METHODS FOR ESTIMATING MISSING DATA ON MIGRANTS IN THE 1991 CENSUS

Philip Rees and
Oliver Duke-Williams

WORKING PAPER 95/20

SCHOOL OF GEOGRAPHY • UNIVERSITY OF LEEDS

CONTENTS

Abstract
List of Tables
List of Figures

1 INTRODUCTION

2 THE PROBLEM AND A FORMAL NOTATION

2.1 The problem
2.2 A formal notation
2.3 Methods used

3 LOGICAL DATA PATCHING

3.1 Preparation: patching of the arrays C^{ljk} and D^{iJk}
3.2 Test 1: Single case of F^{ijk} suppression in iJ or Ij
3.3 Test 2: single case of marginal > 0
3.4 Test 3: partial unsuppression
3.5 The 'county' level checks and 'deep reiteration'

4 LOGICAL DATA PATCHING ILLUSTRATED

5 INTEGER FITTING

5.1 Preparation of the array
5.2 Iterative proportional fitting
5.3 A simple rounding algorithm

6 A CONTROLLED ROUNDING PROCEDURE

7 CONCLUSIONS

REFERENCES

ABSTRACT

The paper discusses the use of suppression to protect data in the Special Migration Statistics, a dataset produced from the 1991 Census, and argues that this procedure prevents accurate analysis of the dataset as supplied. A computer program is described which uses a series of methods are developed to 'recover' data which was suppressed, and to estimate that part of the data which can not be recovered. A process termed *logical data patching* is used to recover data, while a technique termed *integer fitting* is introduced to estimate the remaining suppressed parts of the dataset. The latter process uses a familiar *iterative proportional fitting* procedure, coupled with an innovative three-way *controlled rounding* procedure in order to generate integer-only tables which are consistent with all available totals. The program has been used to recover and estimate data successfully, and results include a sample table of such data.

LIST OF TABLES

1 Migration by district type, 1991 Census: Degree of suppression

2 An aggregated flow matrix constructed from the new data: Black migrants between metropolitan and non-metropolitan regions, 1990-91

LIST OF FIGURES

1 A generalised flow chart of the smsgaps program

2 Migrants from West Yorkshire to Leicestershire: total flows

3 Migrants from West Yorkshire to Leicestershire:
flows for four ethnic groups as released by OPCS with suppression

4 Migrants from West Yorkshire to Leicestershire:
flows for four ethnic groups marked as either known or suppressed

5 Migrants from West Yorkshire to Leicestershire:
logical data patching using a first technique

6 Migrants from West Yorkshire to Leicestershire:
all logical data patching results using the first technique

7 Migrants from West Yorkshire to Leicestershire:
logical data patching using a second technique

8 Migrants from West Yorkshire to Leicestershire:
all logical data patching using the second technique

9 Migrants from West Yorkshire to Leicestershire:
a second iteration of logical data patching using the second technique

10 Migrants from West Yorkshire to Leicestershire:
a third iteration of logical data patching using the first technique

11 Paths in a three-dimensional array

METHODS FOR ESTIMATING MISSING DATA ON MIGRANTS IN THE 1991 CENSUS

Phil Rees and Oliver Duke-Williams
School of Geography, University of Leeds, Leeds LS2 9JT, UK

Paper presented at the International Conference on Population Geography, University of Dundee, 16-19 September 1995

1. INTRODUCTION

One of the most valuable outputs from a national Census of Population is a set of data on the flows of population over time between places. From both the 1981 and 1991 Censuses of Population in Great Britain were produced special sets of interaction statistics covering the journey from home to work (daily interaction in the week prior to the census) and covering the migration from a residence location one year before the census to residence location at the time of the census. Because the Census is designed to capture information from all households and individuals, it is possible to generate very spatially detailed tables of origin to destination flows. The journey to work flows are vital inputs to the definition of Travel-to-Work Areas, while the migration flows between counties and metropolitan districts are essential inputs to subnational projections (in England).

Because flow statistics are best presented and analysed as origin-destination matrices, the number of cells in the statistical tables is potentially very large. This gives rise to concerns that the release of such large tables will compromise the confidentiality of the Census data and inadvertently reveal individual information (though no individual identifiers are present in the tables). Techniques are then used to protect the data. The journey-to-work data (the Special Workplace Statistics) are protected by sampling: only 10% of the census enumeration forms are processed because of the difficulties of recognising and locating the place of work. In the Special Migration Statistics, which use 100%, data an alternative approach is taken: demographic characteristics (age, sex and membership of wholly moving households) are regarded as non-confidential and flow tables for age and sex groups, and for wholly moving households are released (Flowerdew and Green 1993; Rees and Duke-Williams 1995a). But for other

migrant characteristics such as ethnic group membership or economic position, which are regarded as sensitive characteristics by the Census Offices, tables are only released for flows in which ten or more individual migrants or 10 households (depending on the population base of the table) participate. This means that there are a great many holes in the data array (see Rees and Duke-Williams 1994 for a visualisation). The majority of migrants may be reported but not necessarily the majority of origin-destinations flows.

Table 1 illustrates the problem involved. The districts of Great Britain are classified into the OPCS district types arranged roughly in density order from most to least dense. The net migration into these district types is summarised in the third and fourth columns. The final column reports the percentage of out-migrants from these regions which is subject to suppression. In the densest groups of districts (London, the Principal cities of metropolitan counties, the Large non-metropolitan cities) some 95% or more migrants are captured but in the three least densely settled types this percentage is less than 90. Attempts to analyse or aggregate such data are doomed to miserable failure.

One concession was made by the Census Offices when the specification of the Special Migration Statistics (SMS) was designed which make the 1991 Census SMS much more usable than the 1981 version. It was agreed that the total of internal out-migrants from origin districts and the total of internal in-migrants to destination districts would be produced. By internal migration is meant migration, the origin and destination of which are both located within Great Britain. This means that some analysis of total migration exchanges can be carried out but, as Rees and Duke-Williams (1995a) demonstrate, totals for spatial systems aggregated from the basic building brick of local government district have awkward meanings. For example, if we had aggregated the district statistics to regions, we would not obtain the total of out-migrants from regions but the sum, for districts in the region, of out-migrants from districts to elsewhere in Great Britain outside the district. In other words the region out-migrants include within region migrants. Confused? You bet!

Table 1: Migration by district type, 1991 Census: degree of suppression

OPCS type	Population	%	Net migration	Net rate	% out migrants not suppressed
Inner London	2,504,451	*4.57*	-31,009	*-12.38*	95.09
Outer London	4,175,248	*7.62*	-21,159	*-5.07*	94.83
Principal cities	3,922,670	*7.15*	-26,311	*-6.71*	98.17
Other Metropolitan	8,345,084	*15.22*	-7,333	*-0.88*	94.20
Large non-metro	3,493,284	*6.37*	-14,040	*-4.02*	95.68
Small non-metro	1,861,351	*3.39*	-7,812	*-4.20*	91.07
Industrial	7,415,515	*13.52*	7,194	*0.97*	90.22
District inc. new towns	2,921,035	*5.33*	3,060	*1.05*	90.91
Resort and retirement	3,544,013	*6.46*	17,223	*4.86*	89.29
Mixed urban / rural	10,139,100	*18.49*	33,579	*3.31*	88.68
Remote largely rural	6,507,093	*11.87*	46,608	*7.16*	87.22
Totals	54,828,844	*100.00*	0	0	

However, existence of these district out- and in-migrant totals make possible the *reconstruction* of the full flow matrix classified by most of the characteristics deemed too sensitive for release in easy to use form. This paper reports on the algorithms used to accomplish this task, which we believe may have relevance outside the narrow application reported here. A brief illustration is provided of part of the reconstructed data, though we describe a detailed analysis of migration by ethnicity made possible by the work in another paper (Rees and Duke-Williams 1995b). We do not claim to have achieved an exact solution to the problem of filling the gaps in the migration array but we have come close. The principal aim of the work is to use available information in the SMS from Tables M01 to M03, which are unsuppressed, and from the out- and in-migrant totals, to estimate missing cells in Tables M04 to M11. To date (September 1995) we have completed the reconstruction of Tables M04, M05, M06, M07 (except for the final cell), M08 and M09. The task will be completed with the reconstruction of Table M10. The final table, M11, which counts Welsh or Gaelic speaking migrants cannot be reconstructed with the same techniques because there is insufficient information available. Illustrations of the techniques are taken from Table M05 which has been used for analysis in Rees and Duke-Williams (1995b).

2. THE PROBLEM AND A FORMAL NOTATION

2.1 The problem

We have a large three dimensional array or 'cube', which contains counts of migrants classified by origin, destination and attribute. Large numbers of counts are missing because of the suppression process. The task at hand is to find a solution which will provide values for the missing data. Such a solution is believed to be possible because the (correct) row, column or layer totals or 'faces of the cube' are known in all cases.

For any of the tables subject to suppression, the following data are available.

(1) Total out-migrants from districts to the Rest of Great Britain by the table class (e.g. ethnic group)
(2) Total in-migrants to districts from RGB by the table class (e.g. ethnic group)
(3) Total migrants between districts in Great Britain (from the total cell in Table M01)
(4) The migrants by table class for inter-district flows which in total include 10 or more migrants or 10 or more households.
(5) Migrants from a district to a county by the table class
(6) Migrants to a district from a county by the table class.

These latter two pieces of information are very useful because the flows are usually larger and therefore less likely to be suppressed. We need to use all of this information in a consistent way to fill in the gaps.

A formal description of the process of logical data patching is now developed. The reader not wishing to follow the technical description should skip the rest of section 2, miss out section 3 and resume reading with the example described in section 4.

2.2 A formal notation

To develop a method that uses all the available information we need to invent a suitable notation.

- i is an origin district
- j is a destination district
- I is an origin county
- J is a destination county

- *Arrays* have three dimensions, x, y and z, which in the case of this paper refer to origins, destinations and migrant types. It is also useful to refer to the array dimensions in general terms as rows, columns and shafts.

- A *slice* is a two dimensional subset or matrix of a 3 dimensional array (i.e. a view of the array where one of the dimensional indices is restricted to a single fixed value). A slice is described as having rows and columns, irrespective of the relationship these terms might have to the origins, destinations and types of the 3D matrix from which it has been taken.

- The following sets of flow arrays are listed using the variable names given to them in the *smsgaps* program.

F^{ijk} the number of migrants from an origin district i to a destination district j with characteristic k, the typical element of the array $\boldsymbol{F}$.

C^{Ijk} the number of migrants from county I to district j with characteristic k, the typical element of the array $\boldsymbol{C}$.

D^{iJk} the number of migrants from district i to county J with characteristic k, the typical element of the array $\boldsymbol{D}$.

For each table to be solved (i.e. SMS M04 upwards), the initial arrays ***F***, ***C*** and ***D*** will contain a mix of data which is either known or unknown, due to the effects of suppression. Unknown values are set to zero, and a distinction may be made between these values and 'true' zeroes (i.e. flows which are known to be zero, because Fij* is zero). For each pair of areas *ij* the flows will either be known for all values of *k* or be unknown for all values of *k*. It is the purpose of the smsgaps program to replace the zeroes which represent missing values with an actual flow.

The following three terms all refer to 'master' tables. A master table is one which holds shaft totals which are known to be correct. In each case the master table gives the total of migrants summed across all values of *k*. The *ij* scale used will be appropriate to the table structure (i.e. will be either district by district, county by district or district by county). The variable used in the master table will be either the total of table M01 (individual migrants), the cell M0201 (the number of wholly moving households) or the cell M0202 (the number of migrants in wholly moving households) as satisfies the base population of the table under consideration.

D2DM_ij the total number of migrants from district *i* to district *j*. The term 'D2DM' stands for 'district to district master'.
(= Fij*)
this is the constraint total

C2DM_Ij the total number of migrants from county *I* to district *j*. The term 'C2DM' stands for 'county to district master'. It should be noted that for cases where district *j* is a component of the county *I*, the flow does not initially include the flow from county *I* to district *j* (i.e. the flow within district *j*).
(= CIj*)

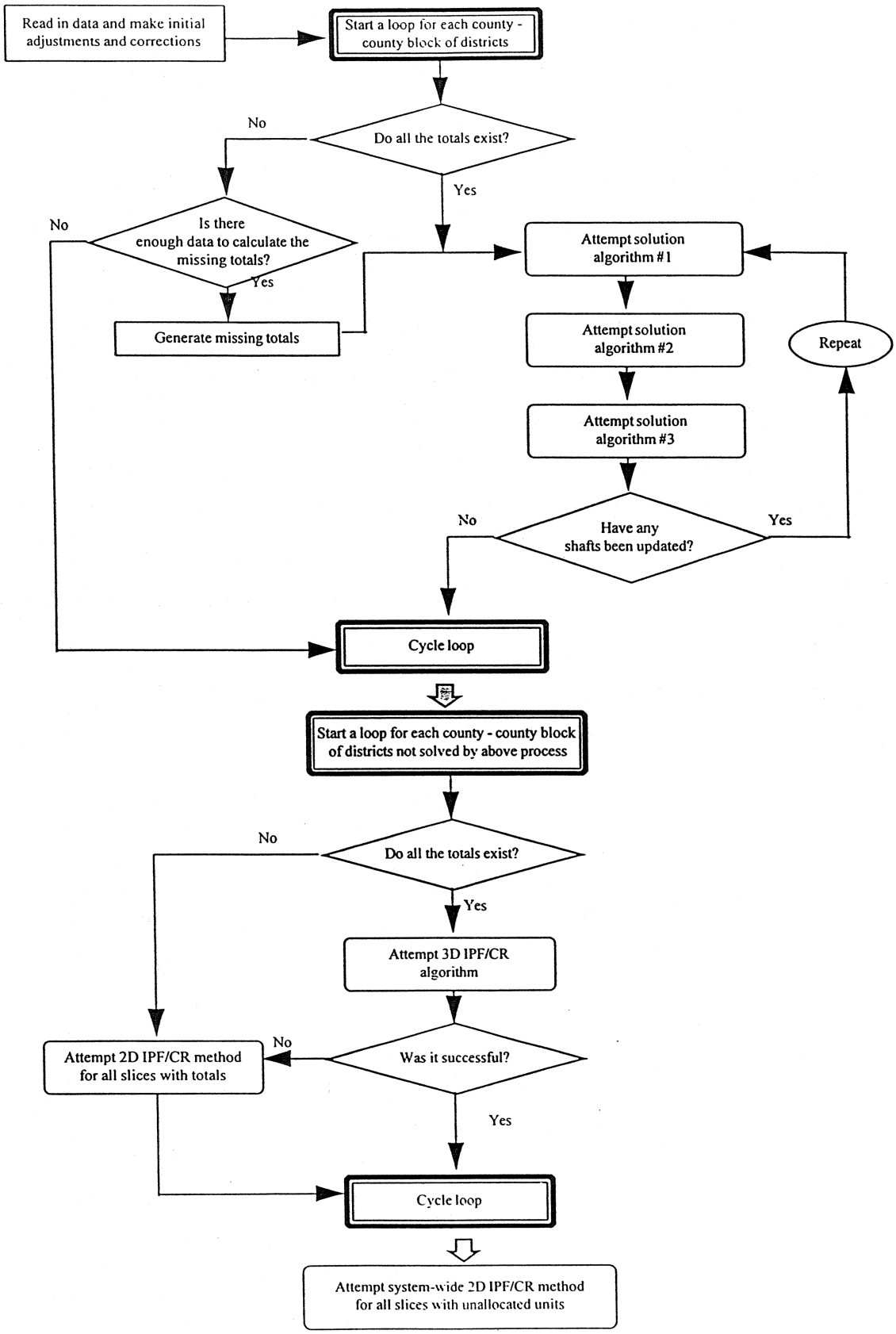
D2CM_iJ the total number of migrants from district *i* to county *J*. The term 'D2CM' stands for 'district to county master'. As with the preceding

definition it should be noted that for cases where district *i* is part of the county *J*, the flow does not include the flow within district *i*.
(= DiJ*)

2.3 Methods used

The process of 'filling the gaps' is one which is made up of two main tasks - *logical data patching* (LDP) and *integer fitting* (IF). The LDP process compares data vectors which should sum to the same total. Where cells have been suppressed differences in totals may allow values associated with suppressed cells to be logically deduced. The IF process uses an *iterative proportional fitting* (IPF) procedure, coupled with a three-way *controlled rounding* (CR) procedure to estimate a set of values which correctly sum to all relevant totals but may not necessarily reflect the 'true' answer. The iterative proportional fitting procedure uses as row, column and layer controls the relevant totals less the sum in the appropriate dimension of the known (unsuppressed) cells including those cells which have been filled in through the simple arithmetic of logical data patching. The methods used have been incorporated in a FORTRAN program, called *smsgaps*, the flow chart of which is set out in Figure 1.

Figure 1: A generalised flow chart of the smsgaps program



3. LOGICAL DATA PATCHING

A large number of unknown flows can be discovered by comparing vectors which should add to the same (known) total. This procedure exploits the county to district and district to county sets of flows present in the SMS to give known sub-totals throughout the matrix.

3.1 Preparation: patching of the arrays C^{Ijk} and D^{iJk}

In all cases where district *i* is **not** a member of county *J*, the totals given by *D2CM_iJ* represent the outflow of district *i* to all districts in county *J*. However, where *i* **is** a member of *J*, then the respective totals show the outflow from *i* to all districts in *J* except *i*. A similar inconsistency occurs in the values of *C2DM_Ij* in cases where *j* is a member of county *I*. The first process carried out adjusts the C2D and D2C totals for cases where the district in question is a constituent of the county in question, by adding in the flow within the district. This makes the relationships between the county subtotals and the district flows consistent throughout the matrix: the subtotal, after this process, always shows the flow from a given district to all districts in the given county, irrespective of whether the district is a part of the county.

For the main part of the LDP process groups of districts are considered in 'county blocks'. A county block *IJ* is a set of districts *i* by *j* where *i* are all members of a given origin county *I*, and *j* are all members of a given destination county *J*. Within each county block a number of tests are made.

3.2 Test 1: Single case of F^{ijk} suppression in *iJ* or *Ij*

Within the county block *IJ* the flows from the origin county *I* to the various destination districts *j* are considered:

If *D2DM_ij* is unsuppressed, then the values of F^{ijk} will be known for all values of *k*. Further, if the number of districts *i* which are suppressed (initially those for which 1

<= $D2DM\_ij$ <= 9) is equal to 1, then if that district is labelled x then the values of for all k can be discovered, by applying the formula:

$$F_k^{xj} = C2D_k^{Ij} - \sum_{i \in I} F_k^{ij}$$

Similarly, unknown values of F^{ijk} for districts which are a member of the destination county J. can be discovered by considering flows from origin districts i to that county, by the formula:

$$F_k^{iy} = D2C_k^{iJ} - \sum_{j \in J} F_k^{ij}$$

where y is the only district in the county J for which flows from i are suppressed.

If any such districts are discovered within a county block then it is possible that these discoveries will have repercussions, creating new cases for which there is only one suppressed district. For this reason if any new values F^{ijk} are discovered within a county block the patching process is repeated for that block (until no new values are discovered). As more district pairs become 'unsuppressed', the initial suppression test - that (1 <= $D2DM\_ij$ <= 9) - becomes invalid, and it is therefore necessary to keep an independent record of which district pairs are suppressed or not.

3.3 Test 2: single case of marginal > 0

In cases where there are too many suppressed districts in a given vector for the above method to be applied, a second test can be made. A slice is considered which represents a portion of the array detailing the flows from a district i to all j in J, for all values of k. All known values in the slice are zeroed and subtracted from the slice totals, which generates a set of marginals - the total of migrants with characteristic k yet to be allocated. If only one of these marginals is greater than zero, then it is clear that all unallocated migrants have the same characteristic, such as membership of a

particular ethnic group or economic activity class. In such cases, the distribution of these 'unknowns' between the various origin districts will be identical to the known *D2DM_ij* totals.

This process requires cases where the *D2C_iJk* values are not suppressed., so that the total values for each *k* will be known. A similar test is made for slices representing the flows from all *i* in *I* to a fixed case of *j*; that requires that the corresponding *C2D_Ijk* totals are not suppressed.

As before, if one or more shafts become unsuppressed during this test there may be repercussions which allow other sets of values to be solved. Therefore if any values are discovered, another iteration of the testing procedure is necessary.

3.4 Test 3: partial unsuppression

The two tests above have involved attempts to unsuppress a complete shaft (or shafts), - i.e. solving a shaft *ij* of ***F*** for all values of *k* - in a single step. The final test (or set of tests) in the LDP stage attempts to solve shafts through a more low-level approach. If values of can be found for single values of *k*, there may eventually be sufficient data known to solve one or more shafts fully. An array ***X*** is constructed from ***F*** where all values are zero, apart from those which are components of shafts which are suppressed, which are set to an arbitrary non-zero value. Row and column totals for ***X*** are found by taking the original totals for that county block (C^{Ijk} and D^{iJk}) and subtracting all known cells. Thus the totals are the total of all unallocated units within the array.

(i) The 'cleaning' of the array.

Where the totals are known, then any positive value of ***X*** which lies on a vector for which $C^{Ijk}=0$ or $D^{iJk}=0$ can obviously be reset to 0. This procedure alone will not render any shafts solvable, but does allow the following stage to be carried out.

(ii) 'Partial unsuppression'.

The array X is considered as a series of 2 dimensional slices. A looped procedure searches for cells within such a slice for cells which have the following characteristics.

- non-zero [i.e. it is part of a suppressed shaft, and has not been set to zero by procedure i) above]

- the only such cell in that row and column [where 'row' and 'column' are considered as the relative addresses in the 2d slice currently being studied]

- the (adjusted) totals C^{Ijk} and D^{iJk} are available.

In such a case, the value of X^{ijk} must equal the value of C^{Ijk} (which must, if the data is internally consistent, be equal to D^{iJk}), because C^{Ijk} is the total of all unknowns for the vector of which X^{ijk} is a member, and X^{ijk} is the only unknown on that same vector. A consistency check is made, and, assuming it is passed, the new value of X^{ijk} is written to a second array Y^{ijk}, and X^{ijk} is reset to zero.

An extended case of the above requirements can also be solved. If there are 1 or more cells on a row or column (with a known total) for which each cell is the only non-zero cell on the respective perpendicular vector , then the totals from the perpendicular vectors can be written into Y^{ijk}, and each appropriate value of X^{ijk} set to 0. The sum of the totals of the perpendicular vectors should be equal to the total of the original vector on which the unknown cells have been identified.

After all slices have been checked, a sweep is made the through X to look for shafts which have become solvable (because only 1 cell (or even none at all) in

the shaft remains unknown). Any such shaft is updated (using the values stored in Y), and marked as unsuppressed.

As with the above tests, any changes made to F^{ijk} by this procedure may allow further shafts to be solved, so the whole patching procedure for the block IJ is repeated.

3.5 The 'county' level checks and 'deep reiteration'

The above sections have described techniques used to unsuppress shafts of F^{ijk}. These however are not the only values which may be suppressed; the totals C^{Ijk} and D^{iJk} might also be subject to suppression. A number of methods are used to unsuppress these values.

(i) Only one value suppressed.

If, for any county block IJ, a single value of C^{Ijk} is suppressed, while all values of D^{iJk} are known, (or vice versa), then the suppressed values can be discovered by subtraction.

(ii) Total suppressed, but all ij flows unsuppressed.

It is possible that as F^{ijk} shafts become unsuppressed, a situation might arise where C^{Ijk} or D^{iJk} is suppressed, but all the ij pairs along the respective vector are unsuppressed. In this case, the unknown totals values are simply the sum of the F^{ijk} values.

(iii) Single case of C or D along whole vector.

The total inflows and outflows between a district and the rest if the country are always known (from the SMS totals 111111, and 222222). If a case is found where any given

set of values C^{Ijk} or D^{iJk} are the only such totals which are suppressed along the whole (national) vector, the those values can be discovered by subtraction.

Previous sections have explained that if any values of F^{ijk} are altered, then the whole county block must be re-examined, in case any further solutions become apparent. Altering a value of F^{ijk} will not affect data outside the block *IJ*. The final test described above, however, might adjust the totals of a county block which has already been solved as far as possible by the LDP process. If a county block which has already been tested is affected in this way then a 'deep reiteration' is necessary - the county block by county block pass which the LDP process makes is sent back to the affected block.

When the LDP process has been completed, a large number of shafts may remain unsolved. A second sets of algorithms is used to attempt to solve these shafts. The overall array F^{ijk} can be considered to contain two sorts of county blocks: those for which all totals are known, and those for which some totals are still suppressed. Those blocks for which all the totals are known are dealt with by the next stage of *smsgaps* - the integer fitting stage.

4. LOGICAL DATA PATCHING ILLUSTRATED

The processes of filling in cells in the array by subtracting the sum of known cells from a marginal total can be illustrated with an example for the flows between the districts of West Yorkshire and those of Leicestershire. The steps involved are shown in Figures 2 to 10.

Figure 2 shows the total number of migrants between the two sets of districts and derive from cell 18 of SMS Table M01 (M0118). The cell numbering follows the expanded table system in the Quanvert SMS database as implemented by Quantime Ltd. on Manchester Computing's Cray CS6400 server - cell 18 gives the overall total of migrants in table M01 (see Quantime, 1995). The values given in the cells are district-to-district flows and are always known. The row and column totals of the table in Figure 2 are taken from the county-to-district and district-to-county flows. These are also available for each ethnic group table shown in Figure 3.

Figure 3 sets out the four ethnic group tables for West Yorkshire to Leicestershire districts. In the tables are recorded the known flows where there are at least 10 migrants in Table M01. The majority of the cells in the table are zero, on first extraction from the SMS. Figure 4 identifies these zeroes as either real zeroes or suppressed counts. Real zeroes are present when there are no migrants in the relevant inter-district cell in Table M01. Cells with known flows are shaded while suppressed cells are tagged with a question mark. The task is to extend shading throughout the four ethnic group tables. Cells the values of which are found at a step in the algorithm are marked with a heavy border.

The process begins by looking at destinations. Figure 5 shows what happens when Charnwood district is examined. The tables gather together the columns from Figure 4 which have a Charnwood destination. Each column has 4 known values and 1 unknown, with a known total. The second table in Figure 5 shows in the column total position the difference between the sum of known cells and the known totals. In third table these values are inserted into the suppressed cells which have Calderdale as their

origin. Figure 6 shows the result of carrying all similar operations on the four ethnic group tables. As well as the Calderdale to Charnwood shaft, we are also able to fill in the Calderdale to Leicester shaft. A similar pass is made through the data considering the flows from specific origins to all destinations. In this example, the flows from Leeds are only suppressed in the case of the destination North West Leicestershire, and so this set of cells can be unsuppressed.

Figure 7 shows the migrants from West Yorkshire origin districts and Blaby district as destination. As there is more than one origin which is suppressed, the first technique used in Figures 5 and 6 cannot be used. However, if we subtract the known values from the totals, we find that all the unallocated values are in one ethnic group, Whites (Figure 7b). The row marginals can thus be inserted into the unknown cells for that ethnic group, with all other cells being fixed as zero (Figure 7c). Figure 8 illustrates the application of this second technique to all possible cases. Values can be placed in the cells of sixteen shafts through this process.

A first round of gap filling creates opportunities (suppressed cells) for further applications of this second technique. Figure 9 shows the result of applying the second technique a second time. A new pass through the data can now be made using the first technique because some destinations have only one unknown cell and known column totals. Its results are shown in the tables of Figure 10. When this has been done the values of all suppressed cells have been found. We now know how many migrants of each ethnic group migrated from each West Yorkshire districts to each Leicestershire district.

Although the process of logical data patching is merely simple arithmetic, the size of the array to be filled (459 origin districts by 459 destination districts by 4 ethnic groups) means that we need a computer program, *smsgaps*, to accomplish the task over several repetitions of the two techniques outlined here.

Does LDP do the whole job? Not quite. As provided by OPCS, table M05 contains 91.77% of migrants (the remainder having been suppressed out of the table). These

constitute 71.09% of all *Mij* flows greater than zero. Thus around 9% of migrants are suppressed, but they account for around 29% of district to district flows. When the LDP process has been completed for table M05, 98.63% of migrants (and 85.10% of flows) are accounted for. The data which has been recovered through the LDP process is said to be *logically correct* - in other words in can be demonstrated that the results reflect the true values prior to the suppression process. To fill in the rest more approximate techniques must be used, to which we now turn.

Figure 2: Migrants from West Yorkshire to Leicestershire: total flows

M0118

	Destinations									
Origins	*Blaby*	*Charnwood*	*Harborough*	*Hinckley and Bosworth*	*Leicester*	*Melton*	*North West Leicestershire*	*Oadby and Wigston*	*Rutland*	
Calderdale	2	8	4	3	2	0	1	0	2	121
Kirklees	5	12	7	9	31	6	5	3	10	22
Leeds	24	26	11	16	73	15	8	20	13	88
Wakefield	7	12	6	3	14	3	7	1	5	206
	59	86	42	37	157	24	24	24	42	58

Figure 3: Migrants from West Yorkshire to Leicestershire: flows for four ethnic groups as released by OPCS with suppression

WHITE

Origins \ Destinations	Blaby	Charnwood	Harborough	Hinckley and Bosworth	Leicester	Melton	North West Leicestershire	Oadby and Wigston	Rutland	
Bradford	21	27	11	0	25	0	0	0	12	105
Calderdale	0	0	0	0	0	0	0	0	0	21
Kirklees	0	11	0	0	19	0	0	0	10	73
Leeds	23	23	11	16	67	15	0	20	9	192
Wakefield	0	11	0	0	9	0	0	0	0	52
	58	80	39	37	121	23	24	23	38	

BLACK GROUPS

Origins \ Destinations	Blaby	Charnwood	Harborough	Hinckley and Bosworth	Leicester	Melton	North West Leicestershire	Oadby and Wigston	Rutland	
Bradford	0	0	2	0	0	0	0	0	0	2
Calderdale	0	0	0	0	0	0	0	0	0	0
Kirklees	0	0	0	0	0	0	0	0	0	0
Leeds	0	1	0	0	1	0	0	0	0	2
Wakefield	0	0	0	0	0	0	0	0	0	0
	0	1	2	0	1	0	0	0	0	

INDIAN PAKISTANI & BANGLADESHI

Origins \ Destinations	Blaby	Charnwood	Harborough	Hinckley and Bosworth	Leicester	Melton	North West Leicestershire	Oadby and Wigston	Rutland	
Bradford	0	1	1	0	12	0	0	0	0	14
Calderdale	0	0	0	0	0	0	0	0	0	1
Kirklees	0	1	0	0	11	0	0	0	0	13
Leeds	1	0	0	0	4	0	0	0	0	5
Wakefield	0	1	0	0	5	0	0	0	0	6
	1	3	1	0	33	0	0	1	0	

CHINESE & OTHER

Origins \ Destinations	Blaby	Charnwood	Harborough	Hinckley and Bosworth	Leicester	Melton	North West Leicestershire	Oadby and Wigston	Rutland	
Bradford	0	0	0	0	0	0	0	0	0	0
Calderdale	0	0	0	0	0	0	0	0	0	0
Kirklees	0	0	0	0	1	0	0	0	0	2
Leeds	0	2	0	0	1	0	0	0	4	7
Wakefield	0	0	0	0	0	0	0	0	0	0
	0	2	0	0	2	1	0	0	4	

Figure 4: Migrants from West Yorkshire to Leicestershire: flows for four ethnic groups marked as either known or suppressed

WHITE

Origins \ Destinations	Blaby	Charnwood	Harborough	Hinckley and Bosworth	Leicester	Melton	North West Leicestershire	Oadby and Wigston	Rutland	
Bradford	21	27	11	?	25	0	?	0	12	105
Calderdale	?	?	?	?	?	0	?	0	?	21
Kirklees	?	11	?	?	19	?	?	?	10	73
Leeds	23	23	11	16	67	15	?	20	9	192
Wakefield	?	11	?	?	9	?	?	?	?	52
	58	80	39	37	121	23	24	23	38	

BLACK GROUPS

Origins \ Destinations	Blaby	Charnwood	Harborough	Hinckley and Bosworth	Leicester	Melton	North West Leicestershire	Oadby and Wigston	Rutland	
Bradford	0	0	2	?	0	0	?	0	0	2
Calderdale	?	?	?	?	?	0	?	0	?	0
Kirklees	?	0	?	?	0	?	?	?	0	0
Leeds	0	1	0	0	1	0	?	0	0	2
Wakefield	?	0	?	?	0	?	?	?	?	0
	0	1	2	0	1	0	0	0	0	

INDIAN PAKISTANI & BANGLADESHI

Origins \ Destinations	Blaby	Charnwood	Harborough	Hinckley and Bosworth	Leicester	Melton	North West Leicestershire	Oadby and Wigston	Rutland	
Bradford	0	1	1	?	12	0	?	0	0	14
Calderdale	?	?	?	?	?	0	?	0	?	1
Kirklees	?	1	?	?	11	?	?	?	0	13
Leeds	1	0	0	0	4	0	?	0	0	5
Wakefield	?	1	?	?	5	?	?	?	?	6
	1	3	1	0	33	0	0	1	0	

CHINESE & OTHER

Origins \ Destinations	Blaby	Charnwood	Harborough	Hinckley and Bosworth	Leicester	Melton	North West Leicestershire	Oadby and Wigston	Rutland	
Bradford	0	0	0	?	0	0	?	0	0	0
Calderdale	?	?	?	?	?	0	?	0	?	0
Kirklees	?	0	?	?	1	?	?	?	0	2
Leeds	0	2	0	0	1	0	?	0	4	7
Wakefield	?	0	?	?	0	?	?	?	?	0
	0	2	0	0	2	1	0	0	4	

Figure 5: Migrants from West Yorkshire to Leicestershire: logical data patching using a first technique

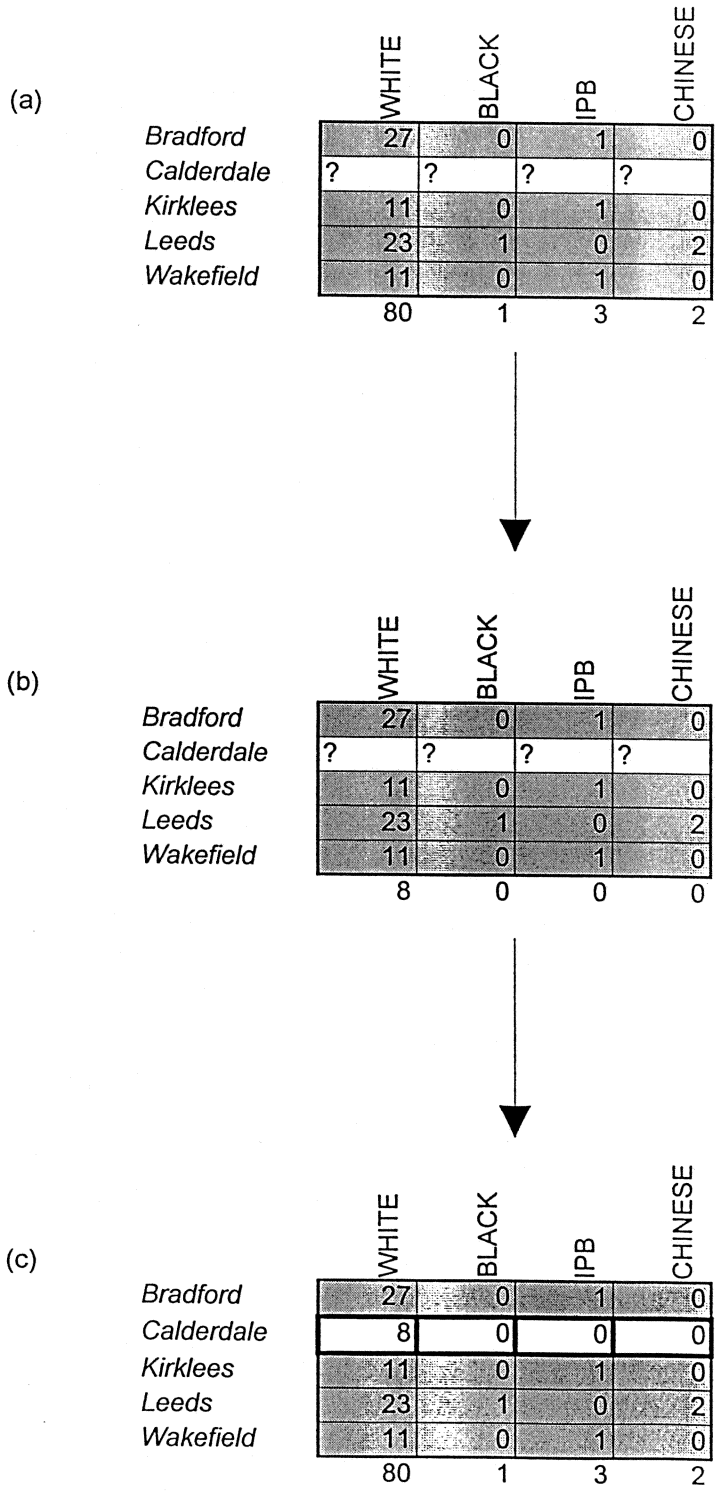


Figure 6: Migrants from West Yorkshire to Leicestershire: all logical data patching results using the first technique

WHITE

Origins \ Destinations	Blaby	Charnwood	Harborough	Hinckley and Bosworth	Leicester	Melton	North West Leicestershire	Oadby and Wigston	Rutland	
Bradford	21	27	11	?	25	0	?	0	12	105
Calderdale	?	8	?	?	1	0	?	0	?	21
Kirklees	?	11	?	?	19	?	?	?	10	73
Leeds	23	23	11	16	67	15	8	20	9	192
Wakefield	?	11	?	?	9	?	?	?	?	52
	58	80	39	37	121	23	24	23	38	

BLACK GROUPS

Origins \ Destinations	Blaby	Charnwood	Harborough	Hinckley and Bosworth	Leicester	Melton	North West Leicestershire	Oadby and Wigston	Rutland	
Bradford	0	0	2	?	0	0	?	0	0	2
Calderdale	?	0	?	?	0	0	?	0	?	0
Kirklees	?	0	?	?	0	?	?	?	0	0
Leeds	0	1	0	0	1	0	0	0	0	2
Wakefield	?	0	?	?	0	?	?	?	?	0
	0	1	2	0	1	0	0	0	0	

INDIAN PAKISTANI & BANGLADESHI

Origins \ Destinations	Blaby	Charnwood	Harborough	Hinckley and Bosworth	Leicester	Melton	North West Leicestershire	Oadby and Wigston	Rutland	
Bradford	0	1	1	?	12	0	?	0	0	14
Calderdale	?	0	?	?	1	0	?	0	?	1
Kirklees	?	1	?	?	11	?	?	?	0	13
Leeds	1	0	0	0	4	0	0	0	0	5
Wakefield	?	1	?	?	5	?	?	?	?	6
	1	3	1	0	33	0	0	1	0	

CHINESE & OTHER

Origins \ Destinations	Blaby	Charnwood	Harborough	Hinckley and Bosworth	Leicester	Melton	North West Leicestershire	Oadby and Wigston	Rutland	
Bradford	0	0	0	?	0	0	?	0	0	0
Calderdale	?	0	?	?	0	0	?	0	?	0
Kirklees	?	0	?	?	1	?	?	?	0	2
Leeds	0	2	0	0	1	0	0	0	4	7
Wakefield	?	0	?	?	0	?	?	?	?	0
	0	2	0	0	2	1	0	0	4	

Figure 7: Migrants from West Yorkshire to Leicestershire: logical data patching using a second technique

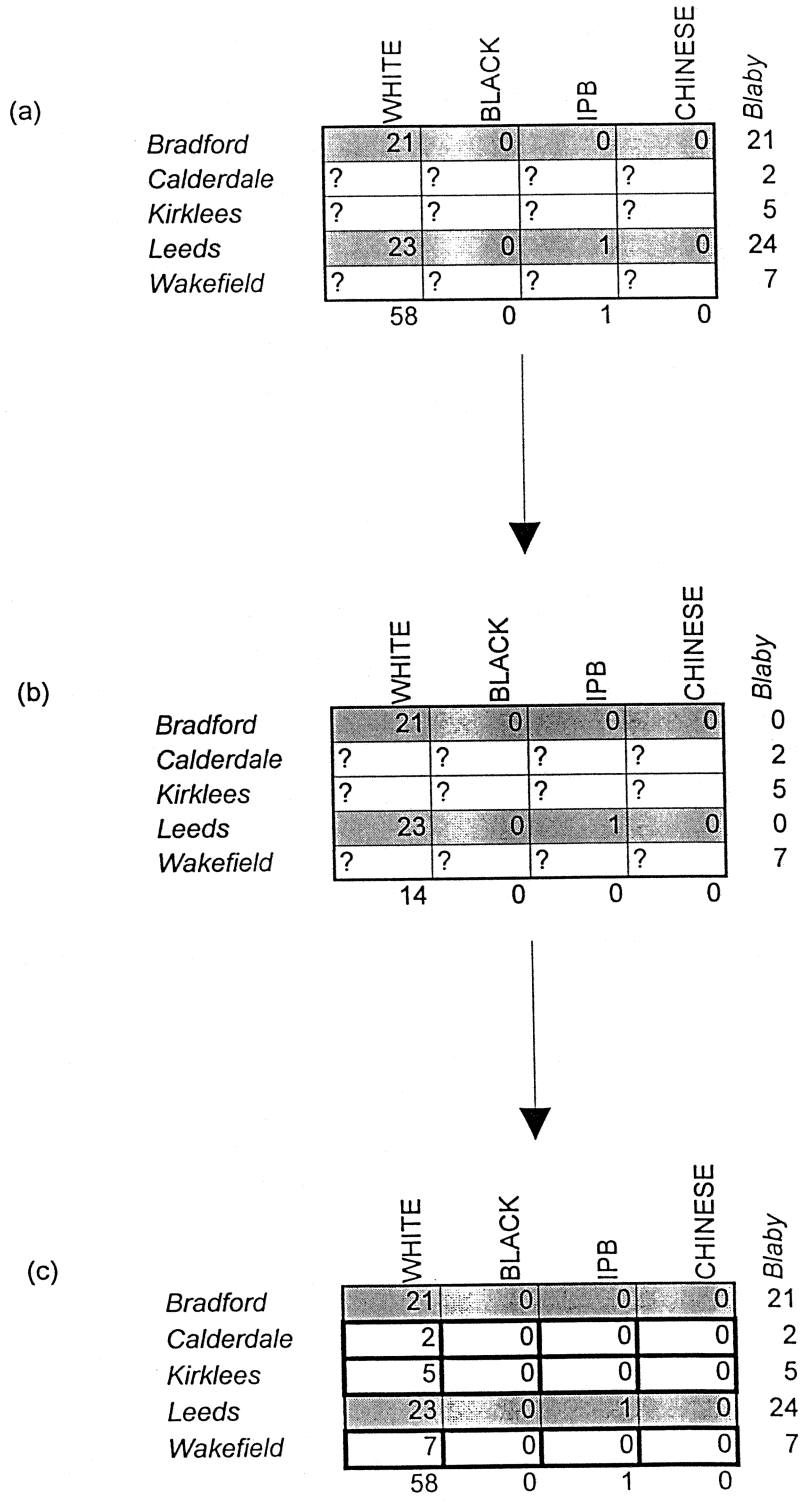


Figure 8: Migrants from West Yorkshire to Leicestershire: all logical data patching using the second technique

WHITE

Origins \ Destinations	Blaby	Charnwood	Harborough	Hinckley and Bosworth	Leicester	Melton	North West Leicestershire	Oadby and Wigston	Rutland	
Bradford	21	27	11	6	25	0	3	0	12	105
Calderdale	2	8	4	3	1	?	1	?	2	21
Kirklees	5	11	7	9	19	?	5	?	10	73
Leeds	23	23	11	16	67	15	8	20	9	192
Wakefield	7	11	6	3	9	?	7	?	5	52
	58	80	39	37	121	23	24	23	38	

BLACK GROUPS

Origins \ Destinations	Blaby	Charnwood	Harborough	Hinckley and Bosworth	Leicester	Melton	North West Leicestershire	Oadby and Wigston	Rutland	
Bradford	0	0	2	0	0	0	0	0	0	2
Calderdale	0	0	0	0	0	0	0	0	0	0
Kirklees	0	0	0	0	0	?	0	?	0	0
Leeds	0	1	0	0	1	0	0	0	0	2
Wakefield	0	0	0	0	0	?	0	?	0	0
	0	1	2	0	1	0	0	0	0	

INDIAN PAKISTANI & BANGLADESHI

Origins \ Destinations	Blaby	Charnwood	Harborough	Hinckley and Bosworth	Leicester	Melton	North West Leicestershire	Oadby and Wigston	Rutland	
Bradford	0	1	1	0	12	0	0	0	0	14
Calderdale	0	0	0	0	1	0	0	0	0	1
Kirklees	0	1	0	0	11	?	0	?	0	13
Leeds	1	0	0	0	4	0	0	0	0	5
Wakefield	0	1	0	0	5	?	0	?	0	6
	1	3	1	0	33	0	0	1	0	

CHINESE & OTHER

Origins \ Destinations	Blaby	Charnwood	Harborough	Hinckley and Bosworth	Leicester	Melton	North West Leicestershire	Oadby and Wigston	Rutland	
Bradford	0	0	0	0	0	0	0	0	0	0
Calderdale	0	0	0	0	0	0	0	0	0	0
Kirklees	0	0	0	0	1	?	0	?	0	2
Leeds	0	2	0	0	1	0	0	0	4	7
Wakefield	0	0	0	0	0	?	0	?	0	0
	0	2	0	0	2	1	0	0	4	

Figure 9: Migrants from West Yorkshire to Leicestershire: a second iteration of logical data patching using the second technique

WHITE

Origins	Blaby	Charnwood	Harborough	Hinckley and Bosworth	Leicester	Melton	North West Leicestershire	Oadby and Wigston	Rutland	
	Destinations									
Bradford	21	27	11	6	25	0	3	0	12	105
Calderdale	2	8	4	3	1	0	1	0	2	21
Kirklees	5	11	7	9	19	?	5	?	10	73
Leeds	23	23	11	16	67	15	8	20	9	192
Wakefield	7	11	6	3	9	8	7	3	5	52
	58	80	39	37	121	23	24	23	38	

BLACK GROUPS

Origins	Blaby	Charnwood	Harborough	Hinckley and Bosworth	Leicester	Melton	North West Leicestershire	Oadby and Wigston	Rutland	
	Destinations									
Bradford	0	0	2	0	0	0	0	0	0	2
Calderdale	0	0	0	0	0	0	0	0	0	0
Kirklees	0	0	0	0	0	?	0	?	0	0
Leeds	0	1	0	0	1	0	0	0	0	2
Wakefield	0	0	0	0	0	0	0	0	0	0
	0	1	2	0	1	0	0	0	0	

INDIAN PAKISTANI & BANGLADESHI

Origins	Blaby	Charnwood	Harborough	Hinckley and Bosworth	Leicester	Melton	North West Leicestershire	Oadby and Wigston	Rutland	
	Destinations									
Bradford	0	1	1	0	12	0	0	0	0	14
Calderdale	0	0	0	0	1	0	0	0	0	1
Kirklees	0	1	0	0	11	?	0	?	0	13
Leeds	1	0	0	0	4	0	0	0	0	5
Wakefield	0	1	0	0	5	0	0	0	0	6
	1	3	1	0	33	0	0	1	0	

CHINESE & OTHER

Origins	Blaby	Charnwood	Harborough	Hinckley and Bosworth	Leicester	Melton	North West Leicestershire	Oadby and Wigston	Rutland	
	Destinations									
Bradford	0	0	0	0	0	0	0	0	0	0
Calderdale	0	0	0	0	0	0	0	0	0	0
Kirklees	0	0	0	0	1	?	0	?	0	2
Leeds	0	2	0	0	1	0	0	0	4	7
Wakefield	0	0	0	0	0	0	0	0	0	0
	0	2	0	0	2	1	0	0	4	

Figure 10: Migrants from West Yorkshire to Leicestershire: a third iteration of logical data patching using the first technique

WHITE

Origins	Blaby	Charnwood	Harborough	Hinckley and Bosworth	Leicester	Melton	North West Leicestershire	Oadby and Wigston	Rutland	
	Destinations									
Bradford	21	27	11	6	25	0	3	0	12	105
Calderdale	2	8	4	3	1	0	1	0	2	21
Kirklees	5	11	7	9	19	0	5	0	10	73
Leeds	23	23	11	16	67	15	8	20	9	192
Wakefield	7	11	6	3	9	8	7	3	5	52
	58	80	39	37	121	23	24	23	38	

BLACK GROUPS

Origins	Blaby	Charnwood	Harborough	Hinckley and Bosworth	Leicester	Melton	North West Leicestershire	Oadby and Wigston	Rutland	
	Destinations									
Bradford	0	0	2	0	0	0	0	0	0	2
Calderdale	0	0	0	0	0	0	0	0	0	0
Kirklees	0	0	0	0	0	0	0	0	0	0
Leeds	0	1	0	0	1	0	0	0	0	2
Wakefield	0	0	0	0	0	0	0	0	0	0
	0	1	2	0	1	0	0	0	0	

INDIAN PAKISTANI & BANGLADESHI

Origins	Blaby	Charnwood	Harborough	Hinckley and Bosworth	Leicester	Melton	North West Leicestershire	Oadby and Wigston	Rutland	
	Destinations									
Bradford	0	1	1	0	12	0	0	0	0	14
Calderdale	0	0	0	0	1	0	0	0	0	1
Kirklees	0	1	0	0	11	0	0	1	0	13
Leeds	1	0	0	0	4	0	0	0	0	5
Wakefield	0	1	0	0	5	0	0	0	0	6
	1	3	1	0	33	0	0	1	0	

CHINESE & OTHER

Origins	Blaby	Charnwood	Harborough	Hinckley and Bosworth	Leicester	Melton	North West Leicestershire	Oadby and Wigston	Rutland	
	Destinations									
Bradford	0	0	0	0	0	0	0	0	0	0
Calderdale	0	0	0	0	0	0	0	0	0	0
Kirklees	0	0	0	0	1	1	0	0	0	2
Leeds	0	2	0	0	1	0	0	0	4	7
Wakefield	0	0	0	0	0	0	0	0	0	0
	0	2	0	0	2	1	0	0	4	

5. INTEGER FITTING

To fill in the remaining unknown cells in the migration array, we resort to the well tried technique of iterative proportional fitting (IPF). In principle, this adjusts interior values of an array to satisfy constraints provided by known marginals of the array. The IPF is not applied to the whole array but only to those cells which are still unknown. Known cells are set to zero (and always continue to be zero through the process) and marginals are adjusted to the original totals minus the sum of known values for cells in the appropriate dimension. IPF works best where the numbers to be estimated are large and rounded real results can be used. In this case IPF will result in a lot of small real numbers well below one migrant. To obtain an integer solution (we don't want part migrants) controlled rounding is used.

County blocks are considered to be potentially solvable by this process if all suppressed cells with the block lie on vectors for which the totals - C^{Ijk} and D^{iJk} are unsuppressed. The shaft totals are always known.

For blocks which meet the requirements a number of procedures are used to attempt to find a solution to the block. Any such solution found will be one that has integer values in cells, such that all three sets of totals (C^{Ijk}, D^{iJk} and $D2DM\_ij$) are satisfied. There may be more than one arrangement of values that satisfies these totals.

An array $\boldsymbol{X}$ is constructed from $\boldsymbol{F}$ such that $\boldsymbol{X}$ has the same dimensions as $\boldsymbol{F}$, and contains zeroes in all cells X^{ijk} apart from those which are suppressed in F^{ijk}. These cells are initially set to an arbitrary non-zero value. X^{ijk} has totals which are mirrored from the arrays C^{Ijk}, D^{iJk} and $D2DM\_ij$. The values of known cells are subtracted from these totals, so that they are equal to zero for any vectors which do not contain any suppressed cells, and equal to the total number of 'unknowns' in all other cases. The purpose of the IF stage is to complete a second array - $\boldsymbol{Y}$ - which contains a set of integer values which are consistent with all three totals. If such an array can be constructed it can then be merged with $\boldsymbol{F}$, in order to complete the block IJ.

5.1 Preparation of the array

The first procedure is similar to the 'partial unsuppression' stage described in the LDP section above. First of all the array is 'cleaned', by re-setting cells which must logically be zero, and then by searching through all possible slices for other cells which can logically be set to a fixed value. Any such cells are set to 0 in X^{ijk}, and the value discovered is written to Y^{ijk}. The cells in the totals arrays for X^{ijk} which are the vectors that the solved cell lies on are adjusted downwards accordingly. As this stage proceeds it is necessary to continually check that all totals are consistent with each other, and sum to the same grand total. Unlike the 'partial unsuppression' stage however, the changes are written to Y^{ijk} on a cell by cell basis, rather than only when a complete shaft can be solved. All values written to Y^{ijk} at this stage have two features: firstly, they make the rest of the IF procedure easier, and secondly, they improve the accuracy of the final solution, because these values are known to be logically correct. Values fixed during the next stages of the IF procedure are known simply to satisfy the totals, but may not necessarily be correct.

5.2 Iterative proportional fitting

When all cells which can be solved on a per-cell basis have been written to Y, the array X is passed to an iterative proportional fitting (IPF) routine. This routine iteratively adjusts all non-zero values in X until all rows, columns and shafts sum to all relevant totals. The routine, however, produces a real number solution, and thus must be adjusted to try and find an integer-only solution. The first part of this process removes all the integer parts of the real numbers produced by the IPF routine, and writes them to Y. Whenever an integer part is removed from a number, the totals arrays for X are revised downwards as appropriate. The integer parts are assumed to be 'correct'. This process results in the array X being one which contains cells which are all either 0, or a real number such that $0 < X^{ijk} < 1$. The totals of this array are all integer values, and equal to the sum of all the real values in the relevant vector. Since the values in X and the totals arrays are adjusted separately (that is to say, the 'totals' arrays function as checks showing what the totals *should be*) consistency checks are

made at this point. The purpose of the remaining stages of the IF procedure is to adjust all cells in X to either 0 or 1, in such a manner that all the totals remain satisfied. If any value is adjusted to 1, then that value is reset to 0, the relevant totals adjusted, and the appropriate cell in Y is incremented. The ultimate goal is thus to convert all cells in X to 0. This will entail Y having been completed as an array which contains integer values which sum to all three totals correctly.

The next stage searches for values in X^{ijk} which are close enough to either 0 or 1 to be considered vulnerable to floating point rounding and accuracy errors. If any value of X^{ijk} or $(1\text{-}X^{ijk})$ is below a critical threshold value, the it is assumed that this value should be either 0 or 1, and is adjusted accordingly.

5.3 A simple rounding algorithm

When any values have been corrected, and a check has been made that a solution has not yet been found, the first rounding algorithm is applied. This copies X^{ijk} into a temporary array, Xt^{ijk}, and then simply rounds all values to the nearest integer (0 or 1). A check is them made to see whether Xt^{ijk} is consistent with the totals of X^{ijk}. If Xt^{ijk} is consistent, the a solution has been found. The values in Xt^{ijk} are then added to Y^{ijk}, and Y^{ijk} is merged with F^{ijk}, completing the block IJ. If, however Xt^{ijk} is not consistent, the routine has failed.

Results so far have shown that the simple rounding algorithm is more successful the fewer the total number of units there are which must be allocated. When there are more than approximately 20 units to be allocated success with the simple algorithm is rare. A simple procedure is used to adjust the results of Xt^{ijk} when the algorithm has failed. When certain conditions apply (a low number of units to be allocated, and the simple algorithm having allocated the right number of units (albeit incorrectly) in the first place), this procedure may be able to adjust Xt^{ijk} so that a solution is found.

6. A CONTROLLED ROUNDING PROCEDURE

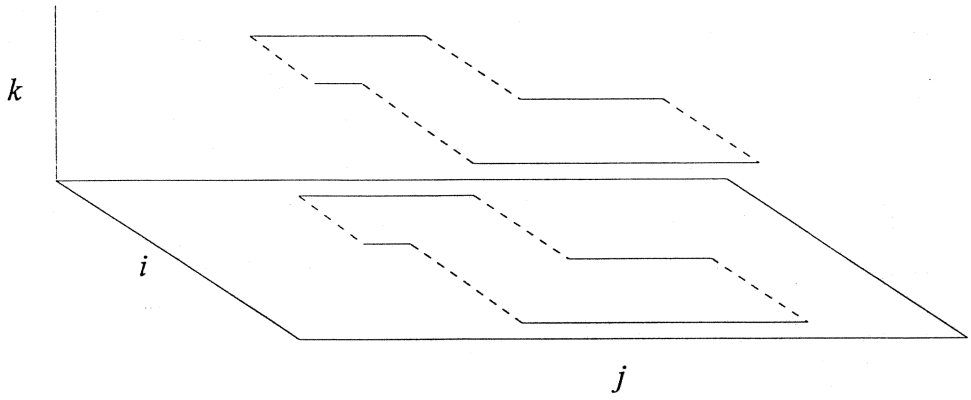
The simple algorithm described above often fails, and a more advanced algorithm is required. The following method is adapted from Cox's' *unbiased controlled rounding* method. Cox (1987) describes a method used to perform a rounding on a two dimensional array which contains cells which have a value of either 0 or a real number between 0 and 1, with integer totals.

The Cox method relies on the fact that each node in the path has 1 'partner' in each dimension; that is, for any given node there will be one node on the same row, and one on the same column. The path is constructed so that any change made to a node will not affect the totals because there will be an equal and opposite change made to the corresponding nodes in both the row and column vectors. When this method is extended into 3 dimensional space a problem arises: any node must, in the same manner, have 'partner' nodes on all vectors (i.e. for any node there should be one node on the same row, one on the same column and on in the same shaft).

In 2 dimensional space it is possible to construct an alternating row-column path to satisfy the method requirements, because any move made (either changing row or column) will correct one total and disrupt a second, which will then be corrected by the next move, until the starting row or column is reached, when the initial error is corrected. However, when a move is made in 3d space, *two* other vectors are disrupted, and clearly a simple extension of the alternating path method can not ameliorate this situation.

A double path is constructed (Figure 11), so that each node in the 3 dimensional space has one (and one only) matching node in each dimension. Values of d- and d+ are derived, and one is picked at random. The nodes on the path are adjusted, with the 'direction of flow' in one path being opposite to the direction in the other.

Figure 11: Paths in a three-dimensional array



All values which have changed on the path are then examined to see whether they have become either 0 or 1 (or, as described above, whether they are sufficiently close to 0 or 1 to be considered as such). If a value has become 1, then Y^{ijk} and the totals arrays will be adjusted accordingly.

The procedure is repeated until either all values have been reduced to zero (i.e. a solution has been found), or until no more paths can be constructed (the 'fail' condition).

If the block fails, then the simple rounding algorithm is applied to the remainder of X^{ijk}. If this is successful then Xt^{ijk} will be added to Y^{ijk} as before, and the block will have been solved. However, this approach is not always successful, and so a 'back up' strategy is required. If the block can not be solved using the above methods, then an attempt is made to solve it as a series of 2d slices. In this case, only two of the three totals can be satisfied; in practice the *smsgaps* program will choose to ignore either the origin set of totals (i.e. producing a destination constrained solution), or ignore the destination set of totals (producing an origin constrained solution), subject to the value of a user-toggleable variable. Any two dimensional matrix with consistent totals should be capable of being solved in this way.

At this stage, any shafts for which the C2D_Ijk and/or D2C_iJk values are known should have been unsuppressed. However, a few shafts remain which so not have these totals available. These are solved using a final sweep through the array, building and solving 2D slices which contain all the 'unknowns' for any origin or destination district to/from the rest of Great Britain.

The above set of solution algorithms will generate results to which varying amounts of confidence may be attached. There are three levels of 'accuracy' - results which are known to be correct, results which satisfy all three sets of totals on a county-block level, and results which only satisfy two sets of totals. In the case of table M05, a typical program run results in 98.63% percent of migrants being in the first 'logically

correct' category. 0.70% being in the second 'three totals satisfied' category, and the remaining 0.67% being in the final 'two totals satisfied' category.

7. CONCLUSIONS

This paper has described a method which has made good a serious deficiency in the supply of 1991 Census migration data. The new full array can now be used for the proper analysis of migration in Great Britain (e.g. Rees and Duke-Williams 1995b) and will be deposited for general use by the academic community in the near future. Table 2 shows a typical output that can now be produced. The tables shows the flows of migrants in the Black ethnic group (Black Caribbean, Black African and Black Other) between districts aggregated to form metropolitan and non-metropolitan regions. The table is consistent and comprehensive unlike the counterpart that would be generated from the official SMS. The data remain confidential. No individual can be recognised from knowledge that only one Black person migrated between Great Manchester and Central Clydeside, for example. You could only recognise the person this refers to if you knew the facts of his or her migration and ethnicity in the first place.

The SMS data were purchased for academic use by the Economic and Social Research Council and the Joint Information Systems Committee of the Higher Education Funding Councils for just over £100,000. The ESRC and JISC were the sole purchasers of the whole national data set. A large portion of that value would have been lost if the data array had not been reconstructed. In negotiating for the purchase of migration data from the 2001 Census the academic community will be seeking to persuade the Census Offices to release the full migration array for districts without suppression.

Table 2: An aggregated flow matrix constructed from the new data: Black migrants between metropolitan and non-metropolitan regions, 1990-91

		Destination																
Origins		GL	GM	ME	SY	TW	WM	WY	SW	RS	EA	NW	NR	EM	WM	YH	WA	SR	CC
Greater London	GL	45697	116	44	49	10	199	98	177	1421	149	27	25	188	46	39	59	52	16
Greater	GM	115	2863	21	5	3	40	20	14	33	1	57	7	31	8	4	4	3	1
Manchester Merseyside	ME	70	22	886	2	0	17	1	6	12	3	18	1	7	4	3	6	1	0
South Yorkshire	SY	59	15	5	944	5	10	22	6	23	4	3	2	26	0	4	1	2	1
Tyne and Wear	TW	37	2	5	3	228	2	2	7	3	2	0	11	1	0	2	2	0	0
West Midlands	WM	234	28	10	12	4	5928	23	46	87	10	11	8	99	124	7	10	17	3
West Yorkshire	WY	107	34	4	16	2	18	1796	12	36	6	11	11	19	8	27	4	3	6
South West	SW	160	18	5	3	7	32	13	1924	135	68	14	6	16	13	8	26	9	0
Rest of South East	RS	1064	48	15	14	17	55	9	155	6502	137	15	12	128	36	20	25	23	0
East Anglia	EA	116	9	1	0	4	11	3	23	88	1657	2	3	36	4	6	3	8	8
North West Rem.	NW	45	56	21	2	0	11	8	8	21	8	479	2	7	7	8	5	0	0
North - Rem.	NR	39	6	3	7	23	9	7	4	11	1	5	177	3	1	9	5	4	0
East Midlands	EM	189	18	2	28	4	76	14	26	126	45	12	6	3032	28	7	6	10	3
West Midlands	WM	66	8	4	1	1	117	13	29	27	6	14	7	36	711	0	11	0	0
Yorks & Humb.	YH	18	15	1	1	3	7	23	9	21	3	6	6	13	3	278	4	6	1
Rem Wales	WA	75	18	7	5	1	9	4	29	23	2	12	2	11	7	2	957	6	0
Scotland - Rem.	SR	61	13	1	2	1	3	5	14	23	15	1	2	11	2	8	11	479	16
Central Clydeside	CC	39	4	0	0	3	0	2	4	13	0	0	2	13	2	0	1	21	212

REFERENCES

Cox L. (1987) A constructive procedure for unbiased controlled rounding. *Journal of the American Statistical Association*, 82, 520-524.

Flowerdew R. and Green A. (1993) Migration, transport and workplace statistics from the 1991 Census. Chapter 10 in Dale A. and Marsh C. (eds.) *The 1991 Census user's guide*. HMSO, London. Pp.269-294.

Quantime Limited (1995) *Special Migration Statistics Special Workplace Statistics Training Course in Quanvert Text*. Manchester Computing

Rees P. and Duke-Williams O. (1994) The Special Migration Statistics: a vital resource for research into British migration. *Working paper 94/20*, School of Geography, University of Leeds, Leeds, UK.

Rees P. and Duke-Williams O. (1995a) The story of the British special migration statistics. *Scottish Geographical Magazine*, 11, 13-26.

Rees P. and Duke-Williams O. (1995b) Inter-district migration by ethnic groups. Paper presented at the International Conference on Population Geography, University of Dundee, Dundee, 16-19 September, 1995.