Secondary Cell Suppression by Gaussian Elimination: An Algorithm Suitable for Handling Issues with Zeros and Singletons

Øyvind Langsrud[0000−0002−1380−4396]

Statistics Norway, P.O. Box 8131 Dep., 0033 Oslo, Norway
Oyvind.Langsrud@ssb.no

This is the author's submitted manuscript version of the work.

This preprint has not undergone peer review or any post-submission improvements or corrections.

The definitive version will be published in the Springer LNCS proceedings of the conference Privacy in Statistical Databases – PSD2024.

See https://crises-deim.urv.cat/psd2024
and https://orcid.org/0000-0002-1380-4396

Secondary Cell Suppression by Gaussian Elimination: An Algorithm Suitable for Handling Issues with Zeros and Singletons

Øyvind Langsrud[0000−0002−1380−4396]

Statistics Norway, P.O. Box 8131 Dep., 0033 Oslo, Norway
Oyvind.Langsrud@ssb.no

Abstract. To protect tabular data through cell suppression, efficient algorithms are essential. Gaussian elimination can be used for secondary cell suppression to prevent exact disclosure. A beneficial feature of this method is that all tables created from the same microdata can be handled simultaneously. This paper presents a solution to the issue where suppressed zeros in frequency tables cannot protect each other. In magnitude tables, it outlines how the algorithm can be tailored to provide protection against singleton contributors using their own data for disclosure.

Keywords: Statistical disclosure control · Confidentiality · Tabular data · Cell suppression · Official statistics.

1 Introduction

Cell suppression is a widely-used statistical disclosure control method for tabular data [3, 4], with several algorithms for secondary suppression implemented in well-known tools [11, 12]. One particular algorithm, Gaussian elimination, is implemented as the function `GaussSuppression` in the R package SSBtools [9]. A practical user interface for this function is provided through another R package, also named GaussSuppression [8]. The method has previously been described briefly in [10], and an application involving more than 50,000 primary suppressed cells can be found in [7]. The present paper describes secondary suppression through Gaussian elimination more thoroughly, detailing how this approach can resolve specific disclosure issues.

This paper utilizes the examples provided in Sect. 2 to describe the underlying theory throughout, while the main algorithm itself is detailed in Sect. 3.

In frequency tables, attribute disclosure can be countered by suppressing all non-structural zeros. In this scenario, however, suppressed zeros cannot protect each other. Sect. 4 explores solutions to this issue.

Table 1. The example frequency table with row and column totals. Symbols indicate suppression: • for primary suppression, ∗ for simple secondary suppression and × for secondary suppression considering zeros.

Age	Iceland	Portugal	Spain	Total
Young	$0^{\bullet}$	$0^{\bullet}$	$5^{\times}$	5
Old	$6^{*\times}$	$3^{*\times}$	$4^{\times}$	13
Total	6	3	9	18

Table 2. The example frequency table with special totals. Total_1 is the overall total excluding Young~Iceland, and Total_2 is the overall total excluding Young~Portugal. Symbols indicate suppression: • for primary suppression, ∗ for ordinary secondary suppression and △ for secondary suppression using the conservative subspace method.

Age	Iceland	Portugal	Spain
Young	$0^{\bullet}$	$0^{\bullet}$	5
Old	6	$3^{*\triangle}$	4
$\text{Total}_1 = 18$		$\text{Total}_2 = 18^{\triangle}$	

Sect. 5 addresses the singleton problem in magnitude tables, aiming to prevent all scenarios where individual contributors can use their own data for disclosure. Finally, Sect. 6 presents some concluding remarks.

2 Illustrative Examples

The frequency table to be used as an example in this paper is given in Table 1. To avoid disclosing the age of the individuals from Iceland and Portugal, the zeros have been primary suppressed. The standard solution from Gaussian elimination and other secondary suppression algorithms is that two cells need to be secondary suppressed. However, this solution does not provide sufficient protection, as it reveals that all five young individuals are from Spain. Since negative frequencies cannot occur, zeros cannot protect each other. Therefore, Table 1 needs four secondary suppressed cells to ensure adequate protection.

As we will explore further, various algorithms can be employed to achieve this result. The small example in Table 1 is not sophisticated enough to highlight the differences between alternative methods of addressing this issue. Therefore, as shown in Table 2, we have made a special modification to the example.

In Table 3, we present a magnitude table example. The values are intended to represent aggregated sales data from different companies across four specific sectors in three different countries. In this example, five cells

Table 3. The example magnitude table. Symbols indicate suppression: • denotes primary suppression, used with A or B to signify a singleton contributor ID. ∗ denotes simple secondary suppression, × denotes secondary suppression with singletons handled by virtual primary suppressed cells, ∘ denotes secondary suppression with singletons handled within Gaussian elimination, # denotes secondary suppression with singletons handled within two parallel Gaussian eliminations, and △ denotes secondary suppression with singletons handled within Gaussian elimination considering combinations.

Sector	Iceland	Portugal	Spain	Total
Agriculture	0.0	$3.9^{\circ\#\triangle}$	$122.3^{\circ\#\triangle}$	126.2
Entertainment	$2.7^{\bullet\mathsf{A}}$	$12.2^{\bullet\mathsf{A}}$	$109.1^{\times\triangle}$	124.0
Governmental	0.0	$64.8^{*\times\circ\triangle}$	$15.5^{\bullet\mathsf{A}}$	$80.3^{\#\triangle}$
Industry	$45.6^{*\times\circ\#\triangle}$	$83.9^{\bullet\mathsf{B}}$	$2.3^{\bullet\mathsf{B}}$	$131.8^{\#\triangle}$
Total	48.3	164.8	249.2	462.3

require primary suppression. These are singleton cells, i.e., each with a single contributor. Among these five primary suppressed cells, the contributions come from just two distinct companies, identified as A and B.

The two zeros in this table result from there being no contributing company. These are considered known zeros, and therefore, these cells cannot be subjected to secondary suppression. Without taking singletons into account, only two cells need to be secondary suppressed, as marked with ∗ in Table 3. However, this solution is insufficient. For instance, by examining only the figures from Spain, it is clear that A and B can reveal each other's values. This table is designed to illustrate various methods and challenges in preventing the disclosure of sensitive information by singleton contributors. As will be discussed below, the method offering the most effective protection requires a total of seven cells to be secondary suppressed.

3 The Main Gaussian Elimination Algorithm

The vector z, representing all cells in the table(s) to be published, can be expressed as

$$z = X^T y \tag{1}$$

where X is a matrix consisting of only 0s and 1s, and y is the vector of all inner cells obtained by crossing all the dimensional variables. Efficient R functionality for creating such X-matrices in connection with hierarchical tables was recently described in [5].

To obtain protection against exact disclosure, we will ensure that the value of any primary suppressed cell cannot be computed as a linear combination of nonsuppressed cells. This means that none of the columns of X

representing primary suppressed cells can depend linearly on the columns of X representing nonsuppressed cells.

There are many methods for calculations dealing with linear dependence. In this context, limiting memory usage is important by ensuring the matrices involved remain sparse, allowing them to be stored compactly. This is one of the reasons to use Gaussian elimination, which, when expressed as matrix factorization, is known as LU decomposition.

As described in the documentation for the Matrix package in R [1], the LU decomposition of X is generally expressed as

$$P_1 X P_2 = LU \tag{2}$$

where P_1 and P_2 are permutation matrices for row and column interchanges, respectively. Here, L and U are lower and upper trapezoidal matrices, respectively. To be precise, we have used the term trapezoidal since triangular is limited to square matrices. To preserve the triangular structure in cases with linearly dependent columns, P_2 should be defined such that the first $\mathsf{rank}(X)$ columns are linearly independent.

Here, we are not primarily interested in the complete decomposition of X, but rather in using the elimination process to achieve secondary suppression. For this purpose, we will employ a specific order (P_2) of the columns, sorting them according to the utility of publishing the cell. The last columns will represent the primary suppressed cells. In Fig. 1, we have illustrated this process in the case of simple secondary suppression of the frequencies in Table 1. The first matrix in this figure represents a version of X where the columns and rows are sorted in a specific order. The columns are arranged by decreasing frequency values. In cases of equal frequency values, marginal cells are prioritized over inner cells.

From left to right, each column is sequentially used as the starting point for elimination. The row selected for elimination is chosen as high up as possible. Each column with an element in this row is changed by subtracting a multiple of the starting column, where the multiple itself can be negative. Besides the column that initiated the process, it is also possible for other columns to be eliminated due to linear dependence. As shown in the figure, column 6 was eliminated during the second round of elimination, a result of the fact that Young~Total can be computed from Total~Total and Old~Total. In this process, we will follow this rule: *Columns representing primary suppressed cells cannot be completely eliminated.* Therefore, we have to skip columns that cannot be used as starting points for elimination. The cells corresponding to these skipped columns become secondary suppressed. In the fifth elimination, we had to skip Old~

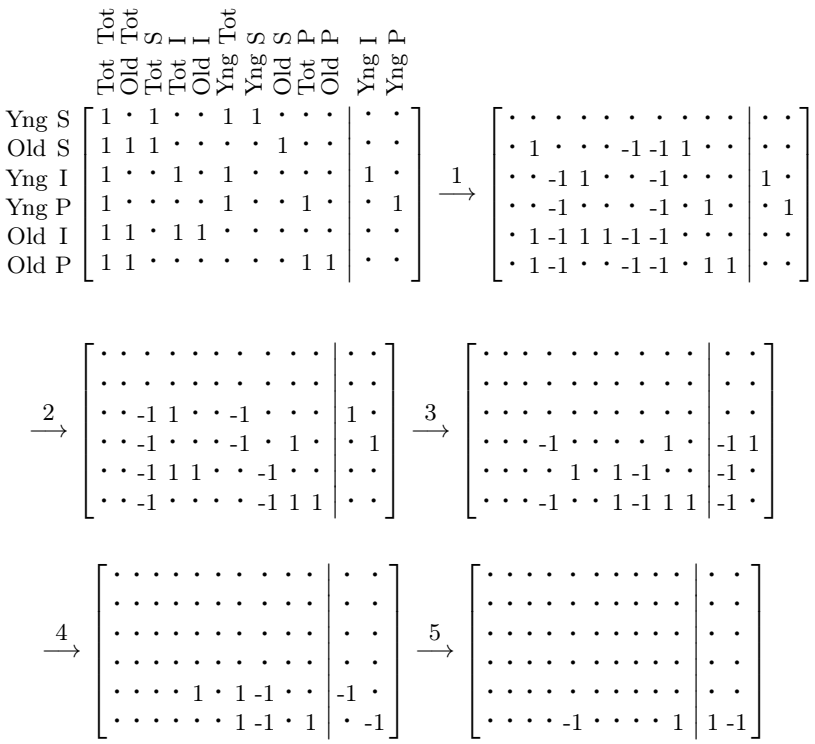

Fig. 1. Gaussian elimination steps leading to simple secondary suppression marked as $*$ in Table 1. The zeros are shown as dots. The columns for primary suppressed cells are to the right of the vertical line. Compared to Table 1, the categories for age and country have been shortened to three and one letter respectively.

Iceland to prevent the elimination of Young~Iceland. The final matrix in Fig. 1 shows two secondary suppressed cells as the result. The deviation from the first matrix can be expressed in LU form as

$$\begin{bmatrix} 1 & \cdot & \cdot & \cdot & \cdot \\ 1 & 1 & \cdot & \cdot & \cdot \\ 1 & \cdot & 1 & \cdot & \cdot \\ 1 & \cdot & 1 & 1 & \cdot \\ 1 & 1 & 1 & \cdot & 1 \\ 1 & 1 & 1 & 1 & 1 \end{bmatrix} \times \left[\begin{array}{cccccccc|cc} 1 & \cdot & 1 & \cdot & \cdot & 1 & 1 & \cdot & \cdot & \cdot & \cdot \\ \cdot & 1 & \cdot & \cdot & \cdot & -1 & -1 & 1 & \cdot & \cdot & \cdot \\ \cdot & \cdot & -1 & 1 & \cdot & \cdot & -1 & \cdot & \cdot & \cdot & 1 & \cdot \\ \cdot & \cdot & \cdot & -1 & \cdot & \cdot & \cdot & \cdot & 1 & \cdot & -1 & 1 \\ \cdot & \cdot & \cdot & \cdot & 1 & \cdot & 1 & -1 & \cdot & \cdot & -1 & \cdot \end{array}\right] \tag{3}$$

In this case, the row order in Fig. 1 was chosen to achieve a triangular structure.

In this example, one row in the X-matrix (Fig. 1) was required for each of the six inner cells (Table 1). When it comes to structural zeros,

which are never involved in suppression, their corresponding rows can be omitted. Below, in Fig. 4, it therefore holds with ten rows in the X-matrix even though there are 12 inner cells in Table 3. In general, we have a nice way to distinguish between non-structural and structural zeros. Rows for the former are included in the X-matrix, but not for the latter.

In the small examples in this paper, the matrices within the elimination process only consist of the numbers -1, 0, and 1. Generally, it will be other numbers. In order to keep integers so that exact calculation is ensured, the columns can be scaled along the way. This is implemented in the `GaussSuppression` function in SSBtools. In the text below, we describe the columns as being made up only of the rows with nonzero elements. This description corresponds to the implementation with sparse column vectors.

4 Handling Zeros in Frequency Tables

4.1 Virtual Cells of Zeros

One way to ensure that the zeros in Table 1 cannot be revealed is by creating a virtual primary suppressed cell that sums the two zero-frequency cells. In Fig. 2, we have introduced such a cell into the elimination process. In this instance, we also changed the row order. This change does not affect the result here, but we will demonstrate in the subsections below how it can be useful. The final matrix in Fig. 2 shows four secondary suppressed cells, as shown in Table 1. In more advanced tables, this method can be challenging because, to be sure, all possible combinations of zeros in y must be taken into account.

4.2 The Subspace Method

In Fig. 2, the rows corresponding to zero frequencies in y (*zero rows*) are arranged at the bottom. Then, no zero rows are eliminated until a column consisting exclusively of zero rows serves as the starting point for the elimination. Assuming that the zero rows are sorted at the bottom, an alternative algorithm that avoids the use of virtual cells is as follows: *Elimination of any of the zero rows is prohibited.* We can see from Fig. 2 that this approach achieves the same result.

However, this is a conservative method that may produce unnecessary secondary cells. Table 2 presents a small toy example that is suitable to illustrate this phenomenon. As shown in Fig. 3, standard Gaussian elimination results in a single secondary suppressed cell. However, if we

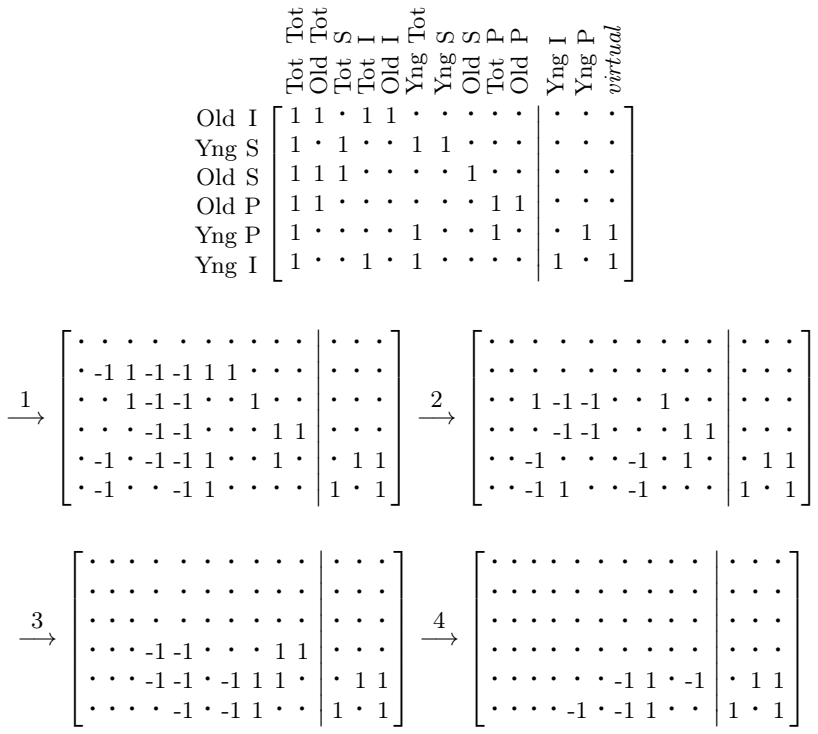

Fig. 2. Gaussian elimination steps with virtual cells leading to secondary suppression marked as × in Table 1.

follow the rule that neither of the two zero rows can be eliminated, then Step 2 in the elimination process shown in Fig. 3 cannot be carried out. Consequently, Total_2 must also be suppressed.

4.3 The Any-Sum Method

The column for Total_2 in the matrix after step 1 in Fig. 3 represents a difference. A difference known to be zero does not mean that the underlying numbers can be revealed. Therefore, we will modify the rule in Sect. 4.2 as follows: *A column consisting solely of elements in zero rows and with the same sign cannot be the starting point for elimination.* One issue is that allowing such a column with differing signs to be used for elimination impacts other columns. Specifically, this affects imaginary columns representing sums of zeros in y. By an imaginary column, we mean a column that is hypothetical to discuss what would have happened if it had been included in the elimination process. After such an elimination,

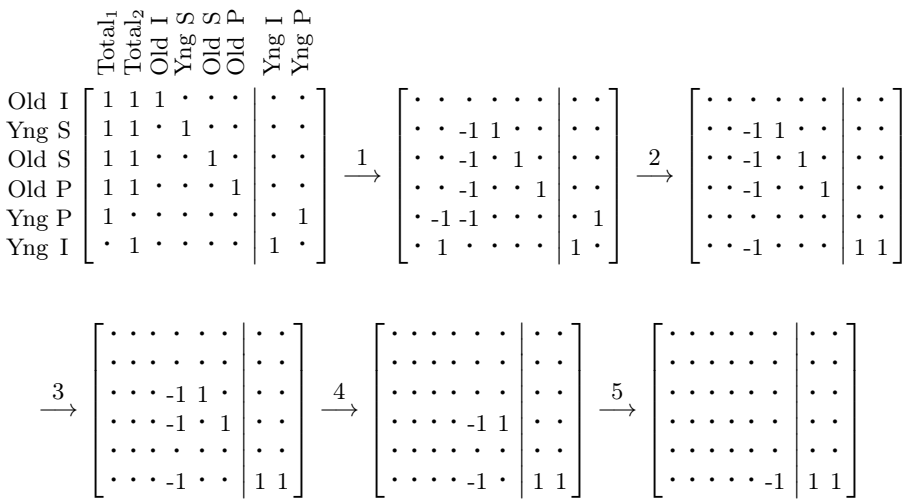

Fig. 3. Gaussian elimination steps leading to ordinary secondary suppression marked as $*$ in Table 2.

such an imaginary column may have different signs, thus the sign rule can no longer be relied upon. To ensure safety, an exception must be introduced: *Elimination based on such a column with differing signs will also not be allowed if it involves rows that have been affected by an elimination step where a zero row is eliminated.* With this rule, we have developed a robust and practical method that effectively addresses the problem of zeros. Specifically, there is no need to construct virtual cells with sums of zero-frequency cells. Regardless of which zero-frequency cells we combine into a virtual cell, the rule described here will handle these without them needing to be included in the process. It is thus a requirement that the zero rows are sorted at the bottom in this algorithm.

5 Handling Singletons in Magnitude Tables

Singleton cells are primary suppressed cells with a single contributor. These cells therefore correspond to specific columns in the X-matrix. Unit contributions also correspond to specific rows in the X-matrix. Thus, some of the columns and some of the rows of X can be labeled with the IDs of the contributors, also known as holding indicators [12]. In general, the same contributor ID may appear on multiple columns and rows. In Fig. 4, the rows are labeled on the right.

	Agr I	Gov I	Tot Tot	Tot S	Tot P	Ind Tot	Agr Tot	Ent Tot	Agr S	Ent S	Gov Tot	Gov P	Tot I	Ind I	Agr P	Ent I	Ent P	Gov S	Ind P	Ind S	
Ind I	·	·	1	·	·	1	·	·	·	·	·	·	1	1	·	·	·	·	·	·	
Agr P	·	·	1	·	1	·	1	·	·	·	·	·	·	·	1	·	·	·	·	·	
Gov P	·	·	1	·	1	·	·	·	·	·	1	1	·	·	·	·	·	·	·	·	
Agr S	·	·	1	1	·	·	1	·	1	·	·	·	·	·	·	·	·	·	·	·	
Ent S	·	·	1	1	·	·	·	1	·	1	·	·	·	·	·	·	·	·	·	·	
Ind P	·	·	1	·	1	1	·	·	·	·	·	·	·	·	·	·	·	·	1	·	B
Ind S	·	·	1	1	·	1	·	·	·	·	·	·	·	·	·	·	·	·	·	1	B
Ent I	·	·	1	·	·	·	·	1	·	·	·	·	1	·	·	1	·	·	·	·	A
Ent P	·	·	1	·	1	·	·	1	·	·	·	·	·	·	·	·	1	·	·	·	A
Gov S	·	·	1	1	·	·	·	·	·	·	1	·	·	·	·	·	·	1	·	·	A

Fig. 4. The initial Gaussian elimination matrix associated with Table 3. In the matrix, the columns for structural zeros are placed furthest to the left. The singleton contributors for rows are labeled on the right. Figures 5 and 6 show some steps of two alternative eliminations.

5.1 Virtual Cells with Singleton

One approach to handling singletons is to create virtual cells, as done in the Modular approach in Tau-Argus [12]. This rule can be formulated as follows: *If there are exactly two primary suppressed cells in a row or column of a subtable, with at least one being a singleton, then a new primary suppressed virtual cell is created, which represents the sum of these two cells.*

Thus, the example in Table 3 involves four virtual cells, resulting in three secondary suppressed cells, as shown in the same table. However, there are some aspects that this approach does not address. Notably, disclosures made via secondary suppressed cells are not taken into account. For example, contributor A can utilize its own values to reveal Entertainment~Spain, which is secondary suppressed, and subsequently Industry~Spain, which is primary suppressed.

5.2 The Within Gaussian Elimination Method

In Sect.3, the rule for Gaussian elimination is that none of the primary suppressed cells can be completely eliminated. Problems can be seen by continuing the algorithm using the primary suppressed columns from a single contributor as starting columns. If another primary suppressed column is then completely eliminated, then we have uncovered that this single contributor can reveal this particular primary cell. To prevent such

disclosure, we now introduce an additional requirement: *The remaining non-zero rows for columns corresponding to primary cells must not originate solely from one contributor.* An essential exception to this rule is needed: *A contributor must be able to reveal cells belonging to themselves.*

Figure 5 illustrates some of the steps in this process. In the seventh elimination step, we must skip, i.e., secondary suppress, the Agriculture~Spain column because using this column could allow A to reveal B's primary cell, Industry~Spain, which is the primary column on the rightmost side. After elimination based on the Agriculture~Spain column seen in the matrix after step 6, the Industry~Spain column would contain only elements from the A rows. Finally, Agriculture~Portugal must also be secondary suppressed for the same reason, as observed after the last step.

With this method, not all disclosures of primary cells are prevented. If the elimination continues after step 7 in the figure, using B's primary cells (the two farthest to the right) as starting points, A's two Entertainment cells will be completely eliminated. Consequently, these cells could be revealed by B. This oversight occurred because one of the B rows was eliminated earlier. Specifically, after the row for Industry~Portugal was eliminated in step 4, the column for Industry~Portugal consisted of rows from both A and B. The method can thus sometimes overlook disclosures due to rows that have been eliminated. To prevent this problem from becoming unnecessarily large, it is important to place the rows for the singleton contributors at the bottom of the X-matrix.

5.3 The Within Parallel Gaussian Eliminations Method

In the general Gaussian algorithm in Sect. 3, different initial sorting of the rows, or alternatively, different choices of elimination rows along the way, will yield the same result. To reduce the problem of overlooked disclosures, we can run two parallel eliminations with differently sorted rows. At each step, we can check for singleton disclosures in both of the parallel eliminations.

As mentioned previously, after the elimination of the Industry~Portugal row, the corresponding column consisted of rows from both A and B. This row was eliminated in step 4, when Industry~Total, as seen in the matrix after step 3 (Fig. 5), was the starting column. Now, imagine that the elimination occurs with the five singleton rows at the bottom sorted in the opposite order. In this scenario, it would be the Entertainment~Iceland row that is eliminated in step 4. However, this would mean that the Entertainment~Iceland column consists solely of two rows belonging to B, enabling B to reveal this primary cell. Therefore, we must skip this

elimination and instead, secondary suppress Industry~Total. As shown in Fig. 6, all eliminations starting with the fourth step are different. The result is five secondary suppressed cells, and now all disclosures of primary cells are avoided.

In general, when two parallel eliminations are conducted, overlooked disclosures are rare, and in practice, the method works very well.

5.4 Considering Combinations

In Sect. 5.3, we addressed the protection of primary cells, but we observe that it is still possible to disclose sums from the same contributor, in this case, the sum of A's two Entertainment cells. One way to prevent this is to introduce virtual cells similar to the descriptions in Sect. 4.1 and 5.1. However, managing all possible sums in this manner can be challenging. Moreover, we may also prevent the disclosure of differences and other linear combinations calculated from cells from the same contributor.

A simple way to prevent many such disclosures is to introduce a new rule: *The column that serves as the starting point for elimination must not consist solely of rows from one or two contributors.* Upon reviewing the matrix after step 5 in Fig. 6, this rule alone suffices to determine that the elimination cannot proceed, as the remaining columns consist only of rows from A and B. Consequently, there are seven secondary suppressed cells.

In the matrix after step 5, the Entertainment~Spain column, comprising two rows from A, shows that elimination using this column would completely eliminate a imaginary cell consisting of the sum of the two A cells. Therefore, Entertainment~Spain must be secondary suppressed to prevent the disclosure of the sum of A's two Entertainment cells.

From the Governmental~Portugal column, we can see that more complex disclosures are prevented. If this column is used to eliminate a imaginary cell consisting of the sum of the two B cells, it becomes evident that A can disclose the sum since, after elimination, the cell would consist solely of rows from A. Additionally, since A's cells have opposite signs, B could reveal the difference between Entertainment~Iceland and Governmental~Spain. Thus, disclosure is mutual. Each contributor can use their own numbers to reveal a linear combination of the other's numbers.

Since this method does not protect primary cells other than singleton cells, it serves as a supplement to the methods described in Sect. 5.2 and 5.3. For this method, disclosures can also be overlooked due to eliminated rows. Thus, parallel eliminations become significantly useful again. In the

example in Table 3, no rows are overlooked by single elimination, and parallel eliminations do not produce any different results.

6 Concluding Remarks

As described in Sect. 3, Gaussian elimination is based on a general X-matrix; there are no requirements for a hierarchical structure or any specific layout. Several tables can be included in the same X-matrix, and therefore, the handling of linked tables is automatically built into the Gaussian elimination algorithm.

Sect. 4 addressed the issue that arises when zeros are primary suppressed. Sometimes, methods are used where zeros are never suppressed, but other small frequencies are primary suppressed. In this scenario, the ones cannot protect each other because frequencies can only be integers. This problem can be resolved in the same way as the problem with zeros. Therefore, the rules in Sect. 4 can be directly translated by replacing "zero rows" with *"one rows"*. In situations where zeros can be secondary suppressed, a more complex method than the one described above is necessary. The method required to handle this is beyond the scope of this paper, but it is implemented in the `GaussSuppression` function in SSBtools. As discussed in Sect. 3, structural zeros can be managed by excluding rows from the X-matrix.

Sect. 5 focused on the singleton problem, and in particular, Sect. 5.3 proposed two parallel eliminations to reduce the issue of eliminated rows. However, there is no guarantee that this will resolve all issues. A possible improvement could be to use more than two parallel eliminations. Another option is to combine the methods described in Sect. 5.3 and 5.4 with the use of virtual cells, as discussed in Sect. 5.1. To prevent unnecessary secondary cells, virtual cells should be treated differently from other primary cells. A virtual cell does not require protection against disclosure by a singleton contributor. By integrating the methods from Sect. 5.1, 5.3 and 5.4, we have developed a robust and efficient routine that prevents almost all scenarios in which a singleton contributor can reveal sensitive numbers exactly. This option is available in the `GaussSuppression` function in SSBtools.

Examples illustrating the methods in Sect. 4 and 5 can be found in the appendix.

To achieve interval protection, Gaussian elimination can be combined with other methods; however, this issue is beyond the scope of this paper.

$$
\cdots \xrightarrow{3}
\left[\begin{array}{cccccccccccccc|ccccc}
\cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\
\cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\
\cdot & \cdot & \cdot & \cdot & \cdot & \cdot & -1 & \cdot & \cdot & \cdot & 1 & 1 & \cdot & \cdot & -1 & \cdot & \cdot & \cdot & \cdot \\
\cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\
\cdot & \cdot & \cdot & \cdot & \cdot & \cdot & -1 & 1 & -1 & 1 & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\
\cdot & \cdot & \cdot & \cdot & \cdot & 1 & -1 & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & -1 & \cdot & \cdot & 1 & \cdot \\
\cdot & \cdot & \cdot & \cdot & \cdot & 1 & -1 & \cdot & -1 & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & 1 \\
\cdot & \cdot & \cdot & \cdot & \cdot & -1 & \cdot & 1 & \cdot & \cdot & \cdot & \cdot & \cdot & -1 & \cdot & 1 & \cdot & \cdot & \cdot \\
\cdot & \cdot & \cdot & \cdot & \cdot & \cdot & -1 & 1 & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & -1 & \cdot & 1 & \cdot & \cdot \\
\cdot & \cdot & \cdot & \cdot & \cdot & \cdot & -1 & \cdot & -1 & \cdot & 1 & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & 1 & \cdot
\end{array}\right]
$$

$$
\cdots \xrightarrow{6}
\left[\begin{array}{cccccccccccccc|ccccc}
\cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\
\cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\
\cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\
\cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\
\cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\
\cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\
\cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & -1 & \cdot & \cdot & \cdot & \cdot & \cdot & 1 & \cdot & \cdot & -1 & 1 \\
\cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & 1 & -1 & \cdot & \cdot & \cdot & -1 & -1 & 1 & \cdot & \cdot & 1 & \cdot \\
\cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & 1 & -1 & \cdot & \cdot & \cdot & \cdot & -1 & \cdot & 1 & \cdot & \cdot & \cdot \\
\cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & -1 & \cdot & \cdot & -1 & \cdot & \cdot & 1 & \cdot & \cdot & 1 & \cdot & \cdot
\end{array}\right]
$$

$$
\xrightarrow{7}
\left[\begin{array}{cccccccccccccc|ccccc}
\cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\
\cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\
\cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\
\cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\
\cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\
\cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\
\cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & -1 & \cdot & \cdot & \cdot & \cdot & \cdot & 1 & \cdot & \cdot & -1 & 1 \\
\cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\
\cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & 1 & \cdot & -1 & 1 & \cdot & -1 & \cdot \\
\cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & -1 & \cdot & \cdot & -1 & \cdot & \cdot & 1 & \cdot & \cdot & 1 & \cdot & \cdot
\end{array}\right]
$$

Fig. 5. Some Gaussian Elimination steps, including the last one, conducted with the matrix in Fig. 4 as a starting point. The treatment of singletons here leads to secondary suppression marked as ∘ in Table 3.

Appendix

Figures 5 and 6 illustrate steps from two alternative eliminations starting from the matrix in Fig. 4. These figures are referred to in Sect. 5.

To illustrate the methods with realistic examples, we consider so-called hypercubes, according to the European Census 2021 [2]. We utilize data of 1,000,000 synthetic individuals based on Norwegian data, with a synthetic numeric variable and approximately 650,000 unique IDs added for use in

$$
\cdots \xrightarrow{4}
\left[\begin{array}{cccccccccccccc|ccccc}
\cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\
\cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\
\cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\
\cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\
\cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & 1 & -1 & 1 & -1 & -1 & \cdot & \cdot & 1 & \cdot & \cdot & \cdot & \cdot \\
\cdot & \cdot & \cdot & \cdot & \cdot & 1 & \cdot & \cdot & \cdot & \cdot & -1 & -1 & \cdot & \cdot & \cdot & \cdot & \cdot & 1 & \cdot \\
\cdot & \cdot & \cdot & \cdot & \cdot & 1 & \cdot & \cdot & -1 & \cdot & -1 & -1 & \cdot & \cdot & 1 & \cdot & \cdot & \cdot & 1 \\
\cdot & \cdot & \cdot & \cdot & \cdot & -1 & \cdot & 1 & \cdot & \cdot & \cdot & \cdot & \cdot & -1 & \cdot & 1 & \cdot & \cdot & \cdot \\
\cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & 1 & \cdot & \cdot & -1 & -1 & \cdot & \cdot & \cdot & \cdot & 1 & \cdot & \cdot \\
\cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & -1 & \cdot & \cdot & -1 & \cdot & \cdot & 1 & \cdot & \cdot & 1 & \cdot
\end{array}\right]
$$

$$
\xrightarrow{5}
\left[\begin{array}{cccccccccccccc|ccccc}
\cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\
\cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\
\cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\
\cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\
\cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\
\cdot & \cdot & \cdot & \cdot & \cdot & 1 & \cdot & \cdot & \cdot & \cdot & -1 & -1 & \cdot & \cdot & \cdot & \cdot & \cdot & 1 & \cdot \\
\cdot & \cdot & \cdot & \cdot & \cdot & 1 & \cdot & \cdot & -1 & \cdot & -1 & -1 & \cdot & \cdot & 1 & \cdot & \cdot & \cdot & 1 \\
\cdot & \cdot & \cdot & \cdot & \cdot & -1 & \cdot & \cdot & 1 & -1 & 1 & 1 & \cdot & -1 & -1 & 1 & \cdot & \cdot & \cdot \\
\cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & 1 & -1 & \cdot & \cdot & \cdot & \cdot & -1 & \cdot & 1 & \cdot & \cdot \\
\cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & -1 & \cdot & \cdot & -1 & \cdot & \cdot & 1 & \cdot & \cdot & 1 & \cdot
\end{array}\right]
$$

$$
\cdots \xrightarrow{7}
\left[\begin{array}{cccccccccccccc|ccccc}
\cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\
\cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\
\cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\
\cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\
\cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\
\cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\
\cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & -1 & \cdot & \cdot & \cdot & \cdot & \cdot & 1 & \cdot & \cdot & \cdot & -1 & 1 \\
\cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\
\cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & 1 & \cdot & -1 & 1 & \cdot & -1 & \cdot \\
\cdot & \cdot & \cdot & \cdot & \cdot & -1 & \cdot & \cdot & -1 & \cdot & 1 & \cdot & \cdot & \cdot & 1 & \cdot & \cdot & 1 & -1 & \cdot
\end{array}\right]
$$

Fig. 6. Some Gaussian Elimination steps, including the last one, conducted with the matrix in Fig. 4 as a starting point. Singletons are handled by two parallel eliminations, but the figure only shows the elimination sequence from one of these. The result is the secondary suppressed cells marked as # in Table 3. This figure also illustrates the singleton method which considers combinations. Then the process stops after step 5 and the secondary cells are those marked with △ in Table 3.

magnitude tables [6]. Tables 4 and 5 provide examples of how the methods described in Sect. 4 and 5 affect the number of secondary suppressed cells. Each of the first three table groups, consisting of four, three, and three linked hierarchical tables, respectively, is analyzed using the GaussSuppression package [8].

Table 4. Synthetic data examples illustrating methods for handling ones, equivalent to methods for handling zeros. Each of the first three hypercube groups, according to the European Census 2021, is considered.

	Group 1	Group 2	Group 3
		Secondary suppressed cells	
No method	552	10443	2167
Virtual cells of ones	560	10671	2285
The subspace method	625	14305	3063
The any-sum method	604	12718	2464
Cells to be published, $\mathsf{ncol}(X)$	11356	60971	40141
Primary suppressed cells	114	2563	2093
Primary suppressed ones	54	1053	726
Inner cells, $\mathsf{nrow}(X)$	2186	20396	43219

Table 5. Synthetic data examples illustrating methods for handling singletons. Each of the first three hypercube groups, according to the European Census 2021, is considered.

				Group 1	Group 2	Group 3
Virtual	Within	Parallel	Combinations		Secondary suppressed cells	
-	-	-	-	516	10150	1985
X	-	-	-	576	11394	2238
-	X	-	-	588	12378	2410
X	X	-	-	588	12401	2416
-	X	X	-	588	12501	2451
X	X	X	-	588	12514	2451
-	X	-	X	588	12378	2410
X	X	-	X	588	12418	2416
-	X	X	X	588	12513	2451
X	X	X	X	588	12514	2451
Cells to be published, $\mathsf{ncol}(X)$				11356	60971	40141
Primary suppressed cells				96	1877	1503
Primary suppressed cells that are singletons				54	1053	726
Inner cells, $\mathsf{nrow}(X)$				2186	20396	43219
Unique singleton contributors to inner cells				31	1693	6838

In Table 4, illustrating the methods in Sect. 4, we chose to primary suppress ones, twos, and threes, while zeros remained unprotected and were considered structural. Thus, these methods address the suppression of ones, which, as described in Sect. 6, is analogous to handling zeros. This approach results in similar examples for frequency and magnitude tables.

The primary suppression of the magnitude tables in Table 5 is based on the p%-rule [12], with $p = 5$. Note that in the GaussSuppression package, virtual cells are created by examining the X-matrix. This may differ somewhat from the Tau-Argus method described in Sect. 5.1. The beginning of Sect. 5 mentioned that both rows and columns of the X-matrix can be labeled with singleton contributor IDs. Generally, there may be IDs on rows without a matching column. Extra primary cells representing these *missing* IDs may be added so that all singleton contributions are considered sensitive. This is included in the singleton methods used in Table 5.

References

1. Bates, D., Maechler, M., Jagan, M.: Matrix: Sparse and Dense Matrix Classes and Methods (2024), https://CRAN.R-project.org/package=Matrix, r package version 1.7-0
2. European Commission: Commission regulation (eu) 2017/712 of 20 april 2017 on statistical data and metadata for population and housing censuses. Official Journal of the European Union (2017), https://eur-lex.europa.eu/eli/reg/2017/712/oj
3. Fischetti, M., Salazar, J.J.: Solving the cell suppression problem on tabular data with linear constraints. Management Science **47**(7), 1008–1027 (2001), http://www.jstor.org/stable/822485
4. Hundepool, A., Domingo-Ferrer, J., Franconi, L., Giessing, S., Nordholt, E.S., Spicer, K., de Wolf, P.P.: Statistical Disclosure Control. John Wiley & Sons, Ltd (2012). https://doi.org/10.1002/9781118348239.ch1
5. Langsrud, Ø.: Sparse model matrices for multidimensional hierarchical aggregation. The R Journal **15**, 150–166 (2023). https://doi.org/10.32614/RJ-2023-088
6. Langsrud, Ø.: About the Norwegian Hypercubes for the 2021 Census. https://github.com/statisticsnorway/sdc-census-2021-hypercubes (2024)
7. Langsrud, Ø., Bøvelstad, H.M.: Synthetic decimal numbers as a flexible tool for suppression of post-published tabular data. In: Domingo-Ferrer, J., Laurent, M. (eds.) Privacy in Statistical Databases. pp. 105–115. Springer International Publishing, Cham (2022)
8. Langsrud, Ø., Lupp, D.: GaussSuppression: Tabular Data Suppression using Gaussian Elimination (2024), https://CRAN.R-project.org/package=GaussSuppression, r package version 0.8.5
9. Langsrud, Ø., Lupp, D.: SSBtools: Statistics Norway's Miscellaneous Tools (2024), https://CRAN.R-project.org/package=SSBtools, r package version 1.5.2
10. Lupp, D. P. and Langsrud, Ø.: Suppression of directly-disclosive cells in frequency tables (2021), Joint UNECE/Eurostat Expert Meeting on Statistical Data Confidentiality, 1–3 December 2021, Poznań, Poland
11. Meindl, B.: sdcTable: Methods for Statistical Disclosure Control in Tabular Data (2023), https://CRAN.R-project.org/package=sdcTable, r package version 0.32.6
12. de Wolf, P.P., Hundepool, A., Giessing, S., Salazar, J.J., Castro, J.: tau-ARGUS user's manual, version 4.1. Tech. rep., Statistics Netherlands (2014), https://github.com/sdcTools/tauargus