j \neq y_j$. Now (11) is satisfied if and only if

$$\left(\alpha_i (x_i - y_i) + \alpha_j (x_j - y_j)\right) \bmod 11 = 0 \qquad (12)$$

How many of all double errors stay undetected? One could argue that there are $\binom{n+1}{2}$ possible combinations of two places out of n+1, and that in each place there are nine possible errors leading to $9.9.\binom{n+1}{2}$ double errors. (This would give 2268 cases for the ISSN (n=7) and 3645 cases for the ISBN.) Yet, this reasoning is not correct. Indeed, criterion (12), which is the only necessary and sufficient condition for non-detection of a double error, is dependent on the values of i and j (via α_i and α_j) and on the values x_i, y_i, x_j and y_j. Of course, only the differences x_i-y_i and x_j-y_j matter, but these occur with different frequencies, and hence should be weighted differently. These weights are given in Table 1.

Indeed, x_i-y_i = 9 can only come from x_i = 9 and y_i = 0, while e.g. x_i-y_i = -7 is obtained if x_i = 0 and y_i = 7, or x_i = 1 and y_i = 8, or if x_i = 2 and y_i = 9. The other weights are similarly obtained. All this leads to the following Lemma 1.

Table 1 Weights for differences x_i-y_i or x_j-y_j

difference	weight
±9	1
±8	2
±7	3
±6	4
±5	5
±4	6
±3	7
±2	8
±1	9

Lemma 1

There are 8100 n (n+1) possible double errors

Proof. We have four parameters in (12): i, j $\in$ {1, ...,n}, $x_i - y_i$, $x_j - y_j \in$ {-9,..., -1, 1, ...,9} with weights as described above. Hence this gives a total of

$$(n+1)n.2^2 \sum_{k,l=1}^{9} kl = 8100\,n(n+1) \quad cases. \qquad (13)$$

□

This amounts to 729000 cases for the study of ISBNs (n=9) and 453600 cases for the study of ISSNs (n=7).

We will next answer the question of how many double errors stay undetected. First we need another lemma.

Lemma 2

For any $\alpha \in \{-1,1,...9\}$, fixed, the function

$$v \text{ ---------> } (\alpha\, v) \bmod 11 \qquad (14)$$

with $v \in \{-9,...,-1,1,...9\}$ weighted as in Table 1, is surjective with range $\{0,...,10\}$, attaining each value exactly 9 times.

Proof. The proof proceeds by considering all cases. First we take $\alpha = -1$. Then we have the following values (with weights between accolades): 9{1},8{2},7{3},6{4}, 5{5},4{6}, 3{7},2{8},1{9},-1 = 10 mod 11 {9}, -2 = 9 mod 11 {8}, -3 = 8 mod 11 {7}, -4 = 7 mod 11 {6}, -5 = 6 mod 11 {5}, -6 = 5 mod 11 {4}, -7 = 4 mod 11 {3}, -8 = 3 mod 11 {2}, -9 = 2 mod 11 {1}. This proves that, working mod 11, yields every possible result, each exactly nine times.

We will next check formula (14) for $\alpha = 9$. Again we write weights between brackets. All calculations are mod 11. This gives the following results: 9*(-9) = 7 {1}, 9*(-8) = 5 {2}, 9*(-7) = 3 {3}, 9*(-6) = 1 {4}, 9*(-5) = 10 {5}, 9*(-4) = 8 {6}, 9*(-3) = 6 {7}, 9*(-2) = 4 {8}, 9*(-1) = 2 {9}, 9*1 = 9 {9}, 9*2 = 7 {8}, 9*3 = 5 {7}, 9*4 = 3 {6}, 9*5 = 1 {5}, 9*6 =10 {4}, 9*7 = 8 {3}, 9*8 = 6 {2} and finally 9*9 = 4 {1}. This proves (14) for $\alpha = 9$. We leave the other cases to the reader □.

Theorem 2

The multilinear error-checking method (10) with p = 11 always detects 90% of all double errors. This result is true whatever the precise, but different, values of $\alpha_i \in$

$\{1,,\ldots,9\}$, $i = 1,\ldots,n$, and with $\alpha_{n+1} = -1$. Consequently, $810n(n+1)$ double errors are always undetected.

Proof. We use criterion (12) to check when $\alpha_i(x_i - y_i) + \alpha_j(x_j - y_j)$ is a multiple of 11. By Lemma 2, $\alpha_i(x_i - y_i)$ leads to nine times the values $\{1,\ldots,10\}$. Similarly, $\alpha_j(x_j - y_j)$ gives nine times each value of $\{1,\ldots,10\}$. It is now easy to see that exactly 10% of all sums is a multiple of 11. These errors will stay undetected. By lemma 1, this number is $810\ n\ (n+1)$ □.

Note 1. The above result implies that there is no best method (set of α's) amongst the acceptable ones (cf. Theorem 1). Consequently, we wonder why the ISO did not choose the simplest method (6) as a standard.

Note 2. It is possible to find codes that detect a smaller percentage of double errors. Indeed, take $\alpha_1 = 1$, $\alpha_2 = 11 = 0$, $\alpha_3 = 10$, $\alpha_4 = 9$, $\alpha_5 = 8$, $\alpha_6 = 7$, $\alpha_7 = 6$, $\alpha_8 = -1$. Then we have calculated (by computer) that only 34020 double errors, i.e. 7.5%, stay undetected (for the ISSN), while the official method leaves 10% double errors undetected. The reason for this surprising result is that, because $\alpha_2 = 0$, double errors for which $i = 2$ and $j \neq 2$ are all detected. This decreases the number of undetected double errors. Yet, there is a price to be paid: all single errors occurring at the second digit are undetected! We, therefore, do not consider such methods.

The arguments leading to Lemma 2 and Theorem 2 also yield precise information on the number of undetected triple and quadruple errors.

Theorem 3

Under the conditions of Theorem 2, 9% of all triple errors are undetected

Proof. By Lemma 2 and the argument of Theorem 2 we see that (12) yields 0 mod 11 in 10% of the cases and each other remainder in exactly 9% of the cases. For triple errors we have to check

$$\alpha_i(x_i - y_i) + \alpha_j(x_j - y_j) + \alpha_k(x_k - y_k) \qquad (15)$$

where $\alpha_i,\alpha_j,\alpha_k \in \{-1,1,\ldots,9\}$, $x_i,y_i,x_j,y_j,x_k,y_k \in \{0,\ldots,9\}$, $x_i \neq y_i$, $x_j \neq y_j$, $x_k \neq y_k$; $i,j,k \in \{1,\ldots,n+1\}$. If the first two terms form a multiple of 11 we know, by Lemma 2, that (15) is never an 11-multiple. In each of the ten cases that the first two terms are not an 11-multiple (90% of the cases) we see that (15) is an 11-multiple in 10% of the cases. Hence, we conclude that (15) is an 11-multiple in 9% of the cases □.

Note that more triple errors than double errors are detected. We can also show that 9.1 % of all quadruple errors are undetected (see Appendix C).

Detecting double errors: the general case

In this section we will study the multilinear coding algorithm based on

$$\left(\sum_{k=1}^{n+1} \alpha_k \, x_k \right) \bmod p = 0 \qquad (10)$$

where p is any prime larger than or equal to 11. Recall that $\alpha_1,\ldots,\alpha_n \in \{1,\ldots,p-2\}$, $\alpha_{n+1} = -1$. Suppose we have a double error at $i \neq j$, $i,j \in \{1,\ldots,n+1\}$. As in the beginning of Section 3, we can deduce, via (11), that (12) is the criterion to know

whether a double error is detected, but now with 11 replaced by p. This means we have to check when

$$\left(\alpha_i(x_i - y_i) + \alpha_j(x_j - y_j)\right) \bmod p = 0 \qquad (16)$$

Equation (16) is a generalisation of formula (2.2) in [9]. Note that the weights of Table 1 also apply here (they are independent of the divisor p). Also Lemma 1 applies here: there still are 8100n(n+1) possible double errors. We have shown that if p = 11 there were always (independent – with some mild restrictions- of the choice of the α's) 10%, i.e. 810n(n+1) undetected double errors. This result is not anymore true in general. To illustrate this behaviour, some examples suffice.

We first introduce some notation. We will refer in this section to method I,II and III, by which we will mean the following:

Method I: $\alpha_i = i$ ($i = 1,\ldots,7$), $\alpha_8 = -1$ (this is sometimes called a positional check product [9])

Method II: $\alpha_i = i+2$ ($i = 1,\ldots,7$), $\alpha_8 = -1$

Method III: $\alpha_i = p-9+i$ ($i = 1,\ldots,7$), $\alpha_8 = -1$.

Note that methods II and III represent the official ISSN in case p = 11.

Using a PASCAL program we were able to compute that, for p = 13, method I does not detect 37760 double errors (8.325%), method II does not detect 37772 cases (8.327%) and method III does not detect 37752 double errors (8.323%). Although the percentage differences are small, it is remarkable that the absolute numbers are

different. Recall that for p = 11 we always had exactly 10% of undetected double errors. Anyway, for p = 13, method III performs best.

We repeated the computation for p = 17. Here method I does not detect 27936 cases, method II 27932 cases and method III 27776 cases, i.e. somewhat more than 6%. Here again, all methods differ and method III performs best. However, this is not always the case. For, e.g., p = 97 we computed the following numbers of undetected double errors: 8104 for method I (1.787 %), 4964 for method II (1.094%) and 6584 for method III (1.451 %). Here differences are larger and method II performs best. From our computations we see that method III performs best up to p = 17, for p = 19 and 23, methods II and III perform equally well, and from p = 29 on, method II is best. The simple method I is never best (except of course for p = 11, where all methods perform equally well).

Our intuition is that the overall detecting capacity should increase with increasing p. We found this true as a general trend, but there were some small deviations from monotonicity (e.g. for p = 103). We refer the reader to Appendix D for complete tables of undetected double errors. These tables run through all primes larger than or equal to 11 (the fifth prime) up to a certain limit beyond which no changes occur anymore (see more on this further on in section five). The graphs corresponding to these methods can be found in Figs.1 and 2.

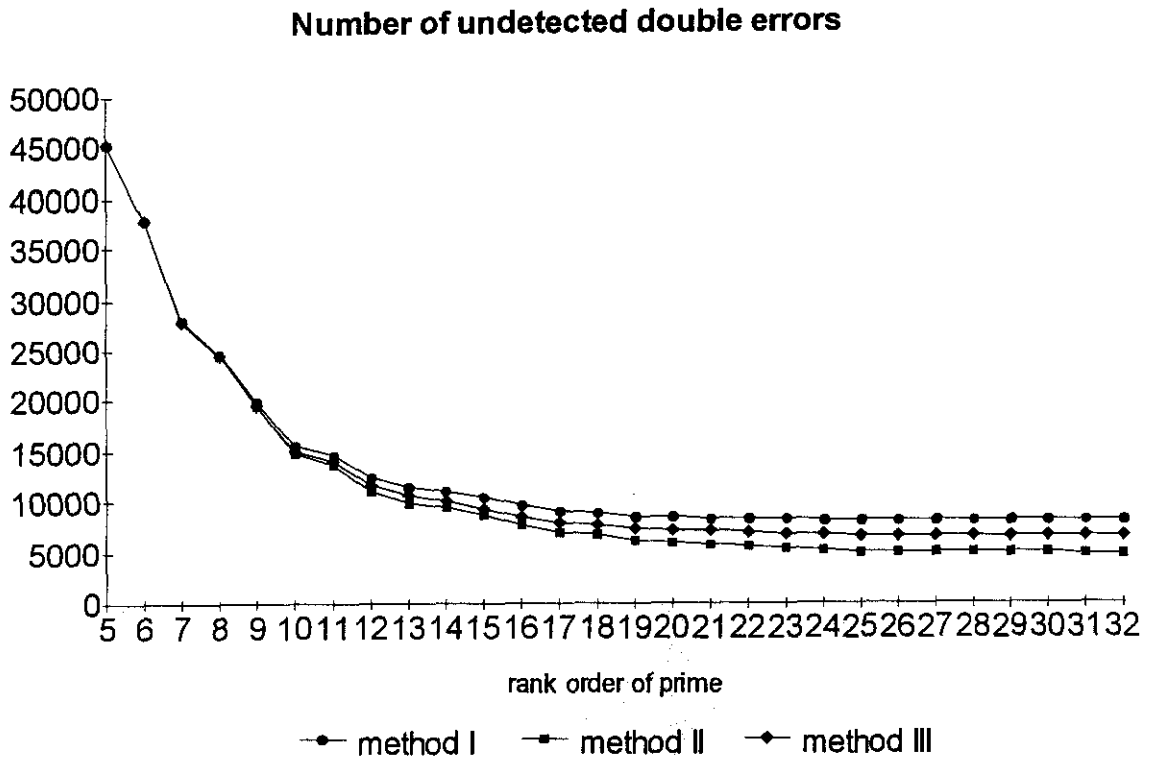

Fig.1 Number of undetected double errors for the three methods, beginning with 11 the fifth prime) and ending with 131 (the 32nd prime)

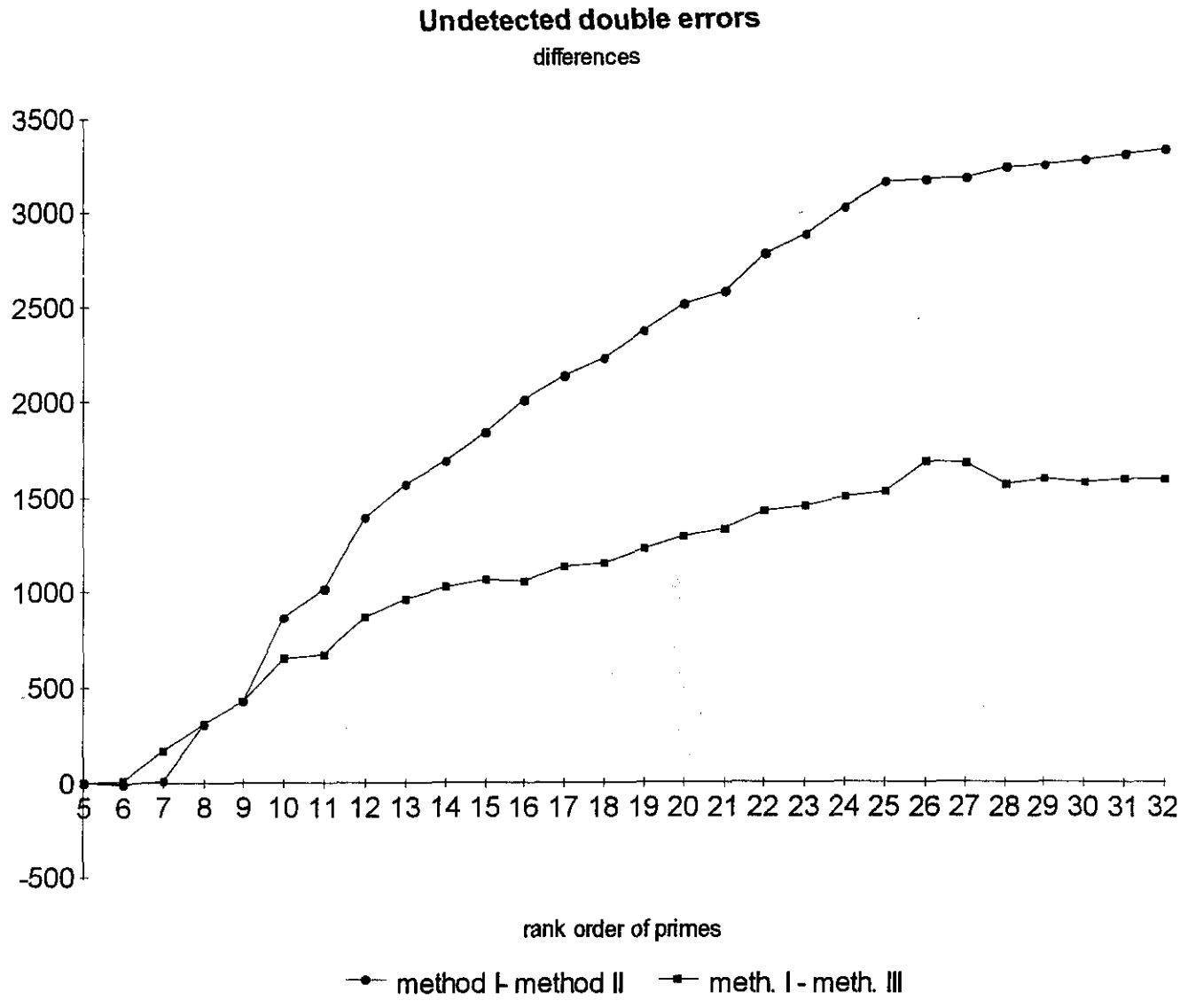

Fig.2 Differences between the number of undetected double errors:
method I – method II and method I – method III

Based on our observations we would advice that if, in the future, one would change the method to compute ISSNs, method III is preferred for p = 13. Note that p = 13 is a natural extension involving only check digits of the form 0,1, ...,9,X,Y,Z. There is, moreover a clear gain (10% - 8.323% = 1.677%) in detected double errors. This

means that an additional 16.77% of the undetected double errors (for p = 11) are now detected (p = 13). Of course, other methods, besides the three presented here, could be tried (in view of Theorem 1). Appendix E shows some results that we have computed for p = 13. It seems (although we can not prove this) that the best value of 37752 undetected double errors (as in method III) can not be improved upon. A similar result was found for p = 17 (Appendix E).

Bounds for undetected double errors and some open problems

One might think that, if the divisor p is taken high enough, the number of undetected double errors can be taken as low as one wishes. This is not so. Indeed, all the cases where

$$\alpha_i(x_i - y_i) + \alpha_j(x_j - y_j) = 0 \qquad (17)$$

for $\alpha_i, \alpha_j \in \{-1,1,\ldots,9\}$, $x_i,y_i,x_j,y_j \in \{0,\ldots,9\}$, $x_i \neq y_i$, $x_j \neq y_j$, $i,j \in \{1,\ldots,n+1\}$ are never detected, whatever the value of p. Note the zero on the right hand side, not zero mod p. Let us examine this for the three methods studied in the previous section (n = 7). In method I the highest possible value for (17) is 7*9 + 6*9 = 117. So from p = 127 on (the first prime large than 117) we have reached the case that (12) can only be a p-multiple if (17) is true. Our computations show that there are 8080 undetected double errors for p = 127, so this is the absolute minimum of undetected double errors for method I. Note also that for this argument to be true the divisor does not have to be a prime. Indeed, our computations confirm that from 118 on the number of undetected double errors, using method I, is 8080. Of course, it may happen (and it usually does) that this number is already observed for a smaller value of the divisor.

Similarly, for method II, we have a maximum value for (17) equal to 9*9 + 8*9 = 153. The minimum number of undetected double errors is here 4776. We can not perform a similar calculation for method III as p is involved in the value of the coefficients α. We can show, however, that the number of undetected errors for method III is the same as the number of undetected errors for (2,3,...,8), i.e. $\alpha_i = i+1$, i=1,...,7 (see appendix F). Hence, the maximum value for (17), using method III is 8*9 + 7*9 = 135. Our computer algorithm yields a limiting value of 6492 undetected double errors. See also the tables in Appendices D and E.

A method to detect all double errors

From the previous reasoning it is clear that the multilinear approach with one check digit will never detect all double errors. Yet, using two check digits, namely one obtained for p = 11 and one for p = 13 (with e.g. method I) finds all double errors, as is readily seen. Note that the method proposed in [9] is unrealistic. It is shown in [10] that a general solution for double error correcting can be obtained using Reed-Solomon codes.

Finally, we end this section by stating some open problems

P1. Formulate a new algorithm, with one check digit (or letter), such that all single and double errors are detected (or prove that this is not possible).

P2. Determine and prove formulae for the number of undetected double errors for p larger than 11 (cf. Theorem 2).

P3. Explain why, for lower p, method III performs best, while, for larger p, method II is better.

P4. Explain why the detecting capacity for double errors is not monotone in p.

P5. Explain the bounds described above. What other 'minimal' values (with respect to a certain method or group of methods) can be determined?

P6. Can the observations made in this article be obtained from more general group-theoretic results?

Conclusion

Coding of the ISBN and ISSN was studied, and alternatives were formulated. A minimum requirement for a useful code is that all single errors as well as all permutations of two symbols must be detectable. The strength of alternative codes, in particular with respect to the detection of double errors was investigated. We gave a complete description of the method based on division by 11 (the ISO norm). In this case we were also able to describe the power of the method with respect to the detection of three or four errors. We have shown that, using division by 11, all coding methods detect the same percentage of errors. In case larger prime numbers are used as divisor numerical experiments were performed showing that different methods have different detection capabilities for double errors. Best methods were experimentally determined. This article illustrates the use of a computer as a heuristic and experimental device as advocated e.g. in [11].

References

[1] ISO (Genève), ISO 3297-1975 (E). Documentation – International standard serial numbering (ISSN). 191-195.

[2] J. McQueen, Record matching: ISBNs and ISSNs, Information Today, (1993), March, 48-49.

[3] ISO (Genève), ISO 2108-1972 (F). Documentation – Système international pour la numérotation des·livres (ISBN).

[4] L. Egghe, A note on two ISBN-checking algorithms. *Journal of Information Science,* 11 (1985) 41-42.

[5] G. McMurdo, An alternative ISBN checking algorithm. *Journal of Information Science*, 3 (1981) 235-237.

[6] J.A.D. MacCormack, Letter to the editor. *Journal of Information Science*, 5, (1982), 172.

[7] R. Hill, *A first course in coding theory.* Oxford Applied Science Series (1986). Oxford: Clarendon Press.

[8].A. Simmonds, The 21st century ISBN, *The Bookseller*, 1997, 5 December, 20-22.

[9] C.K. Chu, A note on multiple error detection in ASCII numeric data communication. *Journal of the ACM*, 28 (1981), 265-269.

[10] D.V. Sarwate, A note on "A note on multiple error detection in ASCII numeric data communication". *Journal of the ACM,* 30 (1983), 33-35.

[11] F. Leimkuhler, On bibliometric modeling. In: *Informetrics 87/88* (L. Egghe & R. Rousseau, eds.) 1988. Amsterdam: Elsevier. 97-104.

Appendix A

In this appendix we prove the equivalence of methods (4) and (5) to calculate the check digit of the standard ISSN. The official method first computes

$$\sum_{k=1}^{7}(11-(k+2))x_k \qquad (4)$$

This expression is equal to 11x + y = 11(x+1) + (y-11), where x $\in \mathbb{N}$ and y $\in$ {0,1, ...,10} is the remainder of the division of (4) by 11. The check digit is 11-y, unless y = 0 in which case the check digit is 0.

The second method calculates

$$\sum_{k=1}^{7}(k+2)x_k \qquad (5)$$

Expression (5) is equal to 11x' + y', where x' $\in \mathbb{N}$ and y' $\in$ {0,1, ...,10} is the remainder of the division of (5) by 11. The check digit is y'.

Summing (4) and (5) clearly yields a multiple of 11, which is equal to 11(x+1) + (y-11) + 11x' + y'. Consequently, y - 11 + y' is also a multiple of 11. However, we also know that:

$$-11 \leq y - 11 + y' < 11$$

This implies that y -11+y' is either equal to –11 or to 0. If y – 11 + y' = -11 then y + y' = 0. This can only happen if y = y' = 0, and in both cases this is the check digit. If y – 11 + y' = 0, then y' = 11 – y. Now, y $\neq$ 0, since y' $\neq$11. Hence, whatever the value of y $\in$ {0,1, ...,10} , both methods yield the same check digit. □

Appendix B

Theorem 1

Let n be a natural number, $x_i \in \{0,1,\ldots,9\}$ for i = 1,...,n and let p be a prime number, larger than or equal to 11. Assume further that $\alpha_i \in \{1,\ldots,p-2\}$, i = 1,...,n. If all α_i are different, then the check digit x_{n+1} for the code characterised by $X = (x_1,\ldots,x_n)$, and calculated as

$$y = \left(\sum_{k=1}^{n} \alpha_k x_k \right) \bmod p \qquad (9)$$

is capable of detecting single errors as well as permutations of different symbols.

Proof. It is easy to see that the algorithm used in formula (9) is equivalent with checking if

$$\sum_{k=1}^{n+1} \alpha_k x_k \qquad (17)$$

with $\alpha_{n+1} = -1$, is a multiple of p. Stated otherwise: the algorithm checks if

$$\left(\sum_{k=1}^{n+1} \alpha_k x_k \right) \bmod p = 0 \qquad (10)$$

Note that, using (10) implies that we also check the check digit! Assume now that a single error has occurred, say at x_j, $j \in \{1,\ldots,n+1\}$. Assume that x_j has been replaced by y_j. This mistake will be detectable if we can prove that

$$\left(\sum_{\substack{k=1 \\ k \neq j}}^{n+1} \alpha_k x_k + \alpha_j y_j \right) \bmod p \neq 0 \qquad (18)$$

By (10) this is equivalent with

$$\alpha_j (x_j - y_j) \bmod p \neq 0 \qquad (19)$$

But $\alpha_j \in \{-1,1,\ldots,p-2\}$ and $x_j - y_j \in \{-9,-8,\ldots -1,1,\ldots,9\}$. Since p is a prime number larger than or equal to 11, $\alpha_j (x_j - y_j)$ can never be prime. Hence (19) is satisfied.

Assume now that two different symbols have been interchanged, say x_j and x_l, $j,l \in \{1,\ldots,n+1\}$. Then this mistake is detected if we can show that

$$\left(\sum_{\substack{k=1 \\ k \neq j,l}}^{n+1} \alpha_k x_k + \alpha_j x_l + \alpha_l x_j \right) \bmod p \neq 0 \qquad (20)$$

Again this is equivalent with showing that

$$(\alpha_j - \alpha_l)(x_j - x_l) \bmod p \neq 0 \qquad (21)$$

Assuming that $\alpha_j > \alpha_l$ (one of the two must be the largest), we know that $\alpha_j - \alpha_l \in \{1,\ldots,p-1\}$ and $x_j - x_l \in \{-9,\ldots,-1,1,\ldots,9\}$. As p is a prime number larger than or equal to 11, (21) is satisfied. This proves the theorem □.

Note that there is nothing special about primes larger than or equal to 11. A similar result can be proved for smaller primes: we only have to restrict x_i to values in the set $\{0,1,\ldots,p-1\}$. Note further that the theoretical background of the above proof is that the finite sets $\mathbf{Z}_p$ (all remainders after division by p) are finite fields if and only if p is prime. The important issue being that in fields the multiplication of two non-zero elements can never be zero ([3], Lemma 3.12 and Theorem 3.5).

Appendix C

Under the conditions of Theorem 2, 9.1% of all quadruple errors stay undetected.

Proof. Now we have a situation where

$$\sum_{j=1}^{4} \alpha_{i_j}(x_{i_j} - y_{i_j}) \qquad (22)$$

must be checked for 11-multiples.

We partially repeat the argument for double or triple errors since we have to know exactly how many times the numbers 0,...,10 (mod 11) appear. If the first two terms yield 0 (mod 11) (10% of the cases) then the first three terms are never a multiple of 11 (Lemma 2). We note here that the numbers 1,...,10 (mod 11) are equally possible. Using again Lemma 2 for the fourth term, we see that in 10% of these cases we have a multiple of 11.

If the first two terms yield 1 (mod 11) (in 9% of the cases), we have an 11-multiple in the first three terms in 10% of the cases, hence never an 11-multiple for (22), by Lemma 2. In the other case we have not an 11-multiple (90% of the cases) but all numbers 1,...,10 (mod 11) are equally possible. Adding the fourth term, using Lemma 2, yields an 11-multiple in 10% of the cases. Exactly the same argument can be given in the case that the first two terms yield 2,...,10 (mod 11).

Hence, the overall conclusion is: we have a multiple of 11 in (22) in 0.1*0.1 + 10*0.09*0.9*0.1 = 0.091 = 9.1% of the cases □.

Appendix D

Numbers of undetected double errors with methods I,II and III, for p prime and larger than or equal to 11, up to a stabilising value of p.

Method I

11	45360	13	37760	17	27936	19	24664
23	19932	29	15636	31	14640	37	12480
41	11488	43	11108	47	10400	53	9604
59	9048	61	8904	67	8552	71	8408
73	8352	79	8240	83	8196	89	8136
97	8104	101	8088	103	8100	107	8080
109	8080	113	8080	127	8080	131	8080

Method II

11	45360	13	37772	17	27932	19	24360
23	19504	29	14772	31	13628	37	11088
41	9920	43	9424	47	8564	53	7596
59	6920	61	6684	67	6192	71	5908
73	5788	79	5480	83	5336	89	5136
97	4964	101	4940	103	4948	107	4872
109	4860	113	4832	127	4808	131	4776
149	4776	151	4776				

Method III

11	45360	13	37752	17	27776	19	24360
23	19504	29	14984	31	13968	37	11620
41	10532	43	10088	47	9340	53	8552
59	7920	61	7760	67	7336	71	7124
73	7028	79	6828	83	6748	89	6648
97	6584	101	6512	103	6532	107	6524
109	6500	113	6516	127	6500	131	6492
149	6492	151	6492				

Appendix E

Numbers of undetected double errors for diverse methods (allowable coefficients according to Theorem 1)

p = 13

α_1	α_2	α_3	α_4	α_5	α_6	α_7	# undetected double errors	
1	2	3	4	5	6	7	37760	
3	4	5	6	7	8	9	37772	
5	6	7	8	9	10	11	37752	best
4	3	5	6	7	8	9	37772	
4	3	6	5	7	8	9	37772	
4	3	6	5	7	9	8	37776	
6	5	7	8	9	1	2	37764	
6	5	7	8	9	10	11	37752	
6	5	7	8	3	1	2	37772	
4	5	7	8	3	1	2	37752	best
4	5	6	7	3	1	2	37760	
4	5	6	7	3	2	1	37760	
2	4	6	7	5	3	1	37760	
2	4	6	9	7	5	3	37752	best
2	3	4	5	6	7	8	37752	best
9	7	5	3	1	2	4	37752	best
12	11	10	9	8	7	6	37760	

The first three cases correspond to methods I, II and III

p = 17

α_1	α_2	α_3	α_4	α_5	α_6	α_7	# undetected double errors	
1	2	3	4	5	6	7	27936	
3	4	5	6	7	8	9	27932	
9	10	11	12	13	14	15	27776	best
2	4	6	9	7	5	3	27776	best
9	7	5	3	1	2	4	27936	
12	11	10	9	8	7	6	28240	
4	5	7	8	3	1	2	27936	
2	3	4	5	6	7	8	27776	best

Appendix F

Proposition

The number of undetected errors using method III is the same as for the method based on $\alpha_i = i+1$, $i=1,\ldots,7$, $\alpha_8 = -1$.

Proof. The mapping

$$J : \{0,...9\}^7 \to \{0,...,9\}^7 : (x_1,...,x_7) \mapsto (x_7,....,x_1)$$

is clearly a bijection. Putting $Y = (y_1,\ldots,y_7) = J(X)$, we will show that for every $X = (x_1,\ldots,x_n)$, $x_i \in \{0,1,\ldots,9\}$ and every p, prime larger than or equal to 11

$$\sum_{i=1}^{7}(p-9+i)x_i = kp$$

$k \in \mathbf{Z}$ (the integers), implies

$$\sum_{j=1}^{7}(1+j)y_j = lp$$

$l \in \mathbf{Z}$, and vice versa.

Indeed, if

$$\sum_{i=1}^{7}(p-9+i)x_i = kp \;\; then also \sum_{i=1}^{7}(i-9)x_i = k'p$$

Putting j = 8-i yields:

$$\sum_{j=1}^{7}(j+1)x_{8-j} = k''p \quad or \sum_{j=1}^{7}(j+1)y_j = lp$$

This proves that there are as many undetected (single, double, triple, ...) errors for method III as for the method based on $\alpha_i = i+1$, $i=1,\ldots,7$, $\alpha_8 = -1$ □.