Manual for ZO version 2.3

Tom A.B. Snijders [*]

July 2017

Abstract

ZO is a computer program that carries out simulation and/or enumeration analysis of graphs with given degrees. It is also possible to fix the number of mutual ties in a digraph and to restrict the analysis to connected graphs. The program can be run as a stand-alone program, or from within the StOCNET environment.

[*] I am grateful to Peter Boer and Mark Huisman for their cooperation in the development of the StOCNET program. Authors' address: Tom Snijders, ICS, Department of Sociology, Grote Rozenstraat 31, 9712 TG Groningen, The Netherlands, `t.a.b.snijders@rug.nl`.

Contents

1 ZO

ZO ('Zero-One') is a computer program that carries out simulation and/or enumeration analysis of graphs with given degrees according to the algorithms of Snijders (1991) and Molloy and Reed (1995). For undirected graphs the degrees are held constant; for directed graphs (digraphs) the in-degrees as well as the out-degrees are held constant; for digraphs also the number of mutual ties may be held constant; and it is possible to restrict the analysis to connected (di)graphs.

Such analyses can be useful, e.g., for testing the degree of reciprocity, the degree of transitivity, or the eigenvalue centrality of vertices, controlling for the observed degrees.

In addition, ZO can be used to simulate and/or enumerate square or rectangular 0-1 matrices with given marginal sums without structural zeros – which can be interpreted as bipartite graphs with given degrees; as well as such matrices with an arbitrary predetermined set of structural zeros.

This manual gives information about the version, ZO 2.3 (July 2002). The program was programmed in Delphi by Tom Snijders on the basis of earlier DOS programs. It can be run under Windows, see Section 8. The manual was updated in 2013 and again, slightly, in 2017, with improved explanation for the execution under Windows.

The program and this manual can be downloaded from the web site,
`http://www.stats.ox.ac.uk/~snijders/socnet.htm`

One way to run it is as part of the StOCNET program collection (Boer, Huisman, Snijders, & Zeggelink, 2002), which can be downloaded from website
`http://www.gmw.rug.nl/~stocnet/StOCNET.htm`
For the operation of StOCNET, the reader is referred to the corresponding manual.

This manual consists of two parts: the user's manual and the programmer's manual.

Part I
User's manual

The user's manual gives the information for using ZO. It is advisable also to consult the user's manual of StOCNET because normally the user will operate ZO from within StOCNET.

2 The purpose of ZO

The ZO package is for the determination of probability distributions of statistics of random (undirected) graphs with given degrees, and random digraphs with given in- and out-degrees. Such graphs and digraphs can be represented by their adjacency matrices, which are square matrices with 0 and 1 elements and with an all 0 diagonal. The program is called Z(ero)-O(ne) in an allusion to this representation by 0-1 matrices. The number of vertices is denoted by n. The matrix is denoted

$$X = \Big(X_{ij}\Big)_{1 \leq i,j \leq n} \ ,$$

where X_{ij} is 0 or 1 accordingly to whether there is not, or there is, an arc from vertex i to vertex j. It is assumed that there are no self-relations: $X_{ii} = 0$ for all i. In other words, the diagonal of the adjacency matrix is structurally zero.

The in- and out-degrees are just the row and column sums of this adjacency matrix. Using a + sign for summation over the index, they are denoted by X_{+i} and X_{i+} , respectively. The vertices of the graph or digraph correspond to the rows and columns of the adjacency matrix. For undirected graphs the adjacency matrix is symmetric, and there is no distinction between in- and out-degrees. Instead of arc, the term edge then also is used.

In addition, ZO also can determine the distribution of statistics for more general random 0-1 matrices with given row and column sums: both for arbitrary rectangular 0-1 matrices, and for rectangular 0-1 matrices with the restriction that a given set of cells has all 0 entries, the so-called *structural zeros*. For graphs and digraphs, the set of structural zeros is the diagonal of the matrix.

For graphs, the extra requirement can be imposed that they be connected; for digraphs, that they be weakly connected. A graph (digraph) is disconnected (i.e., *not* connected) if there are at least two non-empty subsets of vertices without edges (arcs) between them. For digraphs, the extra requirement can be imposed that also the number of mutual dyads

$$M = \sum_{i<j} X_{ij} \, X_{ji} \tag{1}$$

be equal to a given number.

In all these cases the distribution of random 0-1 matrices considered in ZO is uniform, i.e., each matrix satisfying the restrictions has the same probability (cf. Wasserman and Faust, 1994, Chapter 13). The ZO package contains two programs: the most important is zo_sim for analysis by Monte Carlo simulation; in addition there is the program zo_enum for analysis by enumeration, but this is feasible only for very small graphs (no more than 7 to 9 vertices, depending on the distribution of the degrees and the speed of the computer). E.g., for nine vertices with the in-degree sequence (2, 2, 2, 2, 2, 2, 2, 1, 1) and the out-degree sequence (2, 2, 2, 2, 2, 1, 1, 2, 2), the total number of possible digraphs is 132,272,868. This is close to the limit of what can still be enumerated by zo_enum. The main purpose of zo_enum is to check the results of zo_sim – and, of course, the fun of being able to enumerate such a set of matrices.

2.1 Various testing problems, conditional on degrees

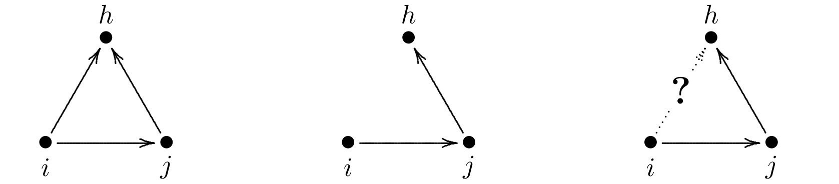
The main purpose in Social Network Analysis for using graph distributions conditional on the degrees, is to test structural properties of an observed network, while controlling for the degrees. The importance of this is discussed, among others, in Katz and Powell (1957), Wasserman (1977), Rao and Bandyopadhyay (1987), Snijders (1991), and Roberts (2000).

2.1.1 Testing reciprocity in digraphs

A first testing problem for directed relations is to test whether reciprocated choices occur more frequently than expected by chance, given the in- and out-degrees. The number of mutual dyads M is the obvious test statistic. Controlling for the in- and out-degrees means that, in the notation commonly used in social network analysis (cf. Wasserman and Faust, 1994, Chapter 13), this statistic is tested in the $\mathcal{U} \mid X_{i+}\,,\, X_{+i}$ distribution. The p-value for testing reciprocity, conditional on the in- and out-degrees, is the probability of obtaining in the $\mathcal{U} \mid X_{i+}\,,\, X_{+i}$ distribution a value of M at least as large as the value observed in reality. This probability (estimated by simulation) can be found in the output of ZO.

2.1.2 Testing transitivity in digraphs

Testing transitivity in digraphs under the $\mathcal{U} \mid X_{i+}\,,\, X_{+i}$ and $\mathcal{U} \mid X_{i+}\,,\, X_{+i}\,,\, M$ distributions is discussed extensively in Karlberg (1999). Transitivity of choice in directed relations means that the presence of ties $i \rightarrow j$ and $j \rightarrow k$ increases the likelihood of the existence of the tie $i \rightarrow k$. A triplet, i.e., an ordered triple of vertices (i, j, k), is called positive transitive if $X_{ij} = X_{jk} = X_{ik} = 1$; intransitive if $X_{ij} = X_{jk} = 1$, $X_{ik} = 0$; vacuously transitive if $X_{ij} = 0$ or $X_{jk} = 0$; and potentially transitive if $X_{ij} = X_{jk} = 1$. Note that in this definition, and in the picture, the existence of arcs from j to i, k to j, or k to i, is immaterial; that is why the terminology is about *triplets* rather than *triads*.



Transitive triplet Intransitive triplet Potentially tr. triplet

Transitivity may usually be considered as an effect of higher order than reciprocity. Therefore, it makes sense in a test for transitivity to control not only for the degrees but also for the number of mutual dyads. Controlling for the observed in- and out-degrees and also for the observed number of mutual dyads means that the $\mathcal{U} \mid X_{i+}\,,\, X_{+i}\,,\, M$ distribution defines the null hypothesis.

Transitivity can be measured by the numer of positive transitive triplets,

$$Tr \;=\; \sum_{i,j,k} X_{ij}\, X_{jk}\, X_{ik}\ ,$$

but also, inversely, by the number of intransitive triplets,

$$Intr \;=\; \sum_{i,j,k;\, i\neq k} X_{ij}\, X_{jk}\, (1 - X_{ik})\ .$$

The sum of these, $Tr + Intr$, is the total number of potentially transitive triplets. A measure for transitivity which is attractive because it is always between 0 and 1 is defined by the ratio

$$R_{\text{trans}} = \frac{Tr}{Tr + Intr} \tag{2}$$

of positive transitive to potentially transitive triplets. For undirected graphs, this index was used also by Frank and Harary (1982) and by Karlberg (1997) – but without conditioning on the degrees. For directed graphs, and with conditioning on the degrees, it was used by Karlberg (1999, see p. 230).

If we control for the degrees and the number of mutuals, these three are equivalent test statistics, since the sum $Tr + Intr$ is a function of the degrees and the number of mutuals:

$$\begin{aligned} Tr + Intr &= \sum_{i,j,k;i\neq k} X_{ij}\, X_{jk}\, (X_{ik} + 1 - X_{ik}) && (3) \\ &= \sum_{i,j,k} X_{ij}\, X_{jk} - \sum_{i,j} X_{ij}\, X_{ji} = \sum_{i,j,k} X_{ij}\, X_{jk} - M && (4) \\ &= \sum_{j} X_{+j}\, X_{j+} - M\ . && (5) \end{aligned}$$

Therefore, if we condition only on the degrees, which implies that M remains a random variable, then there can be slight differences between the p-values produced by these three test statistics. But if we condition on the degrees as well as the number of mutuals, then the p-values obtained for these three test statistics are equal.

The enumeration program in ZO presents the distributions of Tr and $Intr$; the simulation program presents the (estimated) distribution of the ratio R_{trans}. Depending on the specification given by the user, this is the distribution under the $\mathcal{U} \mid X_{i+}, X_{+i}, M$ or the $\mathcal{U} \mid X_{i+}, X_{+i}$ distribution. From this distribution, the p-value for the test of transitivity can be obtained as the probability of obtaining a random value for the test statistic Tr or R_{trans} which is at least as large as the value observed in reality.

2.1.3 Testing transitivity in undirected graphs

In undirected graphs, reciprocity is not an issue – it is a definition. Transitivity can be defined and tested as in the discussion above, with the restriction that all ties are reciprocated by definition ($X_{ij} = X_{ji}$ for all i, j). The transitivity ratio R_{trans} was proposed already by Frank and Harary (1982) for testing transitivity in a graph. ZO can be used to test transitivity in undirected graphs, conditioning on all degrees, analogous to what was explained above. The null distribution canbe indicated as the $\mathcal{U} \mid X_{i+}$ distribution for undirected graphs.

2.1.4 Testing eigenvector centrality

Centrality of vertices in graphs can be measured in various ways, cf. Bonacich (1972a,b) and Freeman (1979). Degrees are one way. Another way is the eigenvector centrality defined by Bonacich (1972a) for undirected graph as the first eigenvector of X, and by Bonacich (1972b) for bipartite graphs as the first eigenvector of $X'X$. It is argued in Bonacich et al. (1998) that these eigenvector centrality value tend to be highly correlated with the degrees, and therefore it can be advisable to control their values for the degrees. This can be done using ZO, for bipartite as well as undirected and directed graphs. For directed graphs, however, the matrix is symmetrized before calculating the eigenvector centralities. Eigenvector centralities can be requested, and their null distributions calculated under the probability distributions available in ZO. Examples are presented in Bonacich et al. (1998). Since eigenvector centralities have the most clear interpretation for graphs with only one connected component, it may be advisable when using these centralities to restrict the simulations to connected graphs.

The eigenvector centralities in ZO are calculated differently for square matrices and for non-square matrices.

For square matrices, they are defined as the elements of the eigenvector belonging to the largest eigenvalue of $X^{(s)}$, where $X^{(s)}$ is the symmetrized version of the matrix X. The symmetrization is defined by

$$X_{ij}^{(s)} = \max\{X_{ij},\, X_{ji}\}\ .$$

The mentioned eigenvector is the vector a which is the solution of

$$X^{(s)}a = \lambda a$$

where λ (the eigenvalue) is the largest real number for which this system of equations has a solution. The eigenvector a is normed in such a way that the sum of squared elements of a (i.e., its squared Euclidean length) is equal to the square of λ.

For non-square matrices, the eigenvector centralities are defined as the elements of the eigenvector belonging to the largest eigenvalue of $X'X$. In this case, the eigenvector is normed to have unit length.

2.2 Other statistics

A number of descriptive graph statistics are programmed in ZO and are calculated for all generated graphs. For users interested in other statistics, several options are open. Those who can program in Delphi could extend the Delphi code (see Section 10). Another possibility may be to propose to Tom Snijders to include important additional statistics in the next version of ZO. Finally, as mentioned in Section 5.4, it is possible to write all generate matrices to a file. This can then be used to calculate the desired statistics; to estimate distributional properties, it will be necessary to take into account the weights that are also written to this file.

Number of digraphs with the given degrees

The number of digraphs with given degrees, or, more generally, the number of 0-1 matrices with given marginal sums and perhaps a given set of structural zeros, can also be calculated by different means than complete enumeration. This provides a way of independently checking the enumeration algorithm. The Enumerate option will calculate some formulae for obtaining these numbers. Which formulae are calculated depends on the type of matrix (see Section 5.1: unrestricted, digraph, graph, or general). For undirected graphs no calculations of this kind are made.

For the other types of matrix, an approximate formula of Bender (1974) is calculated that approximates the number of 0-1 matrices with given marginals and a given set of structural zeros. This formula is a good approximation if the matrix is sparse (i.e., has a small fraction of 1 entries) and if the number of rows and columns is large.

For unrestricted rectangular matrices, Sukhatme's (1938, p. 386–390) formula is calculated that gives the exact number of 0-1 matrices with these marginals.

For digraphs, it is possible to request the calculation of the formula of Katz and Powell (1954, 1957) for the number of digraphs with these in- and out-degrees. However, this is a highly recursive formula and the program unfortunately is liable to run into an error except for quite low values of the degrees. Therefore this option is available only as a curiosity for the cases where it does work, and not intended for general use.

3 Operation of the program

The program works very simply. What it needs from the user is the specification of the type of matrix (i.e., type of graph) and the degrees. The degrees can be given by means of a whole matrix (from which the degrees then will be copied), or just by giving a file with only the degrees; see Section 6.1.

The user must choose between the two options, Simulate and Enumerate. The first option, simulation, is the most important from a practical point of view.

The second option enumerates all matrices satisfying the constraints given. Sometimes it is helpful for understanding the structure of the $\mathcal{U} \mid X_{i+}, X_{+i}, M$ or $\mathcal{U} \mid X_{i+}, X_{+i}$ distribution to know exactly all possible graphs for a given sequence of degrees, and this is what the Enumeration option does. However, this is feasible only if the numbers of rows and columns are very small. ZO automatically disables the Enumerate option for more than 15 rows or columns, but this is a very high threshold. Usually about 8 rows and columns are the maximum for which enumeration is a practical option (the exceptions are some very skewed degree distributions, where out-degrees are strongly negatively correlated with out-degrees). Otherwise the number of possible adjacency matrices just is too large.

For the Simulate option, it is necessary to specify how many simulation runs should be carried out. The default value in StOCNET is 10,000.

During the operation of the program, you see a counter that indicates how many matrices were produced up to now. For the simulation option, another counter is shown that indicates how many attempts were made to simulate a matrix. This is because the algorithm can make trials that fail – after which it tries again. This happens especially often when simulating digraphs with a fixed number of mutual dyads, if this number is relatively high or low given the degrees. It often happens often when employing the Molloy-Reed algorithm (cf. Section 4 about the algorithms used).

It is possible to request early termination of the program. With the Enumerate option, this means that no meaningful results are obtained – the population then has been enumerated only incompletely. With the Simulate option, it implies only that less simulations were carried out than originally planned, so the precision of the results is less, but the results are stil meaningful.

The program writes results to an ASCII file. This includes the mean, variance, standard deviation, a histogram, and the cumulative distribution function of a number of graph statistics. To obtain a unified and relatively simple output layout, the program focuses on statistics with integer values from 0 to 100. The statistics are described in Section 5.2.

4 Algorithms

4.1 Enumeration

The enumeration algorithm in the Enumerate option constructs the enumeration tree as discussed in Snijders (1991). For constructing each consecutive matrix, only the differences with respect to the previously constructed matrix are traced, which makes for a very fast algorithm. The constraints of the integer type in Delphi programming imply that no more than 2,147,483,647 matrices can be enumerated. To give an impression how many digraphs there are with given degrees, the following table gives, for digraphs with g vertices and all in- and out-degrees equal to 1, the number N of digraphs and the result of Bender's (1974) approximate formula. This illustrates the high degree of precision of Bender's formula for sparse digraphs already for a low number of vertices. For higher average degrees there can be considerable deviations between this formula and the exact number.

g	Bender	N
7	1,854	1,854
8	14,833	14,833
9	133,496	133,496
10	1,334,961	1,334,961
11	14,684,570	14,684,570
12	176,214,841	176,214,841
13	2,290,792,932	??

The enumeration possibilities are limited by the computing time necessary for enumerating these possibly huge numbers of matrices; and by the depth of the enumeration tree (a constant in the program).

4.2 Simulation

The combinatorial difficulties of constraining both the row sums and the column sums of a 0-1 matrix are such that there is no straightforward direct algorithm for randomly simulating such matrices. ZO offers the choice between two simulation algorithms. In both algorithms, there is the possibility that at some point an attempted construction of a matrix with the given constraints has to be discarded, because the constraints cannot be satisfied given the cells of the matrix determined up to this point. This will then be indicated in the ZO Simulation interface by the number of attempts being greater than the number of simulated matrices. This has no consequences for the quality of the obtained results except that it will increase the computing time required for generating a given number of matrices.

The first algorithm is the one proposed by Snijders (1991). It generates matrices not with uniform but with varying probabilities. The varying nature of the probabilities is corrected when estimating the probability distribution of the statistics, so that (practically) unbiased results are obtained for the uniform distributions even though the sampling is not uniform. This is discussed in Snijders (1991) and, less extensively, in Karlberg (1999). The former reference also indicates how the sampling method is defined in view of the purpose of having probabilities that are not strongly variable. The inverses of the probabilities are used as weights in the estimation of the probability distribution (see equation (6). If these inverse probabilities are extremely highly variable, the estimates are averages of a very long-tailed distribution, which may lead to unreliable estimation. In that case, a warning message is printed in the output file.

For graphs and digraphs the rows and columns are reordered to increase efficiency. (Of course, rows and columns are reordered in the same way to retain the graph structure.) The reordering is so that the value of $\mid g - 1 - 2x_{i+} \mid$ is non-increasing, where g is the number of vertices. If this is done, it is mentioned in the output and the new vertex order is given. The statistics defined in Section 5.2 that refer specifically to vertices 1 and 2 are not affected by this reordering, i.e., they refer to the vertices that had number 1 and 2, respectively, in the original order.

The second algorithm was proposed by Molloy and Reed (1994). It starts with generating graphs, digraphs or bipartite graphs allowing loops (i.e., edges or arcs from a vertex to itself) and multiple edges or arcs. In other words, a matrix is generated with nonnegative integer elements without the constraint of the structural zeros (the diagonal in the case of graphs and digraphs) and without the constraint that the elements may not be larger than 1. Constructed matrices with 1 entries in at least one of the structurally zero cells or with entries larger than 1 are discarded ('failed attempts'). Molloy and Reed (1994) proved that for sparse undirected graphs, the fraction of failed attempts will be relatively low, and in this case this algorithm will be more efficient than that of Snijders. For directed graphs, however, the Molloy-Reed algorithm can lead to a fraction of failed attempts close to 100%, and therefore can be very inefficient. The interpretation is that for symmetric sparse 0-1 matrices, the structurally zero diagonal is not a very severe restriction, but for non-symmetric 0-1 matrices it is.

The advice for directed graphs therefore is to use the Snijders algorithm. For undirected graphs both algorithms can be used; if the graph density is low, the Molloy-Reed algorithm will be more efficient. If one wishes to have an actual random sample from the uniform distribution with given marginals for which no reweighting is necessary (a sample which can be made available by writing it to file as indicated in Section 5.4), then the Molloy-Reed algorithm is the only option.

Another algorithm exists which is not implemented in ZO. This is a Markov chain Monte Carlo algorithm proposed by Rao, Jana, and Bandyopadhyay (1996) and explained also by Roberts (2000).

5 Options

The previous section already discussed the choice between Simulate and Enumerate, and mentioned the number of simulation runs that must be specified. But there are more options.

5.1 Distributions

There is a choice between four types of matrix, or of graph:

1. Unrestricted: an arbitrary rectangular matrix – in other words, the adjacency matrix of a bipartite graph.
2. Directed graph: a square matrix with structurally zero diagonal.
3. Undirected graph: a symmetric square matrix with structurally zero diagonal.
4. General: an arbitrary rectangular matrix with also an arbitrary set of structural zeros. The previous three cases are obviously special cases of this general form. For this type of matrix, it is required that the user specifies the set of structural zeros by means of an adjacency matrix, where the '1' entries indicate the locations of the structural zeros.

The StOCNET interface will make a guess for this choice on the basis of the input provided – on the basis of whether there are as many rows and columns, or not.

In all cases, the degrees are fixed. For undirected graphs there is only one vector of degrees. For the other three types of matrix, where symmetry is not imposed, the in-degrees as well as the out-degrees are fixed.

5.1.1 Extra requirements

Some extra requirements can be imposed. For these extra requirements, the generation of the matrices is done just like the generation for fixed in- and out-degrees; after generating a matrix by this method, it is checked whether it satisfies the extra condition, and the matrix is retained only if it does. This means that there will usually be a lot of rejected matrices, implying that the algorithm will take longer.

When the matrix specified is a directed or undirected graph, it is possible to restrict the simulations or enumerations to *connected* graphs or digraphs. A graph is connected if for every two vertices there is a path (a sequence of joined edges) connecting them. For digraphs, the restriction is to *weakly connected* digraphs, which is defined similarly but without regard for the directions of the arcs making up the path.

For directed graphs, it is possible to request that the number of mutual dyads,

$$M = \sum_{i<j} X_{ij} \, X_{ji} \ ,$$

is fixed. Note that the user may request a number of mutuals that is incompatible with the given degree sequences. This compatibility is not always easy to verify. The enumeration option will find out whether the degrees and the number of mutuals are compatible (in case of incompatibility, it will find no matrices at all). The simulation option will find this out only in extreme cases; in other cases it just keeps on trying, the number of attempts increases more and more and the number of simulated matrices does not lift off...

5.2 Choices for statistics

For several statistics, the mean, variance, standard deviation, and cumulative distribution function is calculated (exactly by Enumerate, by Monte Carlo approximation by Simulate). This is done for a set of statistics; the user can choose between several sets. Each set of statistics is called a *version*. The available statistics are different for Simulate and Enumerate, and also are different depending on whether the generated matrices are square (as many rows as columns – this is always the case for graphs and digraphs) or non-square rectangular.

Note that the output format of ZO is such that all statistics are treated as having integer values, often bounded by the values 0 and 100.

5.2.1 Statistics for Simulate

For square matrices, the following sets of statistics are available.

i. Version 1.
   1. Number of mutual dyads as defined in (1);
      for undirected graphs without the restriction to connected graphs, this is replaced by the connectivity indicator (1 for connected, 0 for disconnected graphs).
   2. Transitivity index (2) multiplied by 100;
   3. Indicator of the arc from vertex 1 to vertex 2 (1 if the arc is present, 0 if it is absent).

ii. Version 2.
   For each vertex the eigenvector centrality defined in Section 2.1.4, multiplied by 100. (The product is rounded to an integer value.)

iii. Version 3.
   For each vertex the eigenvector centrality defined in Section 2.1.4, multiplied by 1,000. (The product is rounded to an integer value.)
   When the eigenvector centralities tend to be less than 0.1, version 3 will be more informative (less information loss due to rounding) then version 2, since the cumulative distribution function in the output file is shown only for values up to 100.

iv. Version 4.
   1. Indicator of the arc from vertex 1 to vertex 1 (1 if the arc is present, 0 if it is absent). (For graphs and digraphs, this always is 0).
   2. Indicator of the arc from vertex 1 to vertex 2.
   3. Number of matches between rows 1 and 2: this is the number of vertices which have an incoming arc from vertex 1 as well as vertex 2; in matrix terminology, the number of columns j such that there is a 1 entry in cells $(1, j)$ as well as $(2, j)$. Columns $j = 1$ and 2 also are counted (of course, these columns do not contribute for graphs and digraphs.)

ix Version 9.
   For directed graphs, the code 9 will calculate the statistics for version 1, but also attempt to calculate the formula of Katz and Powell (1954, 1957) for the number of digraphs with the given in- and out-degrees. It was mentioned in Section 2.2 that there is a possiblity here that the program will run into an error.

For non-square matrices, the following sets of statistics are available.

i. Version 1.
   1. Indicator of the arc from vertex 1 to vertex 1 (1 if the arc is present, 0 if it is absent).
   2. Indicator of the arc from vertex 1 to vertex 2.
   3. Number of matches between rows 1 and 2 (defined as in option iv above).

ii. Version 2.
   For each vertex the eigenvector centrality defined in Section 2.1.4, multiplied by 100. (The product is rounded to an integer value.)

iii. Version 3.
For each vertex the eigenvector centrality defined in Section 2.1.4, multiplied by 1,000. (The product is rounded to an integer value.)

5.2.2 Statistics for Enumerate

For square matrices, the following sets of statistics are available.

i. Version 1.
   1. Number of mutual dyads as defined in (1);
      for undirected graphs without the restriction to connected graphs, this is replaced by the connectivity indicator (1 for connected, 0 for disconnected graphs).
   2. Number of transitive triplets *Tr*.
   3. Number of intransitive triplets *Intr*.
   4. Transitivity index (2) multiplied by 100.

ii. Version 2.
For each vertex the eigenvector centrality defined in Section 2.1.4, multiplied by 100. (The product is rounded to an integer value.)

iii. Version 3.
   1. Indicator of the arc from vertex 1 to vertex 1 (1 if the arc is present, 0 if it is absent). (For graphs and digraphs, this always is 0).
   2. Indicator of the arc from vertex 1 to vertex 2.
   3. Number of matches between rows 1 and 2, defined as above.

For non-square matrices, the following sets of statistics are available.

i. Version 1.
   1. Indicator of the arc from vertex 1 to vertex 1 (1 if the arc is present, 0 if it is absent).
   2. Indicator of the arc from vertex 1 to vertex 2.
   3. Number of matches between rows 1 and 2 (see above).
   4. Number of matches between columns 1 and 2.

ii. Version 2.
For each vertex the eigenvector centrality defined in Section 2.1.4, multiplied by 100. (The product is rounded to an integer value.)

5.3 Linear combinations of the triad census

In addition to the statistics mentioned above, in Simulate it is also possible to determine the distributions of linear combinations of the triad census, as defined in Holland and Leinhardt (1975) and described extensively in Wasserman and Faust (1994), Chapter 14. This is meaningful only for digraphs, since for graphs under the $\mathcal{U} \mid X_{i+}$ distribution, the information in the triad census is equivalent to the number of transitive triads, which is available already in the statistics presented in the previous section.

In Holland and Leinhardt (1975) the $\mathcal{U} \mid M, A, N$ distribution is considered for the triad census. In ZO, of course, the triad census is considered under the $\mathcal{U} \mid X_{i+}\,,\, X_{+i}$ and the $\mathcal{U} \mid X_{i+}\,,\, X_{+i},\, M$ distributions. Mostly when testing propositions expressed by the triad census, it is advisable to control for the dyad census M, A, N, or for more statistics – but not for less. Therefore the $\mathcal{U} \mid X_{i+}\,,\, X_{+i},\, M$ distribution will be in most cases more relevant for testing linear combinations of the triad census than the $\mathcal{U} \mid X_{i+}\,,\, X_{+i}$ distribution (note that the dyad count M, A, N is not a function of the degrees only, but it is a function of the degrees together with M).

ZO calculates the number of triads of each of the 16 equivalence classes defined in Holland and Leinhardt (1975) and also given, e.g., in Wasserman and Faust (1994). The triad names are given in the usual way (a code starting with three digits indicating the number of mutual, asymmetric, and null dyads in the triad; plus a letter if that is required for identification). The user of ZO can calculate the number of occurrence of the 16 triads (the 'triad census') but also, if desired, linear combinations of the triad census. The order in which the triads are given in the triad census is as given here. (The literature mentioned gives the meanng of these codes.) When a linear combination of the triad census is desired, the weights should be given according to this order. If a given number k of linear combinations is requested but no matrix of weights is supplied to ZO, then the numbers of occurrences of the first k triads are calculated. (In other words, the default value for the k'th weight vector is the k'th unit vector.)

1.	003
2.	012
3.	102
4.	021D
5.	021U
6.	021C
7.	111D
8.	111U
9.	030T
10.	030C
11.	201
12.	120D
13.	120U
14.	120C
15.	210
16.	300

For the linear combinations specified, the means and the covariance matrix is calculated. In addition, the probability is calculated that the linear combination is at least as large as a given threshold. This can be used for testing network propositions using linear combinations of the triad census as test statistics, and the $\mathcal{U} \mid X_{i+}\,,\, X_{+i}$ or $\mathcal{U} \mid X_{i+}\,,\, X_{+i},\, M$ distribution as the null distribution.

5.4 Writing all generated matrices to file

If desired, the user can write all generated matrices to a file. Note that this may lead to very long files. The file is self-explaining. This can be useful, e.g., if you wish to check the definition of the statistics: this file contains not only the generated matrices but also the corresponding calculated statistics.

E.g., for the enumeration of directed graphs with 7 vertices and all degrees equal to 1, the start of such a file is as follows. For obtaining this output, version 1 of the statistics was used (see Section 5.2).

```
This file contains all found 0-1 matrices
with the desired row and column totals.
Mutual dyads are indicated by M.

Row totals :      1  1  1  1  1  1  1
Column totals :   1  1  1  1  1  1  1
Directed graph.
Following each matrix are the following function values:
 1. Number of mutual dyads
 2. # of transitive relations in triads
 3. # of intransitive relations in triads
 4. 100 * #trans / (#trans + #intrans)
```

```
1.
XM00000
MX00000
00XM000
00MX000
0000X10
00000X1
000010X
*   2.00   0.00   3.00   0.00
```

```
2.
XM00000
MX00000
00XM000
00MX000
0000X01
00001X0
000001X
*   2.00   0.00   3.00   0.00
```

In all matrices, the letter M should be read as a 1 entry with the additional information that it is part of a mutual dyad. The letter X stands for a structural zero. E.g., in the first generated matrix it can easily be seen that each row and each column has one 1 entry, and that there are two mutual dyads, as indicated also by the first mentioned function value. Each set of function values is preceded by the symbol *.

For the simulation option, this file contains for every generated matrix one additional piece of information. This is, up to a multiplicative constant, the inverse of the probability of generating this particular matrix. Thus, each matrix x satisfying the given constraints has a probability $p(x)$ of being generated, and the number $c/p(x)$ is given for some constant c (c is independent of x). The reason for incorporating the number c is given in Snijders (1991). If the user wishes to estimate the expected value of some function $f(x)$ which is not included in the statistics mentioned in Section 5.2, then this can be done as follows. Denote the number to be estimated by

$$\mu = \mathcal{E} f(X) ,$$

and the values $c/p(x)$ by $w(x)$ (interpreted as a weight). Then from the file with the generated matrices, the user has to calculated the function $f(x)$ for all matrices listed, and estimate μ by

$$\hat{\mu} = \frac{\sum w(x)\, f(x)}{\sum w(x)} , \tag{6}$$

where the sums in numerator and denominator extend over all generated matrices x.

The start of the file with the generated matrices may be as follows (the precise values depending on chance...).

```
This file contains all found 0-1 matrices
with the desired row and column totals.
Mutual dyads are indicated by M.

Row totals :      1  1  1  1  1  1  1
Column totals :   1  1  1  1  1  1  1
Directed graph.
Following each matrix are the following function values:
 1. Number of mutual dyads
 2. 100 * #trans / (#trans + #intrans)
 3. Arc indicator from 1 to 2

After this follows a constant divided by the probability
for obtaining this matrix.
For unbiased estimation, weights must be used
that are proportional to these numbers.

1.
X000010
0X00100
01X0000
100X000
0010X00
00000X1
000100X
@   0.00   0.00   0.00     0.98843815

2.
X100000
0X01000
00X0M00
100X000
00M0X00
00000XM
00000MX
@   2.00   0.00   1.00     0.77553903
```

Each set of function values is preceded by the symbol @. E.g., for the first generated matrix, the number of mutual dyads is 0; 100 times the transitivity index R_{trans} is 0; the arc indicator x_{12} is 0; and the weight $w(x) = c/p(x)$ is equal to 0.98843815.

6 Input – Output

6.1 Input data

The main information needed by the program is the degrees; for an undirected graph this is one degree vector, for the other (non-symmetric) options two degree vectors are required: in-degrees (column sums) and out-degrees (row sums). The degrees can be given to ZO in two ways: either as a full adjacency matrix from which ZO will calculate the degrees, or as a file containing the in-degrees in the first line, and (for the non-symmetric options) the out-degrees in the second line. For either way of giving the data, the data file must be an ASCII file containing integer numbers separated by blanks (not by tabs). For the adjacency matrix format, each row of the adjacency matrix must be terminated by a hard return.

Missing data are not allowed. In the adjacency matrix format, each positive entry in the data file is converted to a `1` (presence of an arc), each non-positive entry to a `0` (absence of an arc).

ZO can be executed as a stand-alone program under Windows. How to do this is explained in Section 8. When using the StOCNET interface, the StOCNET program will automatically recognize whether data is given as an adjacency matrix or as the vectors of in-and out-degrees, and pass this information on to ZO.

If the general option is used (see Section 5.1), a file with the positions of the structural zeros is also needed. This must be an adjacency matrix of the same dimensions as the matrices to be generated; the `1` values in this matrix have to indicate the positions of the structural zeros. It is read according to the same specifications as the adjacency input matrix mentioned above.

If linear combinations of triad counts are requested with other than unit weights (See Section 5.3), then also a matrix is required with the weights. This must be a matrix with as many lines as the number of requested linear combinations. Each line must contain 16 real numbers, separated by blanks, giving the weights of the 16 types of triad in the order given in Section 5.3. After this, the line may contain a 17^{th} number which is the critical value for the linear combination. The probability that the linear combination is larger than or equal to this critical value will also be estimated. It is allowed, however, that the line ends immediately after the 16^{th} number, in which case a default critical value of 1 is used, or the observed value in the input graph (if a full adjacency matrix is given as input).

6.2 Output file

The output file is self-explaining. It is an ASCII file. The output is divided in sections using the symbol @ followed by a number. The number indicates the sectioning level. The name of the output file is *pname*`.out`, where *pname* is the project name. If all matrices are written to file, then this is to a file named *pname*`.prn`. For both kinds of output file, if ZO discovers that a file exists with this name, it will append the new output to the existing file. If an option is chosen with many calculated statistics, intermediate results will be written to a file named *pname*`.pqr`. This does not require the user's attention.

7 Examples

As a first example, consider a very small digraph: with 7 vertices and all in- and out-degrees equal to 1. The output file of Enumerate for such an input data file (of which the degrees are given in the file zo7dma.dat distributed with the ZO program) and with version 1 for the evaluated statistics contains, among others, the following results.

```
@2
Enumeration results.
--------------------

Total number of matrices found      1854

@3
Number of mutual dyads
Mean                 0.4984
Variance             0.4765
Standard deviation   0.6903

Outcome        Number of matrices
      0              1140
      1               504
      2               210

0.6250|
0.5000|  XX
0.3750|  XX
0.2500|  XX  XX
0.1250|  XX  XX
0.0000|  XX  XX  XX
      +------------
          0   1   2
```

Thus, there are 1854 digraphs having these in- and out-degrees; 1140 of them have no mutual dyads, 504 have one, and 210 have two mutual dyads. The corresponding probabilities for the distribution of the number of dyads are 1140/1854 = 0.615 for 0 mutual dyads, 504/1854 = 0.272 for 1 mutual dyad, and 210/1854 = 0.113 for 2 mutual dyads. These three probabilities are indicated (roughly) in the simple histogram.

To check this with the outcomes of the simulation algorithm, the same data set can be used for the Simulate option. With 1000 simulation runs carried out, I obtained the following result for the number of mutual dyads.

```
@3
Number of mutual dyads
Estimated mean                 0.5198     (standard error   0.0231 )
Estimated variance             0.4998
Estimated standard deviation   0.7069

0.6250|
0.5000|  XX
0.3750|  XX
0.2500|  XX  XX
0.1250|  XX  XX  XX
0.0000|  XX  XX  XX
      +------------
          0   1   2

Minimum value found 0; maximum 2 .

Estimated cumulative ( <= ) probabilities with standard errors.
    0      1      2
 0.6052 0.8749 1.0000
 0.0159 0.0109 0.0000
```

The simulated mean is about one standard error different from the exact mean obtained from the enumeration, which is a reasonable deviation. The output of the simulation does not give the estimated probabilities of each possible value, but the estimated cumulative probabilities. The first cumulative probability is, of course, the same as the probability of the value 0. Here also, the estimated value 0.6052 is about one standard error different from the exact value $1140/1854 = 0.6149$. The second cumulative probability is $\mathrm{P}\{M \leq 1\}$, estimated as 0.8749, with an exact value $(1140 + 504)/1854 = 0.8867$.

Now suppose that we would like to have the values precise up to about 0.001. This would mean a standard error of about 0.0005. The standard error for the first cumulative probability, 0.0159, is about 32 times to large. Since standard errors decrease proportionally to the square root of sample size, the sample size should be about $32^2 \approx 1{,}000$ times as big. With 1,000,000 runs (not more than a few minutes were required), the following result was obtained.

```
@3
Number of mutual dyads

Estimated mean                 0.4982     (standard error   0.0007 )
Estimated variance             0.4769
Estimated standard deviation   0.6906

0.6250|
0.5000|  XX
0.3750|  XX
0.2500|  XX  XX
0.1250|  XX  XX
0.0000|  XX  XX  XX
      +------------
          0   1   2

Minimum value found 0; maximum 2 .


Estimated cumulative ( <= ) probabilities with standard errors.
    0      1      2
 0.6152 0.8865 1.0000
 0.0005 0.0003 0.0000
```

It can be verified that the standard errors are about 30 times as small, and the standard errors of the cumulative probabilities are indeed 0.0005 and less. The errors are less than 0.001.

7.1 Test of reciprocity

In this and the following subsections, Krackhardt's data set on friendships of high-tech managers is used that is also used in Wasserman and Faust (1994) for the same questions. Therefore, the reader is referred to this book (Section 13.8.3 for the test of reciprocity) for more background information for this data set and these examples.

In the first place, a test for reciprocity is carried out by testing the number of mutual dyads, M, in the $\mathcal{U} \mid X_{i+}\,,\, X_{+i}$ distribution. The observed number of mutual dyads is 23. With 10,000 simulation runs, the following result was obtained.

```
@3
Number of mutual dyads

Estimated mean                14.2075     (standard error   0.0356 )
Estimated variance             3.6960
Estimated standard deviation   1.9225

      |
0.2100|               X
      |               X
0.1800|               X
      |             X X X
0.1500|             X X X
      |             X X X X
0.1200|             X X X X
      |           X X X X X
0.0900|           X X X X X
      |           X X X X X X
0.0600|           X X X X X X
      |         X X X X X X X
0.0300|         X X X X X X X
      |       X X X X X X X X X
0.0000| X X X X X X X X X X X X X X X X
      +--------------------------------
               11        16        21

Minimum value found 7; maximum 22 .

Estimated cumulative ( <= ) probabilities with standard errors.
    7      8      9     10     11     12     13     14     15     16
 0.0001 0.0005 0.0067 0.0261 0.0744 0.1851 0.3571 0.5691 0.7453 0.8831
 0.0001 0.0002 0.0020 0.0031 0.0052 0.0075 0.0092 0.0095 0.0083 0.0054

   17     18     19     20     21     22
 0.9611 0.9881 0.9967 0.9993 1.0000 1.0000
 0.0029 0.0016 0.0008 0.0004 0.0000 0.0000
```

The observed number of 23 is so high under this null distribution, that such high values did not even occur once in 10,000 simulations. It can be concluded that there is a very significant tendency toward reciprocity, given the in- and out-degrees. Incidentally, the histogram shows that the null distribution of M has a shape rather resembling a normal distribution.
(In Wasserman and Faust, p. 551, a somewhat lower value for the expected value of M was found; this is a chance deviation, associated with the lower number of simulation runs used there, in view of the more limited computing power available at that time.)

7.2 Test of transitivity

Next, it is tested in the same data set if there is a tendency toward transitivity. Given that a strong tendency to reciprocity was found above, it makes sense to control not only for the degrees but also for the number of mutual dyads. This means that the null distribution will be the $\mathcal{U} \mid X_{i+}\,,\, X_{+i},\, M$ distribution, with $M = 23$.

The transitivity index R_{trans} of (2) for Krackhardt's friendship data set is equal to 0.46 (note that ZO reports this value multiplied by 100, so 46).

Simulating the $\mathcal{U} \mid X_{i+}\,,\, X_{+i},\, M$ with a value for M that is so unlikely under the $\mathcal{U} \mid X_{i+}\,,\, X_{+i}$ distribution is a bit time-consuming, because the number of failed attempts (see Section 4.2) may be quite large. In this case, more than 300 were necessary on average for generating one matrix with the required characteristics. Therefore the number of runs was taken as 1000; however, it still took no more than a few minutes to generate 1000 matrices. The following result was obtained.

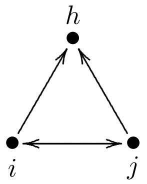
```
@3
100 * #trans / (#trans + #intrans) (rounded to integer)
Estimated mean                42.0593     (standard error   0.1554 )
Estimated variance             0.9270
Estimated standard deviation   0.9628

0.7200|
      |                              XX
0.6400|                              XX
      |                              XX
0.5600|                              XX
      |                              XX
0.4800|                              XX
      |                              XX
0.4000|                              XX
      |                              XX
0.3200|                              XX
      |                              XX
0.2400|                              XX
      |                              XX
0.1600|                              XX
      |                              XX  XX
0.0800|                              XX  XX
      |                          XX  XX  XX
0.0000|  XX  XX  XX  XX  XX  XX  XX  XX  XX  XX  XX  XX  XX  XX  XX
      +------------------------------------------------------------
         35  36  37  38  39  40  41  42  43  44  45  46  47  48  49

Minimum value found 35; maximum 49 .
Estimated cumulative ( <= ) probabilities with standard errors.
   35     36     37     38     39     40     41     42     43     44
 0.0000 0.0000 0.0016 0.0041 0.0155 0.0526 0.1280 0.8285 0.9496 0.9627
 0.0000 0.0000 0.0013 0.0027 0.0117 0.0356 0.0789 0.1201 0.0431 0.0390

   45     46     47     48     49
 0.9990 0.9993 0.9999 1.0000 1.0000
 0.0008 0.0007 0.0001 0.0000 0.0000
```

The transitivity index R_{trans} is strongly concentrated arond the value of 0.42. Value of 0.46 or higher are very unlikely. Thus, this network has a strongly significant tendency toward transitivity, when we control for the in-degrees, the out-degrees, and the number of mutual dyads. The standard errors are small enough to warrant this conclusion.

It may seems a bit strange that the transitivity index has such a small standard deviation. This is understandable, however, given that (1) the information that is held constant (degrees and number of mutuals) is considerable; (2) the graph has a rather high density so that there are a lot of potentially transitive triplets, and the law of large numbers then leads to a small standard deviation for the fraction of transitive triplets among them (even though the triplets are not statistically independent).

7.3 Triad census

Finally, the similarity/attraction hypothesis discussed in Holland and Leinhardt (1975) and in Section 14.4 of Wasserman and Faust (1994) is tested for the same data set. This hypothesis states that if two persons have a strong tie, understood here as a mutual tie, then it is likely that they have similar relations to other (third) persons; in other words, mutual closeness leads to a tendency toward structural equivalence with respect to outgoing relations. Wasserman and Faust (1994, Section 14.4.2) indicate various configurations of ties that express some aspect of this hypothesis. We focus on one of these, viz., the configuration of four ties $X_{ij} = X_{ji} = X_{ih} = X_{jh} = 1$. The similarity/attraction hypothesis entails that this configuration should be observed more frequently than expected by chance, given the number of mutual dyads and given the total number of ties. Holland and Leinhardt (1975) and Wasserman and Faust (1994, Section 14.4.2) explain that this configuration occurs once in the triads `120U` and `210` and thrice in the triad `300`, which implies the weight vector (0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 1 3). This configuration is indicated in the following figure.



Configuration: mutual friends agreeing

As for most hypotheses tested by a linear combination of the triad census, it is most meaningful here to control in any case for the dyad count – i.e., the numbers (M, A, N) of mutual, asymmetric, and null dyads. Controlling only for the dyad count can be carried out using the program Triads of Walker and Wasserman (1987).[1] In this case, the observed linear combination with these weights is 66, and the program Triads (with the changed weight vector – see the footnote) gives the results that its expected value under the $\mathcal{U} \mid M, A, N$ distribution is 24.77, the standard deviation is 5.22, and the resulting standardized test statistic is $\tau = (66 - 24.77)/5.22 = 7.90$. This statistic is to be tested in the standard normal distribution, yielding a highly significant result. Thus, many more mutual friends agree on both having the same third person as a friend than would be expected given the dyad count.

[1] This program, however, seems to contain an error as it produces the results for the transpose adjacency matrix, i.e., for the graph with reversed directions of all arcs. This means that in the triad census, all triads with a code `U` must be interchanged with those with a code `D`, in order to obtain the correct results. E.g., the weight vector used here must be presented to the Triads program as (0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 1 3), to give results for the untransposed matrix. Corresponding to this, the results presented in Wasserman and Faust (1994) on pp. 574, 582, and 595-596 contain some errors.

It is also meaningful, however, to control not only for the dyad count but also for the degrees. This will indicate whether the observed frequency of the configuration $X_{ij} = X_{ji} = X_{ih} = X_{jh} = 1$ is higher than expected by chance given the number of mutual dyads and all in- and out-degrees. This requires for ZO that the $\mathcal{U} \mid X_{i+}\,,\, X_{+i}\,,\, M$ distribution be used – in other words, not only the degrees but also the number of mutual dyads must be fixed, just like in the preceding section. Like above, 1,000 simulation runs were used. The results in the output file are as follows.

```
Critical values
          66.0000
Estimated means (with standard errors) of linear combinations.
            1

mean     61.7270

s.e.      2.3716

Covariance matrix of linear combinations
           1

  1     54.721814

Exceedance probabilities (p-values)
(i.e., probabilities that linear combinations are at least as large
       as the critical values)
                    1

p-values          0.340

standard errors   0.161
```

(The index 1 used in various places refers to the number of the linear combination; if more than one linear combination would have been requested, it would have counted further than 1.)

The observed linear combination is 66 is used as the critical value. For the $\mathcal{U} \mid X_{i+}\,,\, X_{+i}\,,\, M$ distribution, the expected value is estimated as 61.73, with standard error 2.37; and the standard deviation as $\sqrt{54.72} = 7.40$, with the resulting standardized test statistic $\tau = (66 - 61.73)/7.40 = 0.56$. The ZO output also provides an estimated p-value that does not rely on an approximation by a normal distribution, and which here is estimated as 0.34 with standard error 0.16. All these standard errors are relatively large. This has to do with the extremely high value of $M = 23$ mutual dyads for these in- and out-degrees. For such extreme values the simulation algorithm used in ZO can lead to quite skewed weights used for the estimation (see Section 4.2) which leads to high standard errors. The best solution is to use more simulation runs. For 10,000 runs the expected value is estimated as 58.56, with standard error 1.08; the standard deviation is estimated as $\sqrt{40.58} = 6.37$, with the resulting standardized test statistic $\tau = (66 - 58.56)/6.37 = 1.17$. The estimated p-value is 0.16 with standard error 0.046. Still the results are not as precise as we would like. They suggest, however, that the frequency of occurrence of this configuration is somewhat high but not significantly so. The conclusion then is that the frequency of the configuration "mutual friends agreeing on friendship witha third person" cannot be explained by the number of mutual and asymmetric ties, but it can be explained when also the in- and out-degrees are taken into account. It must be noted that especially the out-degrees have a rather high dispersion (as is not unusual with self-reported ties).

8 Running ZO stand-alone; Basic information file

To run ZO as a stand-alone program under MS-Windows, the best procedure is to do this through a batch file. This is a text file with extension name `bat`, and it is executed in the usual way, e.g., by double clicking. Such a file can be made by any text editor; the file must be saved as a raw text or ASCII file. For example, this could be a file called `zosim.bat`. It should be in the directory containing also the executable files `zo_enum.exe` and/or `zo_sim.exe`. The contents of this file can be, e.g.,

```
zo_sim infile
```

where `infile.in` is the name of the input file. If the enumeration program is used, it should be

```
zo_enum infile
```

The zipped file distributing ZO contains a number of examples of input files. Note that the ending '`.in`' should not be given – then ZO would look for a file `infile.in.in`. If no name is given, the input file is expected to be called `ZO.in`.

8.1 Basic information file

The meaning of the information to be given in the basic information file was treated already in Section 6.1. The basic information file must be named ***pname*.in**, where ***pname*** is the project name. It should be an ASCII (i.e., raw text) file containing the commands for the computation part. If the program cannot find the file ***pname*.in**, it will look for the file ZO.in . The structure of the basic information file for zo_sim is equal to the structure for zo_sim, plus some additional lines at the end.

After the required information, the lines are allowed to contain at least one blank and after that more text, which can be used for helpful information. However, lines with names of data files should not contain additional text.

The basic information file for zo_enum must have the following lines (each line giving information about one variable or filename). The suggested defaults are also given.

1. f, the number of rows.
2. g, the number of columns.
3. *GraphMode*, an integer number between 1 and 4:
   1 for an $f \times g$ matrix without structural zeros, 2 for a digraph, 3 for a graph, 4 for an $f \times g$ matrix with an arbitrary set of structural zeros.
   Default *GraphMode* $= 2$.
4. *WriteMode*, a number 0 or 1:
   1 if all produced matrices should be written to file, 0 otherwise.
   Default *WriteMode* $= 0$.
5. *FileMode*, a number 0 or 1:
   1 if the data file contains an adjacency matrix, 0 if it contains only two lines: first the required row sums, then the required column sums.
6. The name of the data file.
7. *ConnectedMode*, a number 0 or 1:
   1 if the extra requirement is made that the generated graphs should be connected (or weakly connected for digraphs), 0 otherwise.
   Default *ConnectedMode* $= 0$.

8. Only for *GraphMode* = 2:
   MutualMode, a number 0 or 1:
   1 if the extra requirement is made that the generated graphs should have a given number of mutual dyads, 0 otherwise.
   Default *MutualMode* = 0.
9. Only for *MutualMode* = 1:
   PrescribedMutuals, an integer number at least 0:
   the prescribed number of mutual dyads.
10. Only for *GraphMode* = 4: optional:
    StrucZeroFilename, the name of the file with an $f \times g$ adjacency matrix that indicates which are the structural zeros.
    If there is not a line with such a filename, it is assumed that there are no structural zeros (which reduces option *GraphMode* = 4 to results equivalent to *GraphMode* = 2.)
11. *FuncVersion*, an integer number at least 1, maximum value depending on whether zo_enum or zo_sim is used, and on whether $f = g$:
    the version of the statistics used by the program.
    Default *FuncVersion* = 1.

For zo_sim , the following extra lines must also be given.

12. *nrun*, the number of simulation runs (at least 0). Default *nrun* = 10000.
13. *numl*, the number of linear combinations of triad counts that are to be calculated as statistics.
    This has an effect only if *FuncVersion* = 1.
    Default *numl* = 0.
14. *Option*, a number 1 or 2:
    1 if standard weights for these linear combinations are used, 2 if the weights are to be read from a file.
    Default *Option* = 1.
15. Only for *numl* > 0, *Option* = 2:
    WeightFileName, the name of the file with the weights for the linear combinations of the triad counts.
    This file must contain *numl* lines, each line containing 16 real number, with an optional 17$^{\text{th}}$ number which then is the critical value for which exceedance probabilities of this linear combinations will also be determined.

A further difference is that for zo_sim the value of *GraphMode* may also be 11, 12, or 13, indicating the same type of matrix as *GraphMode*−10, but now using the algorithm of Molloy and Reed (1995) instead of that of Snijders (1991).

A possible basic information file for zo_sim is

```
20
20
2
0
0
zo20.dat
0
0
2
1000
0
1
```

with the data file zo20.dat containing, e.g., the two lines

```
2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
```

A possible basic information file for zo_enum is

```
7
7
2
0
0
zo7.dat
0
0
2
```

with the data file zo7.dat given, e.g., by

```
2 2 2 2 2 1 1
1 1 2 2 2 2 2
```

The distribution of ZO in a zip file contains some examples of basic information files.

Part II
Programmer's manual

The programmer's manual will not be important for most users. It is intended for those who wish to have a look inside the source code.

The program consists of a basic computation part programmed by the author in Turbo Pascal and Delphi 5; and the StOCNET windows shell, programmed by Peter Boer in Delphi, with Mark Huisman (earlier Evelien Zeggelink) as the project leader (see Boer, Huisman, Snijders, and Zeggelink, 2002). The computational part can be used both directly and from the StOCNET shell. The StOCNET windows shell is easier for data specification and model definition.

9 Parts and units

The calculations of the program are carried out by the executable files zo_sim.exe and zo_enum.exe. which read the basic information file and executes the calculations accordingly. No user interaction is required, although there are possibilities of early termination.

If you wish to run ZO outside of StOCNET, the project name *must* be given in the command line, e.g.

```
zo zo20
```

if `zo20` is the name of the project, and there exists a `zo20.in` file. This `zo20` is called a *command line parameter*. There are the following three ways to specify a command with a command line parameter in Windows. The command line can be given at the DOS prompt (in a Windows environment); it can be given in the Windows "Run" command (for Windows 98 and higher); and it can be indicated in the "target" in the "properties" of a shortcut.

The source code is divided into the following units.
For zo_sim :

1. ZOS_Form, which is the basic form;
2. ZOMat, which contains the data storage routines;
3. Simul, which contains the basic routines for simulating the 0-1 matrices;
4. FunctSim, which contains the basic functions for calculating the statistics of which the distributions are determined;
5. Triads, containing procedures for calculating triad counts, a special type of statistics, used in FunctSim.
6. Rand3, which is a short unit for random number generation;
7. ZOLib, which contains some utilities for use in zo_sim and zo_enum ;
8. ZOSimLib, which contains some utilities for use in zo_sim , among others the data input procedure;
9. SimPlus, which contains a routine for postprocessing intermediate results if these are too voluminous to be stored in memory.

For zo_enum :

1. ZOE_Form, which is the basic form;
2. Enum_Cons, which contains some constant declarations;
3. Enum, which contains the basic routines for enumerating the 0-1 matrices;

4. FunctEn, which contains the basic functions for calculating the statistics of which the distributions are determined;
5. Numbers, with two functions for directly calculating the number of 0-1 matrices with given marginal sums, which are *not* used in the main enumeration algorithm and only present for the possibility of checking enumeration results;
6. ZOLib, which contains some utilities for use in zo_sim and zo_enum ;
7. ZOELib, which contains some utilities for use in zo_enum , among others the data input procedure.

10 Programming additional statistics

For experienced Pascal or Delphi programmers it should not be too hard to include additional statistics for the calculations in ZO. For the Simulate option, this will require only to change the FunctSim unit and recompile ZO. To retain the consistency in the operation of the program and in the output, however, it will be important to change various procedures in this unit in a consistent manner. The following procedures will require attention.

1. Func, the definition of the statistics. It uses the generated matrix as the variable `mat` which is of the type `Adjac` defined in unit `ZOMat`. The arc variables are referred to as `mat(i,j)`.
2. Tex, which gives the text describing the statistic.
3. Numversion, the number of sets of statistics available for simulation; these are the 'versions' described in Section 5.2.
4. Numfunc, the number of statistics calculated within each of these sets.

If statistics are added to the program, either Numversion or Numfunc, or both, will have to be changed.

11 References

Bender, E.A. 1974. The asymptotic number of non-negative integer matrices with given row and column sums. *Discrete Mathematics*, **10**, 217–223.

Boer, P., Huisman, M., Snijders, T.A.B., and E.P.H. Zeggelink. 2002. *StOCNET: An open software system for the advanced statistical analysis of social networks. Version 1.3.* Groningen: Pro*GAMMA*/ICS. Available from http://stat.gamma.rug.nl/stocnet/.

Bonacich, P. 1972a. Factoring and weighting approaches to status scores and clique identification. *Journal of Mathematical Sociology*, **2**, 113–120.

Bonacich, P. 1972b. Techniques for analyzing overlapping memberships. *Sociological Methodology - 1972*, 176–185. San Francisco: Jossey-Bass.

Bonacich, P., Oliver, A., and T.A.B. Snijders. 1998. Controlling for size in centrality scores. *Social Networks*, **20**, 135–141.

Frank, O., and F. Harary. 1982. Cluster inference by using transitivity indices in empirical graphs. *Journal of the American Statistical Association*, **77**, 835–840.

Freeman, L. 1979. Centrality in social networks. *Social Networks*, **1**, 155–168.

Holland, P.W., and S. Leinhardt. (1975). Local structure in social networks. In D. Heise (ed.), *Sociological Methodology-1976.* San Francisco: Jossey-Bass.

Karlberg, M. 1997. Testing transitivity in graphs. *Social Networks*, **19**, 325–344.

Karlberg, M. 1999. Testing transitivity in digraphs. In M.E. Sobel and M.P. Becker (eds.), *Sociological Methodology – 1999*, 225–251.

Katz, L., and J.H. Powell. 1954. The number of locally restricted directed graphs. *Proceedings of the American Mathematical Society*, **5**, 621–626.

Katz, L., and J.H. Powell. 1957. Probability distributions of random variables associated with a structure of the sample space of sociometric investigations. *Annals of Mathematical Statistics*, **28**, 442–448.

Molloy, M., and B. Reed. 1995. A critical point for random graphs with a given degree sequence. *Random Structures and Algorithms* **6**, 161–179.

Rao, A.R., and S. Bandyopadhyay. 1987. Measures of reciprocity in a social network. *Sankhya* ser. A, **49**, 141–188.

Rao, A.R., Jana, R., and S. Bandyopadhyay. 1996. A Markov chain Monte Carlo method for generating random (0,1) matrices with given marginals. *Sankhya* ser. A, **58**, 225–242.

Roberts, J.M., jr. 2000. Simple methods for simulating sociomatrices with given marginal totals. *Social Networks* **22**, 273–283.

Snijders, T.A.B. 1991. Enumeration and simulation methods for 0-1 matrices with given marginals. *Psychometrika*, **56**, 397–417.

Sukhatme. 1938. On bipartitional functions. *Philosophical Transactions of the Royal Society of London*, series A, **237**, 375–409.

Walker, M.E., and S. Wasserman. 1987. *TRIADS: A computer program for triadic analyses.* Urbana, IL: University of Illinois.

Wasserman, S. 1977. Random directed graph distributions and the triad census in social networks. *Journal of Mathematical Sociology*, **5**, 61–86.

Wasserman, S., and K. Faust. (1994). *Social Network Analysis: Methods and Applications.* New York and Cambridge: Cambridge University Press.