

Topic (ii): software and computing developments

## **STRUCTURE OF THE ACS AUTOMATED CELL SUPPRESSION SYSTEM**

Submitted by Gordon Sande<sup>1</sup>

### **Invited paper**

#### **I. INTRODUCTION**

1. The ACS Automated Cell Suppression system provides a comprehensive suite of software for dealing with the cell suppression problem used by statistical agencies to protect the confidentiality of business statistics. Cell suppression is one of many stages in the production of business statistics. It is a very distinctive stage as it addresses the competing requirements that the data collected from the respondents must be held confidential while also producing a useful publication for the end users all within a reasonable balance of various costs and delays. An automated system provides both a theoretical prescription of what has to be done as well as a practical recipe for doing what has to be done. The theoretical prescription involves the choice of analytical model and the mathematical framework for dealing with the model. The practical recipe involves the dealing with the other steps in the production of the statistical product as well as the many untidy aspects of the practical problem. These untidy aspects of the problem are various pragmatic issues such as the desire for historical continuity of suppression patterns for periodic publications or the need to accommodate subject matter concerns into the design of a suppression pattern.

2. The ACS system operates as a free standing system that is provided micro data by and returns cell status information to the surrounding statistical production system. The data descriptions required are sufficiently commonplace that they may be either automatically supplied by some systems or readily manually adapted from existing data descriptions. The analytical nature of the cell suppression process is rather different from other statistical production processes, so that a tight integration of a cell suppression system into the rest of the process would pose many problems. However the interaction of the cell suppression system should not disrupt that process. The structure of the ACS system follows the form of the problem which it is addressing. The structure is often most easily understood by first describing the problem and then describing the solution used by ACS.

---

<sup>1</sup> Gordon Sande is a consultant in Statistical Data Protection with Sande & Associates, United States.

## II. SYSTEM STRUCTURE

3. The analytical content of the cell suppression problem is quite distinctive and separate from the other processing steps in a business survey. This has had the practical effect of leaving cell suppression out of the software systems that supported the processing of the typical business survey. The concern of the survey managers for confidentiality protection was not left out of the survey operations, it was just not supported as an integral part of the survey automation. The cell suppression operations were often manual, so that the survey automation had to provide some support for the manual operations as well as acting on the results of the manual processing. The automated cell suppression systems such as ACS act to replace the manual procedures. There is also a design principal that technologies should not be mixed unless there are good reasons for doing so. The combination of history and design separation lead to cell suppression systems as stand alone systems.

4. The input will be the microdata which is used for the other processing and the necessary metadata needed to define the microdata. Cell suppression would appear to process tabular data and the need for microdata is not immediately apparent. If the survey automation could produce the required sensitivity measures for all appropriate aggregations, then there would be no requirement for the microdata. The problem arises both in the evaluation of the sensitivity measures and in the defining of all appropriate aggregations. The sensitivity measures requires the identification of the common ownership of the respondents contributing to all defined aggregations. The typical aggregation rollup does not preserve this information and would require additional explicit processing to preserve it. In some publications, there may be an other, or some similar notion, category which is not displayed. A cell suppression system would be concerned with all the defined cells, including the ones which are not displayed. Unfortunately, these not displayed cells may be the ones which require more care in their protection. The cell suppression system will also require aggregations which do not naturally appear in the publication. The precise definition of which ones are required may even be data dependent. These analytical requirements combine to suggest that the cell suppression system should be left to form any and all aggregates which it finds necessary. The presence of yet another tabulation system would appear to be unnecessary redundancy. The systems normally labeled as tabulation systems are usually convenient combinations of data management, data retrieval, data selection, data aggregation and aggregation presentation operating within a single command structure. Data selection and data aggregation are the simpler operations in this composite. A cell suppression system requires a simpler data selection and a more elaborate data aggregation capability than is provided by a tabulation system. The more difficult tasks of data management, data retrieval, data selection and aggregation presentation are not required activities of a cell suppression system and would be left to the survey automation. The design of a cell suppression pattern requires a coordinated processing of all of the intended publication. This is often said to be an offline problem. A single business survey may present its output in small pieces as the combined output may be rather bulky and have few users who are interested in the entire combined output. The processing of the small pieces could be organized as several online tasks for many sensible reasons. Independent cell suppression of each of the several tasks would not provide confidentiality protection as there would be no coordination of the separate cell suppression patterns. The online tasks could extract the needed control information from the output of a previously completed offline cell suppression pattern. The design principal of separation of technologies, the analytical requirements of the aggregations and the offline nature of the problem all suggest that keeping cell suppression as a separate activity is a sound decision.

5. These design requirements can be translated into operational requirements. We will want to process all of the data at an early stage in the overall survey processing. We will be interested in all the cells which are defined, but will need to know which are intended to be displayed in the survey publication. In practice, we may wish to do experimental processing on nearly final files and then do

final processing on final files with short deadlines. The large files and the need to have scripted final production indicate batch processing with textual interfaces rather than interactive processing with graphical interfaces. The textual interface needs to be both unambiguous and readily understood by both the system and its operators.

6. The ACS system has five components. Its interface with the surrounding survey automation involve two points of contact and three sets of files. The confidential data is the input represented in two files. One input file is the microdata file which will be repeatedly passed from the survey automation to the cell suppression system for each processing cycle. Some of the processing cycles may be experimental as they are of nearly final files only. The microdata file will probably have a simple dedicated conversion utility associated with it. In most cases it will be nothing more than a database query and a file write in a general purpose processing package, and would hardly seem to warrant being called a dedicated utility. The other input file will be a metadata files describing the structure of the microdata file. This file will be prepared once. The information content will be similar to other metadata files but the format will almost surely be different. This file will probably be prepared manually by textual modification to some other metadata file. The protected data output is represented by one file. Again, there will probably be a simple dedicated conversion utility associated with it. Some version of this utility may already exist to deal with the results of older manual procedures.

7. There are three processing components of ACS are associated with the aggregation of the microdata, the determination of cell suppression patterns and the auditing of cell suppression patterns. The aggregation, ACSTabul, and cell suppression, ACSuprs, components work with collections of tables with all values present and various flags and data items to indicate which cells are sensitive and what their publication status is. These tables are only for internal consumption as they contain much sensitive information. ACSTabul reads microdata and outputs a collection of tables. ACSSuprs read a collection of tables and outputs a collection of tables with revisions to the status flags. The auditing, ACSAudit, component works with collections of tables in which withheld cells have no values given. These tables could potentially be released as publications. The threshold needed to distinguish between disclosures and permitted releases may optionally be present, in which case the tables are only for internal consumption. ACSAudit reads collections of tables with status flags and outputs a collections of tables with revised status flags and data values, as well as audit reports.

8. There are two utility components. One, ACSExtr, is used to convert released spreadsheets into collections of tables for auditing. This utility would not normally be used in a production data release environment, but rather only during some forms of testing of otherwise released publications. The other utility, ACSUtil, supports the various miscellaneous actions that may be applied to a collection of tables. ACSUtil reads collections of tables and outputs collections of tables with revised status flags and data values which may sometimes be in other formats. The utility actions include the recording of user specifications of which values are to be displayed, which are republished, which may be withheld

### **III. SOFTWARE ENGINEERING**

9. Cell suppression is an offline problem for which textual interfaces appear natural. Cell suppression systems will operate in conjunction with various survey automation systems. The interfaces should be devoid of unnecessary difficulties including that the input should be unambiguous to both the program and its users. Free format small specialized languages have now become a readily available technology. The ACS system uses such an approach. It applies it to both the user controls and to its internal communication files.

10. For the system, there is no distinction between user control input and communication file input. For the user, there is a very practical distinction that the user control input is listed in the output but

communication file input is only indicated by the name of the file that it is read from. (There are override options to request listing of all inputs should the user find that appropriate.) The include command of many programming languages is an example of this mechanism. The ACS system also names its command include.

11. The communication files are human legible, readily transported between computing platforms, self describing and easily implemented as a result of this simple implementation strategy. Simple implementation also benefits the quality of the system. Each of the components reads both commands unique to it and the common commands found in the communication files. The style of the commands is modeled on the TPL commands. TPL provides a simple clear model although the needs of the ACS analysis cause enough deviations to make the relationship one of imitation only.

12. Commands may extend over multiple input lines with a simple transparent continuation convention. Multiple commands may be packed into a single line by use of a semicolon command separator. The system generated communication files are formatted with one command per line with continuation only used when necessary to limit the line length. The communication files contain both metadata describing a collection of tables including simple formatting information for the tables and the data contents of the tables. A single data item may play differing roles in different tables. The metadata serves to both establish the identifying information for each data item, its name within the system, as well as the structure of the data, or how the various data items are related to each other. These relationships define the structure of the tables.

13. The data is classified by a number, one to seven are currently permitted, of classification variables. In business statistics these would be variables such as Standard Industrial Classification (SIC) codes or Geographical codes. The typical SIC code describes industrial divisions, industry groups, major industries and industries. It is a hierarchical code. The various levels of hierarchy are all within a single classification variable. The terminology in practical use is often quite badly confused and it would be easy to confuse the notion of hierarchy level with the notion of classification variable. A particularly awkward confusion arises when only part of a hierarchical name is specified.

14. Within an SIC there could be both an other industry group, of wildly disparate industrial activities, as well as an other industry, of only vaguely similar industrial activities, within some major industry. A discussion of other, without further context, quickly leads to confusion. The ACS command structure for this situation is intended to avoid this confusion. First the name of the classification variable must be given, and then all the possible values of the classification variable. The second use of other would be declared as an error and no processing would occur. The resolution of this silly example would be to have other group and other industry as two distinct names. In practice, SIC codes tend to be purely numerical with the number of digits indicating the hierarchical depth.

15. The text would be only descriptive and not used for classification. The requirement for unique names within a classification variable is common good practice. The requirement to list all the names can appear a burden the first time the metadata is prepared if such lists are not already available. In real situations such lists are available, but they can be a nuisance in some demonstration situations. The list becomes part of the communication files and need never be prepared a second time.

16. The list of names establishes the set of legal names. We must then specify the structure which relates the names as well as the name of the total. All of this takes the form

```
var SIC
#
code 0
code 10
code 11
```

```

code 12
code 17
code 20
code 21
code 24
code 28
#
aggr 0 = 10, 20
aggr 10 = 11, 12, 17
aggr 20 = 21, 24, 28
#
total 0

```

for a component of a purely numerical SIC. Not all codes are tidy hierarchical codes. Historical continuity across revisions, the pragmatics of real geography and other operational reasons may lead to nonhierarchical coding structures.

17. If we had a historical continuity requirement (implausible but we need an example!) for an aggregation of 11, 12 and 24 we would define a new aggregation hist by

```

code hist
aggr hist = 11, 12, 24

```

except that we have implicitly defined an aggregation of 17, 21 and 28. The ACS analyzer would grumble about the incomplete coding structure.

18. We would also require

```

code nonhist
aggr 0* = hist, nonhist
aggr nonhist = 17, 21, 28

```

where we have now told the ACS system that we have two definitions of the code 0. We would say that there is a single code 0 with two aggregation structures of 0 and 0\*.

19. The two definitions must ultimately be the same but the internal groupings can be different. This is a complete logical structure needed for the cell suppression analysis but which we would never exhibit for publication purposes. It also provides an example of a nonhierarchical coding structure. Much more common examples arise in geography with metropolitan areas being quite common. Groupings of countries by continent and trading affiliation, such as OPEC, leads to nonhierarchical coding structures.

20. The other classification variables are given similar descriptions and follow one behind the other. A code with more alternative definitions would be indicated by having additional \*s. By their position, we can refer to the first, second or whatever classification variable. The names within the variables may overlap, and popular words like total can even be used. The maximum number of classification variables has been set as seven, although that can be readily changed. The maximum is an extent of a data structure and can be easily changed, as it is just a source parameter, to any other number. Seven was presumed to be so large as to represent infinity but the first large scale trial with ACS required five variables with examples having six classification variables known. When there are a large number of classification variables, they tend to be very simple as there is not enough data to fill all the cells that a complex multiple classification variable problem would define.

21. The convenient order for the classification variables may not be the convenient ordering of the variable for printing of tables. A usage command allows the printing to be controlled. Thus we have

```
usage heading Var2 stub Var3 by Var1
```

to indicate the role of the variables in organizing the tables and

```
table 'story' heading total stub top by 0
```

to indicate the various aggregates to be used for the various tables. There is a default sequencing of tables, with no heading text, which the system will provide.

22. The user can reorder the tables and provide heading text to increase legibility. Stub lengths, column counts and column widths can be specified for all tables to provide minor control over the table layouts. The basic printing layout is fixed and directed at the logical structure of the table. Publication quality presentation is the task of other systems. The intent within ACS is to provide a working table which is good enough to be useful but not to distract from the main task of designing a suppression pattern. This is sometimes called the “coffee cup test” as the output will be good enough to sometimes have coffee spilled on it as a working document and not just put on the shelf as useless output.

23. The major content of the communication file will be the table cell contents. We use

```
( total, xxx ) v = 32068, f = p
```

to indicate that the cell for total of the first classification variable and xxx of the second classification variable in a two classification variable problem has the value of 32068. Other attributes of the cell would be indicated by other keyword values. In this case the cell is flagged as prepublished. If the first keyword is v, it may be omitted as it is often the only keyword.

24. A missing value would be indicated by the absence of the keyword or by a value of x. A cell may play differing roles in differing tables as it may be either a table total, a marginal entry or an internal entry in a various tables when there is hierarchy in the classification variables. In each of these roles, the cell will have the same value and attributes. The differing roles are really just a presentation issue and do not effect the definition of the cell which need only be specified once.

25. A central concern of cell suppression are the various aggregates that can be defined. Many aggregates are defined by the structure of the classification variables and are themselves cells. Some aggregates are not so tidy. These miscellaneous aggregates are just lists of cells. We use

```
{ ( sam, xxx ), ( joe, xxx ) } v = 14028, ut = 15033
```

to form an aggregate of two cells, (sam,xxx) and (joe,xxx), which may then have attributes like any other cell. Here the list has a value and an upper tolerance to indicate that this miscellaneous aggregation is sensitive.

26. The remaining content of the communication files is free form text commentary. This will be a mixture of documentation supplied by the user and processing time stamps added by the system. The initial user content would be

```
text Final(?) test run, only chemicals are still being edited!  
;
```

Each time a file is processed from input to output, the system component doing the processing will add to the text by inserting its name, the time and the names of all files used. This processing history can be used to help sort out confusions that may arise as to what processing has been applied to a file. Of

course such processing blunders will never arise and this is just a redundant facility although it is proven to be useful upon occasion.

#### IV. TABULATION COMPONENT

27. The first component that will process the data will be the tabulation component ACSTabul. This component is distinctive in requiring both a control file and a data file as input. The data file is much more interesting. There will be a field for each of the classification variables. There will also be the data. For an example such as a census of manufacturing these fields will be the SIC code, a geographic code and the value of the manufacturing activity. We assume that the businesses will be happy to have their existence known, which is why businesses advertise, so the values of the classification variables are assumed to be known. The value of the manufacturing is confidential to the business, but we assume that interested observers have some amount of prior knowledge about the value.

28. This is the same collection of fields as we would expect to find in demographic data, such as a population census. Economic data has an additional field which makes the data much more elaborate. That field is the enterprise identification field. The terminology which we will use is that business activity is carried out by establishments which are the store fronts and factories which we commonly see. However many establishments will carry the same name as they are just pieces of larger enterprises. The same enterprise can have many establishments and we must treat the contribution of the enterprise to any cell as the item which we are seeking to protect. The possible common ownership of many establishments by a single enterprise makes economic data much more elaborate than demographic data. The enterprise identification field will not appear directly in the publication although its influence will dominate the cell suppression activity.

29. The purpose of the publication is to provide aggregations of the data. The forming of aggregates for publication is a fairly easy data processing application. If the publication is so detailed that there are no respondents in the cell, then the resulting value of zero is publishable and all the interested observers will know that there are no respondents so no new information is provided. When there is exactly one establishment, the value will be confidential and we must withhold the value. The same will be true if there are several establishments but they are all part of the same enterprise. If a cell has only two establishments, from separate enterprises, then the cell value would allow each establishment to calculate the value of the other, and we must withhold the cell value.

30. The situation is the same if there are many establishments but they are parts of only two enterprises. If the cell has two enterprises where one is large and the other is small, the interested observers would be able to make informed estimates of the larger enterprise by subtracting their estimate of the smaller enterprise from the cell value, if it were to be published. The situation of one large enterprise and two small enterprises is basically the same and not included in the situation of two enterprises. This simple analysis has used both the enterprise structure and the availability of prior knowledge to discover situations in which we must withhold the cell value. This type of analysis led to the notion of a concentration rule to determine if a cell were sensitive. As the analysis was improved, this was extended to subadditive sensitivity measures. The most basic property that we ask of a sensitivity measure is that if we pool two nonsensitive cells, the result should also be nonsensitive.

31. The mathematical terminology for this property is that the measures be subadditive. This includes an assumption that we have arranged the measures so that a cell is sensitive when a measure has a positive value and is nonsensitive when a measure has a negative value. We could have taken the negative of the measures, but that would lead us to different terminology which would not appear to be quite as natural. It is also convenient if we arrange the scaling so that comparisons between differing measures make some sense of the notion of more sensitive.

32. It is convenient if all the measures would have the same value for the extreme case of a cell composed of many small enterprises. The convenient value is the negative of the cell value. To use this machinery we need the list of enterprises in a cell sorted by the size of the contributions of the enterprises to the cell. The value of the sensitivity measure is determined by applying weights to this list of contributions. The forms are quite simple.

33. The concentration rule which classifies a cell as sensitive if the largest  $n$  enterprises contribute more than  $k$  percent of the total, a so called  $n$ - $k$ % rule, has a coefficient of  $(100-k)/k$  to the largest  $n$  enterprises and a coefficient of  $-1$  to the remaining enterprises. Concentration rules are only of historical interest as they were the first subadditive rules. They do not directly account for the prior knowledge of the interested observers. Analysis demonstrates that concentration rules can be viewed as corresponding to a range of prior levels of knowledge depending upon the internal configuration of the sizes of the largest contributors.

34. This analysis also leads to  $c$  times improvement rules, or  $p/q$  rules where  $p/q$  is the improvement ratio, which have a coefficient of  $c$  for the largest enterprise,  $0$  for the second largest enterprise and  $-1$  for the remaining enterprises. The analysis also shows that such rules are also subadditive. Other rules can also be proposed. A minor variant on the  $c$  times improvement rules is to have the coefficient of the third largest enterprise be  $-1/2$ . This would be called a linear sensitivity rule with user specified coefficients. The ACSTabul component permits concentration rules,  $c$  times improvement rules and linear sensitivity rules.

35. The technical subtlety in these rules is that they are based on the ordered sizes of the enterprises while the data is for the establishments. Regular tabulation activities determine the cell values and can use the notion of rolling up the cell values to determine the value of an aggregate from its components. If we were to view a cell as being a vector of values, indexed by the enterprises, then we could use the rolling up notion as well. But most enterprises do not contribute to most cells, so we would be mostly rolling up zeros and in practice would find the process inefficient in terms of intermediate storage.

36. It would also not provide the sorted enterprise values, which can change markedly under aggregation. There are examples of firms that are the largest enterprise nationally but are not the largest enterprise in any of the many regions in which they operate. The query processes for multiple key retrievals, with the  $k$ - $d$  tree methods providing an effective balance of efficiency and storage overhead, allow us to construct the list of establishments for each cell directly. The establishments can be combined into enterprises and the enterprise values can be sorted. The cost of the tabulation process can be attributed to the reading of the data and checking that it is coded according to its specifications, the tabulation and the output of the results.

37. The input and output could be made to operate more quickly by using native machine specific formats, the usual binary representations, that are not human legible for minor savings in the absolute amount of time used. The tabulation process, even when based on retrieval of each cell, operates in comparable amounts of time. This would not be true for retrieval methods, such as selection on the full file, which are not as efficient as the  $k$ - $d$  tree methods.

38. The tabulation process also has to look at application of its output and be more than just a cell by cell tabulation. The very simple case of a table row with two single enterprise cells illustrates the problem. If the two cells were of similar size they might each be used as complementary suppressions for the other so they might be the only cells suppressed in the table row. It would be natural for any interested observer to subtract all the published values in the row from the row total and obtain the total for the two cells. Some forms of analysis would suggest that this should be done.

39. This miscellaneous total is an aggregate that is not one naturally suggested by the aggregation structure of the publication. There would now be an aggregate with a known total and either one, if they were the same, or two, if they were separate, enterprises contributing to the value. Under any reasonable sensitivity measure this would be a sensitive aggregate. The extension is to examine the pooling of all the sensitive cells contained in any of the natural aggregations. This pool may require care in its protection as some of the nonsensitive cells may not be providing as much protection as their nominal value would indicate. Some sensitive aggregates may be constructed of the pool of sensitive aggregates and nonsensitive cells in the aggregation structure.

40. When a cell or an aggregation is sensitive, an indication of this must be included in the output communication file. A possible value would be the numerical value of the sensitivity measure. Another natural value would be the boundary between acceptable and unacceptable, namely too close, estimates of the cell or aggregate value. The use of these boundary values, which are given the name of tolerances, seem to be slightly more natural. We can also allow for the prior knowledge of the users. The arithmetic is simple as an upper tolerance is the cell value plus half the sensitivity value, where the half represents the 50 to 150 percent approximations that the prior knowledge provides.

41. The user input to ACSTabul consists of the descriptions of the variables, their usage and the tables which we have seen in the communication file. The input data file must also be described. The input data fields will be the variables as well as the identification and data fields. The ordering and widths of the fields in the records of input data file must also be specified. There can also be ignored filler fields between the useful data.

```
42.  ident entno
      data value
      record 6, sic 4, 4, geo 6, entno 10, value 15
```

This would be a two classification example with the data record of 6 filler characters, 4 sic characters, 4 fillers, 6 geo characters, 10 entno characters and 15 characters of value. The data fields can be in any order, not just the natural order that they appear in this example. The meaning which associates data values to classification codes is given by an extension of the form of the variable's code which appears only in ACSTabul. Each code specifies a range of data values which contribute to that code value.

```
43.  code tot = 11 - 12, 21 - 22
      code a = 11 - 12
      code aa = 11 ( "aa" )
```

This specifies that tot is a set of ranges, a is a range and aa is a single value which may alternatively be coded as an alphabetic string. A similar convenience is allowed on the ident which can be specified to a fixed type which corresponds to the known establishment coding formats of some statistical agencies and which must be converted to its corresponding enterprise number. These conveniences are intended to lower the fuss level of providing the input data files. The conversion of establishment numbers to enterprise numbers often requires awkward conversions for standard database query packages, although it is standard part of this data use. Purely numerical codes, such as the lowest level SIC codes, would be self defining.

44. The user supplied input commands would be the variable descriptions, the variable usages and tables, the descriptions of the input data records and some option statements to choose the sensitivity rule and control the amount of printed output. This would be about ten lines of input if the variable descriptions, and tables if the defaults are not adequate, are available in separate files. The variable

descriptions would be minor variations on standard metadata prepared either automatically or by manual modification of other files. The output would be a communication file.

## V. CELL SUPPRESSION COMPONENT

45. The central component in the ACS system is ACSSuprs. This is the cell suppression activity. It is more accurate to say that this component completes cell suppression patterns. One extreme case is when the supplied pattern is quite incomplete as it is just the sensitive cells and aggregations. Another extreme case is when the supplied pattern is expected to be complete as it is the previous pattern from a periodic publication. In practice there will often be prespecified cells for both publication and withholding. No mere program is likely to be able to correctly account for the various subject matter subtleties of a real survey. The program will have information of which cells are large, are small or are sensitive, all of which is a large part, but not all, of the subject matter. The completed cell suppression pattern will reflect the technical consequences of the analysts subject matter knowledge. The ability of the analyst to try alternative patterns will enable the requirements of the subject matter to be more fully addressed.

46. The basic operations of the cell suppression component are very simple. Some sensitive cell is temporarily changed to its tolerance value and then other cells are changed to bring the collection of tables back into balance. The changes must be permitted by the assumptions on the prior user knowledge. This means that the changed value of any cell must be in the range of 50 to 150 percent of its nominal value. This is the same set of assumptions that were used to construct the tolerance values. Any cell which is subject to change will become a suppressed cell. There are many patterns possible and we must have a rule for choosing which pattern we want. There are many sensitive cells and we must also have rule for choosing which one we are protecting. This has set up a mathematical programming problem. The components are the constraints which indicate what it means for the tables to be in balance and to satisfy the prior knowledge assumptions. The rule for choosing which pattern we want would be called a figure of merit or an objective function. We also require some sequencing rules for subproblems.

47. The objective function would appear to be the easiest decision. The immediate suggestion is to just suppress the fewest cells. But manual practice knew that this lead to the use of large complements which were undesirable in subject matter practice. The next suggest is to suppress the least total value of cells. But again manual practice know that this could lead to the suppression of many small cells which was also undesirable in subject matter practice. The manual practice had been to suppress few cells but to avoid large cells. The common justifications for the suggestions of either fewest cells or least value of cells are quite similar and equally convincing. The safest view is probably that this is a sure sign that the problem is more difficult that it might initially appear.

48. The objective function for the fewest cells is an equal weight to the cell suppression indicator while the objective function for the least total value is a weight of the cell value to the cell suppression indicator. A functional form which does not grow as fast as linear is logarithm. For small values the logarithm may have the technical problem of being negative. The started logarithm, or logarithm of  $1+x$  rather than just  $x$ , avoids this difficulty. The same functional form arises in matching weights in record linkages and in various data fitting problems, where it is called maximum Berg entropy. In practice it does the job. Sensitive, presuppressed or complementary suppression cells would be given zero coefficients in the objective function.

49. The other part of the objective function is whether the suppression indicator should be treated as a continuous variable between zero and one or as a discrete variable having only the values zero and one. There are two classes of arguments. If the suppression of a value means that there is no further information about value then the discrete values would be appropriate. When a marginal entry is

suppressed, not all of the internal entries will be suppressed and there will be obvious lower bounds on the possible values of the suppressed value. If a large cell has a small complement, as it should if it is only slightly sensitive, then it should be possible to obtain small bounds on the ambiguity of the suppressed value. Both of these examples would appear to be contrary to the motivation for the discrete variables.

50. The second class of argument relate to the technical issues of cost and quality of the solution. As techniques improve our notion of acceptable tradeoffs will change. The cost and quality issues also arise in the context of sequential or all at once methods and of whole or partitioned problem methods. The constraints serve to keep the complements of any sensitive cell localized to the same part of the collection of tables. In practice this means that the continuous variables take on relatively few values and the distinction between discrete and continuous variables is much smaller in practice than it might be expected to be.

51. The differences seem to be mainly in the handling of the embedded knapsack problems and common two pass approximations are quite effective. A second pass to address knapsack issues in cell suppression is quite simple. After the initial solution is found, a second problem is defined. All of the nonsuppressed cells are treated as fixed so the only active computation will involve the cells identified as complements in the initial solution.

52. The objective function is set to prefer large cells as complements, the value divided by the started logarithm is used although the reciprocal of the value would have much the same properties, and a new solution is found. (If such an objective function with its preference for large complements were used on the full problem, we would find that the grand total would be suppressed as it should be under the immediate assumptions but contrary to most notions of subject matter common sense.) It will tend to leave some number of smaller cells unsuppressed. These are the cells which are over suppressions due to knapsack issues in the initial solution. They are not needed as complements.

53. The simplest sequencing rule is to protect the cell needing the largest deviations. When this step is completed, the chosen cell will no longer require protection as we will know that deviations to protect it are possible. We may also know that some other cells have been protected in passing. We would repeat this choice rule until all the cells have been protected. This is a heuristic sequencing rule which is often called the greedy algorithm in operations research.

54. When the solution has been obtained we would like to be able to explain why various complements have been used. We could examine the processing steps which lead to the final solution. The use of protection in passing will mean that in a few spurious cases some cells will be listed as complements of other cells when not actually required. Some early protection steps might benefit from later suppressions and we would not notice this. An explicit phase in which each sensitive cell is protected using only the identified complements avoids these technical problems. This phase closely resembles the knapsack issue solution in use of only suppressed cells and in its use of an objective function preferring large complements. The user is presented with a list of the complements required by each sensitive cell as well as the same information organized to show the sensitive cells that each complement assists in protecting.

55. The default action in ACSSuprs is to run the whole problem with the started logarithm objective function followed by a cleanup pass for the knapsack issues. An explain step follows to indicate the usage of complements of sensitive cells. In its simplest form the user would specify the communication file. In practice, a couple option statements would be used to control the amount of output and whether the all passes are used.

56. When the problems become large, they can be partitioned into several smaller components. The several problem partitions can be specified, and executed, within a single computer task. (Long description omitted at draft stage.)

## **VI. AUDITING COMPONENT**

57. The checking or auditing component is ACSAudit. Checking only makes sense if there are some cell values missing. For this to occur, the input must either come from outside the ACS system or be the result of explicit utility action to withhold cell values. When input comes from outside the ACS system, it is often either independently rounded or subject to capture errors and the resulting tables do not have their rows, columns and other aggregates correctly adding up to their nominal totals. The tables must be adjusted back to being additive as a first step.

58. The adjustment is determined by minimizing a least absolute deviations objective function. For independent rounding all the adjustments will be small, although the results will not be the same as a controlled rounding of the unknown original data. For capture errors there will some adjustments that will not be small. With a correctly adding tables it is possible to determine the smallest and largest values which any missing value could have without violating the constraints implied by the tables.

59. These lower and upper bounds are informative but for many purposes we would prefer a single number for each missing entry. The determined numbers must satisfy all the constraints implied by the tables. We would like the numbers to be inside the bounds and to add to the various totals. Midpoints of the bounds will not add up but do provide a set of starting values for Iterative Proportional Fitting (IPF). The IPF process will adjust the new values to match the existing marginal totals. When the marginal entries also have missing values, we can organize a sequential fitting order and fill in all the missing values.

60. The first fits would be of the least disaggregated marginals. The next fits would be of the next least disaggregated marginals. This provides a lattice ordering of types of marginal entries which allows the fitting to be well defined and applied to all dimensions. The IPF does not usually fully converge to exactly additive cell values and there is a final small adjustment by the least absolute deviation procedure to obtain an exactly adding set of tables.

61. The default action in ACSAudit is to run the adjustment, bounding and fitting steps. In its simplest form the user would specify the communication file. In practice, a couple option statements would be used to control the amount of output and whether the all steps are used. The adjustment step would not be needed if the input was prepared by other ACS components.

## **VII. UTILITY COMPONENT**

62. The utility component is ACSUtil. Most of the user interaction with the system is through this utility component. The values and flags set by ACSUtil provide the fine scale control of the actions of the other components. The untidy features of a real cell suppression problems may require various forms of utility action. Some of these actions make little sense, or often none at all, when mixed with other types of utility actions. The utility actions can take several forms.

63. The utility action may be associated with data input. Undefined cells may be defined if they were absent for data from other sources.

64. The utility action may be associated with the data values. Cell values, including flags, may be changed or set to missing. This may be applied to individual cells or the to cells which are part of some

patterns of cells. The changes for individual cells may be to match the flagging of an external file, so that a periodic survey pattern can be copied to the new round of the survey. The cell values may be set to missing if the flagging indicates that the cell is suppressed. The names, or just spellings, of either variables or codes can be changed. The number of classification variables may be changed by dropping all but one value of an existing variable or by adding a new variable with only one value.

65. The utility action may be associated with the amount of data output. Only a subset of the data may be output so it can be compatible for other uses.

66. The utility action may be associated with the format of the data output. The output file may be reformatted as a flat file for return to the surrounding survey automation. The output may be reorganized so that a spreadsheet program would form an organized spreadsheet from the comma separated values. The order of the variables may be changed so it can be compatible for other uses.

67. To specify that the cell (tot,abc) is to be prepublished and that all possible disaggregation codes of tot with other are to be treated as hidden cells we would use

```
update ( tot, abc ) to f = p
mask ( tot=>, other ) to f = h
```

To copy the flagging information from the last month to the current month of a monthly survey we would use

```
scan "last-month-file"
```

The total amount of user input can vary from small for activities such as the scan to fairly bulky if the user specifies a large number of updates. The practice of having trial runs on almost final files followed by production runs with tighter deadlines makes scripted operations the practical operating mode.

## VIII. EXTRACTION COMPONENT

68. Externally prepared tables for auditing are often made available in the form of spreadsheets. ACSExtr is the component which can extract values from a symbolic link formatted Excel (SYLK) spreadsheet. If the spreadsheet has been prepared with explicit row and column labels, the individual cell values can be located and extracted by ACSExtr. Many spreadsheets already have adequate labeling to allow easy extraction, and most others can be readily modified to achieve this level of labeling. The output can be cleaned up with ACSUtil and audited with ACSAudit.

69. This activity tends to be associated with a preliminary stage at which there are no ACS prepared files and only historic files are being audited. Various bad examples have been illustrated by this mechanism. The original publishers are often not aware of the problems with the publication. This is an interesting adjunct activity to cell suppression but does not warrant further discussion of this component beyond its existence and rather arcane application.

## IX. EXTENSIONS

70. Three experimental variants on ACSSuprs are available. One of the problems with the sequential heuristics is that some of the early suppression steps may benefit from suppressions which are made at later stages. This arises when a large complement is avoided initially but is finally forced to be used by a small sensitive cell at a later step. One variant is a look ahead strategy. The basic observation is that large complements are likely to be the same for all orderings of the protection steps. So the first action is to find the large complements using the basic strategy. Once they have been found, they can be treated as prespecified and the computation repeated.

71. All protection steps will have the benefit of the large complements and the choice of small complements may be modified. A slightly different diagnosis of the problem leads to a reordering strategy. The observation is that the steps should be ordered so that the biggest complements are needed first. So the first action is to determine the size of the complements needed for each sensitive cell. There would be no protection in passing or other reuse shortcuts during this initial activity. Once the processing order has been determined a standard suppression activity can be undertaken. Both of these have shown small improvements on real problems, but further experience would be required before more definitive recommendations could be made.

72. An alternate view is that the solution to the sequential computation problems is to do the problems simultaneously. One defines a coordinating problem which bounds the separate problems and which carries the objective function. All that is required is representing all the separate problems and the variables needed to coordinate them. When memory was a limiting resource this was not really practical. Small problems are now within reason. This is also a single problem, that does not benefit from restarting, so that interior point methods can be used. Toy problems have been solved but no experience has been gained with production sized problems.

73. Experimental versions of ACSUtil and ACSAudit are directed at volume / price / revenue applications. The experimental application was with gasoline volumes and gasoline prices. Volume has the structure of the standard application. Volume weighted prices provide additional information about any withheld volumes. The implicit revenues defined as the product of volume and price have an additive structure like the volume. This allows for a linear analysis. Bounds on prices can be obtained by examining octagonal regions in the volume / revenue planes.

## **X. FURTHER DEVELOPMENTS**

74. The use of range publications is a minor adaptation of the existing technology. (See end of Kirkendall & Sande.) The problems of dealing with after publication ad hoc tabulation requests is an ongoing nuisance for many survey operations. An organized use of these tools would be able to address many of the problems.

## **XI. SUMMARY**

75. The design of the ACS system reflects experience gained from the implementation of the CONFID system. Suitable generalizations and repackaging are present in ACS. There is just one system to handle any number of classification variables with a simpler design as the required generality was known in advance. The additional functions of the explain capability for the cell suppression and the fitted values for the auditing extend the capability.

76. The grouping of the partitioning capability with the cell suppression makes problem partitioning much more convenient. The grouping of adjustment, bounding and fitting into a single component makes auditing more convenient. Bringing the utility functions into a single component lowers the confusion level somewhat.

77. The use of a special purpose small language makes the input less arcane. The input can also serve as documentation as it uses the subject matter terms. The error detection and diagnostics can be improved so use is less frustrating.

78. A system component for tabulation has proven to be very useful as the analytical complexity of the miscellaneous aggregations can be increased without having to rely on other systems. The use of a communication file seems to be appropriate to the problem. There is no difficulty with portability. The

sequence of transformations applied to the communication file do not appear to be sufficiently disciplined to allow it to be treated as a database which is being updated.