REFERENCE MANUAL

for the

PARAFAC ANALYSIS PACKAGE

by

Margaret E. Lundy

and

Richard A. Harshman

Scientific Software Associates
48 WILSON AVENUE
LONDON, ONTARIO, CANADA N6H 1X3

REFERENCE MANUAL FOR THE PARAFAC ANALYSIS PACKAGE

TABLE OF CONTENTS

PREFACE

This manual attempts to present, using nontechnical language, the basic information necessary to use the programs in the PARAFAC Analysis Package and to understand their output. It is assumed throughout that the user is reasonably familiar with his/her own computer system; no attempt has been made to include, for instance, system control language with any of the example input decks.

This manual does not discuss the PARAFAC model in detail, nor does it present proofs. It is strongly recommended that you read the references cited, especially Harshman (1970) and Harshman and Lundy (1984a, 1984b) for more theoretical discussion and background about the model.

Some information in this manual is optional, depending on the user's level of expertise and what his/her purposes are. Most users will omit Appendix A (matrix notation). Appendices B and C (formulas, etc.) are provided for those who are interested, but are not required reading; the same can be said for Sections 6.5 and 6.6 (special loadings interpretations, etc.). Users who do not synthesize data will not need to refer to Sections 2.5 and 2.6, nor Chapter 7.

The authors invite suggestions for improvement to the manual. Comments may be addressed to them C/O Scientific Software Associates.

CHAPTER 1

GENERAL INTRODUCTION

1.1 MANUAL ORGANIZATION

Chapter 1 is devoted to a summary of the manual contents (this section), a brief description of the PARAFAC Analysis Package programs (Section 1.2), a discussion of terminology that will be used in the program documentation (Section 1.3), and a comparison of PARAFAC to some other three-way factor analysis procedures (Section 1.4). If you are not already familiar with the PARAFAC model, you should read Section 1.2 and the references cited there, so that you know whether or not PARAFAC is appropriate for your data. You should also read Section 1.3 so that you will understand what is meant by various terms that appear elsewhere in the manual.

The remainder of the manual is documentation for the programs in the PARAFAC Analysis Package. Chapter 2 describes PARAFAC input for data analysis and data synthesis. Examples of input and output are provided. Chapter 3 describes input and output for the utility programs DIMS, PFPLOT, CMPARE and DISTIN, and supplies examples for each.

Chapters 4-8 contain explanations for practical and theoretical issues that arise when using PARAFAC. Chapter 4 discusses various types of preprocessing and should be referred to before you analyze your data. Chapter 5 is a description of the PARAFAC output. Chapter 6 elaborates on diagnostic indicators and on questions of interpretation. Data synthesis is covered in Chapter 7, and informative messages output by PARAFAC (e.g., warning messages) are listed in Chapter 8.

Additional information is contained in several appendices. Appendix A is a short review of matrix notation; most users may omit it. Appendix B lists the computational formulas used by PARAFAC. Included are formulas for fit values (e.g., Mean Square Error, Stress), data preprocessing (e.g., centering, normalization), factor relationships (e.g., cross-products, correlations, etc.).

Appendix C describes the random number generator used by PARAFAC and gives the equations used to simulate various distributions from which random factor loadings, error components, etc. are selected.

Appendices D and E (if included) give general instructions for installing the PARAFAC Analysis Package programs. This information is not required by those using a program that is already installed.

1.2 PARAFAC ANALYSIS PACKAGE SUMMARY

The PARAFAC Analysis Package is a set of Fortran batch programs which enable the user to perform factor analysis and multidimensional scaling of two- and three-way data arrays.

The main program in the Package is, of course, PARAFAC, which does the actual analysis. PARAFAC (for Parallel Factors) fits the three-way factor analysis model based on the Principle of Proportional Profiles (Cattell, 1944):

$$x_{ijk} = a_{i1}b_{j1}c_{k1} + a_{i2}b_{j2}c_{k2} + \dots + a_{in}b_{jn}c_{kn} + e_{ijk}$$

where x is a data value in a three-way array, a, b and c are loadings for factors 1,2,...,n in the three modes of the data; and e is random error. For example, this model could be applied to an array of people's ratings of stimuli on various scales; the three modes of such an array would be ratings scales, stimuli and subjects. (Throughout this manual, a distinction is not explicitly made between the PARAFAC model and the PARAFAC program; the word "PARAFAC" is used for both. However, the context usually makes it clear which is meant.)

The mathematical bases of the model are discussed by Harshman (1970, 1972, 1976), Harshman and Berenbaum (1981), and Harshman and Lundy (1984b). PARAFAC makes the same assumption as two-way factor analytic procedures do (i.e., that the data can be decomposed into the additive effects of a few basic underlying factors). It also makes assumptions that are related to the three-way nature of the data:

1. The same common factors are present in several two-way slices of the three-way array (although some factors can be missing from some slices). These factors need not have the same relative importance within different slices.

2. System variation occurs in the data (i.e., changes in the underlying factors are proportional across levels of the third mode; see Harshman, 1970,

pp.19-21).

3. For analysis of raw or profile data, but not for covariance analysis, measurements must be of the same objects (or people) on the same variables for every level of the third mode (e.g., for every occasion or condition).

The user should check that the PARAFAC model is appropriate for his/her data. PARAFAC can be directly applied to the raw data if all three assumptions are fulfilled. Otherwise, covariances may have to be computed from the raw data and analysed by PARAFAC (covariance analysis is discussed in Chapters 4.4 and 6.7).

The user should also select preprocessing that will make the data more suitable for PARAFAC analysis. For example, interval scale data should be centered (a PARAFAC option will do this) to remove constants and transform it to ratio scale data, since PARAFAC expects ratio scale data. (Centering and other types of preprocessing are discussed in more detail in Chapter 4.)

Alternating Least Squares, with overrelaxation to accelerate convergence, is the fitting method used. It provides a least squares best fit with the important "intrinsic axis property" (i.e., the axis orientation is <u>unique</u> when the data fulfil certain conditions, such as those described in Assumption 1 above). This uniqueness property eliminates the problem of factor rotation that occurs in two-way factor analysis.

PARAFAC has numerous options which permit considerable flexibility in the use of the program. Default values are assigned for most of the options if no values are specified by the user. It is recommended that the user take the defaults if he/she is unsure of which value to use.

Some particulariy useful features of PARAFAC are:

1. Missing data capability

2. Preprocessing options which allow centering and/or normalizing of the data on one, two or three modes (see Chapter 4)

3. Orthogonality or zero-correlation constraints on the factors in one, two or three modes

4. Data synthesis capability (see Chapter 7)

It should be noted here that PARAFAC can also be used to perform traditional two-way factor analysis, if the data array has only one level in Mode C. However, as with other two-way factor analysis procedures, the solution will be rotationally indeterminate; the factors will have to be rotated using some external criterion. This two-way analysis capability will not be stressed, since the main contribution of PARAFAC is in the domain of three-way data analysis.

In addition to PARAFAC, four "utility" programs are supplied (see Chapter 3). They are included mainly to complement PARAFAC, but three of the four also have more general uses. The utility programs are as follows:

1. DIMS redimensions PARAFAC arrays and can also be used to alter PARAFAC I/O to suit the user's installation. It is written specifically for use with PARAFAC.

2. PFPLOT plots points on a 130-column lineprinter, along a single axis or on the plane defined by two axes. Normally, it is used to help interpret factors from a PARAFAC analysis, but it also accepts input of a more general nature.

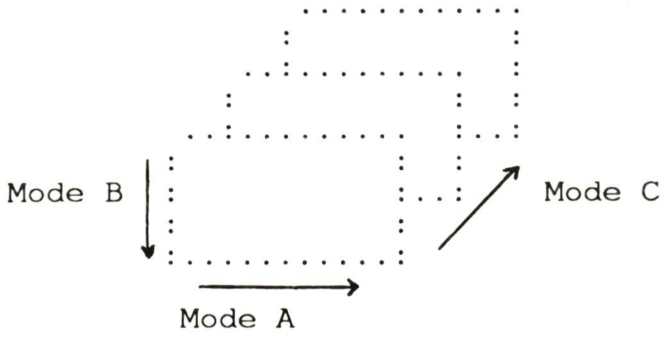
3. CMPARE compares factors from several different PARAFAC solutions by merging them into one file and computing cross-products and correlations among all the merged factors. Like PFPLOT, it also accepts more general input.

4. DISTIN preprocesses data (usually similarity or dissimilarity measures) so that PARAFAC can then be used to do multidimensional scaling. Or, DISTIN may be used to transform data for some other analysis procedure.

1.3 DEFINITION OF TERMS

1.3.1 PARAFAC Notation

Any matrix can be thought of as having two "ways" or "directions" associated with it (i.e., down and across) --hence the term "two-way" factor analysis. PARAFAC, which fits a three-way model, is an extension of two-way factor analysis. Instead of one matrix, a series of data matrices is analysed. The data can be visualized as a block if the matrices are thought of as being stacked one behind the other, as shown below:



It is evident that such an arrangement, called a data array, has three directions associated with it (i.e., down, across, and "back"). These ways or directions are referred to in the manual as "modes". (Some, e.g. Carroll and Arabie, 1980 have made a distinction between "ways" and "modes", but here we do not.) The letters "A", "B" and "C" designate the specific directions: "Mode A" is across, "Mode B" is down, and "Mode C" is back.

As defined, "Mode A", "Mode B" and "Mode C" are general terms which refer to any three-way data set. Of course, any given data array has a specific meaning for each of its three modes, depending on what was measured (e.g., rating scales, stimuli, people) and on how the data are arranged in the array. Each mode has a unique meaning if raw data is input, but if covariance or scalar product matrices are input instead, then Modes A and B have the same meaning.

Mode A is always regarded as the "first" mode, Mode B the "second", and Mode C the "third". This order reflects the data input sequence to PARAFAC (i.e. by rows within matrices). Otherwise, the order has no special significance beyond providing a consistent method for referring to the ways of the data, since during analysis PARAFAC generally treats all three modes in the same way. Indeed, the data can be rearranged so that the rows and columns (for example) are interchanged, but the PARAFAC solution is unchanged except that the Mode A and B factor loadings are reversed.

The size of a PARAFAC data array is specified by the phrase "n by m by p" or "n X m X p". n is the number of columns or levels of Mode A and m is the number of rows or levels of Mode B in each matrix; p is the number of nXm matrices or levels of Mode C. Note that here n X m denotes n columns by m rows, to be consistent with the PARAFAC mode order, even though this is opposite to common matrix notation.

The total number of cells in the data array is the product of n, m and p. Thus a 5x4x10 array, for example, has 200 cells. A cell in the array is the intersection of a column (level of Mode A) and row (level of Mode B) within a specific matrix (level of Mode C). Its location is denoted by the ordered triple (i,j,k), where i is the column number, j the row number and k the matrix number within the data array. Such ordered triples are used to indicate the locations of missing values in the PARAFAC data array.

1.3.2 Two-way Analysis

In the typical situation involving two-way factor analysis, measures on variables of some kind (e.g., personality scales) are collected from a group of subjects, correlations between the variables are computed across the subjects, and the matrix of correlations is factor analysed. The factor analytic procedure attempts to mathematically explain (predict) this correlational data in terms of a few general underlying entities called factors. Each factor consists of a vector of loadings or weights, one loading for each variable. Each factor contributes to the observed value of the variable; the amount contributed is indicated by the sign and magnitude of the loadings. The person (subject) loadings, usually called factor scores, may be recovered by using regression techniques. A problem with two-way factor analysis is that the set of loadings which fulfil the mathematical constraints is not unique; different rotations of the same factors may yield ones that are more or less easily interpreted than the unrotated ones.

1.3.3 PARAFAC Analysis

Three-way PARAFAC analysis has the same purpose as two-way factor analysis, that is, reduction of the data to a few factors that are simpler to interpret than the data itself, but which predict the data reasonably well. Usually it is hoped that these factors reflect underlying real causes or meaningful entities, and so they not only help to explain the current data but also may suggest new predictions that can be tested in further research. It has been argued that the "uniqueness" property of PARAFAC facilitates the discovery of such empirically meaningful factors (see Harshman, 1970, Ch. 1, 2).

In contrast to the two-way case, PARAFAC is often used to analyse raw data, or in some special cases covariance data (see Chapter 4). And, if the assumptions of the PARAFAC model as stated in Section 1.2 above are met, the obtained solution is unique (i.e. there are no rotational indeterminacies). Each PARAFAC factor consists of a vector of weights or loadings, one loading for each level of each mode. If one of the modes refers to people, then the weights that are output for that mode are the "person loadings"; no additional computation is necessary to recover them. We do not often use the special term "factor scores" to refer to the person loadings, since we prefer to regard all three modes, and the loadings associated with them, on an equal basis. Also, note that "dimension" and "factor" are used interchangeably in the manual; for example, by the phrase "one-dimensional solution", we mean that one factor was used to explain the data.

The PARAFAC solution is output as three matrices of factor loadings, one matrix for each mode of the data array. The matrix rows correspond to the levels of the mode and the columns to the factors. The following diagram illustrates the arrangement of a 3-factor PARAFAC solution from an analysis of a 3x4x3 data array:

```
           Factors
            1  2  3
 Mode     :--:--:--:
  A  1    :  :  :  :
          :--:--:--:
     2    :  :  :  :
          :--:--:--:
     3    :  :  :  :
          :--:--:--:

            1  2  3
 Mode     :--:--:--:
  B  1    :  :  :  :
          :--:--:--:
     2    :  :  :  :
          :--:--:--:
     3    :  :  :  :
          :--:--:--:
     4    :  :  :  :
          :--:--:--:

            1  2  3
 Mode     :--:--:--:
  C  1    :  :  :  :
          :--:--:--:
     2    :  :  :  :
          :--:--:--:
     3    :  :  :  :
          :--:--:--:
```

This review of terminology is geared specifically toward use of PARAFAC. See Comrey (1973), Green and Carroll (1976) or Kim and Mueller (1978) for more extensive background about the general theory and procedures pertaining to factor analysis.

1.4 OTHER THREE-WAY ANALYSIS PROGRAMS

There are other three-way analysis procedures besides PARAFAC. Some are: CANDECOMP (Carroll and Chang, 1970), ALSCAL (Takane, Young, and DeLeeuw, 1977), MULTISCALE (Ramsay, 1977), PINDIS (Lingoes and Borg, 1978), ALSCOMP (Sands and Young, 1980) and TUCKALS2 and TUCKALS3 (Kroonenberg and DeLeeuw, 1980). CANDECOMP, ALSCAL, MULTISCALE and PINDIS perform multidimensional scaling (MDS). The others are used mainly for factor analysis. Most of these models, along with others developed for multi-mode data analysis, are included in Law, Snyder, Hattie and McDonald (1984).

The basic form of the CANDECOMP model (Carroll and Chang, 1970) is very similar to PARAFAC; in fact, the model presented in Section 1.2 above is sometimes referred to as the PARAFAC-CANDECOMP model. However, Harshman and Lundy (1984a) have recently shown that the PARAFAC model can be "extended" to a more general one than CANDECOMP when the special PARAFAC preprocessing options are used.

The principal difference between PARAFAC and CANDECOMP is their application. CANDECOMP, incorporated into the INDSCAL program (Carroll and Chang, 1970), has mainly been used for fitting the weighted Euclidean distance model in three-way metric MDS. In contrast, PARAFAC was incorporated into a program specifically designed for three-way factor analysis (although it can accomplish INDSCAL-like MDS too if the data are first preprocessed by the DISTIN program (see Chapter 3). For example, the PARAFAC program has an option for estimating diagonal values that allows the common factor model to be fit to covariance matrices, and its other special preprocessing and orthogonality options help the user to cope with difficulties that arise in three-way factor analysis but not in three-way MDS.

The other MDS programs are related to INDSCAL. According to Carroll and Arabie (1980), ALSCAL is a nonmetric implementation of INDSCAL that in addition allows for missing or replicated data. MULTISCALE performs maximum likelihood metric MDS and permits the definition of confidence regions for both subject weights and stimulus points (Ramsay, 1977, 1978). PINDIS fits an INDSCAL-like weighted Euclidean distance model, and generalizations of it, plus a "vector weighting" or "perspective" model (Lingoes and Borg, 1978).

Tucker's three-mode factor analysis (Tucker, 1964, 1966), implemented in TUCKALS2 and TUCKALS3, provides a more general model for three-way data than PARAFAC does. Therefore, it can be successfully applied to some data for which PARAFAC is inappropriate, or for which PARAFAC gives "degenerate" solutions (see Harshman and Lundy, 1984a). However, it does not possess the "intrinsic axis" property of PARAFAC, and so solutions must be rotated to some position preferred by the analyst.

ALSCOMP is a general program that can do nonmetric as well as metric analysis, and it can handle continuous or discrete data (Sands and Young, 1980). In those cases where a nonmetric procedure is clearly superior to a metric one, ALSCOMP has the advantage over PARAFAC. Often, however, the difference in procedures is likely to be minimal, even when the data are only ordinal scale (see Weeks and Bentler, 1979). Even though PARAFAC was not designed for discrete data, the distinction between continuous and discrete data is not crucial; Sentis, Harshman and Stangor (1983) found in a Monte Carlo study that PARAFAC successfully recovered continuous latent structure from binary data. And, the preprocessing options of PARAFAC allow the model to be "extended" to interval scale or conditional data that would otherwise be appropriate for ALSCOMP but not PARAFAC. Finally, both ALSCOMP and PARAFAC may yield "degenerate" solutions for some types of three-way structure. However, PARAFAC has special orthogonality options that frequently overcome such degeneracies and result in interpretable solutions.

In summary, then, the PARAFAC model is less general than some other three-way analysis models (e.g., Tucker's; ALSCOMP) and very similar to others (e.g., CANDECOMP). However, its unique axis property and the special preprocessing and analysis options make the PARAFAC program generally appropriate for a broad range of data, and therefore quite useful as a data analysis tool.

CHAPTER 2

PARAFAC INPUT

This chapter is mainly a description of PARAFAC input, but it also provides some other information that is useful if you're just beginning to use PARAFAC. Sections 2.1 and 2.2 list the standard array sizes and I/O units and explain how to modify them if necessary. Section 2.3 discusses the arrangement of PARAFAC input, and Table 2.1 gives a detailed description of the input parameters for data analysis. Examples of PARAFAC input are presented in Section 2.4.

Data synthesis is introduced in Section 2.5 and Table 2.3 is a summary of the synthesis input parameters. Examples of input for data synthesis are presented in Section 2.6. Data synthesis is further discussed in Chapter 7 and it is recommended that you read Sections 7.1-7.3 before generating data for a specific purpose.

Example output from a data analysis run appears in Table 2.2 but is not described; Chapter 5 explains such output in detail. Table 2.4 is an example of output from a data synthesis run; it is described in Chapter 7.

2.1 PARAFAC LIMITS

The standard PARAFAC code (i.e., as shipped) has the following limits:

1. Maximum number of levels in Mode A is 18.
2. Maximum number of levels in Mode B is 18.
3. Maximum number of levels in Mode C is 35.
4. Maximum number of missing data values is 50.
5. Maximum number of factors to be extracted during an analysis is 10.

To change any of these limits, the utility program DIMS must be run. See Chapter 3 for more details. If you do not know what is meant by "Mode A", "Mode B", etc. you should refer to Section 1.3.

2.2 PARAFAC I/O UNITS

PARAFAC uses up to 3 different logical input units and up to 4 different logical output units, which are denoted by separate parameter names. The parameters are assigned default values that can be overridden by the user as desired. Appropriate system commands must be included with the job to link the logical units with actual disk files. Listed below are the parameter names used for the I/O units, along with their default values and other information.

Input

1. ISTDIN=5 (standard input unit; default); it can be changed by using the DIMS program. It is used for input of the analysis control parameters.

2. IUNITB=5 (default), or it can be specified by the user on input Card I-3 of the PARAFAC job. It is used for input of the data parameters, data set, and any missing value subscripts.

3. IUNITC=5 (default), or it can be specified by the user on Card I-3 of the PARAFAC job. It is used for input of the starting loadings.

Output

1. ISTDOU=6 (standard output unit; default); it can be changed by using DIMS. It is used for output of the program documentation and step-by-step output of the analysis.

2. IUNITG=7 (default), or it can be specified by the user on Card I-8 of the PARAFAC job. It is used for output of the final loadings from the PARAFAC solution(s). This output can be suppressed by specifying -1 on Card I-8.

3. IUNITD=0 (no output; default), or it can be specified by the user on Card I-8 of the PARAFAC job. It is used for output of the revised (e.g., centered or synthesized) data.

4. IUNITF=0 (no output; default), or it can be specified by the user on Card I-8 of the PARAFAC job. It is used for output of the residuals at the end of each solution.

PARAFAC I/O is pictured in Figure 2.1.

2.3 PARAFAC INPUT

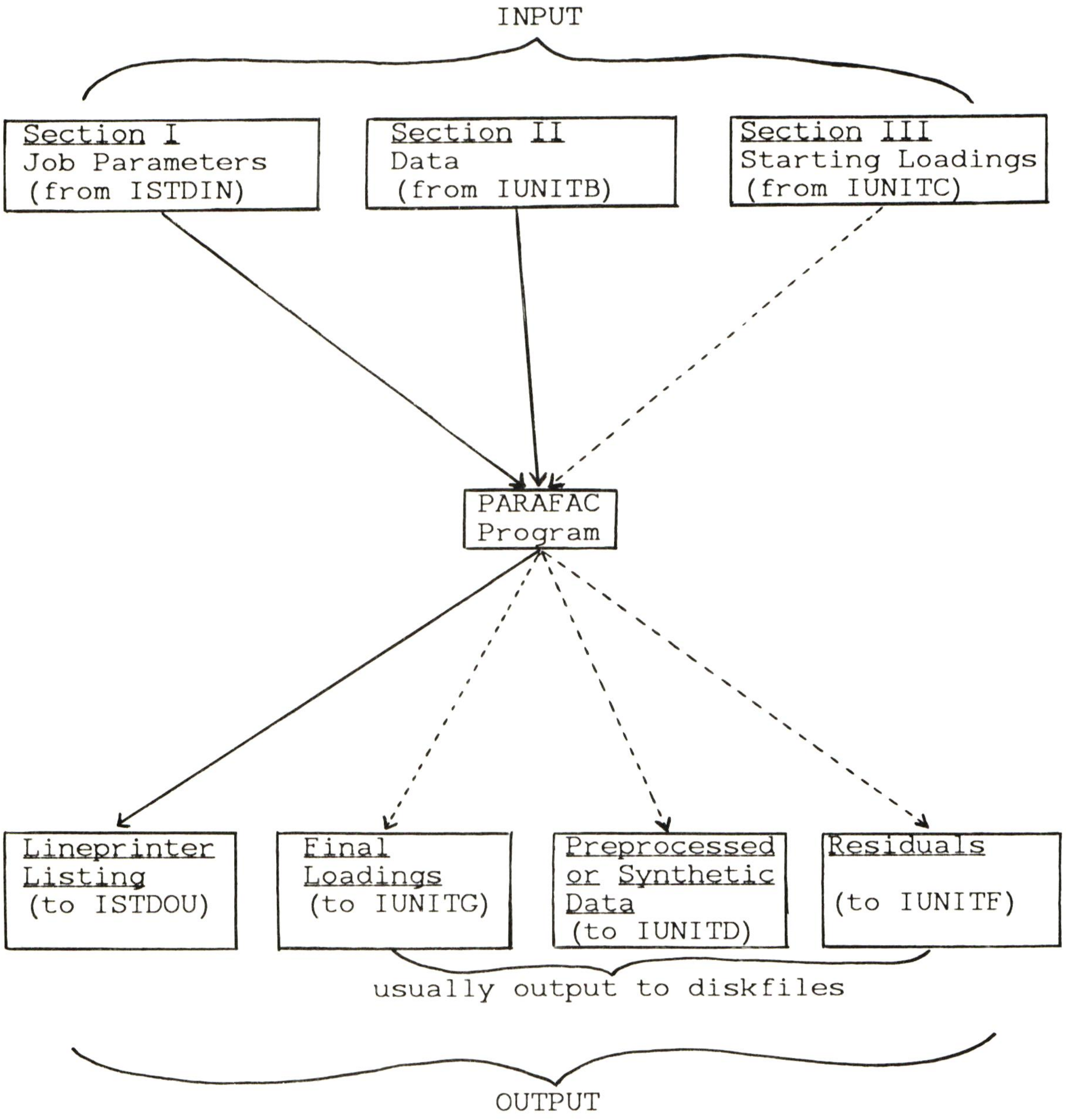
Input to PARAFAC consists of three sections:

1. Job control parameters (Input Section I) read from ISTDIN
2. Data parameters and data set (Input Section II) read from IUNITB
3. Optional starting loadings for the analysis (Input Section III) read from IUNITC

The content and format of all three sections are described in Table 2.1.

Figure 2.2 illustrates the arrangement of the PARAFAC input file. If the data set and starting loadings are to be read from the standard input unit (i.e., IUNITB= IUNITC= ISTDIN), then the file is like the diagram, with Card II-1 immediately following Card I-8 and the starting loadings following Card II-(Y+1) -- do not insert blank records between the input sections. More often, the data set (and starting loadings) will be stored in separate files, and so the input file will contain only the job parameters.

Figure 2.1. PARAFAC Input and Output



(Dotted lines represent input/output that is optional or that can be suppressed.)

Figure 2.2

PARAFAC Input Deck

	Card No. or Record No.		
SECTION I	I—1	Job title	
Analysis Control Parameters	I—2	Task size parameters	
(read from standard	I—3	Input options	
input unit)	I—4	Data preprocessing options	
	I—5	Starting position parameters	
	I—6	Analysis options	
	I—7	Convergence criteria	
	I—8	Output options	
	I—8A	Format for revised data output (optional)	
	I—8B	Format for residuals output (optional)	
SECTION II	II—1	Data title	
Data Parameters	II—2	Data set dimensions	
and Data Set	II—3	Data input format	
(read from standard	II—4		
input unit, or	⋮	Data set	
from disk or tape files)	II—(X)		
	II—(X+1)		
	⋮	Optional table of missing-value subscripts	
	II—(Y)		
	II—(Y+1)	Termination code	
(Optional) **SECTION III**	⋮	Loadings for solution 1	(Optional) One set of loadings
Starting Loadings (read from standard	⋮	Loadings for solution 2	for each solution
input unit, or from disk or tape	⋮	Loadings for solution 3	requested on Card I-2
files)	etc.	etc.	

Table 2.1

PARAFAC INPUT SPECIFICATIONS TABLE

(for PARAFAC version 6H)

SECTION I: ANALYSIS CONTROL PARAMETERS

This first section of parameter input controls the subsequent steps of data input, preprocessing, analysis, and output of results. The analysis control parameters are specified on 8 cards, which are always read from the standard input unit ISTDIN (usually unit 5). For these parameters, if the user leaves blanks instead of specifying particular values, default values are assigned by the program. Integer values specified by the user must be right-justified in their respective fields.

CARD I–1 **FORMAT: (80A1)** (Cols. 1–80)

JOB TITLE: A description of the current job that distinguishes it from other analyses of the same data set. (Note: General information on the data being analyzed need not appear here, as it is included on Record II–I below.) The information on this card is printed out as the first line of every loadings output table and thus helps to document the PARAFAC output. This card also provides an identifying label for the Section I input deck.

CARD I–2 **FORMAT: (4I4)** **TASK SIZE PARAMETERS**.

Column	*Default Value*	*Parameter Name*	*Explanation*
1–4	2	NFACT	Number of factors to be extracted.
5–8	3	NSOLS	Number of "solutions" to be obtained; each "solution" is an analysis from a different starting point. (Set NSOLS=–1 to suppress analysis, e.g. when only doing data preprocessing or data synthesis.)
9–12	2	NOUTS	Maximum number of intermediate outputs of loading tables for each of the solutions. (Fewer outputs may be generated if the solution converges.)
13–16	100	NITER	Number of iterations done between each intermediate output and the next.

CARD I–3 **FORMAT: (2I2, 1X,I1)** **INPUT OPTIONS**. (See also "Note 4: I/O Units" at the end of Section I of this table).

Column	*Default Value*	*Parameter Name*	*Explanation*
1–2	5 or ISTDIN	!UNITB	Input unit for data. (It is only necessary to specify a value if you do not want to use the standard input unit.)
3–4	5 or ISTDIN	IUNITC	Input unit for starting loadings; to be used if ISTART (on Card I–5, below) is 1 or 2. (It is only necessary to specify a value if you do not want to use the standard input unit).
6	0	IFSYMT	Check for symmetry across Modes A and B. A message is printed each time a symmetry failure is detected up to a maximum of 50 messages, and then execution stops. (0 = no symmetry check; 1 = check symmetry.)

Table 2.1 (Continued)

CARD I–4 FORMAT: (3I1,1X,I1, 1X,I1,2X,5G10.4) DATA PREPROCESSING OPTIONS AND MISSING DATA INFORMATION.

Column	*Default Value*	*Parameter Name*	*Explanation*
1	0	IFCENA	Mode A Flag for centering and/or normalization.
2	0	IFCENB	Mode B Flag for centering and/or normalization.
3	0	IFCENC	Mode C Flag for centering and/or normalization.

IFCEN Flag Values are interpreted as follows:

0 = Do not center or normalize the data on the indicated mode.
1 = Center the data on the indicated mode.
2 = Normalize the data on the indicated mode.
3 = Center *and* normalize the data on the indicated mode.
4 = Apply equal-average-diagonal normalization. (If used, this option must be applied to both mode A and B, and in this case, IFCENC can only be 0 or 1).
5 = Apply equal-average-diagonal normalization, followed by centering. (If used, this option must be applied to both Mode A and Mode B, and in this case, IFCENC can only be 0 or 1).

Column	*Default Value*	*Parameter Name*	*Explanation*
5	0	IFCODE	Flag to indicate whether special data values ("missing-data code values") are used to identify missing data cells. This flag is also used to indicate when limits on the valid data range are being specified. 0 = Do not check; 1 = Data will be checked for values equal to missing data codes and for values outside the specified range for valid data. (See Note 1 and Note 2 at the end of Section I for more details.)
7	0	MISEST	Flag to specify how to get starting values for the iterative estimation of the quantities to fill the missing data cells. Starting values for missing data cell estimates are needed at the beginning of computation for each solution. (See Note 3 at the end of Section I for additional details.)

MISEST values are interpreted as follows:

0 = Estimates at the end of one solution are used as the starting estimates for the next solution.
1 = Estimates are re-initialized on iteration 1 of each new solution. Thus, the starting estimates for the missing data are the same for all solutions.

The following parameters are used only if IFCODE is 1:

Column	*Default Value*	*Parameter Name*	*Explanation*
10–19	0.	DMISS1	Missing-data code value.
20–29	10.E30	DMISS2	Missing-data code value.
30–39	10.E30	DMISS3	Missing-data code value.

When IFCODE is 1, a data cell is considered to be missing data if its initial value matches any one of the three missing-data code values. If zero is used as a code value, it must be put in cols. 10–19. If zero is not a code value, put some other number in cols. 10–19 (e.g., a missing-data code value or a number that is outside the range of the data). Note that the value of 10 raised to the 30th power is outside the range of most data sets and therefore will not cause any additional data cells to be treated as missing values.

Column	*Default Value*	*Parameter Name*	*Explanation*
40–49	–10.E30	DLOWR	Lower limit for valid data range.
50–59	10.E30	DUPPER	Upper limit for valid data range.

If specified, the lower limit must have a value that is smaller than that of the upper limit. Otherwise, an error message is printed and execution stops. If IFCODE is 0, these limits are ignored.

Table 2.1 (Continued)

CARD I–5 FORMAT: (I1,1X,D16.0) STARTING POSITION PARAMETERS FOR THE ANALYSIS.

Column	*Default Value*	*Parameter Name*	*Explanation*
1	0	ISTART	Starting position type. ISTART values are interpreted as follows: 0 = Random loadings are generated at the beginning of each solution 1 = Standard continuation. Starting loadings for each solution are supplied by the user, and input according to a format supplied by the program. Usually these loadings are the output from a previous PARAFAC analysis. 2 = Analysis with variable format input of the loadings. Starting loadings for each solution *and* input format for the loadings are supplied by the user.
3–18	0219843517.D0	SEED	Seed for the random number generator (used only if ISTART is 0). SEED must be a double precision whole number in the inclusive range (2.D0 to 2147483646.D0). If it is outside this range, the default value is used by the program. Note: After the string of one to 10 digits, the end of the number should consist of a decimal point, followed by the letter "D", followed by the numeral 0. This defines the seed as a double precision whole number.

CARD I–6 FORMAT: (3I1,1X,I1,1X,3I1,1X,I3) ANALYSIS OPTIONS

Column	*Default Value*	*Parameter Name*	*Explanation*
1	1	IORTHA	Factor independence-dependence constraint for Mode A.
2	1	IORTHB	Factor independence-dependence constraint for Mode B.
3	2	IORTHC	Factor independence-dependence constraint for Mode C. IORTH flags for Modes A, B, and C are interpreted as follows: 1 = Allow oblique factor loadings in this mode during all iterations. 2 = Require uncorrelated factors in this mode during early iterations (when solution is far from convergence) 3 = Require uncorrelated factors in this mode up to middle stage of iterations (until solution is less than one power of 10 from the convergence criterion). 4 = Require uncorrelated factors in this mode during all iterations. 5 = Require orthogonal factor loadings in this mode during early iterations. 6 = Require orthogonal factors in this mode up to middle stage of iterations (when solution is less than one power of 10 from the conversence criterion). 7 = Require orthogonal factor loadings in this mode during all iterations. Note: orthogonal and zero correlation constraints have the same effect when factors have a mean loading of zero in the given mode (e.g. when the data has been centered in the given mode).
5	0	IGDIAG	Option to ignore data diagonals (i.e., treat them as missing values). Values in diagonal cells will be iteratively reestimated during the analysis. The initial estimates are the values from the data array. Estimates at the end of a solution are used as the starting estimates for the next solution. (0 = do not estimate diagonals; 1 = estimate diagonals.)
7	0	IFHLDA	Option to hold Mode A fixed during analysis.
8	0	IFHLDB	Option to hold Mode B fixed during analysis.
9	0	IFHLDC	Option to hold Mode C fixed during analysis. Flags IFHLDA, IFHLDB, IFHLDC allow the user to hold the loadings for a given mode or modes at their initial value (except for normalization) while using PARAFAC to estimate optimal values for the other mode(s), given the fixed mode(s). IFHLD flags for Modes A, B, and C are interpreted as follows: 0 = Iteratively compute new loadings for the indicated mode. 1 = Hold the initial values of the loadings for the indicated mode constant during analysis (except for renormalization). 2 = Set all loadings to 1.0 for the indicated mode, and hold them constant during the analysis (except for renormalization).
11-13	5	IRINTV	Interval (i.e. number of iterations) between successive computations of the fit value R (correlation between data and predictions of the model) during the analysis. Each time R is computed it is checked to see that it is still increasing, and changes in the analysis procedure are made if it is not. After each check, R is written out along with related information. (Set IRINTV to −1 if you want to suppress this checking and outputting of R.)

Table 2.1 (Continued)

CARD I–7 FORMAT: (3G10.4) **CONVERGENCE CRITERION FOR EACH MODE.** For a given mode, this specifies the largest percentage change allowed on any loading from one iteration to the next, if the solution is to be considered "converged." (Percentage change values are computed by dividing the absolute value of each loading change by the root-mean-square loading value for that factor in that mode, and then multiplying by 100.) A convergence message is printed when all three convergence criteria are met; the values of the loadings at that iteration are then output, and the program proceeds immediately to the next solution.

Column	*Default Value*	*Parameter Name*	*Explanation*
1–10	0.1	DIFMXA	Mode A convergence criterion (i.e., the maximum percentage change allowed in Mode A).
11–20	value of DIFMXA	DIFMXB	Mode B convergence criterion (i.e., the maximum percentage change allowed in Mode B).
21–30	value of DIFMXA	DIFMXC	Mode C convergence criterion (i.e., the maximum percentage change allowed in Mode C).

CARD I–8 FORMAT: (3I2,1X,I1) **OUTPUT OPTIONS.** (See also "Note 4: I/O Units" at the end of Section I of this table)

Column	*Default Value*	*Parameter Name*	*Explanation*
1–2	7 or ISTDLD	IUNITG	Output unit for special disk, punched or tape copy of the final loadings from each solution. (This copy is in addition to the one on the listing.) The output format is supplied by the program. These loadings may be used as the starting loadings for a continuation of the analysis in a subsequent run (with ISTART on Card I–5 set to 1 in the subsequent run). To suppress output of this special copy, specify "–1" for IUNITG.
3–4	—	IUNITD	Output unit (if any) for revised data (e.g., centered data or data with missing values estimated). The data are output at the end of each solution. The default is to not write them out. (This unit is also used for output of any synthesized data.) If IUNITD is specified, DATFMT (Card I–8A) must also be specified.
5–6	—	IUNITF	Output unit (if any) for residuals. They are output at the end of each solution. The default is to not write them out. If IUNITF is specified, RSDFMT (Card I–8B) must also be specified.
8	3	ISTANM	Flag to indicate method of standardization of loadings before output. Except when ISTANM=4, the standardization causes the loadings for two modes to have a mean squared loading of 1.0 for each factor, with compensatory rescaling of the loadings in the other mode to reflect the scale of the data. 1 = Mode A reflects the scale of the data, Modes B and C standardized. 2 = Mode B reflects the scale of the data, Modes A and C standardized. 3 = Mode C reflects the scale of the data, Modes A and B standardized. 4 = Modes A and B jointly reflect the scale of the data; Mode C has a mean square of 1.0. 5 = Do not standardize any mode.

The following card is used *only* if IUNITD on Card I–8 is specified. When required, this card follows Card I–8.

(Card I–8A) FORMAT: (80A1) (Cols. 1–80) **"DATFMT": FORMAT FOR OUTPUT OF REVISED OR SYNTHESIZED DATA.** This is the format for one row of data (see explanation of Record II-3 and II-4 below for more details on what is meant by "one row"). It must be enclosed in parentheses, and must specify F, E or G format. (If the data is to be written on a line printer, include a carriage control character in DATFMT.)

The following card is used *only* if IUNITF on Card I–8 is specified. When required, this card follows either Card I–8A, or (if I–8A is not used) it immediately follows Card I–8.

(CARD I–8B) FORMAT: (80A1) (Cols. 1–80) **"RSDFMT": FORMAT FOR OUTPUT OF RESIDUALS.** This is the format for one row of residuals (see explanation of Record II-3 and II-4 below for more details on what is meant by "one row"). It must be enclosed in parentheses and must specify F, E or G format. (If residuals are to be written on a line printer, include a carriage control character in RSDFMT.)

Table 2.1 (Continued)

SECTION I NOTES

NOTE 1: MISSING OR INVALID DATA VALUES. Any data cells which, on input, contain missing-data code values or values which are outside the specified range limits will be considered to have missing data. Their location in the data set is entered into the array of subscripts of missing data cells, and the values in these locations are iteratively re-estimated during the analysis.

NOTE 2: MISSING-DATA SUBSCRIPTS. If the user wishes to identify certain locations in the data array as containing missing data *without* using code values, then he may have the subscripts identifying these locations entered *directly* into the array of subscripts of missing-data cells (see Record II-(X+1) to Record II-Y).

NOTE 3: STARTING VALUES FOR MISSING-DATA VALUE ESTIMATES. Locations of missing-data cells may be specified directly by a subscript table immediately following the data and/or indirectly by missing-data code values. The initial estimate for any data cell whose location is listed directly in the subscript table is the value contained in that cell upon input, i.e. the value read from the data set. In contrast, the initial estimate for any data cell containing a missing—data code value is the mean computed from all other valid data values in the same "tube" (i.e., all points on the same Mode A and mode B level and across all levels of mode C). Therefore, two or more cells containing missing-data code values which are located in the same tube will have the same initial estimate. Any data point which is listed in the subscript table *and* which contains a missing-data code value is treated like the others in the table (i.e., its first estimate is the value read from the data set and *not* the mean computed across all points in the same tube).

These initial estimates are used on iteration 1 of the first solution. For all subsequent iterations, the missing-data estimates do not depend on whether the cell locations were specified directly or indirectly. Starting estimates used for iteration 1 of the following solutions depend on the value of MISEST.

NOTE 4: I/O UNITS.

a. At most installations, the operating system provides "standard" units for Fortran input and output (e.g. 5 is often the standard input unit and 6 the standard output unit). When the program reads and writes using the "standard" units, no extra system control cards are required. The standard version of PARAFAC assumes that 5 is the standard input unit and 6 the standard output unit. However, the DIMS program can be used to create a revised version of PARAFAC where ISTDIN (the number of the standard input unit) and ISTDOU (the number of the standard output unit) can be set to any values desired, to make PARAFAC compatible with the conventions at the user's installation. The first 8 cards (I-1 to I-8) are always read from unit ISTDIN, and the main PARAFAC output listing is written on ISTDOU.

b. When the user requests special input and/or output on units different from the standard (usually 5 and 6, respectively), the appropriate system control cards must be used to access the input files and to save the output files. The form of these cards depends on the particular computer and operating system at the user's installation.

c. Users with a Cyber installation may specify only units 1—9 inclusive (unless they change the "Program" statement at the beginning of the PARAFAC Fortran source).

Table 2.1 (Continued)

SECTION II: DATA PARAMETERS AND DATA SET

This second input section provides the data set and information about the data. It is read from IUNITB (see Card I–3). Preceeding the data set one must put a data title, followed by information on the size and format of the data array. Following the data set is an optional list of the subscripts of any data cells for which data is missing. If for IUNITB the user selects the standard input unit (usually 5), the Section II information is read from cards put immediately after Card I–8 (no end-of-file or other separation is used). The normal (and recommended) procedure, however, is to keep Section II information in a separate file on tape or disk, with Record II–1 the first record of that file.

RECORD II–1 | **FORMAT: (80A1)** (Cols. 1–80)

DATA TITLE: A general description of the data set that (a) identifies it easily for future reference and (b) distinguishes it from any other versions of the data which may also be (or have been) analyzed. This record provides identifying information that stays with the data. It is also printed out as the second line at the top of every loading output table and thus helps to document the output of a PARAFAC analysis.

RECORD II–2 | **FORMAT: (3I4)**

DATA SET DIMENSIONS. The data set is NAS by NBS by NCS points. NAS gives the number of items per row of the data matrix (even if this involves more then one Fortran record). NBS gives the number of rows per "slice" or matrix (i.e. per level of Mode C). NCS gives the number of "slices" or matrices assembled into the three-way data set.

Column	*Parameter Name*	*Explanation*
1–4	NAS	Number of levels or items in Mode A.
5–8	NBS	Number of levels or items in Mode B.
9–12	NCS	Number of levels or items in Mode C.

RECORD II–3 | **FORMAT: (80A1)** (Cols. 1–80)

"VARFMT": DATA INPUT FORMAT. Format for reading one row of the data matrix (this may be a multi-record format). The format must be enclosed in parentheses, and must specify F, E or G format for input of the data. To input data stored as integers, use F format with zero places to the right of the decimal point (e.g. use F2.0 to read two-column integer data). VARFMT should provide for reading all the levels of Mode A at a fixed level of Mode B and C. (See also comment for Record II–4.)

RECORDS II–4 to II–X (NBS by NCS sets of records) | **FORMAT: VARFMT**

DATA ARRAY. There are NCS blocks of records. Every block contains NBS sets of records, and each set is read according to VARFMT. In Fortran, the data points are input in the following way:

```
     DO 10 K=1, NCS
     DO 10 J=1, NBS
  10 READ (IUNITB, VARFMT) (POINT (I,J,K), I=1, NAS)
```

} in PARAFAC source; do not input with data

RECORD II–(X+1) to II–Y | **FORMAT: (3I4)**

MISSING-VALUE SUBSCRIPTS TABLE: Optional. Each record indicates the location of one data cell with missing data.

Column	*Parameter Name*	*Explanation*
1–4	I	Level of Mode A; maximum value is NAS.
5–8	J	Level of Mode B; maximum value is NBS.
9–12	K	Level of Mode C; maximum value is NCS.

} Subscripts for a POINT (I,J,K) with missing data.

RECORD II–(Y+1) | **FORMAT: (I4)** (Cols. 1–4)

TERMINATOR RECORD FOR SECTION II. –001 in columns 1–4 is a fixed code which marks the end of the subscripts. It must *always* be included, even if no missing value subscripts are specified.

Table 2.1 (Continued)

SECTION III: STARTING LOADINGS (Optional)

When it is required, this third input section is used to provide nonrandom starting loadings for the analysis. It is read from IUNITC (see Card I–3). When the loadings are read from the same unit as the data, i.e. when both IUNITB and IUNITC are the same (for example if both are set to be the standard input unit) the loadings information is placed immediately after the "–001" record of Section II. When IUNITB is not the standard input unit, but IUNITC is, the loadings immediately follow Card I–8 of Section I. Usually, all three sections are in separate files. In Section III, there should be NSOLS (see Card I–2) complete sets of loadings, one set for each solution.

This section is necessary only if the user chooses to read in starting loadings rather than generate them randomly, i.e., if ISTART (Card I–5) is 1 or 2. Since the form of this input depends on whether ISTART is 1 or 2, it will be discussed separately for the two cases.

(a) Standard Continuation (ISTART = 1)

For standard continuations, the loadings are read according to a standard format provided internally by the program. Generally, they are the output from a previous PARAFAC analysis. Loadings from another source may be used only if the format is identical to that described in the table below.

The table describes the format for one complete set of loadings. It must be repeated NSOLS times (see Card I–2), so that there is a set of starting loadings for each solution requested. The PARAFAC loadings file may be used exactly as it was written or punched out (by IUNITG) if NSOLS has the same value for both the initial and continuation analyses.

Record No.	*Format*	*Explanation*
III–1 to III–6	–	Information about the loadings sets: analysis and data titles, predicted data fit values, Mode A heading, etc. This information is printed out during the input phase of PARAFAC, to document which loadings set is being used. However, since these labels are not used in the actual analysis, blank records can be placed here instead. In either case, there must always be *6* records preceding the loadings.
III–7 to – (NAS sets of records)	**(5X,6G12.4)**	Mode A factor loadings. Each of the NAS sets of records consists of the loadings on NFACT factors for one level of Mode A.
(2 Records)	–	Blank record and "Mode B" heading. Two blank records may be substituted.
(NBS sets of records)	**(5X,6G12.4)**	Mode B factor loadings. Each of the NBS sets of records consists of the loadings on NFACT factors for one level of Mode B.
(2 Records)	–	Blank record and "Mode C" heading. Two blank records may be substituted.
(NCS sets of records)	**(5X,6G12.4)**	Mode C factor loadings. Each of the NCS sets of records consists of the loadings on NFACT factors for one level of Mode C.

Repeat this pattern for second and successive solutions

Table 2.1 (Continued)

(b) Continuation with Variable Input Format for the Loadings (ISTART = 2)

This type of loadings input is generally used when the loadings are from a source other than PARAFAC (e.g., to input theoretical values determined by the user). The input format for the loadings in each mode must be specified individually. These formats are always read from the same unit as the loadings.

The table below describes the format for one complete set of loadings. It must be duplicated NSOLS times, so that there is a set of starting loadings for each solution requested.

Record No.	*Format*	*Parameter Name*	*Explanation*
III–1	**(80A1)** **(Cols. 1–80)**	**FORMTA**	Format for input of a single level of Mode A loadings. It must be enclosed in parentheses, and must specify F, E or G format. (Integers should be read using F*n*.0 e.g. F2.0 for two-column integers.)
III–2 to III–X (NAS sets of records)	**FORMTA**	–	Mode A factor loadings. Each of the NAS sets of records consists of the loadings on NFACT factors for one item or level of Mode A.
III–(X+1) (1 record)	**(80A1)** **(Cols. 1–80)**	**FORMTB**	Format for input of a single level of Mode B loadings. It must be enclosed in parentheses, and must specify F, E or G format. (Integers should be read using F*n*.0 format, e.g. F2.0 for two-column integers.)
III–(X+2) to III–Y (NBS sets of records)	**FORMTB**	–	Mode B factor loadings. Each of the NBS sets of records consists of the loadings on NFACT factors for one item or level of Mode B.
III–(Y+1) (1 record)	**(80A1)** **(Cols. 1–80)**	**FORMTC**	Format for input of a single level of Mode C loadings. It must be enclosed in parentheses, and must specify F, E or G format. (Integers should be read using F*n*.0, e.g. F2.0 for two-column integers.)
III–(Y+2) to III–Z (NCS sets of records)	**FORMTC**	–	Mode C factor loadings. Each of the NCS sets of records consists of the loadings on NFACT factors for one item or level of Mode C.

Repeat this pattern for second and successive solutions

2.4 PARAFAC EXAMPLES (FOR DATA ANALYSIS)

In this section, examples are given of the input that is described in the PARAFAC Input Specifications Table. First, individual cards from Input Section I are shown; then, examples of Input Sections I, II and III are presented.

2.4.1 Examples Of Individual Cards/Records

Card I-1 (80A1) -Job Title

You can leave this card blank, and a blank line will be output (i.e., no default info is inserted). However, it is better to input some sort of documentation here to indicate what distinguishes the current run from other analyses.

Card I-2 (3I4) -NFACT, NSOLS, NITER, NOUTS

```
   0  -1
```

(1) No data analysis will be done, but data preprocessing as requested on Card I-3 will be carried out; used most often when synthesizing data.

```
   0   0   4  50
```

(2) Maximum 200 iterations per solution; loadings will be output on the lineprinter listing after iterations 50, 100, 150 and 200 of the analysis, unless the solution converges before. 2 factors will be extracted; 3 different starting positions will be used (defaults).

```
   0   0   1 200
```

(3) Same as example 2, except factor loadings are output only once -- after 200 iterations.

Card I-3 (2I2,1X,I1) -IUNITB, IUNITC, IFSYMT

```
 1 2
```

(4) Data (Input Section II) to be read from logical unit 1; starting loadings for data analysis or synthesis (Input Section III) to be read from logical unit 2; appropriate system commands must be included to access these files.

Card I-4 (3I1,1X,I1,1X,I1,2X,5G10.4) -IFCENA, IFCENB, IFCENC, IFCODE, MISEST, DMISS1, DMISS2, DMISS3, DLOWR, DUPPER
(See Chapter 4 before specifying IFCEN- values.)

```
312 1 1  -9.0      0.0       0.0       -6.0      6.0
```

(5) Data to be centered on Modes A and B, normalized on Modes A and C; a value of -9.0 indicates missing data; data with values outside the range -6.0 to 6.0 to be treated as missing even if they are not equal to -9.0 (e.g., keypunch errors); solution 1 starting estimates for missing values are tube means, solution 2 starting estimates are final estimates from solution 1, etc.

```
312 1 1  -9.0      0.0       0.0       0.0       0.0
```

(6) Same as 5, except the valid data range is virtually unbounded (i.e., -10.E30 to 10.E30 by default).

```
312 1 1  0.0       0.        0.0       1.0       100.0
```

(7) Same as example 5, except 0.0 is the missing data code, and the valid data range is 1.0 to 100.0.

```
312 1 1  10.E30    0.0       0.0       1.0       100.0
```

(8) Same as 7, except no missing data codes are used (or the missing data code is 10.E30).

```
312 1 1  0.0       -9.0      0.0       1.        100.
```

(9) Same as 7, except both 0.0 and -9.0 are missing data codes. Note that 0. must be the <u>first</u> missing data code specified (cf example 5).

<u>Card</u> <u>I-5</u> (I1,1X,D16.0) -ISTART, SEED

```
0 0003401260.D0
```

(10) Random starting loadings to be used for the analysis; seed for random number generator is specified.

<u>Card</u> <u>I-6</u> (3I1,1X,I1,1X,3I1,1X,I3) -IORTHA, IORTHB, IORTHC, IGDIAG, IFHLDA, IFHLDB, IFHLDC, IRINTV

```
700 0 000   1
```

(11) Factors in Mode A constrained to be orthogonal throughout the entire analysis; change in fit value checked and output after every iteration; default values to be assigned for other parameters.

<u>Card</u> <u>I-7</u> (3G10.4) -DIFMXA, DIFMXB, DIFMXC

```
0.05
```

(12) Convergence criterion for all three modes to be twice as stringent (percentage change allowed is half as big) as the default value.

<u>Card</u> <u>I-8</u> (3I2,1X,I1), <u>I-8A</u> and <u>I-8B</u> (80A1) -IUNITG, IUNITD, IUNITF, ISTANM (and DATFMT, RSDFMT)

```
 0 0 0 0
```

(13) Final loadings written to logical unit 7 and Mode C reflects the magnitude of the data values (defaults); no format line required. Appropriate system commands must be included with the job to save the unit 7 file.

```
-1 0 6
(1H ,6F12.7/1H ,4F12.7)
```

(14) No output to disk files; residuals are to be listed on the standard output unit according to the format shown on the second line. Note the inclusion of the carriage control characters. This format is for a data array that has 10 values in each row (i.e., 10 levels of Mode A).

```
-1 3 6
(5G15.7/5G15.7)
(1H ,6F12.7/1H ,4F12.7)
```

(15) Same as 14, except the preprocessed data is to be written on logical unit 3 according to the format on the second line; format for the residuals is on the third line. Appropriate system commands must be included with the job to save the unit 3 file.

2.4.2 Examples Of Input Section I

2.4.2.1 Analysis Of Two-way Data -

(16) <u>Principal components analysis (direct fitting)</u>
The input data is a matrix of raw scores arranged as variables (Mode A) by people (Mode B). The data should be transformed to z-scores by normalizing across variables and centering across people (IFCENA=2, IFCENB=1), and the variables factor weights should reflect the scale of the data (ISTANM=1). Orthogonality constraints are imposed on both modes (IORTHA=IORTHB=7) to obtain the principal components axis orientation. These parameters are circled in the deck setup below. Others are either assigned example values or left blank (zero).

```
CARD
 I-1 EXAMPLE 2-D PRINCIPAL COMPONENTS ANALYSIS
 I-2     2
 I-3  1
 I-4 210
 I-5 0 0230821715.D0
 I-6 771
 I-7
 I-8  7 0 0 1
```

(17) Principal components analysis (indirect fitting)
The input data is a (symmetric) matrix of correlations or covariances. The input deck is the same as 16 above, except no preprocessing of the data is necessary, and so Card I-4 is as follows:

```
000
```

Or, if you input covariances but want correlations, you can obtain correlations by requesting EAD normalization on Card I-4 as follows:

```
440
```

(18) Common factor analysis
As for example 17, the input data is a matrix of correlations or covariances. The analysis differs from 17 in that it involves iteration on the diagonal (IGDIAG=1). The pertinent parameters are circled in the deck setup below, while some of the others have been given example values.

```
CARD
 I-1  EXAMPLE 3-D COMMON FACTOR ANALYSIS
 I-2     3
 I-3   1
 I-4  000
 I-5  0 1142785923.D0
 I-6  771 1
 I-7
 I-8   7 0 0 1
```

2.4.2.2 Direct Fitting Of Three-way Data -

See examples 22 and 23 below. Usually, of course, you don't input everything from the file as shown in example 22. Neither do you usually input starting loadings for the analysis unless continuing a previous run, as illustrated in example 23. Thus you would usually specify some of the parameters (e.g., IUNITB, IUNITC, ISTART) differently than shown in the examples.

2.4.2.3 Indirect Fitting Of Three-way Data -

(19) Covariance analysis (see Sections 4.4 and 6.7)
The input data are covariance matrices that are symmetric across Modes A and B (IFSYMT=1), have EAD normalization applied to Modes A and B (IFCENA= IFCENB=4), and the data scale should be jointly reflected in Modes A and B of the factor loadings

(ISTANM=4). These parameters are circled in the example deck below; other parameters have been assigned example values or left blank.

```
CARD
 I-1 |EXAMPLE COVARIANCE ANALYSIS; 2-D SOLUTION
 I-2 |    2
 I-3 | 1 0 1
 I-4 |440
 I-5 |0 0010348205.D0
 I-6 |000 0 000  -1
 I-7 |
 I-8 | 7 0 0 4
```

(20) Multidimensional scaling (see Sections 4.6 and 6.8) The input data are scalar product matrices (possibly output from the DISTIN program) that are symmetric across Modes A and B (IFSYMT=1), and they may be normalized on Mode C (IFCENC=2). The Mode C factor weights reflect the scale of the data (ISTANM=3) -- (cf covariance analysis).

```
CARD
 I-1 |EXAMPLE 2-D MDS ANALYSIS
 I-2 |    2
 I-3 | 1 0 1
 I-4 |002
 I-5 |0 0000428295.D0
 I-6 |000 0 000  -1
 I-7 |
 I-8 | 7 0 0 3
```

2.4.2.4 Three-way Data Preprocessing For Direct Fitting -

(21) One-cycle preprocessing (see Section 4.3)
Suppose you have 3-way ratings data with 10 levels in Mode A. You want to center Modes A and B, normalize Modes A and C, and then do a 2-D analysis. You ordinarily accomplish this in one job, by specifying NFACT=2 on Card I-2 and IFCEN-=312 on Card I-4. To do this via "one-cycle" preprocessing, however, requires two steps.
Step 1: The input data is raw score data. Center the data as required (IFCENA= IFCENB=1 in this example) and save the centered data without analysing it (NSOLS=-1; IUNITD=2). Include system commands to save the centered data.

```
CARD
 I-1 ONE-CYCLE PREPROCESSING -- STEP 1
 I-2          -1
 I-3  1
 I-4 110
 I-5
 I-6
 I-7
 I-8 -1 2
I-8A (5G15.7/5G15.7)
```

Step 2: The input data is the centered data from step 1. Normalize the data as required (IFCENA= IFCENC=2 in this example) and analyse as usual.

```
CARD
 I-1 ONE-CYCLE PREPROCESSING -- STEP 2
 I-2     2
 I-3  1
 I-4 202
 I-5 0 0000025173.D0
 I-6
 I-7
 I-8  7 0 0 3
```

Alternatively, in step 2 you could suppress the analysis (NSOLS=-1) and save the normalized data. In a third run, you would then analyse the data that had been saved in the second step, but no data preprocessing would be done.

2.4.2.5 User-supplied Starting Loadings -

(22) Nonstandard loadings format
The following example illustrates the deck arrangement when both the data (Input Section II) and starting loadings (Input Section III) are read from the standard input unit (IUNITB= IUNITC=5). So that the example may be presented on one page, the entire data array is not shown; however, the card numbers indicate how many lines of data there would be altogether. Missing data subscripts are included with the data set (in contrast to using missing data codes on Card I-4); the two values specified as missing are circled in the data. The starting loadings are in nonstandard format (ISTART=2). Only one set of two factors is included and so only one 2-D solution is requested (NFACT=2, NSOLS=1).

```
   CARD
    I-1 |EXAMPLE OF EVERYTHING READ FROM UNIT 5
    I-2 |    2    1
    I-3 | 5 5
    I-4 |
    I-5 |2
    I-6 |
    I-7 |
    I-8 |
   II-1 |EXAMPLE DATA SET
   II-2 |    7    5    8
   II-3 |(1X,7F7.3)
   II-4 |   3.404 -2.777  1.567 -2.586  0.662  0.814 -0.001
      : |  -3.217  2.626 -1.509  2.440 -0.660 -0.759  3.832
      : |   2.144 -1.743  0.917 -1.637  0.334  0.537 -2.322
      : |  -1.963  1.600 -0.899  1.491 -0.376 -0.471  2.280
      : |  -2.443  1.989 -1.070  1.863 -0.410 -0.603  2.711
      : |   1.540 -1.240  0.507 -1.196  0.057  0.440 -1.268
      : |  -0.297  0.282 -0.639  0.159 -0.660  0.108  1.665
      : |   3.771 -2.934 -0.068 -3.103 -1.432  1.544  0.330
      : |  -1.073  0.857 -0.266  0.845  0.064 -0.338  0.656
      : |  -3.303  2.579 -0.060  2.703  1.111 -1.310  0.024
      : |      :      :      :      :      :      :      :
      : |
      : |   0.250 -0.195  0.001 -0.205 -0.088  0.100  0.007
      : |   0.417 -0.318 -0.086 -0.354 -0.253  0.199  0.243
      : |   1.737 -3.976 -0.205 -1.453 -0.869  0.774  0.608
      : |  -0.249  0.193  0.013  0.206  0.105 -0.105 -0.044
  II-43 |  -1.419  1.093  0.161  1.186  0.701 -0.629 -0.479
      : |    7    1    1
      : |    2    3    8
  II-46 |-001
  III-1 |(2G12.4)
  III-2 |  -1.422       -1.652
      : |   1.160        1.271
      : | -0.6579       0.2142
      : |   1.080        1.384
      : | -0.2806       0.8487
      : | -0.3386      -0.7424
      : |   1.670      -0.6281
  III-9 |(2G12.4)
      : |  -1.274      -0.1911
      : |   1.222      -0.4658
      : | -0.7584       -1.685
      : |  0.7315       0.2138
      : |  0.8799        1.365
 III-15 |(2G12.4)
      : |   1.872       0.4502E-01
      : |  0.6582        1.100
      : |  0.2688       0.9783
      : |   2.189       0.7985
      : |   1.426       0.8334
      : |  0.2107       0.8974
      : |  0.4916       0.9061
 III-23 |   1.502       0.1187
```

(23) Standard loadings format
The following illustrates the continuation of a PARAFAC analysis, using final loadings from the first run as starting loadings in the second. You would do this whenever solutions do not converge before the maximum allowed number of iterations has been reached, or when you want to impose more stringent convergence criteria on (converged) solutions that you already have. For the continuation run, the data must be preprocessed in the same way and NFACT must have the same value; NSOLS will usually also have the same value. The values of other parameters may be different, however.
Job 1: Three 2-D solutions are requested, and a maximum of 200 iterations is allowed for each. The data are input from logical unit 1. Random starting loadings are used (ISTART=0). Final loadings are written on logical unit 7; system control commands must be included to save this file.

```
CARD
 I-1 |JOB 1:  3 2-D SOLUTIONS
 I-2 |   2
 I-3 | 1
 I-4 |312
 I-5 |0 0030518619.D0
 I-6 |
 I-7 |
 I-8 |
```

Job 2: Continuing the analyses begun in job 1, the maximum allowed number of iterations is increased to 300. The final loadings saved in job 1 are input from logical unit 2 (IUNITC=2; ISTART=1).

```
CARD
 I-1 |CONTINUATION OF 3 2-D SOLUTIONS FROM JOB 1
 I-2 |   2   3   2 150
 I-3 | 1 2
 I-4 |312
 I-5 |1
 I-6 |
 I-7 |
 I-8 |
```

2.4.3 Examples Of Input Section II

(24) Single-record format and missing-value subscripts
In example 22 above, IUNITB= ISTDIN=5, and so the data are included in the same file as the analysis control parameters. If IUNITB≠ ISTDIN, the information on Cards II-1 through II-46 would be in a separate file.

That data set is a case where values for all levels of Mode A are contained on a single line. It also includes two sets of missing-value subscripts; the circled data values are at the locations indicated by those subscripts.

(25) Multi-record format
The following is an example of a data array where the values for the 11 levels of Mode A span 3 records. Card I-3 specifies a format to read all 3 records, not just the first one. To save space, only the first and last parts of the data file are shown.

```
  CARD
  II-1   EXAMPLE DATA SET
  II-2     11   15   40
  II-3   (4G14.7/4G14.7/3G14.7)
  II-4    -.5319867        1.988933        -.3579961        -.2289961
  II-5    -.3253686E-01    1.048057        -.3314863        -.5062028
  II-6     1.593186       -1.932368         .5049098
    :      1.296400       -.8910265E-01   -1.819356         -.7503637
    :      1.300004         .6574843      -1.367316         -.2070050
    :       .8240627        .7563386      -1.007790
    :          :               :               :                :
    :          :               :               :                :
    :       .7732101       -.6742343        2.136340        -.5338494
    :       .5192412       -.7221175       -.9940822E-01    -.2064337
II-1803   -.9411721       -.3429637       -.5200354
II-1804  -001
```

2.4.4 Examples Of Input Section III

(26) Nonstandard loadings format
Example 22 contains an example of variable format loadings (Cards III-1 to III-23). They are included in the same file as the analysis control parameters because IUNITC= ISTDIN=5 in that example; otherwise, they would be in a separate file. There is only one set of loadings because NSOLS=1 on Card I-2. In contrast, when NSOLS=3, three sets of loadings are required, one for each solution.

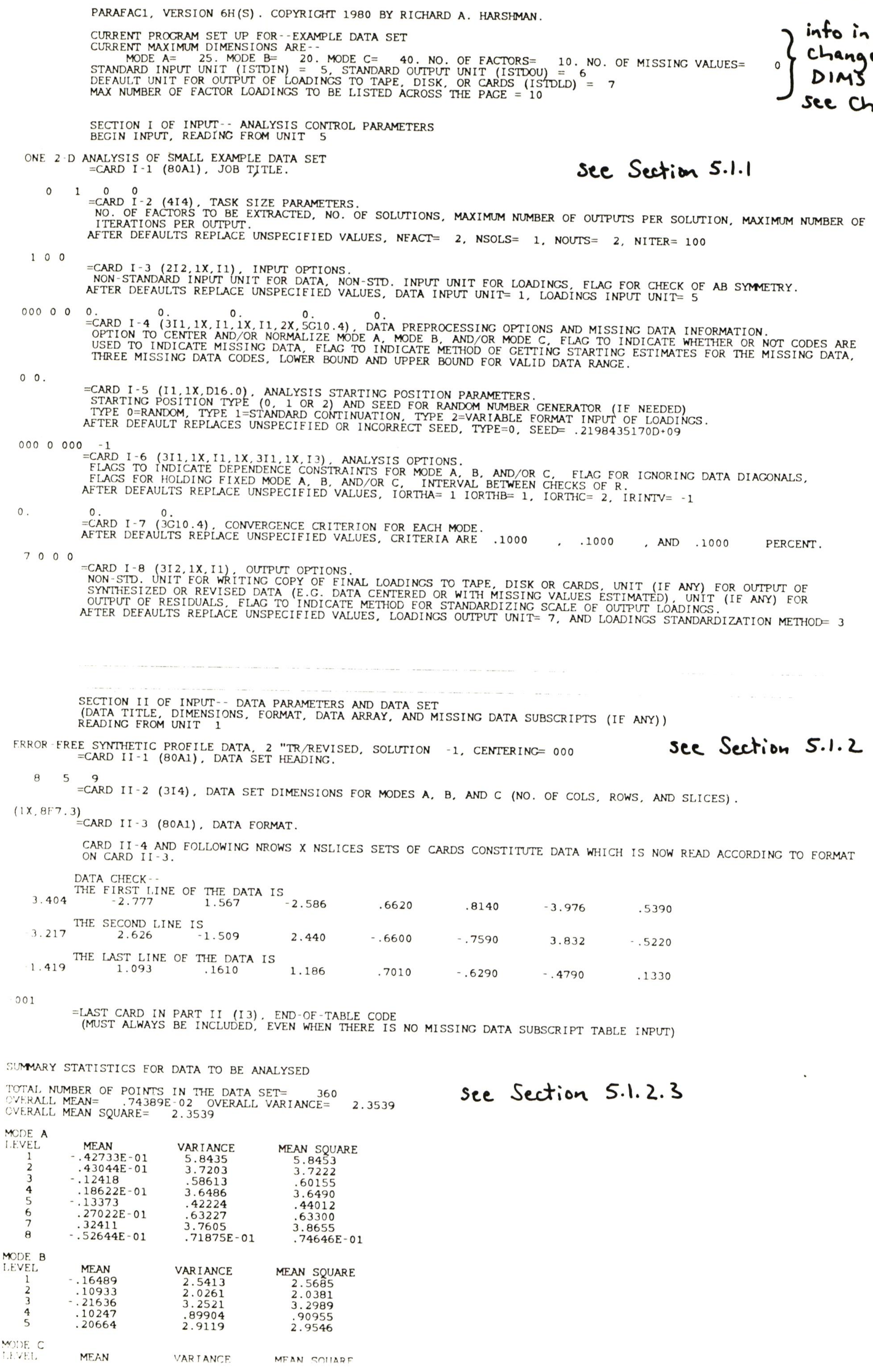
(27) Standard loadings format
The loadings from example 22 are shown below in standard PARAFAC format (ISTART=1). If they had in fact been output by PARAFAC, there would be additional information on Cards II-1 to II-6, II-14, -15, -21 and -22. While this information is useful as documentation, it is not necessary, and so these records have all been left blank below.

```
  CARD
 III-1
 III-2
 III-3
 III-4
 III-5
 III-6
 III-7        -1.422      -1.652
   :           1.160       1.271
   :          -0.6579      0.2142
   :           1.080       1.384
   :          -0.2806      0.8487
   :          -0.3386     -0.7424
   :           1.670      -0.6281
III-14
III-15
III-16        -1.274      -0.1911
   :           1.222      -0.4658
   :          -0.7584     -1.685
   :           0.7315      0.2138
   :           0.8799      1.365
III-21
III-22
III-23         1.872       0.4502E-01
   :           0.6582      1.100
   :           0.2688      0.9783
   :           2.189       0.7985
   :           1.426       0.8334
   :           0.2107      0.8974
   :           0.4916      0.9061
III-30         1.502       0.1187
```

2.4.5 Example Of PARAFAC Output

Table 2.2 shows the lineprinter output produced by PARAFAC during the analysis of a small data set. The data were synthesized in a previous run (see example 1 in Section 2.6). Chapter 5 explains the listing contents in more detail.

Table 2.2. Example of PARAFAC Lineprinter Output (Analysis)



```
          PARAFAC1, VERSION 6H(S). COPYRIGHT 1980 BY RICHARD A. HARSHMAN.

          CURRENT PROGRAM SET UP FOR--EXAMPLE DATA SET
          CURRENT MAXIMUM DIMENSIONS ARE--
               MODE A=    25. MODE B=    20. MODE C=    40. NO. OF FACTORS=    10. NO. OF MISSING VALUES=      0
          STANDARD INPUT UNIT (ISTDIN) =  5, STANDARD OUTPUT UNIT (ISTDOU) =  6
          DEFAULT UNIT FOR OUTPUT OF LOADINGS TO TAPE, DISK, OR CARDS (ISTDLD) =  7
          MAX NUMBER OF FACTOR LOADINGS TO BE LISTED ACROSS THE PAGE = 10


          SECTION I OF INPUT-- ANALYSIS CONTROL PARAMETERS
          BEGIN INPUT, READING FROM UNIT  5

ONE 2-D ANALYSIS OF SMALL EXAMPLE DATA SET
          =CARD I-1 (80A1), JOB TITLE.

   0   1   0   0
          =CARD I-2 (4I4), TASK SIZE PARAMETERS.
           NO. OF FACTORS TO BE EXTRACTED, NO. OF SOLUTIONS, MAXIMUM NUMBER OF OUTPUTS PER SOLUTION, MAXIMUM NUMBER OF
           ITERATIONS PER OUTPUT.
          AFTER DEFAULTS REPLACE UNSPECIFIED VALUES, NFACT=  2, NSOLS=  1, NOUTS=  2, NITER= 100

 1 0 0
          =CARD I-3 (2I2,1X,I1), INPUT OPTIONS.
           NON-STANDARD INPUT UNIT FOR DATA, NON-STD. INPUT UNIT FOR LOADINGS, FLAG FOR CHECK OF AB SYMMETRY.
          AFTER DEFAULTS REPLACE UNSPECIFIED VALUES, DATA INPUT UNIT= 1, LOADINGS INPUT UNIT= 5

000 0 0  0.        0.        0.        0.        0.
          =CARD I-4 (3I1,1X,I1,1X,I1,2X,5G10.4), DATA PREPROCESSING OPTIONS AND MISSING DATA INFORMATION.
           OPTION TO CENTER AND/OR NORMALIZE MODE A, MODE B, AND/OR MODE C, FLAG TO INDICATE WHETHER OR NOT CODES ARE
           USED TO INDICATE MISSING DATA, FLAG TO INDICATE METHOD OF GETTING STARTING ESTIMATES FOR THE MISSING DATA,
           THREE MISSING DATA CODES, LOWER BOUND AND UPPER BOUND FOR VALID DATA RANGE.

0 0.
          =CARD I-5 (I1,1X,D16.0), ANALYSIS STARTING POSITION PARAMETERS.
           STARTING POSITION TYPE (0, 1 OR 2) AND SEED FOR RANDOM NUMBER GENERATOR (IF NEEDED)
           TYPE 0=RANDOM, TYPE 1=STANDARD CONTINUATION, TYPE 2=VARIABLE FORMAT INPUT OF LOADINGS.
          AFTER DEFAULT REPLACES UNSPECIFIED OR INCORRECT SEED, TYPE=0, SEED= .2198435170D+09

000 0 000  -1
          =CARD I-6 (3I1,1X,I1,1X,3I1,1X,I3), ANALYSIS OPTIONS.
           FLAGS TO INDICATE DEPENDENCE CONSTRAINTS FOR MODE A, B, AND/OR C,  FLAG FOR IGNORING DATA DIAGONALS,
           FLAGS FOR HOLDING FIXED MODE A, B, AND/OR C,  INTERVAL BETWEEN CHECKS OF R.
          AFTER DEFAULTS REPLACE UNSPECIFIED VALUES, IORTHA= 1 IORTHB= 1, IORTHC= 2, IRINTV= -1

0.        0.        0.
          =CARD I-7 (3G10.4), CONVERGENCE CRITERION FOR EACH MODE.
          AFTER DEFAULTS REPLACE UNSPECIFIED VALUES, CRITERIA ARE  .1000     ,  .1000     , AND  .1000     PERCENT.

 7 0 0 0
          =CARD I-8 (3I2,1X,I1), OUTPUT OPTIONS.
           NON-STD. UNIT FOR WRITING COPY OF FINAL LOADINGS TO TAPE, DISK OR CARDS, UNIT (IF ANY) FOR OUTPUT OF
           SYNTHESIZED OR REVISED DATA (E.G. DATA CENTERED OR WITH MISSING VALUES ESTIMATED), UNIT (IF ANY) FOR
           OUTPUT OF RESIDUALS, FLAG TO INDICATE METHOD FOR STANDARDIZING SCALE OF OUTPUT LOADINGS.
          AFTER DEFAULTS REPLACE UNSPECIFIED VALUES, LOADINGS OUTPUT UNIT= 7, AND LOADINGS STANDARDIZATION METHOD= 3



          SECTION II OF INPUT-- DATA PARAMETERS AND DATA SET
          (DATA TITLE, DIMENSIONS, FORMAT, DATA ARRAY, AND MISSING DATA SUBSCRIPTS (IF ANY))
          READING FROM UNIT  1

ERROR-FREE SYNTHETIC PROFILE DATA, 2 "TR/REVISED, SOLUTION  -1, CENTERING= 000
          =CARD II-1 (80A1), DATA SET HEADING.

   8   5   9
          =CARD II-2 (3I4), DATA SET DIMENSIONS FOR MODES A, B, AND C (NO. OF COLS, ROWS, AND SLICES).

(1X,8F7.3)
          =CARD II-3 (80A1), DATA FORMAT.

           CARD II-4 AND FOLLOWING NROWS X NSLICES SETS OF CARDS CONSTITUTE DATA WHICH IS NOW READ ACCORDING TO FORMAT
           ON CARD II-3.

          DATA CHECK--
          THE FIRST LINE OF THE DATA IS
  3.404     -2.777      1.567     -2.586      .6620      .8140     -3.976      .5390

          THE SECOND LINE IS
 -3.217      2.626     -1.509      2.440     -.6600     -.7590      3.832     -.5220

          THE LAST LINE OF THE DATA IS
 -1.419      1.093      .1610      1.186      .7010     -.6290     -.4790      .1330

 001
          =LAST CARD IN PART II (I3), END-OF-TABLE CODE
           (MUST ALWAYS BE INCLUDED, EVEN WHEN THERE IS NO MISSING DATA SUBSCRIPT TABLE INPUT)


SUMMARY STATISTICS FOR DATA TO BE ANALYSED

TOTAL NUMBER OF POINTS IN THE DATA SET=     360
OVERALL MEAN=    .74389E-02  OVERALL VARIANCE=    2.3539
OVERALL MEAN SQUARE=    2.3539

MODE A
LEVEL       MEAN          VARIANCE       MEAN SQUARE
  1      -.42733E-01      5.8435          5.8453
  2       .43044E-01      3.7203          3.7222
  3      -.12418           .58613          .60155
  4       .18622E-01      3.6486          3.6490
  5      -.13373           .42224          .44012
  6       .27022E-01      .63227          .63300
  7       .32411          3.7605          3.8655
  8      -.52644E-01      .71875E-01      .74646E-01

MODE B
LEVEL       MEAN          VARIANCE       MEAN SQUARE
  1      -.16489          2.5413          2.5685
  2       .10933          2.0261          2.0381
  3      -.21636          3.2521          3.2989
  4       .10247           .89904          .90955
  5       .20664          2.9119          2.9546

MODE C
LEVEL       MEAN          VARIANCE       MEAN SQUARE
```

info in headings changed via DIMS program; see Ch. 3

see Section 5.1.1

see Section 5.1.2

see Section 5.1.2.3

Table 2.2 (Continued)

```
  1        .36100E-01      3.5433         3.5446
  2       -.51250E-02      1.9855         1.9855
  3       -.10750E-01      1.1537         1.1538
  4        .30025E-01      6.2557         6.2566
  5        .14400E-01      3.2914         3.2916
  6       -.10625E-01      .93914         .93925
  7       -.52500E-02      1.2736         1.2736
  8        .27600E-01      2.3547         2.3555
  9       -.94250E-02      .38459         .38468


         BEGINNING OF SOLUTION  1

THE STARTING LOADINGS ARE RANDOM NUMBERS.  THE INITIAL SEED FOR THE RANDOM NUMBER GENERATOR IS  .2198435170D+09
```

```
ONE 2-D ANALYSIS OF SMALL EXAMPLE DATA SET
ERROR-FREE SYNTHETIC PROFILE DATA, 2 "TR/REVISED, SOLUTION  -1, CENTERING= 000
SOL 1, ITERATION    0, MEAN SQ ERROR=  2.404702    , STRESS=1.0107307
R= .0216673 RSQ= .0004695 DIFFA=0.          DIFFB=0.         DIFFC=0.
```

see Section 5.2.1

```
MODE A
          1             2
   1    .2390        -.4107
   2   1.448          .6550
   3   -.1475E-01    1.369
   4  -1.328         -.7366
   5   -.8896        2.023
   6    .3179        -.2031E-01
   7   1.200         -.7655
   8   1.322          .5532
MODE B
          1             2
   1  -1.376         1.631
   2   -.1812         .6718
   3    .2421E-01   -1.122
   4   1.477          .4348
   5    .9454        -.6642
MODE C
          1             2
   1    .1089        -.9393E-01
   2   -.3087        -.1387
   3    .3446E-01    -.1486E-01
   4    .3909         .1013
   5    .1398         .1263
   6    .3024         .2937E-01
   7   -.6644E-01    -.2764E-01
   8   -.3503         .2165
   9    .2902E-01     .1007

ROOT MEAN SQUARED CONTRIBUTION FOR EACH FACTOR
       .2357         .1121
```

See Section 5.2.2.1

```
DEPENDENCE CONSTRAINTS IN EFFECT FOR--
MODE C
```

see Section 8.1.2.3

```
CONSTRAINT THAT MODE C FACTORS BE INDEPENDENT WAS DROPPED BEFORE ITERATION   12.  DIFFC= 4.988     PERCENT.


CONVERGENCE CRITERION MET ON ITERATION   19
MODE A MAXIMUM CHANGE =    .901503E-02 PERCENT
MODE B MAXIMUM CHANGE =    .361696E-01 PERCENT
MODE C MAXIMUM CHANGE =    .521205E-02 PERCENT
```

see Section 8.1.3

```
NO DEPENDENCE CONSTRAINTS IN EFFECT
```

Table 2.2 (Continued)

```
ONE 2-D ANALYSIS OF SMALL EXAMPLE DATA SET
ERROR-FREE SYNTHETIC PROFILE DATA, 2 "TR/REVISED, SOLUTION  -1, CENTERING= 000
SOL 1, ITERATION   19, MEAN SQ ERROR=  .1112202E-06, STRESS= .0002174
R=1.0000000 RSQ=1.0000000 DIFFA= .9015E-02 DIFFB= .3617E-01 DIFFC= .5212E-02
```

See Section 5.2.2.3

```
MODE A
          1             2
   1  -1.422        -1.652
   2   1.160         1.271
   3  -.6579          .2144
   4   1.080         1.384
   5  -.2806          .8488
   6  -.3386        -.7424
   7   1.670        -.6288
   8  -.2268          .1663

MODE B
          1             2
   1  -1.274        -.1911
   2   1.222        -.4659
   3  -.7584        -1.685
   4   .7315          .2138
   5   .8799         1.365

MODE C
          1             2
   1   1.872         .4503E-01
   2   .6584         1.100
   3   .2689          .9782
   4   2.189          .7984
   5   1.426          .8334
   6   .2108          .8973
   7   .4918          .9059
   8   1.502          .1186
   9   .3130E-01      .6121

ROOT MEAN SQUARED CONTRIBUTION FOR EACH FACTOR
       1.219          .7828
```

see Section 5.2.3

```
CROSS-PRODUCTS OF NORMALIZED FACTORS
(I.E. COSINES OF ANGLES BETWEEN FACTORS)

MODE A
       1      2
1  1.000   .513
2   .513  1.000
```

see Section 5.2.4

```
MODE B
       1      2
1  1.000   .462
2   .462  1.000


MODE C
       1      2
1  1.000   .563
2   .563  1.000
```

```
CORRELATIONS OF FACTOR LOADINGS
```

see Section 5.2.5

```
MODE A
       1      2
1  1.000   .506
2   .506  1.000


MODE B
       1      2
1  1.000   .498
2   .498  1.000


MODE C
       1      2
1  1.000  -.506
2  -.506  1.000
```

```
ERROR ANALYSIS FOR SOLUTION 1
MODE A
LEVEL   MEAN SQ ERROR
   1      .8624946E-07
   2      .8956451E-07
   3      .9456774E-07
   4      .1102534E-06
   5      .1030322E-06
   6      .1023541E-06
   7      .2216783E-06
   8      .8206155E-07

MODE B
LEVEL   MEAN SQ ERROR
   1      .9007235E-07
   2      .1210248E-06
   3      .1468931E-06
   4      .8456512E-07
   5      .1135454E-06

MODE C
 LEVEL   MEAN SQ ERROR
```

see Section 5.2.6

Table 2.2 (Continued)

```
    1       .8966895E-07
    2       .1588353E-06
    3       .9459584E-07
    4       .1001498E-06
    5       .1174497E-06
    6       .1280869E-06
    7       .1444743E-06
    8       .8311072E-07
    9       .8460999E-07
```

see Section 8.1.3

```
THE SEED FOR THE RANDOM NUMBER GENERATOR AT END OF EXECUTION IS  .6771805000D+09
```

2.5 DATA SYNTHESIS

The PARAFAC input deck has the same general form whether one is doing data analysis or data synthesis (or both in the same run). When doing data synthesis, however, you of course don't input a data set. The data input format (Record II-3) is replaced by a special codeword that indicates data synthesis is to be performed; the following four cards are then used for synthesis parameters rather than data. All other input parameters are the same as for data analysis.

A detailed description of Input Section II for data synthesis is presented in Table 2.3. Chapter 7 contains additional information about the parameters in that table.

CHAPTER 9

PFCORE PROGRAM

PFCORE is a Fortran batch program, written especially for use with PARAFAC. It has been tested fairly thoroughly, and will probably be included with the PARAFAC Analysis Package in the future. Although the code has not been checked against PFORT specifications, PFCORE should be easy to install on most systems.

PFCORE is a further development for dealing with "degenerate" PARAFAC solutions (see Section 6.2 in the *Reference Manual for the PARAFAC Analysis Package*, hereafter referred to simply as the "manual"). As mentioned in the manual, complex data structure that cannot be represented by the PARAFAC model, but can be fit by the Tucker T2 or T3 models, will cause the PARAFAC solution to be degenerate. The Tucker solutions do not possess the unique axis property of PARAFAC, however. What PFCORE does is combine the optimal features of the two models -- PARAFAC unique axes and Tucker generality -- to shed light on *why* the degenerate PARAFAC solution occurred in the first place. The Tucker T2 and T3 models are fit to the data, using the factor weights of a *constrained* PARAFAC solution (see Section 6.2.1.3 in the manual) as estimates for the TUCKALS components. This allows PFCORE to noniteratively compute the corresponding T2 and T3 core arrays.

The core arrays show the across-mode interactions amongst the (constrained) PARAFAC factors, which the PARAFAC model does not allow for. The interaction patterns may indicate a violation of the PARAFAC assumption of angle invariance between any two factors, for example, which would help explain why the unconstrained PARAFAC solution was degenerate. As Lundy, Harshman and Kruskal (1985) did, you can actually interpret the core array to gain understanding of the relationships in the data. This would not be possible if the degenerate PARAFAC solution (with uninterpretable factors) were used instead.

9.1 USING PFCORE

It is recommended that you arrange your data so that Mode C refers to people (Mode C is not reduced to a common space by the T2 model). The T2 core slices will then show the Mode A and B factor interactions for individuals, and the T3 slices will show the interactions for "idealized people".

9.1.1 PFCORE Limitations

The current PFCORE arrays have the following limits (the parameter names printed in upper case are defined in the Input Specifications Table below):

1. Maximum number of points in the data array is 39000 (i.e., NAS*NBS*NCS⩽39000).

2. Maximum number of points in the factor loading matrices for Modes A and B is 250 (i.e., NAS*NFACT⩽250 and NBS*NFACT⩽250).

3. Maximum number of points in the Mode C factor loading matrix is 400 (i.e., NCS*NFACT⩽400).

4. Maximum number of levels in Mode A and in Mode B is 40 (i.e., NAS⩽40 and NBS⩽40).

5. Maximum number of factors is 10 (i.e., NFACT⩽10).

These limits may be changed by modifying the appropriate dimension statements and assignment statements at the beginning of the program.

9.1.2 PFCORE I/O Units

The logical units used for system standard input and output are 5 and 6 respectively. These may be changed by modifying the appropriate assignment statements at the beginning of the program (ISTDIN=5 for input; ISTDOU=6 for output).

9.2 PFCORE INPUT

Input to PFCORE consists of three sections. Listed in order of input, they are:

1. Job parameters (input from logical unit 5)

2. Data parameters and data set (input from IUNITB, a job parameter specified by the user; default is 5)

3. PARAFAC loadings (input from IUNITC, a job parameter specified by the user; default is 5)

Usually your data and PARAFAC solution will be stored in separate diskfiles, but it is possible to input everything from one file (i.e., by taking the defaults for IUNITB and IUNITC). If using separate files, remember to include system control statements with your job to access them.

9.2.1 PFCORE Input Specifications Table

Input Section 1: Job Parameters

The first input section consists of five records read from the standard input unit. They are explained in detail below. Parameter names are printed in upper case. The expression in parentheses that follows each record number is the Fortran input format for parameters on that record. Integer values should be right-justified in their respective input fields.

Record 1 (80A1) Job Title (columns 1-80): Description of the current job, for your own information

Record 2 (I4) NFACT (columns 1-4): Number of factors in the PARAFAC solution

Record 3 (4I2) Input and Scaling Options

Column	Default Value	Parameter Name, Explanation
1-2	5	IUNITB, data input unit
3-4	5	IUNITC, PARAFAC loadings input unit
5-6	3	ISTANM, mode that reflects data scale in PARAFAC solution; explained on page 2-9 in the PARAFAC manual
7-8	0	IDSCOR, option to have core matrix reflect data scale (0=No, 1=Yes)

Record 4 (2I2) Output Options

Column	Default Value	Parameter Name, Explanation
1-2	0	ITCORE, core array(s) to be output 0=T2 (extended) core array only 1=T3 (compressed) core array only 2=both T2 and T3 core arrays
3-4	0	IUNITD, output unit for disk copy of core array(s). THIS OPTION IS NOT FUNCTIONAL IN THIS VERSION OF PFCORE.

Record 5 (80A1) Output Format for Core Array(s) (columns 1-80): This is the Fortran format for outputting one row of the core array (one row contains NFACT values). The format must be enclosed in parentheses, include a carriage control character, and specify F, E or G format for the core array elements. See page 2-16 of the manual for examples.

Input Section 2: Data Parameters and Data Arrays

The data are read from IUNITB, specified on Record 3 above. The format is exactly the same as for the PARAFAC program, described on page 2-11 of the PARAFAC manual. See page 2-22 for an example data set.

There are two points to note about data input:

1. Usually, you want the data to correspond to the PARAFAC solution that you input to PFCORE (i.e., you want to input the data that produced the solution). In most cases, this is not the raw data that was input to the PARAFAC analysis. If you preprocessed the data or specified missing values, you should save the data at the end of the solution and use these data as input for PFCORE (see Card I-4 and the IUNITD parameter on Card I-8, described on pages 2-7 and 2-9 in the manual).

2. If they are included with the data, PFCORE ignores missing value subscripts and the "terminator record" that are written out by PARAFAC. You would have to delete these records from the data set only if the PARAFAC solution was to be read from the same file as the data (i.e., if IUNITC=IUNITB).

Input Section 3: PARAFAC Solution

The PARAFAC factor loadings are read from IUNITC, specified on Record 3 above. Output by PARAFAC, they are already in the format required for input to PFCORE. If you use loadings obtained from some other source, you must arrange them in "standard PARAFAC" format, as described on page 2-12 of the manual; an example is given on page 2-22.

9.3 PFCORE OUTPUT

PFCORE outputs everything to the standard output unit (i.e., logical unit 6). First, the input is documented; then fit values and core array(s) are listed. Input of the data and the factor loadings is verified in the same way as for PARAFAC, described in Sections 5.1.2 and 5.1.3 of the manual.

9.3.1 Fit Values

Fit values output by PFCORE are R, RSQ, MEAN SQ ERROR and STRESS; they are explained in Section 5.2.1 of the PARAFAC manual. These fit values are given for five models: PARAFAC (the most restricted model) and the Tucker T3, T2, T1(A) and T1(B) models. Note that the fit values for the "T" models are not computed using TUCKALS components for Modes A, B and C, but rather using the PARAFAC components that were input.

Except for roundoff error, fit values for the PARAFAC model will agree with the ones listed as documentation for the input PARAFAC solution. If not, check your data. (For some purposes, you might deliberately input a solution that was not obtained by analysis of the input data; then, of course, the fit values would be different.)

You will see that the fit improves as the models become more general, which is due at least partly to the increasing number of parameters of the model. Thus, as you go from PARAFAC to the T1 models, the R and RSQ values increase and MEAN SQ ERROR and STRESS decrease. Both T1 models are general, but one may fit more parameters than the other (depending on NAS and NBS), and so that one would be expected to fit the data better.

Extreme differences in the fit values suggest the presence of data structure that is more general than can be fit by a restricted model like PARAFAC or even T3 (e.g., the T1(A) fit values for the metaphor data in Lundy, Harshman and Kruskal, 1985). To be sure whether the improvement in fit for a less restricted model is evidence of more general structure in the data, and not just due to the greater number of parameters and/or chance, you would of course need to do a Monte Carlo study (briefly described in the Lundy et al paper).

9.3.2 Core Arrays

The core arrays are listed as a series of 2-way matrices or "slices", NFACT slices for the T3 core and NCS slices for the T2 core. Each slice shows what are essentially "interactions" between the Mode A and B components for each "idealized" person (in the T3 case) and for each individual (in the T2 case).

Table 2.3

PARAFAC INPUT FOR DATA SYNTHESIS

SECTION I: ANALYSIS CONTROL PARAMETERS

This first section of input for the generation (and analysis) of synthetic data is set up as described in Section I of the Input Specifications Table in the chapter on PARAFAC input. The same 8 cards can be used here to control the analysis of the synthetic data once it is generated. On the other hand, if no immediate analysis of the generated data is desired, these cards can be mostly left blank. Simply specify a value of –1 (negative one) for the NSOLS parameter on Card I–2 (cols. 5–8). In this case, it is not necessary to specify any of the parameters on Cards I–3 to I–7 (blank cards may be used instead). On Card I–8, however, you need to specify IUNITD (cols. 3–4) and provide a DATFMT (cols. 1–80 on Card I–8A) if you want the synthetic data to be output.

SECTION II: DATA PARAMETERS (AND LOADINGS TO GENERATE DATA)

The organization of the second input section, when doing data synthesis, is similar to that for general data analysis. However, the actual raw data is replaced by parameter cards and externally supplied loadings (if any) that are used in the generation of the synthetic data. The program assigns default values to some of the parameters if they are not specified by the user. All information except the loadings is read from IUNITB (cols. 1–2 on Card I–3). The loadings are input from IUNITC (cols. 3–4 on Card I–3). (Of course, IUNITB and IUNITC can be set to the same unit, e.g. 5 or ISTDIN, if desired.)

RECORD II–1 FORMAT: (80A1) **DATA TITLE:** This record should contain a verbal description of the data set that is to be generated; it will become the first line of the generated data file. See the explanation of Record II–1 in the PARAFAC Input Specifications Table for more details.

RECORD II–2 FORMAT: (3I4) **DESIRED DIMENSIONS FOR GENERATED DATA SET:** The data set to be generated will contain NAS by NBS by NCS points.

Column	*Parameter Name*	*Explanation*
1–4	NAS	Number of levels or items in Mode A.
5–8	NBS	Number of levels or items in Mode B.
9–12	NCS	Number of levels or items in Mode C.

RECORD II–3 FORMAT: (4A1) **CODE WHICH SELECTS SYNTHETIC DATA GENERATION:** "SYNT" in cols. 1–4 is a fixed code which specifies that synthetic data is to be generated by the program. This replaces the data input format (VARFMT) which is put here when data is to be read in.

Column	*Parameter Name*	*Contents of Col. 1–4*
1–4	–	SYNT

RECORD II–4 FORMAT: (I4) **NUMBER OF FACTORS:** This parameter specifies the number of factors to be used to generate the "true" or systematic part of the data. If omitted, the number will be set equal to the number specified for analysis (i.e. NFACT from Card I–2).

Column	*Default Value*	*Parameter Name*
1–4	= NFACT	NFGEN

Table 2.3 (Continued)

RECORD II-5 **FORMAT: (I1,1X,D16.0)** **TYPE OF FACTOR LOADING INPUT FOR DATA GENERATION:** These parameters determine the *source* of the factor loadings which will define the underlying factor structure of the "true" or systematic part of the synthetic data. (These loadings should not be confused with those selected by the starting position parameters on Card I-5. Card I-5 selects the starting loadings for use in the iterative *analysis*, which occurs after the data is generated.) The size of the factor loading tables used to generate the data will be NAS by NFGEN, NBS by NFGEN, and NCS by NFGEN for Modes A, B and C, respectively.

Column	*Default Value*	*Parameter Name*	*Explanation*
1	0	ILDGIN	Type of source for the loadings used to generate the "true" or systematic part of the synthetic data. ILDGIN values are interpreted as follows: 0 = Random loadings are to be generated (rectangularly distributed). 1 = Loadings in standard format are to be read in, as provided by the user. See Section III(a) in the PARAFAC Input Specifications Table for a description of the format. 2 = Loadings with nonstandard format are to be read in, as provided by the user. See Section III(b) in the Input Specifications Table for a description of the arrangement of the input.
3-18	0594420173.D0	SEED2	Seed for the random number generator (used only if ILDGIN is 0). SEED2 must be a double precision whole number in the inclusive range (2.D0 to 2147483646.D0). If it is outside this range, the default value is used by the program. Note: after the string of one to 10 digits, the end of the number should contain a decimal point, followed by the letter "D", followed by the numeral 0. This defines the seed as a double precision whole number.

RECORD II-6 **FORMAT: (I1,1X,3I1, 1X,3I1,1X,I1,1X, I1,G10.4)** **CHARACTERISTICS OF THE TRUE OR SYSTEMATIC PART OF THE DATA.** These parameters determine the general characteristics of the "true" or systematic part of the data to be synthesized. Data type 0 selects profile data, appropriate for factor analysis. Data type 1 selects dissimilarity data, appropriate for multidimensional scaling. (This data would require preprocessing with DISTIN before analysis with PARAFAC.) Data type 3 selects cross-product or covariance type data, appropriate for factor analysis. Unipolarity constraints allow data to be generated with all loadings having the same sign (as in "positive manifold") in any or all modes. Dependence constraints allow the user to specify that the factors be orthogonal or uncorrelated across the levels of any particular mode. The factor size multipliers determine the distribution of relative sizes of the factors used to generate the "true" part of the data, and the data size multiplier determines the scale of the "true" part of the data.

Column	*Default Value*	*Parameter Name*	*Explanation*
1	0	IDATYP	Type of synthetic data to be generated. IDATYP values are interpreted as follows: 0 = Raw score or "profile" data 1 = Dissimilarity data 2 = Cross-product, covariance, or scalar product data
3	*	IFAPOS	Mode A unipolarity option. * (for default value see Note 1; below.)
4	*	IFBPOS	Mode B unipolarity option. * (for default value, see Note 1; below.)
5	*	IFCPOS	Mode C unipolarity option. * (for default value, see Note 1; below.)

Table 2.3 (Continued)

			IFAPOS, IFBPOS and IFCPOS flags are interpreted as follows: 0 = Default value will be assigned depending on the mode and the type of data involved (see "Note 1: Defaults" below). 1 = Allow bipolar (positive and negative) loading patterns in the specified mode. 2 = Constrain loadings to be unipolar (i.e. all positive or all negative for each factor in the specified mode). 3 = "Center" loadings for each factor in the specified mode (i.e. adjust loadings so that, in the specified mode, the mean loading for each factor will be zero).
7	*	IFAORT	Factor independence - dependence constraint for Mode A. * (for default value, see Note 1, below.)
8	*	IFBORT	Factor independence - dependence constraint for Mode B. * (for default value, see Note 1, below.)
9	*	IFCORT	Factor independence - independence constraint for Mode C. * (for default value, see Note 1, below.) IFAORT, IFBORT and IFCORT are interpreted as follows: 0 = Default value will be assigned depending on the mode and type of data involved (see "Note 1: Defaults" below.) 1 = No constraint will be imposed on the factors in the specified mode. 2 = Constrain the factors in the specified mode to be uncorrelated. 3 = Constrain the factors in the specified mode to be orthogonal.
11	0	ISTRMS	Option to standardize the root-mean-square of the true part of the data. 0 = Do not standardize; let the scale of the true part of the data be determined by the number and size of the factors, as well as DSIZE. 1 = Standardize, so that the root-mean-square of the data points, before error is added, is equal to DSIZE; (this option is useful for precisely determining the variance contributed by the true part of the data).
13	0	ISZFAC	Method of selecting factor size multipliers. When ISZFAC = 0 or 1 (i.e. when factors are allowed to have different overall variances or mean squares) a multiplier is randomly selected for each "size" factor, and Mode C loadings for that factor are scaled up or down by this size multiplier. Since, with real data, it is seldom the case that all factors account for the same proportion of the data variance (or mean square), these multipliers are used to simulate naturally occuring variations in the relative sizes or variance contributions of the factors used to generate the true part of the data. If a triangular distribution of sizes is used, factors are more likely to have similar sizes, whereas a rectangular distribution will tend to give a wider range of variations in the relative proportions of variance contributed by the different factors underlying a particular set of synthetic data. 0 = A size multiplier for each factor is randomly selected from a triangular distribution ranging from 0.1 to 1.9, with a mean of 1.0. 1 = A size multiplier for each factor is randomly selected from a rectangular distribution ranging from 0.1 to 1.9, with a mean of 1.0. 2 = The size multiplier for all factors is 1.0.
14–23	1.0	DSIZE	Size multiplier for the "true" part of the data. When data of the type specified by IDATYP is generated from factors which have the characteristics requested via other parameters on this record, the variance of the true part of the data will depend on the factor sizes, factor covariances, and other characteristics of the true part of the data. DSIZE allows this variance to be adjusted. If ISTRMS is zero, each data point is then multiplied by DSIZE (and consequently the variance of the true part is multiplied by the square of DSIZE). If ISTRMS is one, DSIZE is first divided by the root-mean-square of the unadjusted data, before being used as the adjustment factor for each data point (and consequently the original value of DSIZE becomes the root-mean-square of the true part of the data). (See Note 2 for further details).

Table 2.3 (Continued)

RECORD II–7 FORMAT: (I1,1X,D16.0, 1X,4G10.4) CHARACTERISTICS OF THE RANDOMLY VARYING PARTS OF THE DATA.

Column	*Default Value*	*Parameter Name*	*Explanation*
1	0	IERTYP	Type of random error to be added to each synthetic data point. 0 = No error 1 = Constant variance, uniform distribution 2 = Proportional variance, uniform distribution 3 = Constant variance, normal distribution 4 = Proportional variance, normal distribution 5 = "Lognormal" distribution 6 = Constant variance, "slash" distribution 7 = Proportional variance, "slash" distribution
3–18	0807833162.D0	SEED3	Seed for the random number generator (used only if IERTYP or ACONSZ is *not* zero). The range restrictions and special format, noted above for SEED2 (Record II–5), also apply to SEED3.
20–29	0.0	ERRSIZ	Size multiplier for the random error specified by IERTYP. If the default is taken, no random error is added, regardless of the value of IERTYP (except when CONPRB and CONSIZ are nonzero).
30–39	0.0	ACONSZ	Size multiplier for additive constants. If IERTYP is zero, ACONSZ is added to each data point (i.e. the additive constants are all 1.0). If IERTYP is greater than zero, an additive constant is generated for each level of Mode C as a random number uniformly distributed between zero and one. The constant is multiplied by ACONSZ and added to all data points at that level of Mode C (i.e. each "slice" of the data matrix has a different additive constant).
40–49	0.0	CONPRB	Probability that each data point is contaminated by extra large error (i.e. probability that the point is an "outlier"). The distribution type for the error of contaminated points is the same as for the non-contaminated points, i.e. it has the shape determined by IERTYP. However, the expected standard deviation of the error of the contaminated points is different as determined by CONSIZ, described below. (If IERTYP is zero, CONPRB is ignored). Note that since CONPRB is a probability, it can assume only values from zero to one inclusive.
50–59	0.0	CONSIZ	ERRSIZ multiplier for data points that have extra large error (i.e. for "outliers"). CONSIZ is used only if *both* IERTYP and CONPRB are greater than zero. In general, ERRSIZ is the error size multiplier for all points that are not outliers, while the product ERRSIZ*CONSIZ is the error multiplier for the outliers. However, the user may sometimes want the non-contaminated points to be error-free. In such cases, ERRSIZ is zero and CONSIZ is nonzero (and IERTYP and CONPRB are nonzero). With such specifications, the error multiplier that is used for the outliers is CONSIZ (i.e. ERRSIZ is assumed to be 1.0 when calculating error for the outliers, and zero otherwise.) In this case, *only* the "outliers" have error added.

RECORD II–8 to II–X (optional) **OPTIONAL FACTOR LOADINGS USED TO GENERATE THE "TRUE" COMPONENT OF THE DATA.** Data Generation Loadings are needed only if ILDGIN (Record II–5) is 1 or 2. When needed, these loadings are read from IUNITC (Card I–3). The format of this loadings section is as described in Section III of the PARAFAC Input Specifications Table, except that the loadings are read in during Part II and only *one* complete set of loadings is read in. (These values are used only to generate the data, and *not* to analyze it.) The records containing these loadings may be included among the Part II parameter records if IUNITC is the same as IUNITB. Otherwise they are in a separate file. See "Note 3: IUNITC" (below) for further details.

Table 2.3 (Continued)

RECORD II—(X+1) to II—Y **FORMAT: (3I4)** **MISSING-VALUE SUBSCRIPTS TABLE: Optional.** Each record indicates the location of one data cell with missing data. (Naturally, synthetic data is not going to have cells with missing values. However, the user can cause the output synthetic data file to be written with a missing-value subscript table which will cause certain cells to be treated *as if* they contained missing data. This might be useful when simulating real data which contained missing values, or studying the effects of missing data on the solution of synthetic data problems.)

Column	*Parameter Name*	*Explanation*
1—4	I	Level of Mode A; maximum value is NAS.
5—8	J	Level of Mode B; maximum value is NBS.
9—12	K	Level of Mode C; maximum value is NCS.
		Subscripts for a point (I,J,K) which is to be treated as missing data

RECORD II—(Y+1) **FORMAT: (I4)** **TERMINATOR RECORD FOR SECTION II.** —001 in cols. 1—4 is a fixed code which marks the end of the subscript table. It must *always* be included, even if no missing value subscripts are specified.

SECTION III: STARTING LOADINGS FOR ANALYSIS OF SYNTHESIZED DATA (Optional)

This third input section follows the format described in Section III of the Table of Input Specifications. The "Note on IUNITC" below should also be referred to. It is not necessary to include Section III (the section of starting loadings) if NSOLS is —1 or if ISTART is 0.

NOTES

NOTE 1: DEFAULTS. The default values for parameters IFAPOS to IFCORT inclusive on Record II—6 above depend on whether one is generating profile data, distance data, or covariance - like data (i.e. the defaults depend on the value given to IDATYP). These defaults are given in the table below.

IF IDATYP =	0	1	2
	DEFAULT VALUE OF THE PARAMETER		
IFAPOS=	1	3	1
IFBPOS=	1	3	1
IFCPOS=	2	2*	2
IFAORT=	1	1	1
IFBORT=	1	1	1
IFCORT=	1	1*	1

*When IDATYP is 1, the default value for IFCPOS takes precedence over any other value specified by the user; the program issues a message when it resets the value of IFCPOS. Also when IDATYP is 1, a user-specified value of 3 for IFCORT will be reset to 2 by the program, and a message will be printed to inform the user.

Table 2.3 (Continued)

NOTE 2: SPECIAL LIMITATIONS ON DSIZE. (a) For synthetic data which consists entirely of error, the user must specify a very small nonzero number for DSIZE (e.g. 10.E-30) so that the contribution of the "true" component is essentially zero. A DSIZE of exactly zero cannot be used for this purpose, because the default value of 1.0 is assigned if cols. 14–23 contain zero or are left blank.) (b) If IDATYP is one, DSIZE should be positive.

NOTE 3: IUNITC. If IUNITC is not the standard input unit, and factor loadings are to be input by the user for *both* data synthesis (ILDGIN is 1 or 2) *and* analysis (ISTART is 1 or 2), the information on IUNITC must be arranged in the following order:

a. *One* complete set of loadings, which is used to generate the data. The set is arranged as described in Part III of "PARAFAC INPUT," according to either the standard continuation (ILDGIN =1) or the continuation with variable input (ILDGIN =2).

b. *NSOLS* complete sets of loadings, which are used during analysis as starting positions for the different solutions. The sets are arranged as described for a standard continuation (ISTART =1) or as explained for a continuation with variable input (ISTART = 2).

2.6 PARAFAC EXAMPLES (FOR DATA SYNTHESIS)

This section presents examples of the input that is described above in Table 2.3. Whereas Section 2.4 presents separate examples for the three input sections, each example here consists of the entire input deck (usually Input Sections I and II only). More emphasis is placed on Section II, however, as this is where the synthetic data parameters are specified.

(1) This simple example produces the lineprinter output shown in Table 2.4. Default values are assigned for most parameters. The synthetic data is error-free raw score data, with 2 underlying "true" factors. The data and "true" factors are saved (include system commands to do so), but

the data are not analysed.

```
 CARD
  I-1 |GENERATE AND SAVE SMALL EXAMPLE DATA SET
  I-2 |      -1
  I-3 |
  I-4 |
  I-5 |
  I-6 |
  I-7 |
  I-8 | 7 2
 I-8A |(1X,8F7.3)
 II-1 |ERROR-FREE SYNTHETIC PROFILE DATA, 2 "TRUE" FACTORS
 II-2 |   8   5   9
 II-3 |SYNT
 II-4 |
 II-5 |
 II-6 |
 II-7 |
 II-8 |-001
```

(2) This example shows synthesis of raw score data with 2 underlying "true" factors and subsequent 3-D analysis in the same run. The data have normally distributed random error added (IERTYP=4) that contributes approximately 40% of the total variance (ISTRMS=1, DSIZE=1.0, ERRSIZ=0.8165; see Section 7.3). The data are not saved, but the factor loadings are (include system commands to do so). The loadings file will contain the 2-D "true" factor solution first, followed by the three 3-D solutions produced by the analysis. (These solutions were then compared; see the CMPARE output listing in Table 3.4).

```
 CARD
  I-1 |GENERATE DATA WITH 2 FACTORS AND DO 3-D ANALYSIS
  I-2 |   3   3
  I-3 |
  I-4 |
  I-5 |0 0004216538.D0
  I-6 |             -1
  I-7 |
  I-8 | 7
 II-1 |SYNTHETIC RAW SCORES, 2 "TRUE" FACTORS, 40% ERROR
 II-2 |  25  18  35
 II-3 |SYNT
 II-4 |   2
 II-5 |0 0081024173.D0
 II-6 |0 112 111 1 0 1.0
 II-7 |4 0000023149.D0    0.8165
 II-8 |-001
```

(3) In the example below, parameters are specified to generate dissimilarities data with error due to additive

constants. Note that NAS=NBS because for dissimilarities, Modes A and B refer to the same stimuli. The data are not analysed here; they must be converted to scalar products first (see DISTIN input example 1 in Section 3.4.5).

```
CARD
  I-1 GENERATE AND SAVE DISSIMILARITIES DATA
  I-2 }
   :  }  same as example 1
  I-7 }
  I-8 -1 2
 I-8A (1X,6G13.5/1X,6G13.5)
 II-1 DISSIMILARITIES, 3 TRUE FACTORS, ADDITIVE CONSTANTS
 II-2   12   12   20
 II-3 SYNT
 II-4    3
 II-5 0 0042951039.D0
 II-6 1 332 111 0 0 2.0
 II-7 5 0000559183.D0      0.0         1.0
 II-8 -001
```

(4) Suppose you want to generate raw score data with 3 underlying factors and no error added except for outliers; 8% of the data points are to be outliers contaminated with proportional variance error from the slash distribution (CONPRB=0.08, IERTYP=7). You then want to do a 3-D analysis of the data to see how the outliers affect the recovery of the true structure in the data. Both data and solutions are to be saved (include system commands to do so). The PARAFAC input would be as follows:

```
CARD
  I-1 GENERATE AND ANALYSE DATA WITH OUTLIERS
  I-2     3
  I-3  5
  I-4
  I-5 0 0000309158.D0
  I-6               -1
  I-7
  I-8  7 1
 I-8A (1X,5G14.6/1X,5G14.6/1X,5G14.6)
 II-1 SYNTHETIC RAW SCORES, 8% OUTLIERS
 II-2   15   19   14
 II-3 SYNT
 II-4    3
 II-5 0 0145326595.D0
 II-6 0 000 000 0 0 1.0
 II-7 7 2003678995.D0      0.0         0.0         0.0800      1.0
 II-8 -001
```

Table 2.4 is the lineprinter output produced by PARAFAC, given the input shown above in example 1.

Table 2.4. Example of PARAFAC Lineprinter Output (Synthesis)

```
          PARAFAC1, VERSION 6H(S). COPYRIGHT 1980 BY RICHARD A. HARSHMAN.

          CURRENT PROGRAM SET UP FOR--STANDARD PROGRAM
          CURRENT MAXIMUM DIMENSIONS ARE--
               MODE A=   18. MODE B=   18. MODE C=   35. NO. OF FACTORS=   10. NO. OF MISSING VALUES=   50
          STANDARD INPUT UNIT (ISTDIN) =  5, STANDARD OUTPUT UNIT (ISTDOU) =  6
          DEFAULT UNIT FOR OUTPUT OF LOADINGS TO TAPE, DISK, OR CARDS (ISTDLD) =  7
          MAX NUMBER OF FACTOR LOADINGS TO BE LISTED ACROSS THE PAGE = 10
```

} standard array sizes

```
          SECTION I OF INPUT-- ANALYSIS CONTROL PARAMETERS
          BEGIN INPUT, READING FROM UNIT  5

GENERATE AND SAVE SMALL EXAMPLE DATA SET
          =CARD I-1 (80A1), JOB TITLE.
```

see Section 7.4.1.1

```
   0  -1   0   0
          =CARD I-2 (4I4), TASK SIZE PARAMETERS.
           NO. OF FACTORS TO BE EXTRACTED, NO. OF SOLUTIONS, MAXIMUM NUMBER OF OUTPUTS PER SOLUTION, MAXIMUM NUMBER OF
           ITERATIONS PER OUTPUT.
          AFTER DEFAULTS REPLACE UNSPECIFIED VALUES, NFACT=  2, NSOLS= -1, NOUTS=  0, NITER=    0

 5 0 0
          =CARD I-3 (2I2,1X,I1), INPUT OPTIONS.
           NON-STANDARD INPUT UNIT FOR DATA, NON-STD. INPUT UNIT FOR LOADINGS, FLAG FOR CHECK OF AB SYMMETRY.
          AFTER DEFAULTS REPLACE UNSPECIFIED VALUES, DATA INPUT UNIT= 5, LOADINGS INPUT UNIT= 5

000 0 0   0.         0.         0.         0.         0.
          =CARD I-4 (3I1,1X,I1,1X,I1,2X,5G10.4), DATA PREPROCESSING OPTIONS AND MISSING DATA INFORMATION.
           OPTION TO CENTER AND/OR NORMALIZE MODE A, MODE B, AND/OR MODE C, FLAG TO INDICATE WHETHER OR NOT CODES ARE
           USED TO INDICATE MISSING DATA, FLAG TO INDICATE METHOD OF GETTING STARTING ESTIMATES FOR THE MISSING DATA,
           THREE MISSING DATA CODES, LOWER BOUND AND UPPER BOUND FOR VALID DATA RANGE.

0 0.
          =CARD I-5 (I1,1X,D16.0), ANALYSIS STARTING POSITION PARAMETERS.
           STARTING POSITION TYPE (0, 1 OR 2) AND SEED FOR RANDOM NUMBER GENERATOR (IF NEEDED)
           TYPE 0=RANDOM, TYPE 1=STANDARD CONTINUATION, TYPE 2=VARIABLE FORMAT INPUT OF LOADINGS.
          AFTER DEFAULT REPLACES UNSPECIFIED OR INCORRECT SEED, TYPE=0, SEED= .2198435170D+09

000 0 000    0
          =CARD I-6 (3I1,1X,I1,1X,3I1,1X,I3), ANALYSIS OPTIONS.
           FLAGS TO INDICATE DEPENDENCE CONSTRAINTS FOR MODE A, B, AND/OR C,  FLAG FOR IGNORING DATA DIAGONALS,
           FLAGS FOR HOLDING FIXED MODE A, B, AND/OR C,  INTERVAL BETWEEN CHECKS OF R.
          AFTER DEFAULTS REPLACE UNSPECIFIED VALUES, IORTHA= 1 IORTHB= 1, IORTHC= 2, IRINTV=  5

0.        0.         0.
          =CARD I-7 (3G10.4), CONVERGENCE CRITERION FOR EACH MODE.
          AFTER DEFAULTS REPLACE UNSPECIFIED VALUES, CRITERIA ARE  .1000     , .1000     , AND  .1000     PERCENT.

 7 2 0 0
          =CARD I-8 (3I2,1X,I1), OUTPUT OPTIONS.
           NON-STD. UNIT FOR WRITING COPY OF FINAL LOADINGS TO TAPE, DISK OR CARDS, UNIT (IF ANY) FOR OUTPUT OF
           SYNTHESIZED OR REVISED DATA (E.G. DATA CENTERED OR WITH MISSING VALUES ESTIMATED), UNIT (IF ANY) FOR
           OUTPUT OF RESIDUALS, FLAG TO INDICATE METHOD FOR STANDARDIZING SCALE OF OUTPUT LOADINGS.
          AFTER DEFAULTS REPLACE UNSPECIFIED VALUES, LOADINGS OUTPUT UNIT= 7, AND LOADINGS STANDARDIZATION METHOD= 3
(1X,8F7.3)
          =CARD I-8A (80A1), OUTPUT FORMAT FOR REVISED OR SYNTHESIZED DATA.



          SECTION II OF INPUT-- DATA PARAMETERS AND DATA SET
          (DATA TITLE, DIMENSIONS, FORMAT, DATA ARRAY, AND MISSING DATA SUBSCRIPTS (IF ANY))
          READING FROM UNIT  5

ERROR-FREE SYNTHETIC PROFILE DATA, 2 "TRUE" FACTORS
          =CARD II-1 (80A1), DATA SET HEADING.
```

see Section 7.4.1.2

```
   8   5   9
          =CARD II-2 (3I4), DATA SET DIMENSIONS FOR MODES A, B, AND C (NO. OF COLS, ROWS, AND SLICES).
SYNT
          =CARD II-3 (80A1), DATA FORMAT

           CODEWORD SYNT IN COL 1-4 SELECTS SYNTHETIC DATA GENERATION INSTEAD OF DATA INPUT.


   0
          =SPECIAL CARD II-4 (I4)
           NUMBER OF FACTORS TO USE IN GENERATING THE TRUE OR SYSTEMATIC PART OF THE DATA.
          AFTER DEFAULT REPLACES THE UNSPECIFIED VALUE, THE NUMBER OF FACTORS IS    2


0 0.
          =SPECIAL CARD II-5 (I1,1X,D16.0)
           TYPE OF SOURCE FOR DATA-GENERATION LOADINGS (0,1 OR 2) AND SEED FOR RANDOM NUMBER GENERATOR (IF NEEDED).
           TYPE 0=RANDOM NUMBERS, TYPE 1=STANDARD CONTINUATION DECK, TYPE 2=VARIABLE FORMAT INPUT OF LOADINGS.
          AFTER DEFAULT REPLACES UNSPECIFIED OR INCORRECT SEED, TYPE=0, SEED= .5944201730D+09


0 000 000 0 00.
          =SPECIAL CARD II-6 (I1,2(1X,3I1),2(1X,I1),G10.4), CHARACTERISTICS OF THE TRUE OR SYSTEMATIC PART OF THE DATA--
           DATA TYPE, OPTION TO DETERMINE SIGNS FOR MODES A, B AND C, FACTOR DEPENDENCE OPTION FOR MODES A, B AND C,
           OPTION TO STANDARDIZE THE ROOT MEAN SQUARE OF THE TRUE PART, SELECTION OF SIZE MULTIPLIERS FOR FACTORS,
           SIZE MULTIPLIER FOR TRUE PART OF THE DATA.
          AFTER DEFAULTS REPLACE UNSPECIFIED OR INCORRECT VALUES, THE SIGN OPTIONS FOR MODES A,B,C=112,
           THE DEPENDENCE OPTIONS FOR MODES A,B,C=111, THE DATA SIZE MULTIPLIER= 1.000


0 0.                0.        0.         0.         0.
          =SPECIAL CARD II-7 (I1,1X,D16.0,1X,4G10.4), CHARACTERISTICS OF THE RANDOMLY VARYING PARTS OF THE DATA--
           VARIATION TYPE, SEED, SIZE MULTIPLIER FOR ERROR, SIZE MULTIPLIER FOR ADDITIVE CONSTANTS, PROBABILITY THAT
           ANY GIVEN DATA POINT IS CONTAMINATED BY ERROR WITH A DIFFERENT SIZE, AND EXTRA ERROR MULTIPLIER FOR CONTAMINATED
           DATA POINTS.


          BEGINNING OF DATA SYNTHESIS
```

see Section 7.4.1.2.1

```
DATA WILL BE GENERATED FROM RANDOM LOADINGS.  THE INITIAL SEED FOR THE RANDOM NUMBER GENERATOR IS  .5944201730D+09
```

Table 2.4 (Continued)

```
SUMMARY STATISTICS FOR SYNTHETIC DATA COMPONENTS

                             MEAN       VARIANCE    MEAN SQUARE
TRUE PART OF DATA        .7454E-02     2.354        2.354
ADDITIVE CONSTANT       0.            0.           0.
RANDOM ERROR (NOT       0.            0.           0.
 INCLUDING ADDITIVE
 CONSTANT)
TOTAL ERROR             0.            0.           0.
 (INCLUDING ADDITIVE
 CONSTANT)


                      RANDOM ERROR    TOTAL ERROR
                          VS.             VS.
                       TRUE PART       TRUE PART
MEAN CROSS-PRODUCT      0.              0.
COVARIANCE              0.              0.
COSINE                  *******         *******
CORRELATION             *******         *******
```

see Section 7.4.1.2.2

```
GENERATE AND SAVE SMALL EXAMPLE DATA SET
ERROR-FREE SYNTHETIC PROFILE DATA, 2 "TRUE" FACTORS
SYNTHETIC DATA-- MSE=  .7470924E-28,STRESS= .0000000,R=1.0000000,RSQ=1.0000000
THE TRUE OR SYSTEMATIC PART OF THE DATA HAS THE FOLLOWING STRUCTURE

MODE A
          1             2
  1   -1.422        -1.652
  2    1.160         1.271
  3   -.6579          .2142
  4    1.080         1.384
  5   -.2806          .8487
  6   -.3386         -.7424
  7    1.670         -.6281
  8   -.2268          .1662

MODE B
          1             2
  1   -1.274        -.1911
  2    1.222        -.4658
  3   -.7584       -1.685
  4    .7315          .2138
  5    .8799         1.365

MODE C
          1             2
  1    1.872         .4502E-01
  2    .6582        1.100
  3    .2688         .9783
  4   2.189          .7985
  5   1.426          .8334
  6    .2107         .8974
  7    .4916         .9061
  8   1.502          .1187
  9    .3112E-01     .6121

ROOT MEAN SQUARED CONTRIBUTION FOR EACH FACTOR
       1.219         .7829
```

see Section 7.4.1.2.3

```
CROSS-PRODUCTS OF NORMALIZED FACTORS
(I.E. COSINES OF ANGLES BETWEEN FACTORS)

MODE A
      1     2
1  1.000  .513
2   .513 1.000


MODE B
      1     2
1  1.000  .462
2   .462 1.000
```

see Section 7.4.1.2.4

Table 2.4 (Continued)

```
 MODE C
      1      2
1  1.000   .563
2    .563 1.000



CORRELATIONS OF FACTOR LOADINGS

MODE A
      1      2
1  1.000   .507
2    .507 1.000


MODE B
      1      2
1  1.000   .498
2    .498 1.000


MODE C
      1      2
1  1.000 -.506
2  -.506 1.000
```

see Section 7.4.1.2.5

```
          DATA CHECK--
          THE FIRST LINE OF THE DATA IS
  3.404       -2.777       1.567      -2.586       .6617       .8136      -3.976       .5392

          THE SECOND LINE IS
 -3.217        2.626      -1.509       2.440      -.6596      -.7587       3.832      -.5221

          THE LAST LINE OF THE DATA IS
 -1.419        1.093       .1609       1.186       .7013      -.6294      -.4790       .1327


-001
          =LAST CARD IN PART II (I3), END-OF-TABLE CODE
           (MUST ALWAYS BE INCLUDED, EVEN WHEN THERE IS NO MISSING DATA SUBSCRIPT TABLE INPUT)


SUMMARY STATISTICS FOR DATA TO BE ANALYSED

TOTAL NUMBER OF POINTS IN THE DATA SET=     360
OVERALL MEAN=    .74545E-02  OVERALL VARIANCE=    2.3539
OVERALL MEAN SQUARE=    2.3539

MODE A


LEVEL        MEAN           VARIANCE        MEAN SQUARE
   1      -.42780E-01       5.8434          5.8452
   2       .43121E-01       3.7205          3.7224
   3      -.12414            .58602          .60143
   4       .18656E-01       3.6488          3.6492
   5      -.13371            .42229          .44016
   6       .27025E-01        .63226          .63299
   7       .32410           3.7604          3.8655
   8      -.52647E-01        .71887E-01      .74659E-01
MODE B
LEVEL        MEAN           VARIANCE        MEAN SQUARE
   1      -.16492           2.5411          2.5683
   2       .10936           2.0260          2.0380
   3      -.21638           3.2522          3.2990
   4       .10253            .89904          .90955
   5       .20668           2.9121          2.9549
MODE C
LEVEL        MEAN           VARIANCE        MEAN SQUARE
   1       .36135E-01       3.5432          3.5445
   2      -.50986E-02       1.9857          1.9857
   3      -.10777E-01       1.1538          1.1539
   4       .30009E-01       6.2557          6.2566
   5       .14407E-01       3.2913          3.2915
   6      -.10592E-01        .93919          .93930
   7      -.52008E-02       1.2737          1.2738
   8       .27651E-01       2.3548          2.3555
   9      -.94436E-02        .38458          .38466


THE SEED FOR THE RANDOM NUMBER GENERATOR AT END OF EXECUTION IS  .1601272600D+09
```

CHAPTER 3

PARAFAC UTILITY PROGRAMS

This chapter describes the four programs that come with PARAFAC in the PARAFAC Analysis Package. DIMS is discussed first in Section 3.1, CMPARE next in Section 3.2, then PFPLOT in 3.3 and finally, DISTIN in 3.4.

3.1 DIMS PROGRAM

The DIMS program changes array sizes in PARAFAC so that data sets of varying sizes can be analysed. Decreasing the standard dimensions (see Section 2.1 for PARAFAC limits) saves computer memory when executing PARAFAC; increasing them permits analyses of larger data sets. DIMS can also change other features of PARAFAC, such as the standard I/O units, the descriptive header at the top of the output listing, and the maximum width of tables on the listing.

To make the changes, DIMS copies the main routine of the PARAFAC Fortran source code from one file to another, revising dimension statements and/or assignment statements in the code as it does so. The modified source code must then be compiled before PARAFAC can be executed.

3.1.1 DIMS I/O Units

DIMS uses 2 logical input units and 2 logical output units that are denoted by parameter names. Default values are assigned to the parameters via assignment statements in the DIMS source code. (Users who have access to the source code can modify any of the I/O units if necessary by following the instructions given in Appendix E.) Appropriate system commands must be included with the job to link the nonstandard units (IUNITA, IUNITB) with disk files.

<u>Input</u>

1. ISTDIN=5 (standard input unit) is used for input of DIMS parameter values, described in Table 3.1.

2. IUNITA=1 (diskfile); it is used for input of the PARAFAC source code that is to be redimensioned. DIMS leaves this code intact.

Output

1. ISTDOU=6 (standard output unit) is used to document the DIMS run. The parameter input is verified and the amount of storage required by the redimensioned PARAFAC arrays is reported. This number may be useful if core memory is limited; when added to the requirements of PARAFAC compiled without its arrays (which depends on your computer and compiler), you have an estimate of how much core memory is needed for the redimensioned PARAFAC.
2. IUNITB=2 (diskfile); it is like a datafile, used for output of the revised PARAFAC source code. This new source code must be compiled, of course, before executing PARAFAC.

DIMS I/O is pictured below in Figure 3.1.

Figure 3.1. DIMS Input and Output

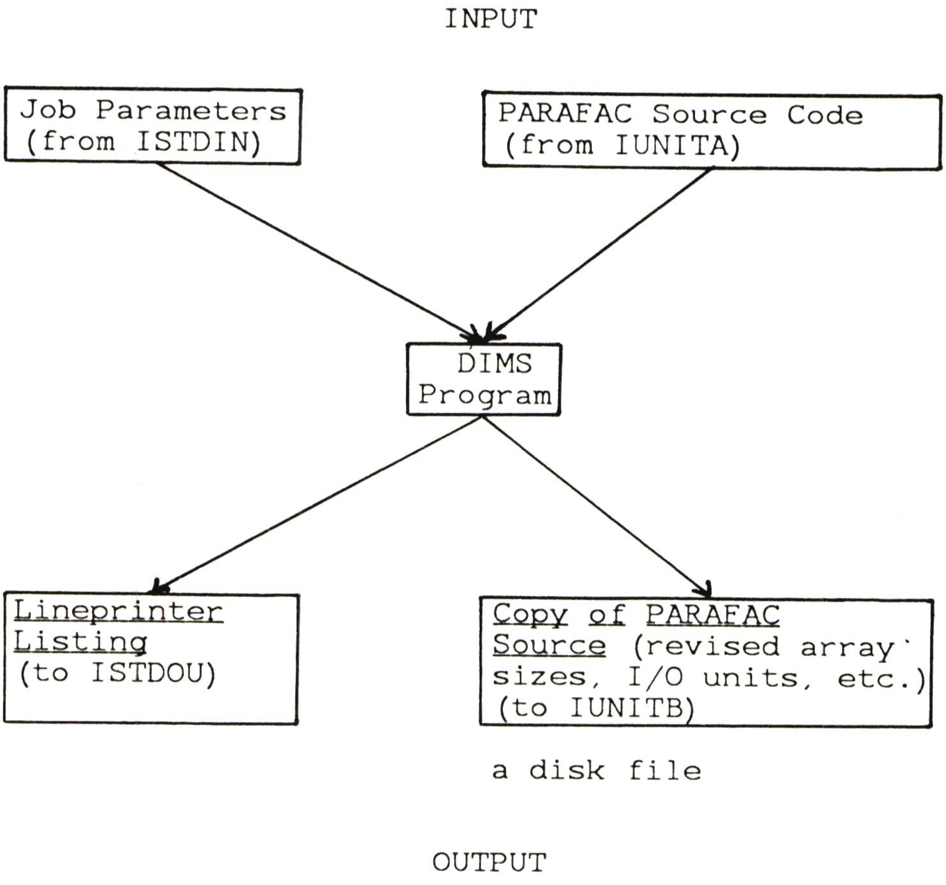

INPUT
Job Parameters (from ISTDIN)
PARAFAC Source Code (from IUNITA)
DIMS Program
Lineprinter Listing (to ISTDOU)
Copy of PARAFAC Source (revised array` sizes, I/O units, etc.) (to IUNITB)
a disk file
OUTPUT

Table 3.1

DIMS INPUT SPECIFICATIONS TABLE

CARD 1 FORMAT: (80A1) REVISION TITLE

Label identifying the researcher and/or applications for which the modified version of PARAFAC is intended, a revision date, and any other useful information. This label will be written out as part of the header at page 1 of each PARAFAC output.

CARD 2 FORMAT: (9I5) PROGRAM REVISION PARAMETERS

Columns	*Explanation*
1–5	Maximum number of levels in Mode A.
6–10	Maximum number of levels in Mode B.
11–15	Maximum number of levels in Mode C.
16–20	Maximum number of factors to be extracted.
21–25	Maximum number of missing values in the data set.
(The following parameters are optional. If not specified by the user, DIMS assigns default values.)	
26–30	Maximum number of factor loadings to be printed on one line across the PARAFAC output listing. (Default = 10; maximum = 11). The best value for this number depends on the width of the page at your printing facility. This parameter affects only text which is output in columns. See the Note below for more details.
31–35	Standard input unit in PARAFAC (Default = 5). This is the value for "ISTDIN," referred to in "PARAFAC I/O Units."
36–40	Standard output unit in PARAFAC (Default = 6). This is the value for "ISTDOU," referred to in "PARAFAC I/O Units."
41–45	Output unit for special tape, disk, or card copy of the final loadings from PARAFAC solutions, unless otherwise specified in the PARAFAC runs (Default = 7). This is the value for "ISTDLD," referred to in "PARAFAC I/O Units."

NOTE: ADJUSTING THE WIDTH OF OUTPUT TABLES TO THE PRINTING FACILITY. PARAFAC outputs simple text (i.e. nontabular information) using up to 124 characters per line. When this material is output to 80-column CRT terminals or 72-column teletypes, some of the lines will be too long to fit these output devices. Most systems will handle this situation by carrying over part of the line onto an additional line. This "carry over" does not impair the readability of simple text. However, tables arranged in columns become very hard to read when output in this way. Therefore DIMS sets a parameter ("NFCOLS") which determines the maximum number of columns per line in a PARAFAC output table. In the output of factor loading tables, each loading on a given line takes 12 columns, and 4 extra columns are needed for the line number (plus 1 for carriage control). Consequently, when the DIMS parameter NFCOLS is set to K, the required page width is 12*k + 4 columns (plus 1 for carriage control). If set to 11, only 136-column line printers will be able to print the output without "carry over" onto the next line. If set to 10, most printers should be able to handle the output, since it will only require 124 columns. If outputting to CRT terminals or 72-columns teletypes, set this parameter to 6 to avoid "carry over."

3.1.2 DIMS Examples

Two examples of DIMS parameters (input from ISTDIN) are given below. Table 3.2 is what is output to the lineprinter (ISTDOU), given the example 1 input.

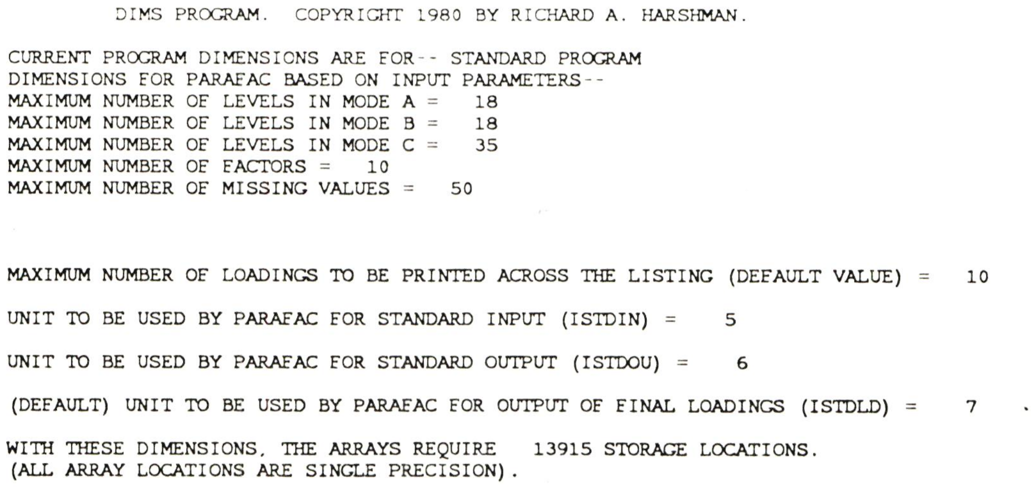
(1) The following will create a copy of the PARAFAC source code with array limits and I/O units identical to those described in Sections 2.1 and 2.2. Note that values must be specified for the parameters in columns 1-25 of Card 2, even if no changes in array sizes are to be made, while the parameters in columns 26-45 need not be specified when no changes are desired.

```
STANDARD PROGRAM
   18   18   35   10   50
```

(2) The following will make a copy of the PARAFAC source code with the following revisions: arrays can accommodate data sets up to 75x75x100 in size, with no missing values; the standard I/O units (ISTDIN and ISTDOU) are 41 and 42 respectively; and only 6 columns of factor loadings will be printed across the page (e.g., for output to 80-column CRTs or narrow paper).

```
MONTE CARLO STUDY, DECEMBER 1985
   75   75  100   10    0    6   41   42
```

Table 3.2



```
          DIMS PROGRAM.  COPYRIGHT 1980 BY RICHARD A. HARSHMAN.

CURRENT PROGRAM DIMENSIONS ARE FOR-- STANDARD PROGRAM
DIMENSIONS FOR PARAFAC BASED ON INPUT PARAMETERS--
MAXIMUM NUMBER OF LEVELS IN MODE A =    18
MAXIMUM NUMBER OF LEVELS IN MODE B =    18
MAXIMUM NUMBER OF LEVELS IN MODE C =    35
MAXIMUM NUMBER OF FACTORS =    10
MAXIMUM NUMBER OF MISSING VALUES =    50


MAXIMUM NUMBER OF LOADINGS TO BE PRINTED ACROSS THE LISTING (DEFAULT VALUE) =    10

UNIT TO BE USED BY PARAFAC FOR STANDARD INPUT (ISTDIN) =    5

UNIT TO BE USED BY PARAFAC FOR STANDARD OUTPUT (ISTDOU) =    6

(DEFAULT) UNIT TO BE USED BY PARAFAC FOR OUTPUT OF FINAL LOADINGS (ISTDLD) =    7   .

WITH THESE DIMENSIONS, THE ARRAYS REQUIRE    13915 STORAGE LOCATIONS.
(ALL ARRAY LOCATIONS ARE SINGLE PRECISION).
```

3.2 CMPARE PROGRAM

The CMPARE program is often helpful when interpreting solutions. It takes as input separate sets of factor loadings and merges them into one large set. The merged set may contain either all the factors that were input, or only some factors specified by the user. CMPARE outputs one or both of the following:

1. cross-products and intercorrelations among all the factors in the merged file
   This permits comparison of factors from different solutions. This is useful for ascertaining that a PARAFAC solution is stable from different starting positions (and thus worth interpreting), for example, or for comparing a PARAFAC solution with theoretically predicted factors, etc.

2. the merged set of loadings
   The merged set is written as one large set of loadings in standard PARAFAC format, which is suitable for input to the CMPARE program again, PFPLOT or PARAFAC. By inputting the merged set to PFPLOT, for example, you can obtain more information about the relationship between factors from different solutions (e.g., by requesting two-way plots for certain pairs of factors).

CMPARE array capacity is indicated in Section 3.2.1 and I/O units in 3.2.2. Input and output are described in Sections 3.2.3 and 3.2.4 respectively. Examples of CMPARE input and output are presented in Section 3.2.5.

3.2.1 CMPARE Limits

Array limits of the standard CMPARE code (i.e., as shipped) are as follows:

1. The maximum number of levels in any mode is 250.

2. The maximum number of factors in the merged set is 75. (The total number of factors input may be greater than 75, but the factors selected for merging and comparison may not exceed 75.)

3. The maximum number of loadings or points in any mode is 8750 (i.e., the number of levels in the mode times the number of factors in the largest loadings set ≤ 8750; the merged set will usually have the most factors, unless the input sets are very large and only a few factors are selected from them for merging).

Instructions for modifying the CMPARE array sizes are given in Appendix E, for users who are permitted access to the source code.

3.2.2 CMPARE I/O Units

CMPARE can take input from up to 25 different logical input units, although you will probably only use a few, and up to two output units. In addition, it uses one temporary file for both input and output. System commands must be included with each job to link any nonstandard units to disk files.

Input

1. ISTDIN=5 (standard input unit) is used for input of CMPARE job parameters.

2. ILDIN(1), ILDIN(2), ... , ILDIN(24) =0 (no output, default), or up to 24 units can be specified by the user if desired on Cards I-4, -4A and -4B. DO NOT SET ANY OF THESE UNITS TO 1. They are used for input of the factor loadings sets.

Output

1. ISTDOU=6 (standard output unit) is used for listing documentation and tables of cross-products and correlations.

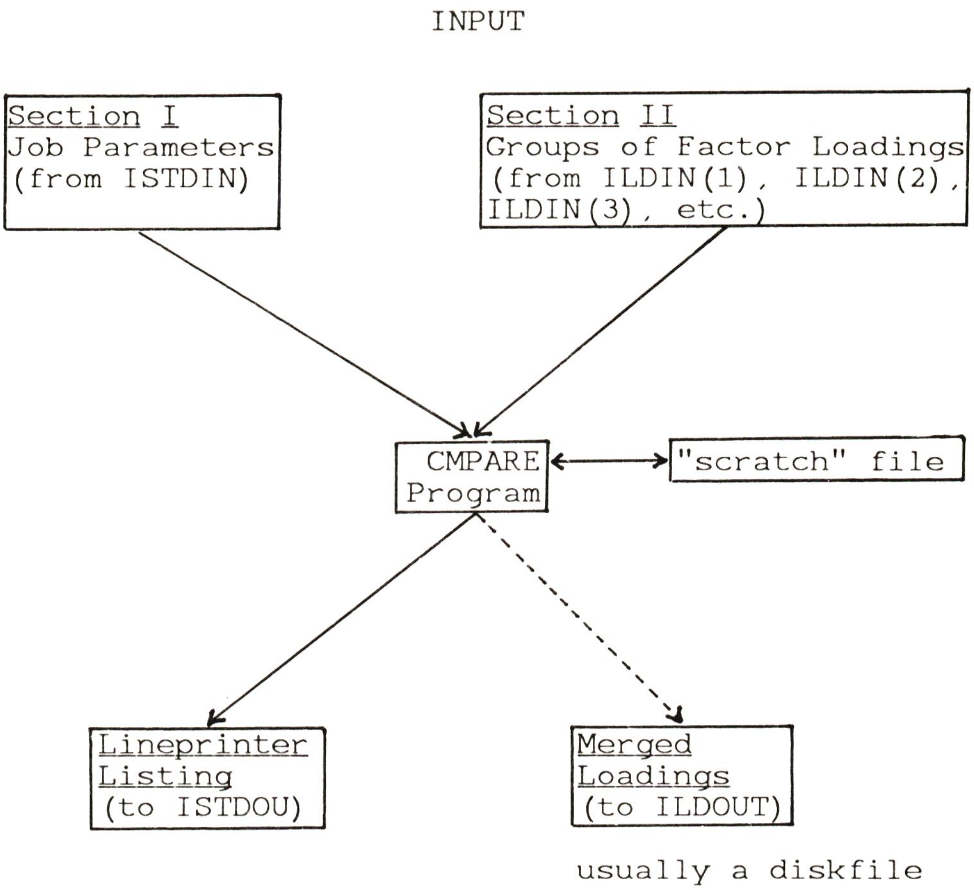
2. ILDOUT=0 (no output, default), or the user can specify a value on Card I-5. DO NOT SET ILDOUT TO 1. It is used for output of the merged set of factors.

Temporary I/O Unit

Logical unit 1 is assigned to a temporary disk file that CMPARE uses for input and output during execution. This "scratch" file is not saved, but appropriate system commands should be included if necessary to allow for I/O to logical unit 1.

The standard I/O units (5, 6) and the temporary unit (1) can be changed if necessary by users who have access to the source code (see instructions in Appendix E). CMPARE I/O. is pictured in Figure 3.2.

Figure 3.2. CMPARE Input and Output



OUTPUT

(Dotted line represents optional output.)

3.2.3 CMPARE Input

The CMPARE input consists of job parameters and loadings sets that are specified in terms of "groups" of sets. All sets within a group must have the same format (i.e., all standard PARAFAC or all variable input) and number of factors. Usually a "group" will be several solutions output from one PARAFAC run and stored in a separate file. Across different groups, the number of factors and the format may vary, but all sets must have the same number of levels in corresponding modes (but see example 3 in Section 3.2.5). Appropriate system commands must be included with the job to link logical units with disk files containing the loadings.

CMPARE input is described below in Table 3.3.

Table 3.3

CMPARE INPUT SPECIFICATIONS TABLE

SECTION I: CONTROL PARAMTERS

The first section of input contains the parameters which control input and merging of the groups of loadings sets. These parameters are always read from the standard input unit ISTDIN (usually Fortran unit 5). Integer parameters must be right justified in their respective fields.

CARD I–1 **FORMAT: (80A1)** (Cols. 1–80)

JOB TITLE. A description of the current job. The information on this card is used as the first line of the file of merged loadings, and thus serves to identify that file.

CARD I–2 **FORMAT: (3I4)** **NUMBER OF LEVELS IN EACH MODE.**

Column	*Parameter Name*	*Explanation*
1–4	NAS	Number of levels in Mode A.
5–8	NBS	Number of levels in Mode B.
9–12	NCS	Number of levels in Mode C.

CARD I–3 **FORMAT: (I3)** **NUMBER OF GROUPS OF LOADINGS SETS TO BE INPUT.** The maximum allowed is 24.

Column	*Parameter Name*	*Explanation*
1–3	NGRPS	See Note 1 at the end of Section I for more details.

Table 3.3 (Continued)

CARD I–4 FORMAT: (8(2I3,2I2)) LOADINGS GROUPS TO BE INPUT.

Column	*Parameter Name*	*Explanation*
1–3	ISOLS(1)	The number of sets of loadings (number of solutions) in the first group.
4–6	IFACT(1)	The number of factors in each set in the first group. (All sets in the group must have the same number of factors).
7–8	ILDIN(1)	The input unit for the loadings sets in the first group. It may *not* be unit 1.
9–10	IFORM(1)	The format type of every set in the first group (all the sets in one group must have the same type of format). 0 = Standard PARAFAC format 1 = Variable (nonstandard) format

Subsequent 10–column fields on this card are used to specify the characteristics of groups two to eight if needed. Characteristics are specified in the same way as shown above for the first group. See Note 2 below for more details.

CARD I–4A FORMAT: (8(2I3,2I2)) CONTINUATION OF CARD I–4. Optional. Include only if NGRPS (Card I–3) is greater than 8. Characteristics of groups 9 to 16 are specified in 10–column fields as shown for the first group on Card I–4.

CARD I–4B FORMAT: (8(2I3,2I2)) CONTINUATION OF CARD I–4A. Optional. Included only if NGRPS (Card I–3) is greater than 16. Characteristics of groups 17 to 24 are specified in 10–column fields as shown for the first group on Card I–4.

CARD I–5 FORMAT: (4I3) CHOICE OF PROCEDURE(S) AND OUTPUT PARAMETERS.

Column	*Default Value*	*Parameter Name*	*Explanation*
1–3	0	IOPT	This specifies the procedure(s) to be performed on the merged file, as follows: 0 = Compute and list cross-products and intercorrelations of factors in the merged loadings file. 1 = Write out merged loadings set to unit ILDOUT (see cols. 7–9 below). 2 = Both list cross-products and intercorrelations, and write merged loadings set to unit ILDOUT.
4–6	0	KFAC	Method of selecting factors for the merged file. 0 = All factors input are to be merged. 1 = For some sets of loadings, only user-specified factors are to be included in the merged file. A list of factors (Cards I–6 to I–Y) will be provided by the user.
7–9	·	ILDOUT	Output unit for the merged loadings. It may *not* be unit 1. Specify only if IOPT is 1 or 2.
10–12	0	INORM	Option to standardize factor loadings before output to unit ILDOUT. Except for INORM = 4, the standardization causes loadings on two modes to have a mean square loading of 1.0 for each factor, with compensatory rescaling of loadings on the other mode to reflect the scale of the data. Specify only if IOPT is 1 or 2. 0 = Do not normalize 1 = Mode A reflects the scale of the data 2 = Mode B reflects the scale of the data 3 = Mode C reflects the scale of the data 4 = Modes A and B jointly reflect the scale of the data, and Mode C has a mean square of 1.0 (normally used with data that is symmetric across Modes A and B, e.g. covariances)

Table 3.3 (Continued)

CARD I–6 to CARD I–(X) **FORMAT: (26I3)** **LIST OF FACTORS TO BE SELECTED FOR MERGING AND COMPARISON. Optional.** Include only if KFAC on Card I–5 is 1. Each card refers to a different loadings set and specifies the group number, the set number within the group, and up to 24 factors in the set that are to be selected for merging. The maximum allowed number of these cards is 47.

The following describes the parameters on Card I–6. All other cards in the list are identical.

Column	*Parameter Name*	*Explanation*
1–3	KGRP(1)	Group number. This identifies a group which contains a set from which factors are to be selected.
4–6	KSOL(1)	Set number. This identifies a loading set (within the group) from which factors are to be selected.
7–9	KFACT(1,1)	Number of the first factor in the set to be included in the merged set.
10–12	KFACT(1,2)	Number of the second factor in the set to be included in the merged set.
etc.	etc.	etc. See Note 3 below for more details.

CARD I–(Y) **FORMAT: (I3)** (Cols. 1–3) **TERMINATION CODE FOR THE LIST OF FACTORS. OPTIONAL.** Include only if KFAC on Card I–5 is 1. The character string –01 in columns 1–3 is a fixed code which specifies the end of the table.

NOTE 1: LOADINGS GROUPS. A "group" of loadings consists of one or more loadings sets; each set in the group has the same number of factors and the same format, and is input from the same unit.

NOTE 2: GROUP SPECIFICATION AND INPUT. All loading sets in one group are input before reading loadings from the next group. The values specified for ISOLS, IFACT, ILDIN and IFORM may vary for different groups. If the groups are on different disk or tape files, each group would be input from a different unit. However, the same unit may be used for input of more than one group. For example, suppose the user wishes to compare two 3-dimensional PARAFAC solutions that were punched on cards and one 3-dimensional solution (also on cards) from another source. The input would consist of two groups read from the standard input unit (usually Fortran unit 5). The first group contains two sets of loadings in PARAFAC format; the second group consists of one set in variable format. This information would be specified on Card I–4 as follows:

2 3 5 0 1 3 5 1

NOTE 3: SELECTED FACTORS. For any set *not* specified in the table, all the factors are added to the merged file. For each set specified in the table, only the factors listed are included in the merged file; these factors are added to the merged file in the order they appear on the card. If the group number and set number (KGRP and KSOL respectively) are specified, but the rest of the card is left blank, no factors from the indicated loadings set are included in the merged file.

Table 3.3 (Continued)

CMPARE INPUT SPECIFICATIONS TABLE

SECTION II: GROUPS OF LOADINGS SETS

The second section of input consists of one or more groups of loadings which may be read from various input units (except unit 1). The number of loadings sets (ISOLS), the input unit (ILDIN), and the input format (IFORM) for each group are specified on Card I–4. Any groups to be input from the standard input unit follow Card I–5 (or the terminator Card I–Y if a list of factors is included). All groups of loadings are input in the order that they are specified on Card I–4. The two possible formats of the loadings sets are given below.

a) Standard PARAFAC Format (IFORM = 0)
See PFPLOT Input Specifications Table, Section IIa for a description of the arrangement of one loadings set with this format. For a group containing more than one set, this arrangement is repeated ISOLS times.

b) Variable (Nonstandard) Format (IFORM = 1)
See PFPLOT Input Specifications Table, Section IIb for a description of the arrangement of one loadings set with this format. For a group containing more than one set, this arrangement is repeated ISOLS times.

3.2.3.1 General Uses Of CMPARE - Most of the time you will use CMPARE with PARAFAC solutions of three-way data. However, the loadings sets can also be from a source other than PARAFAC (e.g., theoretically predicted) and they can be for two modes only or one mode only. With nonstandard loadings you would use CMPARE as follows:

1. Specify 1 for NCS (and NBS) on Card I-2 if data for only two modes (one mode) are input.

2. Set IFORM to 1 on Card I-4.

3. Specify 0, 1, 2 or 4 for INORM on Card I-5 if input data have two modes (0 or 1 if one mode).

4. Input Section II contains format and data for Modes A and B only (Mode A only).

CMPARE prints a table of factor cross-products and correlations for modes with more than one level. Regardless of the loadings format on input, the merged set is always written in standard PARAFAC format, with one row of loadings for the mode(s) with one level. If the recommended values for INORM are used, then the loadings for the single-level mode(s) are all 1.0 and may be discarded. (Otherwise, the scale of the data is reflected in these loadings and so they must be retained to preserve it.)

3.2.4 CMPARE Output

CMPARE output consists of documentation and tables of cross-products and correlations, and/or the merged set of factor loadings. Everything may be listed on the standard output unit (ISTDOU), although usually the merged loadings are saved on a disk file so they can be accessed for input to PARAFAC, CMPARE or PFPLOT.

3.2.4.1 Cross-products And Correlations - Cross-products and correlations for the merged set of factors are computed in the same way as for PARAFAC and are listed just as PARAFAC outputs them (see Sections 5.2.4-5). The additional interpretations described in Chapter 6 do not apply for CMPARE output, however.

CMPARE does not preserve factor (order) numbers as they are in the input sets. Rather, it adds factors to the merged file in the order that they are read, and then numbers them consecutively from 1 as they occur in the file. Thus the (i,j) entry in each matrix is the cross-product or correlation between the ith and jth factors in the <u>merged</u> set. To make it easier to read the output tables, you may want to draw lines on the listing to separate the different solutions (as has been done in Table 3.4).

When PARAFAC solutions have been compared, subsets of the tables will be identical to those on the PARAFAC listing (i.e., where adjacent factors in the merged file belong to the same PARAFAC solution). What CMPARE shows that PARAFAC does not is the relationship between factors from <u>different</u> solutions. Two factors are identical (or very similar) if the magnitude of their cross-products and correlations is 1.0 (close to 1.0) in <u>all</u> <u>three</u> modes. Two solutions are identical (very similar) if they have identical (very similar) factors in three modes.

3.2.4.2 Merged Loadings - The merged loadings set can be listed on the lineprinter, or saved on a disk file in standard PARAFAC format. If written on disk, the user is informed by a message on the CMPARE listing. Appropriate system commands must be included with the job to save the file.

3.2.5 CMPARE Examples

This section consists of three examples of CMPARE input, followed by an example of CMPARE lineprinter output in Table 3.4.

(1) You have six 3-D PARAFAC solutions for 21x32x40 data, stored on a disk file. The PARAFAC output listing shows that four of the solutions converged and are identical. Solutions 3 and 5 seem close to convergence (R values are comparable to the converged solutions, DIFF- values are near the convergence criterion). You want to see if they are similar enough to the converged solutions to be called identical, or if they should be continued. One way to set up the CMPARE input is as follows (remember to include system control commands to access the loadings file):

```
CARD
 I-1 |COMPARISON OF 6 3-D SOLNS, TOTAL DATA
 I-2 |  21   32   40
 I-3 | 1
 I-4 | 6   3 2 0
 I-5 | 0   0   0   0
```

(2) You could make the cross-product and correlation tables for example 1 more compact (18 factors were compared there) by excluding solutions 2, 4 and 6 from the comparison (assuming visual inspection of the PARAFAC listing indicates that they are identical to solution 1). This would be done as follows (note that solution 6 can be excluded simply by not inputting it; see Card I-4):

```
CARD
 I-1 |COMPARISON OF 3-D SOLN 1 WITH SOLNS 3 AND 5
 I-2 |  21   32   40
 I-3 | 1
 I-4 | 5   3 2 0
 I-5 | 0   1   0   0
 I-6 | 1   2
 I-7 | 1   4
 I-8 |-01
```

(3) Suppose Mode C of the data in example 1 refers to people. You split the total data into two equal-sized

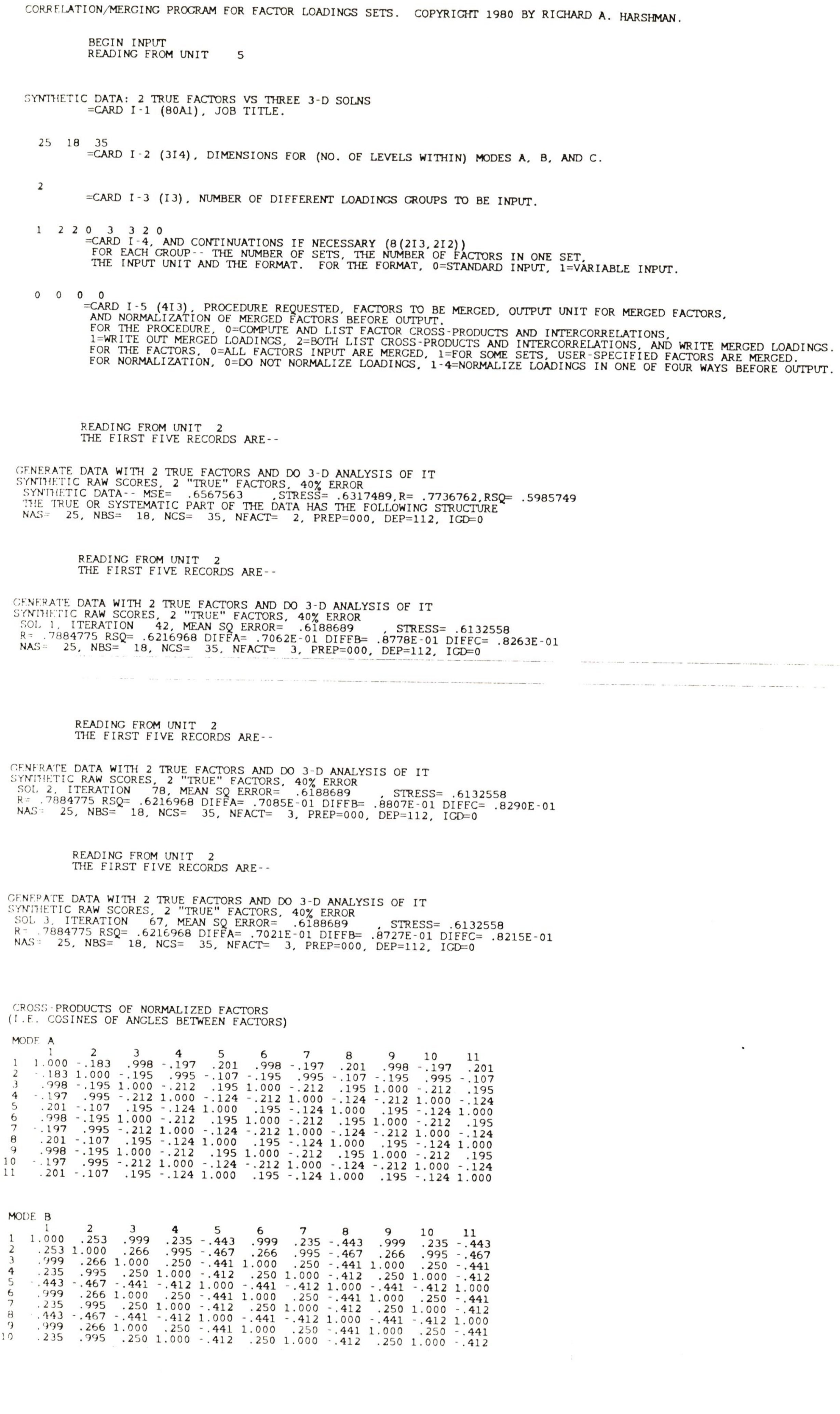
samples, randomly assigning the people to one sample or the other. You then obtain three (converged) 3-D PARAFAC solutions for each of the split halves. You could compare them with each other and with solution 1 of the total data, as follows:

```
CARD
 I-1  COMPARISON OF 3-D SOLNS-- TOTAL VS SPLIT HALF 1 AND 2
 I-2    21   32   20
 I-3    3
 I-4    1   3 2 0   3   3 3 0   3   3 4 0
 I-5    0   0   0   0
```

Note that here comparisons are meaningless in Mode C: you don't expect person loadings to be replicated for different people. However, you hope to see the Mode A and B loadings replicated across the different samples. Note also that to compare any of the other total data solutions with the split half solutions, you would first need to put them in separate files; NCS is not correct for the total data, and subsequent solutions in the same file would not be input properly.

Table 3.4 shows an example of the lineprinter output that is generated by CMPARE. The loadings sets that are compared were produced by PARAFAC, using the input shown for data synthesis example 2 (see Section 2.6). Note that although the sets are all in the same file, they must be treated as two "groups" for the CMPARE run (see Cards I-3, -4). This is because the "true" factor solution (first loadings set in the file) has 2 factors while the other three solutions have 3 factors. Lines have been drawn on the output to indicate the different solutions, so that the pertinent comparisons can be seen more easily.

Table 3.4. Example of CMPARE Lineprinter Output



```
CORRELATION/MERGING PROGRAM FOR FACTOR LOADINGS SETS.  COPYRIGHT 1980 BY RICHARD A. HARSHMAN.

         BEGIN INPUT
         READING FROM UNIT    5


SYNTHETIC DATA: 2 TRUE FACTORS VS THREE 3-D SOLNS
         =CARD I-1 (80A1), JOB TITLE.

  25  18  35
         =CARD I-2 (3I4), DIMENSIONS FOR (NO. OF LEVELS WITHIN) MODES A, B, AND C.

  2
         =CARD I-3 (I3), NUMBER OF DIFFERENT LOADINGS GROUPS TO BE INPUT.

  1  2 2 0  3  3 2 0
         =CARD I-4, AND CONTINUATIONS IF NECESSARY (8(2I3,2I2))
          FOR EACH GROUP-- THE NUMBER OF SETS, THE NUMBER OF FACTORS IN ONE SET,
          THE INPUT UNIT AND THE FORMAT.  FOR THE FORMAT, 0=STANDARD INPUT, 1=VARIABLE INPUT.

  0  0  0  0
         =CARD I-5 (4I3), PROCEDURE REQUESTED, FACTORS TO BE MERGED, OUTPUT UNIT FOR MERGED FACTORS,
          AND NORMALIZATION OF MERGED FACTORS BEFORE OUTPUT.
          FOR THE PROCEDURE, 0=COMPUTE AND LIST FACTOR CROSS-PRODUCTS AND INTERCORRELATIONS,
          1=WRITE OUT MERGED LOADINGS, 2=BOTH LIST CROSS-PRODUCTS AND INTERCORRELATIONS, AND WRITE MERGED LOADINGS.
          FOR THE FACTORS, 0=ALL FACTORS INPUT ARE MERGED, 1=FOR SOME SETS, USER-SPECIFIED FACTORS ARE MERGED.
          FOR NORMALIZATION, 0=DO NOT NORMALIZE LOADINGS, 1-4=NORMALIZE LOADINGS IN ONE OF FOUR WAYS BEFORE OUTPUT.



        READING FROM UNIT  2
        THE FIRST FIVE RECORDS ARE--

GENERATE DATA WITH 2 TRUE FACTORS AND DO 3-D ANALYSIS OF IT
SYNTHETIC RAW SCORES, 2 "TRUE" FACTORS, 40% ERROR
 SYNTHETIC DATA-- MSE=  .6567563     ,STRESS= .6317489,R= .7736762,RSQ= .5985749
 THE TRUE OR SYSTEMATIC PART OF THE DATA HAS THE FOLLOWING STRUCTURE
 NAS=  25, NBS=  18, NCS=  35, NFACT=  2, PREP=000, DEP=112, IGD=0

        READING FROM UNIT  2
        THE FIRST FIVE RECORDS ARE--

GENERATE DATA WITH 2 TRUE FACTORS AND DO 3-D ANALYSIS OF IT
SYNTHETIC RAW SCORES, 2 "TRUE" FACTORS, 40% ERROR
 SOL 1, ITERATION   42, MEAN SQ ERROR=  .6188689    , STRESS= .6132558
 R= .7884775 RSQ= .6216968 DIFFA= .7062E-01 DIFFB= .8778E-01 DIFFC= .8263E-01
 NAS=  25, NBS=  18, NCS=  35, NFACT=  3, PREP=000, DEP=112, IGD=0


        READING FROM UNIT  2
        THE FIRST FIVE RECORDS ARE--

GENERATE DATA WITH 2 TRUE FACTORS AND DO 3-D ANALYSIS OF IT
SYNTHETIC RAW SCORES, 2 "TRUE" FACTORS, 40% ERROR
 SOL 2, ITERATION   78, MEAN SQ ERROR=  .6188689    , STRESS= .6132558
 R= .7884775 RSQ= .6216968 DIFFA= .7085E-01 DIFFB= .8807E-01 DIFFC= .8290E-01
 NAS=  25, NBS=  18, NCS=  35, NFACT=  3, PREP=000, DEP=112, IGD=0

        READING FROM UNIT  2
        THE FIRST FIVE RECORDS ARE--

GENERATE DATA WITH 2 TRUE FACTORS AND DO 3-D ANALYSIS OF IT
SYNTHETIC RAW SCORES, 2 "TRUE" FACTORS, 40% ERROR
 SOL 3, ITERATION   67, MEAN SQ ERROR=  .6188689    , STRESS= .6132558
 R= .7884775 RSQ= .6216968 DIFFA= .7021E-01 DIFFB= .8727E-01 DIFFC= .8215E-01
 NAS=  25, NBS=  18, NCS=  35, NFACT=  3, PREP=000, DEP=112, IGD=0


CROSS-PRODUCTS OF NORMALIZED FACTORS
(I.E. COSINES OF ANGLES BETWEEN FACTORS)

MODE A
       1      2      3      4      5      6      7      8      9     10     11
 1  1.000  -.183   .998  -.197   .201   .998  -.197   .201   .998  -.197   .201
 2  -.183  1.000  -.195   .995  -.107  -.195   .995  -.107  -.195   .995  -.107
 3   .998  -.195  1.000  -.212   .195  1.000  -.212   .195  1.000  -.212   .195
 4  -.197   .995  -.212  1.000  -.124  -.212  1.000  -.124  -.212  1.000  -.124
 5   .201  -.107   .195  -.124  1.000   .195  -.124  1.000   .195  -.124  1.000
 6   .998  -.195  1.000  -.212   .195  1.000  -.212   .195  1.000  -.212   .195
 7  -.197   .995  -.212  1.000  -.124  -.212  1.000  -.124  -.212  1.000  -.124
 8   .201  -.107   .195  -.124  1.000   .195  -.124  1.000   .195  -.124  1.000
 9   .998  -.195  1.000  -.212   .195  1.000  -.212   .195  1.000  -.212   .195
10  -.197   .995  -.212  1.000  -.124  -.212  1.000  -.124  -.212  1.000  -.124
11   .201  -.107   .195  -.124  1.000   .195  -.124  1.000   .195  -.124  1.000


MODE B
       1      2      3      4      5      6      7      8      9     10     11
 1  1.000   .253   .999   .235  -.443   .999   .235  -.443   .999   .235  -.443
 2   .253  1.000   .266   .995  -.467   .266   .995  -.467   .266   .995  -.467
 3   .999   .266  1.000   .250  -.441  1.000   .250  -.441  1.000   .250  -.441
 4   .235   .995   .250  1.000  -.412   .250  1.000  -.412   .250  1.000  -.412
 5  -.443  -.467  -.441  -.412  1.000  -.441  -.412  1.000  -.441  -.412  1.000
 6   .999   .266  1.000   .250  -.441  1.000   .250  -.441  1.000   .250  -.441
 7   .235   .995   .250  1.000  -.412   .250  1.000  -.412   .250  1.000  -.412
 8  -.443  -.467  -.441  -.412  1.000  -.441  -.412  1.000  -.441  -.412  1.000
 9   .999   .266  1.000   .250  -.441  1.000   .250  -.441  1.000   .250  -.441
10   .235   .995   .250  1.000  -.412   .250  1.000  -.412   .250  1.000  -.412
```

Table 3.4 (Continued)

```
11   -.443 -.467 -.441 -.412 1.000 -.441 -.412 1.000 -.441 -.412 1.000


 MODE C
        1      2      3      4      5      6      7      8      9     10     11
 1  1.000   .688   .999   .713   .280   .999   .713   .280   .999   .713   .280
 2   .688  1.000   .679   .993  -.090   .679   .993  -.090   .679   .993  -.090
 3   .999   .679  1.000   .706   .303  1.000   .706   .303  1.000   .706   .303
 4   .713   .993   .706  1.000  -.055   .706  1.000  -.055   .706  1.000  -.055
 5   .280  -.090   .303  -.055  1.000   .303  -.055  1.000   .303  -.055  1.000
 6   .999   .679  1.000   .706   .303  1.000   .706   .303  1.000   .706   .303
 7   .713   .993   .706  1.000  -.055   .706  1.000  -.055   .706  1.000  -.055
 8   .280  -.090   .303  -.055  1.000   .303  -.055  1.000   .303  -.055  1.000
 9   .999   .679  1.000   .706   .303  1.000   .706   .303  1.000   .706   .303
10   .713   .993   .706  1.000  -.055   .706  1.000  -.055   .706  1.000  -.055
11   .280  -.090   .303  -.055  1.000   .303  -.055  1.000   .303  -.055  1.000



CORRELATIONS OF FACTOR LOADINGS

MODE A
        1      2 |    3      4      5 |    6      7      8 |    9     10     11   true
 1  1.000  -.184 |  .998  -.200   .165 |  .998  -.200   .165 |  .998  -.200   .165
 2  -.184  1.000 | -.196   .995  -.104 | -.196   .995  -.104 | -.196   .995  -.104  factors
 3   .998  -.196 | 1.000  -.215   .159 | 1.000  -.215   .159 | 1.000  -.215   .159
 4  -.200   .995 | -.215  1.000  -.122 | -.215  1.000  -.122 | -.215  1.000  -.122  3-D sol'n 1
 5   .165  -.104 |  .159  -.122  1.000 |  .159  -.122  1.000 |  .159  -.122  1.000
 6   .998  -.196 | 1.000  -.215   .159 | 1.000  -.215   .159 | 1.000  -.215   .159
 7  -.200   .995 | -.215  1.000  -.122 | -.215  1.000  -.122 | -.215  1.000  -.122  3-D sol'n 2
 8   .165  -.104 |  .159  -.122  1.000 |  .159  -.122  1.000 |  .159  -.122  1.000
 9   .998  -.196 | 1.000  -.215   .159 | 1.000  -.215   .159 | 1.000  -.215   .159
10  -.200   .995 | -.215  1.000  -.122 | -.215  1.000  -.122 | -.215  1.000  -.122  3-D sol'n 3
11   .165  -.104 |  .159  -.122  1.000 |  .159  -.122  1.000 |  .159  -.122  1.000

MODE B
        1      2 |    3      4      5 |    6      7      8 |    9     10     11   true
 1  1.000   .340 |  .999   .323  -.316 |  .999   .323  -.316 |  .999   .323  -.316
 2   .340  1.000 |  .357   .995  -.560 |  .357   .995  -.560 |  .357   .995  -.560  factors
 3   .999   .357 | 1.000   .342  -.312 | 1.000   .342  -.312 | 1.000   .342  -.312
 4   .323   .995 |  .342  1.000  -.503 |  .342  1.000  -.503 |  .342  1.000  -.503  3-D sol'n 1
 5  -.316  -.560 | -.312  -.503  1.000 | -.312  -.503  1.000 | -.312  -.503  1.000
 6   .999   .357 | 1.000   .342  -.312 | 1.000   .342  -.312 | 1.000   .342  -.312
 7   .323   .995 |  .342  1.000  -.503 |  .342  1.000  -.503 |  .342  1.000  -.503  3-D sol'n 2
 8  -.316  -.560 | -.312  -.503  1.000 | -.312  -.503  1.000 | -.312  -.503  1.000
 9   .999   .357 | 1.000   .342  -.312 | 1.000   .342  -.312 | 1.000   .342  -.312
10   .323   .995 |  .342  1.000  -.503 |  .342  1.000  -.503 |  .342  1.000  -.503  3-D sol'n 3
11  -.316  -.560 | -.312  -.503  1.000 | -.312  -.503  1.000 | -.312  -.503  1.000

MODE C
        1      2      3      4      5      6      7      8      9     10     11


 1  1.000  -.049 |  .996   .005   .346 |  .996   .005   .346 |  .996   .005   .346  true factors
 2   .049  1.000 | -.061   .980  -.332 | -.061   .980  -.332 | -.061   .980  -.332
 3   .996  -.061 | 1.000  -.001   .385 | 1.000  -.001   .385 | 1.000  -.001   .385
 4   .005   .980 | -.001  1.000  -.279 | -.001  1.000  -.279 | -.001  1.000  -.279  3-D sol'n 1
 5   .346  -.332 |  .385  -.279  1.000 |  .385  -.279  1.000 |  .385  -.279  1.000
 6   .996  -.061 | 1.000  -.001   .385 | 1.000  -.001   .385 | 1.000  -.001   .385
 7   .005   .980 | -.001  1.000  -.279 | -.001  1.000  -.279 | -.001  1.000  -.279  3-D sol'n 2
 8   .346  -.332 |  .385  -.279  1.000 |  .385  -.279  1.000 |  .385  -.279  1.000
 9   .996  -.061 | 1.000  -.001   .385 | 1.000  -.001   .385 | 1.000  -.001   .385
10   .005   .980 | -.001  1.000  -.279 | -.001  1.000  -.279 | -.001  1.000  -.279  3-D sol'n 3
11   .346  -.332 |  .385  -.279  1.000 |  .385  -.279  1.000 |  .385  -.279  1.000
```

3.3 PFPLOT PROGRAM

The PFPLOT program takes as input sets of numbers -- usually PARAFAC factor loadings -- and outputs plots to the standard output unit (the lineprinter). Two types of plots can be produced:

1. individual factor plots (i.e., the loadings projected onto each factor axis) and/or

2. two-factor plots (i.e., the loadings projected onto the coordinate plane defined by a pair of factor axes).

Both types of plots can be printed for the factors in one, two or all three modes of a PARAFAC solution. The plots aid in the interpretation of PARAFAC solutions, because you can immediately see which levels have the highest and lowest loadings on each factor and see the relationship between pairs of factors in a particular mode.

PFPLOT array limits are listed below in Section 3.3.1. I/O is discussed in Section 3.3.2, input is described in Section 3.3.3 and output in Section 3.3.4. Example input and output is shown in Section 3.3.5.

3.3.1 PFPLOT Limits

Array limits of the standard PFPLOT code (i.e., as shipped) are as follows:

1. The maximum number of levels in each of Modes A, B and C is 250.

2. In Modes A and B, the maximum value of the number of levels times the number of factors in the solution to be plotted is 1500 (i.e., NAS*NFACT $\leq$1500 and NBS*NFACT $\leq$1500).

3. In Mode C, the maximum value of the number of levels times the number of factors in the solution to be plotted is 3500 (i.e., NCS*NFACT $\leq$3500).

For example, a loadings set with 6 factors and 250 levels in each mode can be accommodated (250x6=1500). Or, a set with 14 factors and 100 levels in Modes A and B and 250 levels in Mode C can be accommodated (100x14=1400; 250x14=3500).

Users who are permitted access to the source code can modify these limits if necessary, by following the instructions given in Appendix E.

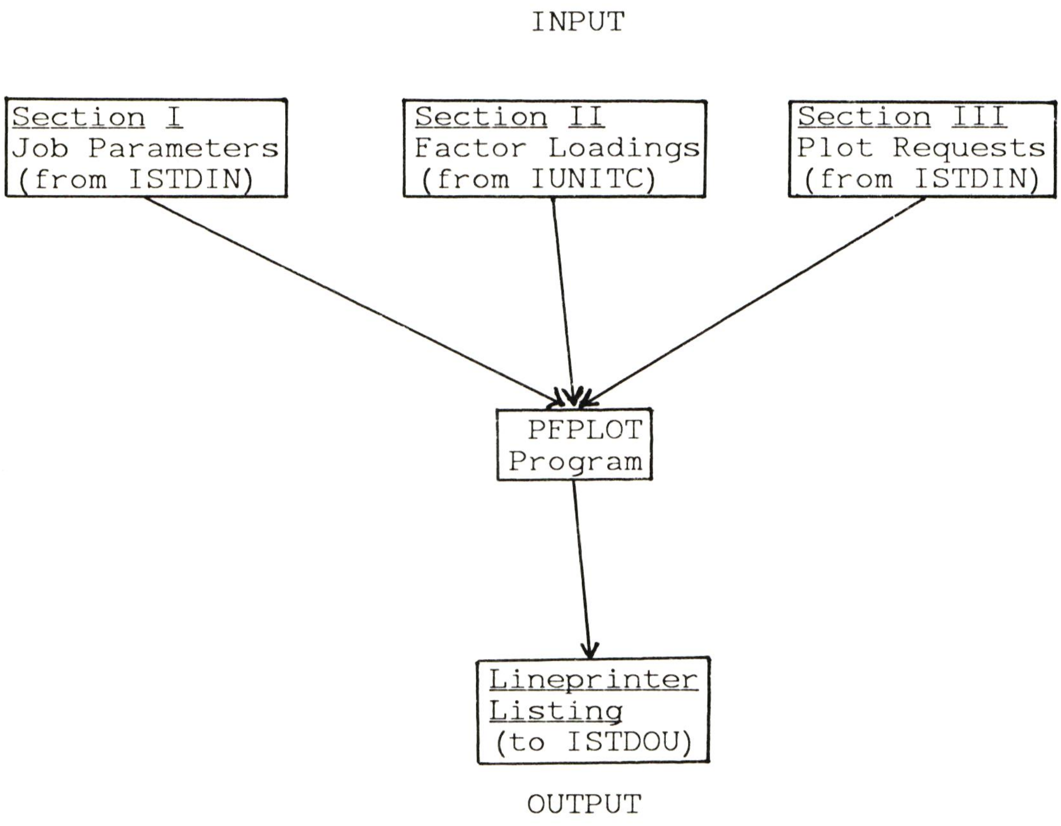
3.3.2 PFPLOT I/O Units

PFPLOT uses up to 2 logical input units and 1 logical output unit; these are denoted by integer values assigned to parameter names. The standard units for input and output are 5 and 6 respectively. They can be changed if necessary by modifying assignment statements in the source code (see Appendix E for instructions). I/O units are as follows:

1. ISTDIN=5 (standard input unit) is used for input of general plotting parameters and specific plot requests.
2. IUNITC=5 (default), or the user can specify another value on Card I-1 of the input file. It is used for input of factor loadings that are to be plotted.
3. ISTDOU=6 (standard output unit) is used for listing documentation and plots produced by PFPLOT.

PFPLOT I/O is pictured below in Figure 3.3.

Figure 3.3. PFPLOT Input and Output



3.3.3 PFPLOT Input

Input to PFPLOT consists of three general sections:

1. General plotting parameters (Input Section I) read from unit ISTDIN

2. Factor loadings set (Input Section II) read from unit IUNITC

3. Specific plot requests (Input Section III) read from ISTDIN

Sections II and III can be repeated to produce plots for more than one loadings set in the same run.

The content and format of the three input sections depends on whether "Quick" or "Detailed" input is used. Quick input requires less user-supplied information but Detailed input gives the user more control over the plots that are produced. The two types of input are described separately, Quick input in Section 3.3.3.1 and Detailed input in Section 3.3.3.2. Examples of both are given in Section 3.3.5. Section 3.3.3.3 briefly discusses how to specify input parameters for more general uses of PFPLOT.

3.3.3.1 Quick Input - Use of Quick input is restricted to PARAFAC output (or loadings in standard PARAFAC format). The plots produced are standardized, whereas Detailed input enables the user to specify certain characteristics of the plots. Quick input is more flexible that Detailed input, however, in that the loadings sets plotted in one run need not be the same size.

The following are standard features of Quick plots, but may be changed using Detailed input:

1. Plot headings specify which mode (A, B or C) is plotted. The rest of the heading is a combination of the first 32 characters from line 1 of the input loadings set, plus the first 32 characters from line 2.

2. For each mode, item labels are the digits 1-99 for levels 1-99 respectively, A0-A9 for levels 100-109, B0-B9 for levels 110-119, etc.

3. The largest and smallest loadings in a given mode determine the endpoints of the scale of the plot.

4. Single-factor plots are double-spaced.

5. The loadings are ordered by absolute value in the table of ranked loadings that accompanies single-factor plots.

6. Two-factor plots include all possible pairs of factors. (The total number of two-way plots is NFACT*(NFACT-1)/2, where NFACT is the number of factors in the loadings set.)

Quick input parameters are described below in Table 3.5.

Table 3.5

PFPLOT INPUT SPECIFICATIONS TABLE (QUICK INPUT)

SECTION I: GENERAL PARAMETERS

The first input section consists of one card only, which is read from the standard input unit ISTDIN (usually unit 5). Integer values must be right-justified in their respective yields.

CARD I–1　FORMAT: (2I3, 1X, A1)　LOADINGS FILE PARAMETERS, AND TYPE OF INPUT.

Column	*Default Value*	*Parameter Name*	*Explanation*
1–3	5 or ISTDIN	IUNITC	Loadings input unit.
4–6	0	LFORM	Format of the loadings. 0 = Standard PARADAC format (usually output from a PARAFAC analysis). LFORM must be 0 for Quick input. (See Section II below for a description of the required loadings format.) Use the Detailed input procedure for loadings sets that are in nonstandard format.
8	D	IPROC	Type of input, Q = "Quick" input, described in this table (The default D refers to "Detailed" input, which is described in the following table.)

Table 3.5 (Continued)

SECTION II: FACTOR LOADINGS SET

The second input section consists of *one* complete set of loadings, in standard PARAFAC format, which is read from IUNITC (see Card I-1 above). If IUNITC is the same unit as ISTDIN, then these leadings follow Card I-1.

The program reads the loadings set according to a standard format. Loadings output from a PARAFAC analysis are in the required form, but loadings from another source may be used if the format is identical to that described below.

Record No.	*Format*	*Explanation*
II-1 to II-4	-	Information about the loadings set: analysis and data titles, predicted fit values, etc. This information only documents the PFPLOT output, and four blank records may be substituted.
II-5	(5X,I4,6X,I4, 6X,I4,8X,I3)	Dimensions of the loadings set.
Column	*Parameter Name*	
6-9	NAS	Number of levels in Mode A
16-19	NBS	Number of levels in Mode B
26-29	NCS	Number of levels in Mode C
38-40	NFACT	Number of factors
II-6	-	"Mode A" heading. One blank record may be substituted
II-7 to — (NAS sets of records)	(5X,6G12.4)	Mode A factor loadings. Each set of records consists of the loadings on NFACT factors for one level of Mode A.
(2 records)	-	Blank record and "Mode B" heading. Two blank records may be substituted.
(NBS sets of records)	(5X,6G12.4)	Mode B factor loadings. Each set of records consists of the loadings on NFACT factors for one level of Mode B.
(2 records)	-	Blank record and "Mode C" heading. Two blank records may be substituted.
(NCS sets of records)	(5X,6G12.4)	Mode C factor loadings. Each set of records consists of the loadings on NFACT factors for one level of Mode C.

Table 3.5 (Continued)

SECTION III: SPECIFIC PLOT REQUESTS

The third section of input controls the production of plots from the loadings set (input Section II). These plot control parameters are always read from the standard input unit ISTDIN (usually unit 5). When IUNITC is the same as ISTDIN, this section follows the loadings in the input file. When the leadings are input from some other unit, these plot requests follow Card I-1 in the input file.

Card III-1 FORMAT: (4A1) PROCEDURE CODE AND PLOT REQUESTS

Column	*Default Value*	*Parameter Name*	*Explanation*
1	-	CHTEST	P = Print plots that are requested via CODEA, CODEB and CODEC values in columns 2-4
2	blank	CODEA	Plot request for Mode A
3	blank	CODEB	Plot request for Mode B
4	blank	CODEC	Plot request for Mode C

CODEA, CODEB and CODEC values are interpreted as follows:

X = No plots for the indicated mode
1 = One-way plots only for the indicated mode
2 = Two way plots only for the indicated mode
blank = Both one- and two-way plots for the indicated mode

CARD III-2 Format: (A1) CONTINUATION/TERMINATION CODE.

The letter E in column 1 indicates the end of the run. This termination code must be on the last card of the input deck.

The letter G in column 1 indicates that another set of loadings is to be read and plotted. This means that input Sections II and III, described above, must be repeated. The PFPLOT run can be continued in this way for as many times as desired (within computer time and printer limits).

3.3.3.2 Detailed Input - Detailed input can be used with either standard PARAFAC loadings or loadings in nonstandard format. If more than one loadings set is to be plotted in the same run, however, all loadings sets must be the same size and have the same format (either all standard PARAFAC or all nonstandard). If you supply labels for levels of one or more modes, use alphanumeric characters only, since some special characters (e.g., * and $) are used to indicate overlap on the plots (see PFPLOT output).

Figure 3.4 shows the Detailed input arrangement of PFPLOT. If IUNITC=ISTDIN, then there is one input file arranged exactly like the diagram (without blanks between the different input sections, of course). Usually the loadings will be in a separate diskfile (i.e., IUNITC≠ISTDIN), though, and so the Section II segments would be one after the other in another file. Table 3.6 describes the Detailed input parameters in detail.

3.3.3.3 General Uses Of PFPLOT - Usually you will use PFPLOT when interpreting PARAFAC solutions of three-way data, but you can also use it to plot other types of data. Section II of the input can also be factor loadings for two modes only or for one mode only, or it can even be raw data. If Section II is PARAFAC output (e.g., from analysis of two-way data), then you can assume standard PARAFAC format and use Quick input format to specify the plot requests. Otherwise, you must use Detailed input as follows:

1. Cards I-1 thru I-8 are all included as usual, but 1 is specified for NCS (and NBS) if data for only two modes (one mode) is input.

2. Input Section II includes format and data for Modes A and B only (Mode A only).

If raw score data is input, PFPLOT can be used to generate scatterplots that may reveal outliers, etc. Each (two-way) matrix is treated by PFPLOT as "factor loadings" for one "mode". For example, if the data consist of scores on the 11 subscales of the WAIS IQ test for 50 people (where each row is an individual's scores), then you would specify NAS=50 and NFACT=11. You could then get scatterplots for scores on one subscale vs another by requesting two-factor plots for specific pairs of factors (use IOPTXY=2; 60 plots would be generated if IOPTXY=1).

Figure 3.4

PFPLOT Deck Setup for Detailed Input

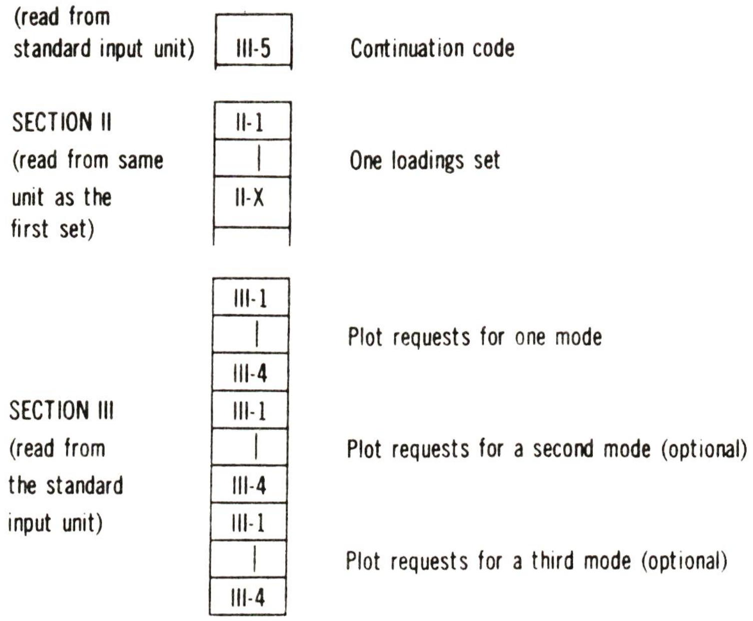
The diagram below illustrates the arrangement of the "Detailed" input parameters for a PFPLOT run. The parameters are described in detail in the table following the diagram. If the unit specified for loadings input is the same as the standard input unit, then the input deck for a PFPLOT run is exactly like the diagram. However, it will generally be more convenient to read them from a separate disk or tape file (where they would be after a PARAFAC run). Then the Section II segments would be deleted from the input deck; these segments would reside on another file, one after the other.

	I-1	Loadings file parameters	
SECTION I	I-2	Number of levels in Mode A	
General	I-3	Labels for levels of Mode A	
Parameters	I-4	Number of levels in Mode B	
(read from	I-5	Labels for levels of Mode B	
standard	I-6	Number of levels in Mode C	
input unit)	I-7	Labels for levels of Mode C	
	I-8	Number of factors	
SECTION II			
Factor Loadings Set	II-2		
(read from disk	|	One loadings set	
on tape file,	II-X		
or from standard			
input unit)			
	III-1	General plot parameters	
SECTION III	III-2	One-way plot parameters	Plot requests
Specific	III-3	Two-way plot parameters	for one mode
Plot	III-4	Factor pairs to be plotted (optional)	
Requests	III-1		
(read from	|	Plot requests for a second mode (optional)	
standard	III-4		
input unit)	III-1		
	|	Plot requests for a third mode (optional)	
	III-4		
SECTION II			
and SECTION III		Input for continuation of the run (optional)	
Repeated		See below for details of this block.	
(read from			
standard input	III-5	Termination code	
unit)			

Figure 3.4 (Continued)

PFPLOT Continuation (How Sections II and II Can Be Repeated)

See diagram above for the entire deck.


(read from standard input unit)
III-5
Continuation code
SECTION II (read from same unit as the first set)
II-1
II-X
One loadings set
SECTION III (read from the standard input unit)
III-1
III-4
Plot requests for one mode
III-1
III-4
Plot requests for a second mode (optional)
III-1
III-4
Plot requests for a third mode (optional)

This arrangement may be repeated as many times as desired.

Table 3.6

PFPLOT INPUT SPECIFICATIONS TABLE (Detailed input)

SECTION I: GENERAL PARAMETERS

This first input section specifies the dimensions of the loadings set to be plotted and the labels to be used on the plots. These general parameters are always read from the standard input unit ISTDIN (usually unit 5). Integer values specified by the user must be right-justified in their respective fields.

CARD I–1 **FORMAT: (2I3,1X,A1)** **LOADINGS FILE PARAMETERS, AND TYPE OF INPUT.**

Column	*Default Value*	*Parameter Name*	*Explanation*
1–3	5 or ISTDIN	IUNITC	Loadings input unit.
4–6	0	LFORM	Format of the loadings. 0 = Standard PARAFAC format (usually output from a PARAFAC analysis) 1 = Variable (nonstandard) format
8	D	IPROC	Type of input. D = "Detailed" input, described in this table

---

CARD I–2 **FORMAT: (I4)** **NUMBER OF LEVELS IN MODE A.** The maximum allowed is 250.

Column	*Parameter Name*
1–4	NAS

CARD I–3 **FORMAT: (40A2)** (Cols. 1–80) **LABELS FOR LEVELS OF MODE A.** Each label occupies two columns; at least one of these two columns must be nonblank. Only letters and integers should be used (i.e. no special characters). If this card is left blank, default labels will be assigned. Default labels are the integers 1, 2, 3, etc.

CARD(S) I–3A (I–3B, I–3C, ..., I–3F) **FORMAT: (40A2)** (Cols. 1-80) **CONTINUATION(S) OF CARD I–3. Optional.** Include *only* if required (e.g. I–3A is included only if NAS is greater than 40; I–3B follows I–3A only if NAS is greater than 80; etc.). A maximum of 6 continuation cards is allowed.

CARD I–4 **FORMAT: (I4)** **NUMBER OF LEVELS IN MODE B.** The maximum allowed is 250.

Column	*Parameter Name*
1–4	NBS

Table 3.6 (Continued)

CARD I–5 **FORMAT: (40A2)** (Cols. 1–80) **LABELS FOR LEVELS OF MODE B.** Each label occupies two columns; at least one of these two columns must be nonblank. Only letters and integers should be used (i.e. no special characters). If this card is left blank, default labels will be assigned. Default labels are the integers 1, 2, 3, etc.

CARD(S) I–5A (I–5B, I–5C, ..., I–5F) **FORMAT: (40A2)** (Cols. 1–80) **CONTINUATION(S) OF CARD I–5. Optional.** Include *only* if required (e.g. I–5A is included only if NBS is greater than 40; I-5B follows I-5A only if NBS is greater than 80 etc.) A maximum of 6 continuation cards is allowed.

CARD I–6 **FORMAT: (I4)** **NUMBER OF LEVELS IN MODE C.** The maximum allowed is 250.

Column	*Parameter Name*
1–4	NCS

CARD I–7 **FORMAT: (40A2)** (Cols. 1–80) **LABELS FOR LEVELS OF MODE C.** Each label occupies two columns; at least one of these two columns must be nonblank. Only letters and integers should be used (i.e. no special characters). If this card is left blank, default labels will be assigned. Default labels are the integers 1, 2, 3, etc.

CARD(S) I–7A (I–7B, I–7C, ..., I–7F) **FORMAT: (40A2)** (Cols. 1–80) **CONTINUATION(S) OF CARD I–7. Optional.** Include *only* if required (e.g. I–7A is included only if NCS is greater than 40; I–7B follows I–7A only if NCS is greater than 80; etc.). A maximum of 6 continuation cards is allowed.

CARD I–8 **FORMAT: (I3)** **NUMBER OF FACTORS.**

Column	*Parameter Name*
1–3	NFACT

Table 3.6 (Continued)

SECTION II: FACTOR LOADINGS SET

The second input section consists of *one* complete set of loadings which is read from IUNITC (see Card I–1). If IUNITC is the same unit as ISTDIN, then these loadings follow Card I–8. Since the form of this section of input depends on whether LFORM (see Card I–1) is 0 or 1, it will be described separately for the two cases.

(a) Standard PARAFAC Format (LFORM=0)

The loadings set is read according to a standard format specified by the program. Generally, it is output from a previous PARAFAC analysis. However, loadings from another source may be used if the format is identical to that described in the table below.

Record No.	*Format*	*Explanation*
II–1 to II–6	–	Information about the loadings set: analysis and data titles, predicted data fit values, Mode A heading, etc. This information is used only to document the PFPLOT output, and blank records can be substituted. However, there must always be 6 records preceding the loadings.
II–7 to – **(NAS sets of records)**	**(5X,6G12.4)**	Mode A factor loadings. Each of the NAS sets of records consists of the loadings on NFACT factors for one level of Mode A.
(2 records)	–	Blank record and "Mode B" heading. Two blank records may be substituted.
(NBS sets of records)	**(5X,6G12.4)**	Mode B factor loadings. Each of the NBS sets of records consists of the loadings on NFACT factors for one level of Mode B.
(2 records)	–	Blank record and "Mode C" heading. Two blank records may be substituted.
(NCS sets of records)	**(5X,6G12.4)**	Mode C factor loadings. Each of the NCS sets of records consists of the loadings on NFACT factors for one level of Mode C.

(b) Variable (Nonstandard) Format (LFORM=1)

This type of format generally occurs when the loadings are from a source other than PARAFAC. The input format for the loadings in each mode must be specified individually. These formats are always read from the same unit as the loadings.

Record No.	*Format*	*Parameter Name*	*Explanation*
II–1	**(80A1)** **(Cols. 1–80)**	**FORMTA**	Format for input of Mode A loadings. It must be enclosed in parentheses, and must specify F, E or G format. (Integers should be read using Fn.0 format, e.g. F2.0 for two-column integers.)
II–2– **(NAS sets of records)**	**FORMTA**	–	Mode A factor loadings. Each of the NAS sets of records consists of the loadings on NFACT factors for one level of Mode A.
(1 record)	**(80A1)** **(Cols. 1–80)**	**FORMTB**	Format for input of Mode B loadings. It must be enclosed in parentheses, and must specify F, E or G format. (Integers should be read using Fn.0 format, e.g. F2.0 for two-column integers.)
(NBS set of records)	**FORMTB**	–	Mode B factor loadings. Each of the NBS sets of records consists of the loadings on NFACT factors for one level of Mode B.
(1 record)	**(80A1)** **(Cols. 1–80)**	**FORMTC**	Format for input of Mode C loadings. It must be enclosed in parentheses, and must specify F, E or G format. (Integers should be read using Fn.0 format, e.g. F2.0 for two-column integers.)
(NCS sets of records)	**FORMTC**	–	Mode C factor loadings. Each of the NCS sets of records consists of the loadings on NFACT factors for one level of Mode C.

Table 3.6 (Continued)

SECTION III: SPECIFIC PLOT REQUESTS

This third section of input controls the production of plots from the loadings set (input section II). These plot control parameters are always read from the standard input unit ISTDIN (usually unit 5). When IUNITC is the same as ISTDIN (i.e. the loadings are read from the standard input unit), this section follows the loadings in the input file. When the loadings are input from some other unit, these plot requests follow Card I-8 in the input file. Integer values specified by the user must be right-justified in their respective fields.

Card III-1 FORMAT: (A1,1X,2G10.4) GENERAL PLOT PARAMETERS.

Column	*Default Value*	*Parameter Name*	*Explanation*

1	—	CHTEST	Mode to be plotted.
			A = Mode A
			B = Mode B
			C = Mode C

3-12	see Note 1	SMAXP	Maximum value to be plotted.
13-22	see Note 1	SMAXN	Minimum value to be plotted. SMAXP and SMAXN apply to all plots printed for CHTEST. See Note 1 below for more details.

CARD III-2 FORMAT: (I1,2X,I1,72A1) ONE-WAY (Y) PLOT PARAMETERS.

Column	*Default Value*	*Parameter Name*	*Explanation*
1	0	IOPTY	Selection of one-way plots (i.e. to print single factor plots for CHTEST and table of ranked loadings) 0 = No plots 1 = Print plots, and rank loadings by algebraic value in the table 2 = Print plots, and rank loadings by absolute value in the table
4	0	ISPACE	Spacing of plot output. (Not necessary if IOPTY = 0) 0 = Single space 1 = Double space
5-76	-	HEADY	Title for the plot output. The default is a blank line. (Not necessary if IOPTY = 0)

CARD III-3 FORMAT: (I1,1X,I2,72A1) TWO-WAY (XY) PLOT PARAMETERS.

Column	*Default Value*	*Parameter Name*	*Explanation*
1	0	IOPTXY	Selection of two-way plots (i.e. to print two-factor plots for mode CHTEST). 0 = No plots 1 = Print plots for all possible pairs of factors 2 = Print plots for only the factor pairs specified by the user
3-4	0	N2FAC	Number of factor *pairs* to be input by the user. *Optional.* Specify only if IOPTXY = 2. The maximum allowed for N2FAC is 30.
5-76	—	HEADXY	Title for the two-way plots. The default is a blank line. (Not necessary if IOPTXY = 0)

Table 3.6 (Continued)

CARD III-4 **FORMAT: (20I4)** **FACTOR PAIRS TO BE PLOTTED: Optional.** Include *only* if N2FAC has a value greater than zero (and IOPTXY = 2). See Note 2 below for more details.

CARD III-4A **FORMAT: (20I4)** **CONTINUATION OF CARD III-4: Optional.** Include *only* if N2FAC has a value greater than 10. The eleventh factor pair to be plotted is specified by the numbers in cols. 1-4 and 5-8 of this card, etc.

CARD III-4B **FORMAT: (20I4)** **CONTINUATION OF CARD III-4A: Optional.** Include *only* if N2FAC has a value greater than 20.

Cards III-1 to III-4 determine the plots for one mode only, as specified to CHTEST. For each of the other modes that you want plotted, repeat card sequence III-1 to III-4. (The parameters do not have to be specified identically for the different modes.)
To terminate these plot requests, insert Card III-5.

CARD III-5 **FORMAT: (A1)** **CONTINUATION/TERMINATION CODE.** Any letter other than A, B, C or G in column 1 indicates the end of the run. This termination code must be on the last card of the input deck. The letter G in column 1 indicates that another set of loadings is to be read and plotted. See Note 3 below for more details.

NOTE 1: PLOT LIMITS. Defaults for SMAXP and SMAXN are the values of the largest and smallest loadings in the selected mode. The appropriate default value will override the user-specified value for SMAXP and/or SMAXN if the user-specified range is too small, i.e., if some loadings would otherwise be excluded from the plot. The value of SMAXP used by the program will either be the user specified value, or the value of the largest loading, whichever is greater. Similarly, SMAXN will be the user specified value, or the value of the lowest loading, whichever is less.

The ability to specify SMAXP and SMAXN explicitly is useful for making plots share the same scale across different solutions and/or modes. If a minimum scale of exactly zero is desired, specify a very small number (e.g. 10.E-30) for SMAXN. Blank or zero cannot be specified directly, since it is replaced by the default.

NOTE 2: USER-SPECIFIED FACTOR PAIRS. On Card III-4, the first pair is specified by the (factor) numbers in cols. 1-4 and 5-8; numbers for the second pair are in cols. 9-12 and 13-16; etc. The first number in each pair represents the factor that will be plotted on the ordinate (y-) axis and the second number is the factor that will be plotted on the abscissa (x-) axis. Ten pairs of factors can be specified on Card III-4 altogether. If N2FAC is greater than 10, continue the factor pair specification on Card III-4A (and Card III-4B if necessary).

NOTE 3: CONTINUATION. To continue the PFPLOT run, input sections II (Factor Loadings Set) and III (Specific Plot Requests) described above must be repeated. Section I (General Parameters) is not repeated; values specified for the first loadings set apply to the second one as well. Thus, successive sets must be the same size and have the same general format (see LFORM on Card I-1) as the first set.

The second loadings set and then the specific plot request follow Card III-5 in the input file if IUNITC (Card I-1) is the same as ISTDIN). If IUNITC is some other unit, the second loadings set follows the first one on that file, while the specific plot requests follow Card III-5 in the input file. Another Card III-5 must be included at the end of the second set of plot requests to specify either termination of the run, or continuation of it with a third set of loadings and plot requests. Each continuation would follow the same format as described above.

There is no limit to the number of continuations allowed in a PFPLOT run (except for computer time and printer limits). Remember that all loadings sets used must be the same size and the same format type, and that the final record in the input file (i.e. after the last set of plot requests) must be a Card III-5 containing a termination code.

3.3.4 PFPLOT Output

The output consists of documentation and plots, both of which are printed on the standard output unit (ISTDOU). The documentation allows a check of the input, and provides a list of the labels used in the plots. An example of PFPLOT output is shown in Table 3.7, and some comments about the plots are also made below.

3.3.4.1 One-way Plots - The one-way plot consists of item labels printed in columns under the headings D1 (dimension or factor 1), D2 (dimension or factor 2), etc. The vertical positions of the labels in each column represent the pattern of factor loadings for the indicated factor, plotted against the scale values that are printed down both sides of the page. The plot is printed on either one page (single-spaced) or two pages (double-spaced); the spacing does not affect the scale values. (Double-spaced plots are useful if you need to make a lot of notations on the plot.)

The scale values are not usually the actual magnitude of the loadings, but are the loadings raised or lowered by a power of 10. This does not affect the relative positions of the plot labels. It does, however, minimize the number of characters needed for the scale labels, which is especially useful when printing the two-factor plots.

For each factor, loadings with the same value (when rounded off by the program) are indicated by labels printed side-by-side in the appropriate column, up to a maximum of three. When more than three have the same value, a special character is printed instead of the first label. The special characters are defined at the top of the plot: + means 2 overlapping points, ++ means 3 overlapping points, * means 4, ** means 5, $ means 6, $$ means 7, and () means more than 7. An "Overlap Table" below the plot provides information about the overlapping loadings.

You may have to add notations when interpreting the plots, since the labels printed on the plot have only two characters at most. Labels that appear at both ends of a bipolar dimension and at the nonzero end of a unipolar dimension indicate the levels that influence the factor most; use them when determining the factor interpretation. You will need to inspect plots for the corresponding factor in all modes before you can assign an overall interpretation to the factor. For an example of factor interpretation, see Harshman and DeSarbo (1984). One solution discussed by them is plotted in Table 3.7.

3.3.4.2 Two-way Plots - Each two-way plot shows the relationship between the two factors indicated at the top of the plot; the one on the vertical axis is Y and the one on the horizontal axis is X. The vertical and horizontal scales are identical and are the same as the scale for the one-way plots in the same mode. This consistency facilitates comparisons across different plots. The "STEP=" value at the top of the plot (ignore the decimal point) is the difference in value between adjacent points on the plot (i.e., it is the step size of the scale). The overlap of two or more labels at the same point on the plot is indicated by the same special characters as for one-way plots, and an Overlap Table gives more information about the overlapping points.

Each two-way plot can be thought to illustrate one plane in the multidimensional space spanned by the solution. While one-way plots are especially helpful when interpreting a solution, two-way plots can sometimes reveal additional useful information about it because they show how pairs of factors are related. A linear trend along either diagonal of the plot indicates highly correlated factors; scattered points reveal that the factors are independent. Nonlinear relationships due to factor interactions may show up; they might suggest the need for a nonlinear factor model for the data. Finally, interesting clusters of points might sometimes be exposed.

3.3.5 PFPLOT Examples

Examples 1 and 2 below show Quick input, examples 3-6, Detailed input. Table 3.7 is an example of PFPLOT output.

(1) Suppose you have a file containing several PARAFAC solutions that are identical, and so you want complete plots for only the first one. The PFPLOT Quick input parameters would be (remember to include system commands to access the loadings file):

```
  CARD
   I-1    1  0 Q
 III-1 P
 III-2 E
```

(2) Suppose you have a file containing 3 PARAFAC solutions. You don't want plots for the first solution (it didn't converge), but you want one- and two- factor plots for Modes A, B and C of the second solution, and one-way plots for Mode B of the third solution. The Quick input parameters would be as follows:

```
 CARD
  I-1 |   1  0 Q
III-1 |PXXX
III-2 |G
III-1 |P
III-2 |G
III-1 |PX1X
III-2 |E
```

In fact, the card containing PXXX can be omitted (i.e., if you want to skip a solution without plotting anything, the continuation code G is sufficient).

(3) Detailed input parameters to produce plots like those for example 1 are as follows (assuming a 2-factor solution for a data array that is 41x35x90):

```
 CARD
  I-1 |   1  0 D
  I-2 |   41
  I-3 |
 I-3A |
  I-4 |   35
  I-5 |
  I-6 |   90
  I-7 |
 I-7A |
 I-7B |
  I-8 |   2
III-1 |A
III-2 |2  1 MODE A  2-D SOLN 1 OF EXAMPLE DATA
III-3 |1    MODE A  2-D SOLN 1 OF EXAMPLE DATA
III-1 |B
III-2 |2  1 MODE B  2-D SOLN 1 OF EXAMPLE DATA
III-3 |1    MODE B  2-D SOLN 1 OF EXAMPLE DATA
III-1 |C
III-2 |2  1 MODE C  2-D SOLN 1 OF EXAMPLE DATA
III-3 |1    MODE C  2-D SOLN 1 OF EXAMPLE DATA
III-5 |E
```

(4) Suppose the factor loadings in example 3 above were not in standard PARAFAC format. The only change in the plotting parameters would occur for Card I-1, which instead would be:

```
   1  1 D
```

(5) The following example illustrates how to specify plot characteristics via Detailed input parameters. Suppose you have a file containing six 5-factor PARAFAC solutions for 41x35x90 data. Inspection (CMPARE runs, visual checks) has revealed 2 different solutions: solutions 1, 2, 3 and 5 are identical, and 4 and 6 are identical. You thus want to plot

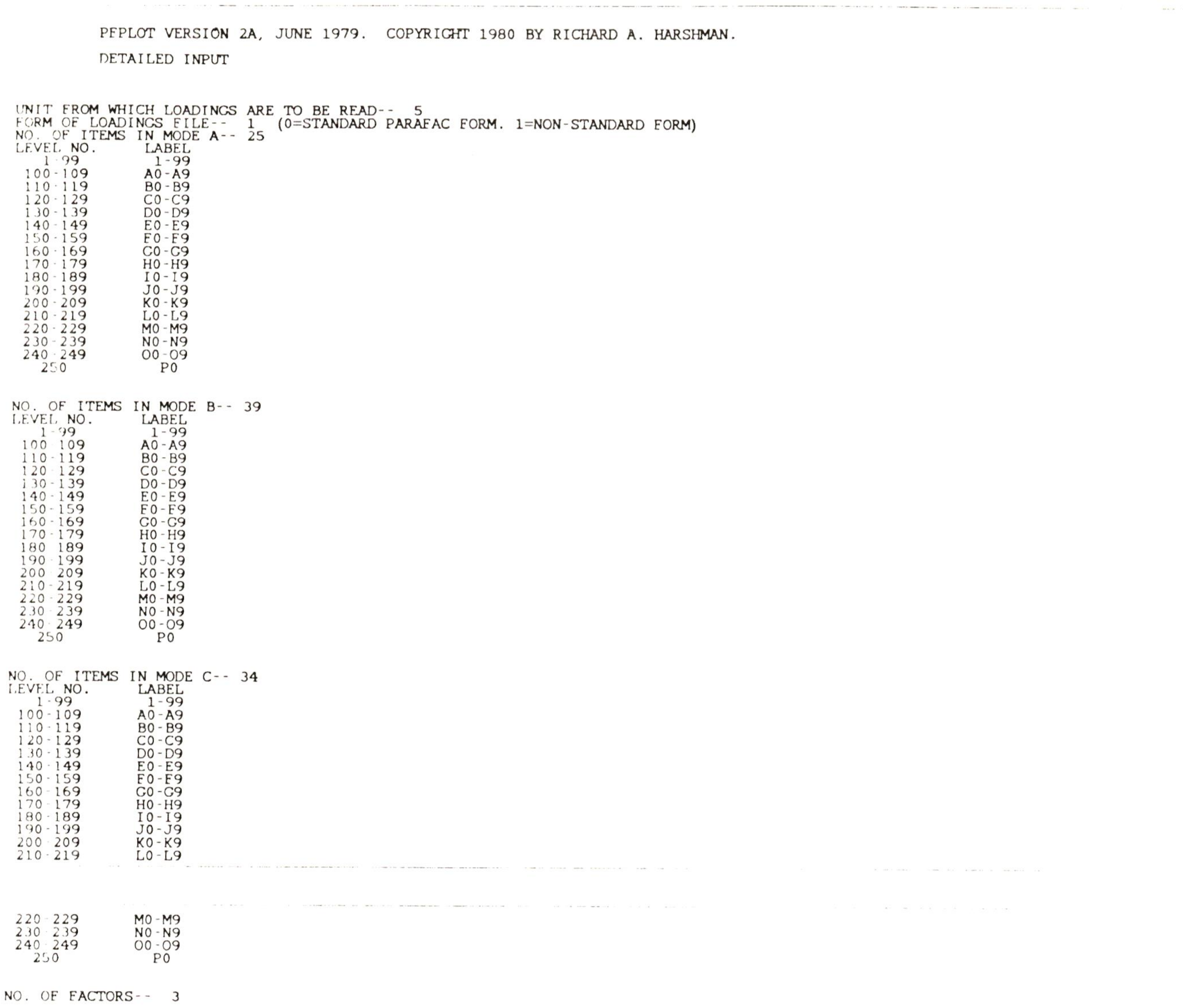
solutions 1 and 4. You want one-way plots of Modes A and B (Mode C refers to subjects whom you don't have any information about, and so you will use Modes A and B to interpret the factors) and two-factor plots of factor 1 vs all the other factors (with factor 1 on the horizontal axis). To make the plots of the two solutions comparable, you want to specify the endpoints of the scales. The plotting parameters would be as follows:

```
 CARD
  I-1  }
   :   }  same as example 3
 I-7B  }
  I-8    5
III-1  A 3.0        -3.0
III-2  1  0 MODE A   5-D SOLN 1
III-3  2  4 MODE A   5-D SOLN 1
III-4     2   1   3   1   4   1   5   1
III-1  B 3.0        -3.0
III-2  1  0 MODE B   5-D SOLN 1
III-3  2  4 MODE B   5-D SOLN 1
III-4     2   1   3   1   4   1   5   1
III-5  G
III-5  G
III-1  A 3.0        -3.0
III-2  1  0 MODE A   5-D SOLN 4
III-3  2  4 MODE A   5-D SOLN 4
III-4     2   1   3   1   4   1   5   1
III-1  B 3.0        -3.0
III-2  1  0 MODE B   5-D SOLN 4
III-3  2  4 MODE B   5-D SOLN 4
III-4     2   1   3   1   4   1   5   1
III-5  E
```

(6) Suppose you have a 3-D PARAFAC solution for 41x35x90 data and suppose Mode C refers to subjects. Your sample consists of 45 males and 45 females, and the data file is organized with data for all the females first, then data for the males. To check for a sex difference on the factors, you could assign labels for the levels of Mode C (F=female, M=male) and request plots for that mode, as follows:

```
 CARD
  I-1  }
   :   }  same as example 3
  I-6  }
  I-7   F F F F F F F F F F ................F F F F F F F F
 I-7A   F F F F F M M M M M ................M M M M M M M M
 I-7B   M M M M M M M M M M
  I-8    3
III-1  C
III-2  1  1 MODE C   3-D SOLN (F=FEMALE, M=MALE)
III-3  1    MODE C   3-D SOLN (F=FEMALE, M=MALE)
III-5  E
```

Table 3.7. Example of PFPLOT Lineprinter Output

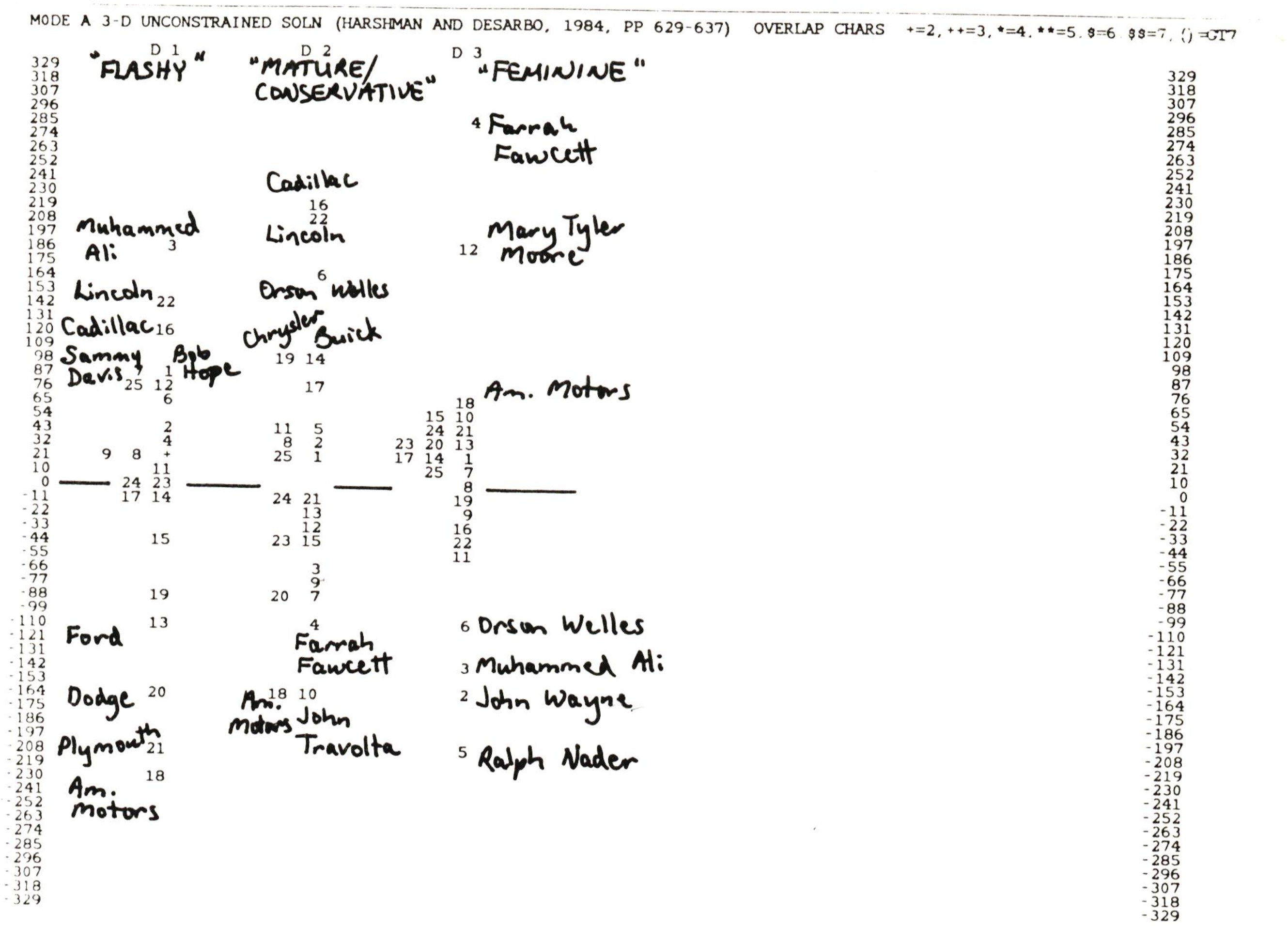


```
          PFPLOT VERSION 2A, JUNE 1979.  COPYRIGHT 1980 BY RICHARD A. HARSHMAN.
          DETAILED INPUT

UNIT FROM WHICH LOADINGS ARE TO BE READ--  5
FORM OF LOADINGS FILE--  1  (0=STANDARD PARAFAC FORM. 1=NON-STANDARD FORM)
NO. OF ITEMS IN MODE A-- 25
LEVEL NO.       LABEL
   1-99          1-99
 100-109        A0-A9
 110-119        B0-B9
 120-129        C0-C9
 130-139        D0-D9
 140-149        E0-E9
 150-159        F0-F9
 160-169        G0-G9
 170-179        H0-H9
 180-189        I0-I9
 190-199        J0-J9
 200-209        K0-K9
 210-219        L0-L9
 220-229        M0-M9
 230-239        N0-N9
 240-249        O0-O9
   250            P0

NO. OF ITEMS IN MODE B--  39
LEVEL NO.       LABEL
   1-99          1-99
 100-109        A0-A9
 110-119        B0-B9
 120-129        C0-C9
 130-139        D0-D9
 140-149        E0-E9
 150-159        F0-F9
 160-169        G0-G9
 170-179        H0-H9
 180-189        I0-I9
 190-199        J0-J9
 200-209        K0-K9
 210-219        L0-L9
 220-229        M0-M9
 230-239        N0-N9
 240-249        O0-O9
   250            P0

NO. OF ITEMS IN MODE C--  34
LEVEL NO.       LABEL
   1-99          1-99
 100-109        A0-A9
 110-119        B0-B9
 120-129        C0-C9
 130-139        D0-D9
 140-149        E0-E9
 150-159        F0-F9
 160-169        G0-G9
 170-179        H0-H9
 180-189        I0-I9
 190-199        J0-J9
 200-209        K0-K9
 210-219        L0-L9

 220-229        M0-M9
 230-239        N0-N9
 240-249        O0-O9
   250            P0

NO. OF FACTORS--  3
```

Table 3.7 (Continued)

```
          SOLUTION  1

 INPUT FORMAT FOR MODE A LOADINGS--
 (3F6.2)

 INPUT FORMAT FOR MODE B LOADINGS--
 (3F6.2)

 INPUT FORMAT FOR MODE C LOADINGS--
 (3F5.2)

 SELECT MODE  (A=MODE A,  B=MODE B,  C=MODE C)-- A

 YPLOT SELECTED=1  (0=NO. 1=YES--RANK LOADINGS BY ALGEBRAIC VALUE.
 2=YES--RANK LOADINGS BY ABSOLUTE VALUE)
 SPACING=0  (0=SINGLE. 1=DOUBLE)

 XYPLOT SELECTED=0  (0=NO. 1=ALL 2-FACTOR PLOTS. 2=USER-SPECIFIED 2-FACTOR PLOTS)
 NUMBER OF USER-SPECIFIED FACTOR PAIRS= 0

 USER-SPECIFIED LIMITS FOR THE PLOTS--MAXIMUM=   3.000     AND MINIMUM= -3.000
 LIMITS USED BY THE PROGRAM--MAXIMUM=   3.000     AND MINIMUM= -3.000


             TABLE OF RANKED LOADINGS

 FACTOR    1
         RANK       ITEM NO.  ITEM LABEL  ITEM VALUE
           1           3          3        1.770
           2          22         22        1.370
           3          16         16        1.180
           4           1          1        .7800
           5           7          7        .7700
           6          25         25        .7400
           7          12         12        .7200
           8           6          6        .6000
           9           2          2        .3700
          10           4          4        .2400
          11           5          5        .2100
          12           8          8        .1500
          13          10         10        .1500
          14           9          9        .1400
          15          11         11        .4000E-01
          16          23         23       -.3000E-01
          17          24         24       -.3000E-01
          18          14         14       -.1800
          19          17         17       -.2100
          20          15         15       -.5300
          21          19         19       -.9300
          22          13         13       -1.180
          23          20         20       -1.700
          24          21         21       -2.160
          25          18         18       -2.310
 FACTOR    2
         RANK       ITEM NO.  ITEM LABEL  ITEM VALUE
           1          16         16        2.150
           2          22         22        2.080
           3           6          6        1.580
           4          19         19        .9600
           5          14         14        .9000
           6          17         17        .6800
           7          11         11        .3700

           8           5          5        .3500
           9           8          8        .2800
          10           2          2        .2700
          11          25         25        .1700
          12           1          1        .1200
          13          21         21       -.2100
          14          24         24       -.2100
          15          13         13       -.2600
          16          12         12       -.3300
          17          15         15       -.4700
          18          23         23       -.4800
          19           3          3       -.7200
          20           9          9       -.8400
          21          20         20       -.9100
          22           7          7       -.9600
          23           4          4       -1.160
          24          18         18       -1.650
          25          10         10       -1.710
 FACTOR    3
         RANK       ITEM NO.  ITEM LABEL  ITEM VALUE
           1           4          4        2.840
           2          12         12        1.770
           3          18         18        .6200
           4          10         10        .4900
           5          15         15        .4500
           6          21         21        .3900
           7          24         24        .3700
           8          23         23        .2800
           9          13         13        .2500
          10          20         20        .2400
          11          14         14        .2200
          12           1          1        .1600
          13          17         17        .1300
          14           7          7        .9000E-01
          15          25         25        .9000E-01
          16           8          8       -.1000E-01
          17          19         19       -.1100
          18           9          9       -.2700
          19          16         16       -.3500
          20          22         22       -.4700
          21          11         11       -.6100
          22           6          6       -1.200
          23           3          3       -1.510
          24           2          2       -1.710
          25           5          5       -2.190
```

Table 3.7 (Continued)



```
OVERLAP TABLE
SCALE PT.      FACTOR NO.   ITEM NO.   ITEM LABEL    ITEM VALUE
    21              1           5           5           .2100
    21              1          10          10           .1500
```

Table 3.7 (Continued)

```
SELECT MODE (A=MODE A, B=MODE B, C=MODE C)-- B

YPLOT SELECTED=1 (0=NO. 1=YES--RANK LOADINGS BY ALGEBRAIC VALUE.
2=YES--RANK LOADINGS BY ABSOLUTE VALUE)
SPACING=0 (0=SINGLE. 1=DOUBLE)

XYPLOT SELECTED=0 (0=NO. 1=ALL 2-FACTOR PLOTS. 2=USER-SPECIFIED 2-FACTOR PLOTS)
NUMBER OF USER-SPECIFIED FACTOR PAIRS= 0

USER-SPECIFIED LIMITS FOR THE PLOTS--MAXIMUM=  3.000     AND MINIMUM= -3.000
LIMITS USED BY THE PROGRAM--MAXIMUM=  3.000     AND MINIMUM= -3.000


               TABLE OF RANKED LOADINGS

FACTOR    1
          RANK        ITEM NO.   ITEM LABEL   ITEM VALUE
            1           20          20          1.360
            2           34          34          1.330
            3           24          24          1.280
            4           39          39          1.250
            5            4           4          1.240
            6            7           7          1.140
            7           16          16          1.120
            8           10          10          1.070
            9           37          37          1.060
           10           18          18          1.030
           11           15          15           .7900
           12            9           9           .7600
           13            1           1           .7000
           14           21          21           .7000
           15            2           2           .6700
           16           29          29           .5800
           17           35          35           .2500
           18           27          27           .2400
           19           30          30           .1600
           20           36          36           .3000E-01
           21           38          38           .1000E-01
           22           22          22          -.3000E-01
           23            3           3          -.4000E-01
           24           19          19          -.1300
           25           12          12          -.2700
           26           17          17          -.4200
           27           13          13          -.6200
           28           26          26          -.8700
           29           11          11          -.8900
           30           32          32          -.9000
           31           33          33          -.9800
           32           14          14         -1.060
           33           28          28         -1.310
           34           25          25         -1.420
           35            5           5         -1.430
           36           23          23         -1.490
           37           31          31         -1.560
           38            8           8         -1.570
           39            6           6         -1.810
FACTOR    2
          RANK        ITEM NO.   ITEM LABEL   ITEM VALUE
            1            3           3          1.910
            2           13          13          1.790
            3           38          38          1.360

            4            9           9          1.330
            5           14          14          1.320
            6           32          32          1.120
            7           29          29          1.100
            8           33          33          1.080
            9           23          23           .8600
           10           21          21           .7700
           11            1           1           .7000
           12           24          24           .6000
           13            2           2           .5500
           14           27          27           .4400
           15           39          39           .4000
           16            5           5           .3800
           17           10          10           .2700
           18           34          34           .2300
           19            6           6           .2200
           20           16          16           .1800
           21           15          15          -.4000E-01
           22           31          31          -.1000
           23            7           7          -.1100
           24           37          37          -.1900
           25           22          22          -.3800
           26           25          25          -.6500
           27            8           8          -.6600
           28           19          19          -.8300
           29           18          18          -.8400
           30           35          35          -.8400
           31           28          28         -1.040
           32           11          11         -1.110
           33           12          12         -1.200
           34            4           4         -1.220
           35           30          30         -1.300
           36           17          17         -1.390
           37           26          26         -1.460
           38           20          20         -1.620
           39           36          36         -1.660
FACTOR    3
          RANK        ITEM NO.   ITEM LABEL   ITEM VALUE
            1           29          29          2.030
            2            1           1          1.790
            3           39          39          1.740
            4           34          34          1.520
            5           10          10          1.270
            6           36          36          1.100
            7           33          33           .9200
            8            5           5           .8400
            9           16          16           .8100
           10           27          27           .8100
           11           12          12           .6300
           12            3           3           .3800
           13            6           6           .3500
           14           23          23           .3500
           15            9           9           .3100
           16           15          15           .3000
           17           26          26           .3000
           18           17          17           .2100
           19           31          31           .2100
           20           11          11           .1900
           21           37          37          -.1000E-01
           22           24          24          -.7000E-01
           23           30          30          -.3900
           24           13          13          -.4900
           25           35          35          -.5000
           26            4           4          -.5300
           27           32          32          -.5500
```

Table 3.7 (Continued)

28	8	8	-.5700
29	14	14	-.5800
30	38	38	-.6500
31	21	21	-.9700
32	28	28	-.9900
33	20	20	-1.030
34	7	7	-1.050
35	2	2	-1.180
36	25	25	-1.190
37	18	18	-1.290
38	19	19	-1.910
39	22	22	-2.140

```
MODE B 3-D UNCONSTRAINED SOLN (HARSHMAN AND DESARBO, 1984, PP 629-637)  OVERLAP CHARS  +=2,++=3,*=4,**=5,$=6,$$=7,()=GT7
                 D 1              D 2              D 3
 329   "FLASHY"          "MATURE/                                      329
 318                                       "FEMININE"                  318
 307                   CONSERVATIVE"                                   307
 296                                                                   296
 285                                                                   285
 274                                                                   274
 263                                                                   263
 252                                                                   252
 241                                                                   241
 230                                                                   230
 219                                                                   219
 208                            Formal       29 Smooth                 208
 197                          3                                        197
 186      Superior           13 Mature        1 Pleasant               186
 175          Graceful                       39 Attractive             175
 164   Attractive               Sophisticated                          164
 153                                         34 Graceful               153
 142        34 20 Active      38  9                                    142
 131     39 24  4 Dynamic        14          10                        131
 120        16  7                32                                    120
 109     37 18 10             33 29          36                        109
  98                                         33                        98
  87           15             23 21    27 16  5                        87
  76      9  2  +                 1                                    76
  65           29                24          12                        65
  54                              2                                    54
  43                       39 27  5    23  6  3                        43
  32        35 27             34 10    26 15  9                        32
  21           30             16  6    31 17 11                        21
  10        38 36                                                      10
   0  ____  22  3  ______     31 15  ____  37 24  ________              0
 -11           19             37  7                                   -11
 -22           12                                                     -22
 -33           17                22          30                       -33
 -44                                   35 13  4                       -44
 -55           13                25    32 14  +                       -55
 -66                              8                                   -66
 -77           26          35 19 18                                   -77
 -88     33 32 11                            21                       -88
 -99           14                28    28 20  7                       -99
-110                          12 11       25  2                      -110
-121           28             30  4          18 Aggressive           -121
-131           25                17 Careless                         -131
-142  Slow  23  5 Usual          26 Simple                           -142
-153        31  8                20                                  -153
-164  Obscure  Plain             36 Active                           -164
-175            6                   Light                            -175
-186                                         19 Hard                 -186
-197           Colorless                                             -197
-208                                         22 Masculine            -208
-219                                                                 -219
-230                                                                 -230
-241                                                                 -241
-252                                                                 -252
-263                                                                 -263
-274                                                                 -274
-285                                                                 -285
-296                                                                 -296
-307                                                                 -307
-318                                                                 -318
-329                                                                 -329
```

Table 3.7 (Continued)

```
OVERLAP TABLE
SCALE PT.    _FACTOR NO.    ITEM NO.    ITEM LABEL     ITEM VALUE
    76            1             1            1            .7000
    76            1            21           21            .7000
   -55            3             8            8          -.5700
   -55            3            38           38          -.6500
```

Note that while -3.0 and 3.0 are specified as the minimum and maximum respectively for the above plots, the scale values range from -329 to 329 instead of from -300 to 300. This is a result of how PFPLOT computes the scale step size, and is not an error.

The solution plotted above, taken from Harshman and DeSarbo (1984, pp. 628-30), is an example of how to deal with the complexities of interpretation that may arise when ratings on bipolar scales are analysed. Mode A refers to various celebrities and automobiles, Mode B to bipolar rating scales, and Mode C to subjects. Compared to the values presented by Harshman and DeSarbo, the sign of the Mode A factor 1 loadings have been reversed for these plots, as have the Mode B loadings for factors 2 and 3. (This is equivalent to reversing the sign of all the data values and then analysing the reversed data.) With this reversal, labels from the "low" instead of the "high" end of the rating scales are assigned to the levels of Mode B. Taken together, these labels make certain aspects of the solution more salient than do the opposite labels with the unreversed data. Harshman and DeSarbo preferred to focus on these aspects in their interpretation.

They assigned factor labels based on the scale labels at the upper end of the factor. The reversal of factor 1 in Mode A rather than in Mode B causes the "flashy" labels to be at the high end in Mode B; hence, the label "Flashy". This was done because they preferred to discuss the solution in terms of positive aspects (e.g., flashiness) rather than negative ones (e.g., dullness, lack of colour).

In general, interpreting a solution obtained from bipolar ratings data may pose problems that are not encountered with other types of data. Sometimes the interpretation may seem unduly complicated and difficult to discuss until you reverse the loadings for each factor in one mode and focus on the opposite scale labels, as was done in the above example.

3.4 DISTIN PROGRAM

The main purpose of the DISTIN program is to transform similarity or dissimilarity data to scalar products, so that PARAFAC can then be used to do a multidimensional scaling analysis (see Section 4.6). The input data is usually a three-way array (a set of two-way matrices), or it can be a two-way array (one matrix only). For each Mode A-by-Mode B matrix (i.e., for each Mode C or frontal slice of the array; see Section 4.2), the following transformations are done:

1. Symmetrization, either by computing the mean of the corresponding off-diagonal entries (i.e., (x(i,j) + x(j,i))/2.) or by Shepard's method (1972) (i.e., (x(i,j)+x(j,i))/(x(i,i)+x(j,j)); DO NOT USE IF DIAGONALS EQUAL ZERO)

2. Similarities changed to dissimilarities (i.e., sign of all data values reversed or reciprocal values computed)

3. Dissimilarities changed to distance measures (i.e., additive constant estimated for the matrix and then data values adjusted by this constant)

4. Distances transformed to scalar products (i.e., each value squared, then the matrix is row- and column-centered)

The output data need not be scalar products. Each of the above transformation steps is a separate option in DISTIN and so any combination of one or more can be requested. You could just symmetrize the data or just compute reciprocals, for example, to make the data appropriate for whatever you want to do.

DISTIN array sizes are listed below in Section 3.4.1 and I/O units in Section 3.4.2. Input is discussed in Section 3.4.3 and output in 3.4.4. Examples of input and output are given in Section 3.4.5.

3.4.1 DISTIN Limits

Array limits of the standard DISTIN code (i.e., as shipped) are as follows:

1. The maximum number of levels in each of Modes A and B is 40.

2. The maximum number of levels in Mode C is 250.

Users with access to the source code can change these limits if necessary. Instructions for doing so are given in

Appendix E.

3.4.2 DISTIN I/O Units

DISTIN uses up to 3 different logical input units and up to 3 different logical output units. They are denoted by integer values assigned to parameter names. The standard units for input and output are 5 and 6 respectively. System commands must be included with each job to link any nonstandard units with disk files. The I/O units are as follows:

Input

1. ISTDIN=5 (standard input unit) is used for input of DISTIN processing options.

2. IUNITB=5 (default), or the user can specify a value on Card I-1. It is used for input of the data array.

3. IUNIT3 = IUNITB (default), or the user can specify a value on Card I-3. It is used for input of the data parameters.

Output

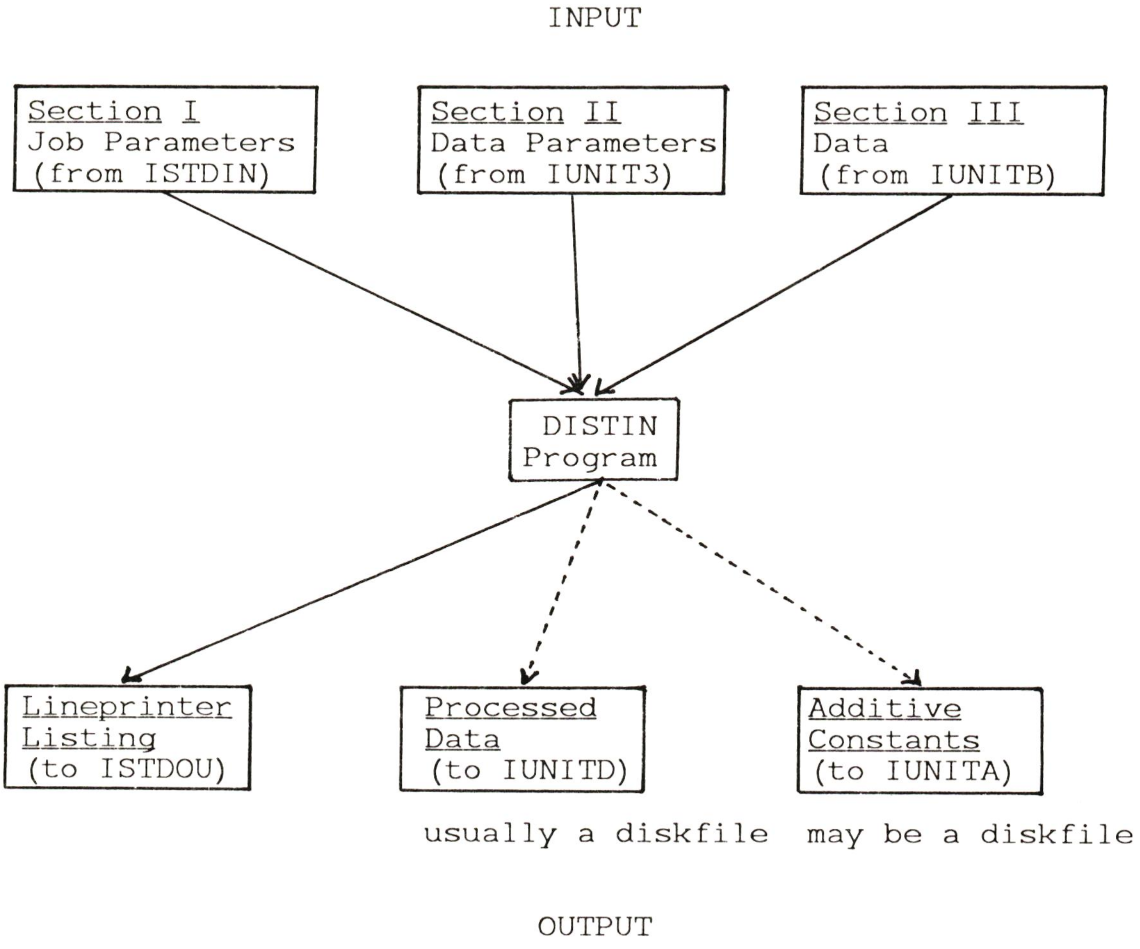
1. ISTDOU=6 (standard output unit) is used for listing documentation of the DISTIN run.

2. IUNITD=7 (disk file, default), or the user can specify a value on Card I-2. DO NOT SET EQUAL TO IUNITB. It is used for output of the transformed data. This output can be suppressed by specifying -1 on Card I-2.

3. IUNITA=0 (no output, default), or the user can specify a value on Card I-3. It is used for output of the additive constants.

The standard I/O units (5, 6) can be changed if necessary; see Appendix E for instructions. Figure 3.5 illustrates DISTIN I/O.

Figure 3.5. DISTIN Input and Output



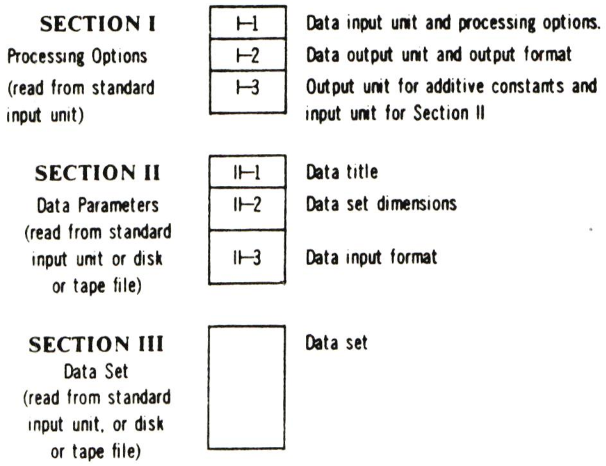
(Dotted lines represent output that is optional or that can be suppressed.)

3.4.3 DISTIN Input

DISTIN input is described below. If both IUNITB and IUNIT3 are set to 5, then the information is arranged on one file as shown in Figure 3.6. A detailed description of the input is given in Table 3.8 and examples are to be found in Section 3.4.5.

Figure 3.6

DISTIN Input Deck



3.4.4 DISTIN Output

Output generated by a DISTIN run includes documentation, the transformed data and a list of the estimated additive constants (if applicable). The documentation, listed on the lineprinter, allows a check of the input and shows the step-by-step transformation of the data for the first and last matrices only (i.e., the first and last frontal slices in the three-way data array). The additive constants are output either to a disk file or to the lineprinter, as requested by the user.

After all the requested transformation processes have been applied, the transformed data set is output in a format compatible with that required by PARAFAC for data input (i.e., Input Section II -- see Table 2.1 for a description; also see example 25 in Chapter 2). Usually these data are written to a disk file. It is recommended that you not set IUNITD equal to the standard output unit, because the actual data listing and the step-by-step documentation for the last matrix may be confused; also, the transformed data would not be saved for subsequent analysis.

Table 3.8

DISTIN INPUT SPECIFICATIONS TABLE

SECTION I: PROCESSING OPTIONS

This first section of input contains parameters which specify the I/O units to be used, and the transformation procedures to be performed on the data. This section is always read from the standard input unit ISTDIN (usually Fortran unit 5).

CARD I–1 FORMAT: (5I3) DATA INPUT UNIT AND PROCESSING OPTIONS.

Column	*Default Value*	*Parameter Name*	*Explanation*
1–3	5 or ISTDIN	IUNITB	Data input unit.
4–6	0	ISYM	Symmetrize data on Modes A and B. (For symmetrization NAS must equal NBS). See Card II-2 below). 0 = No 1 = Use means 2 = Use method described by Shepard (David and Denes, 1972).
7–9	0	IPRE	Preprocess data values 0 = No 1 = Reverse sign 2 = Take reciprocal
10–12	0	IESTC	Estimate additive constant for each level of Mode C (applicable only if input data set contains measures of similarities or dissimilarities) 0 = No 1 = Yes
13–15	0	ISCALP	Transform data to scalar products (normally appropriate only if the data set contains measures of similarities or dissimilarities) 0 = No 1 = Yes

CARD I–2 FORMAT: (I3,1X,76A1) DATA OUTPUT UNIT AND OUTPUT FORMAT.

Column	*Default Value*	*Parameter Name*	*Explanation*
1–3	7 or ISTDTD	IUNITD	Data output unit. To suppress this output, specify –1 for IUNITD.
5–80	(1X,7G11.4)	DATFMT	Output format for one row of the data (i.e. it should provide for writing all levels of Mode A at a fixed level of Modes B and C; it may be a multi-record format). The format must be enclosed in parentheses, and must specify F,E or G format.

CARD I–3 FORMAT: (2I3) OUTPUT UNIT FOR ADDITIVE CONSTANTS AND INPUT UNIT FOR SECTION II.

Column	*Default Value*	*Parameter Name*	*Explanation*
1–3	·	IUNITA	Output unit for additive constants (Optional). The default is to not output the constants.
4–6	= IUNITB	IUNIT3	Input unit for Records II–1, II–2, and II–3.

Table 3.8 (Continued)

DISTIN INPUT SPECIFICATIONS TABLE

SECTION II: DATA PARAMETERS

This second section of input specifies the title, dimensions and input format for the data set. It is input from IUNIT3 (specified on Card I-3). Most often the value for IUNIT3 will be either the same as ISTDIN (usually Fortran unit 5) or the same as IUNITB. In the former case, Section II would follow Section I in the input deck; in the latter, Section II would precede the data set (Section III) in the file on IUNITB.

RECORD II-1 **FORMAT: (80A1)** **DATA TITLE.** A general description of the data set that identifies it and distinguishes it from any other versions of the data. This information is output as the first record of the transformed data set.

RECORD II-2 **FORMAT: (3I4)** **DATA SET DIMENSIONS.** The data set is NAS by NBS by NCS points. NAS gives the number of items per row of the data matrix (even if this involves more than one Fortran record). NBS gives the number of rows per "slice" or matrix (i.e. per level of Mode C). NCS gives the number of "slices" or matrices assembled into the three-way data set.

Column	*Parameter Name*	*Explanation*
1-4	NAS	Number of levels or items in Mode A.
5-8	NBS	Number of levels or items in Mode B.
9-12	NCS	Number of levels or items in Mode C.

RECORD II-3 **FORMAT: (80A1) (Cols. 1-80)** **"VARFMT": DATA INPUT FORMAT.** Format for reading one row of the data matrix (i.e. it should provide for reading all the levels of Mode A at a fixed level of Modes B and C; it may be a multi-record format). The format must be enclosed in parentheses, and must specify F, E or G format. To input data stored as integers, use F format with zero places to the right of the decimal point (eg. use F2.0 to read two-column integer data).

SECTION III: DATA SET

The third section of input is the data set that is to be transformed and/or written in a format suitable for input to the PARAFAC program. It is input from IUNITB (specified on Card I-1). The data set contains NAS by NBS by NCS data values altogether. There are NCS blocks of records. Every block contains NBS sets of records, and each set is one row of the data which is read according to VARFMT (specified on Record II-3) (i.e. the three-way data set consists of NCS matrices; each matrix is NBS rows by NAS columns). In Fortran, the data points are input in the following way:

```
DO 10 K=1,NCS
DO 10 J=1,NBS
10 READ (IUNITB, VARFMT) (DIS(I,J,K), I=1,NAS)
```

} in DISTIN source; do not input with data

3.4.5 DISTIN Examples

Two examples of DISTIN input are given below. Table 3.9 is an example of lineprinter output.

(1) Suppose you had used PARAFAC to generate 12x12x20 dissimilarities data with error due to additive constants (see Section 2.6, example 3). You now want to convert them to scalar products. In this case, you do not need to symmetrize the data because it is already symmetric (no random error was added to destroy the symmetry). The input to DISTIN would be as follows (include system control cards to access the data file and to save the scalar products):

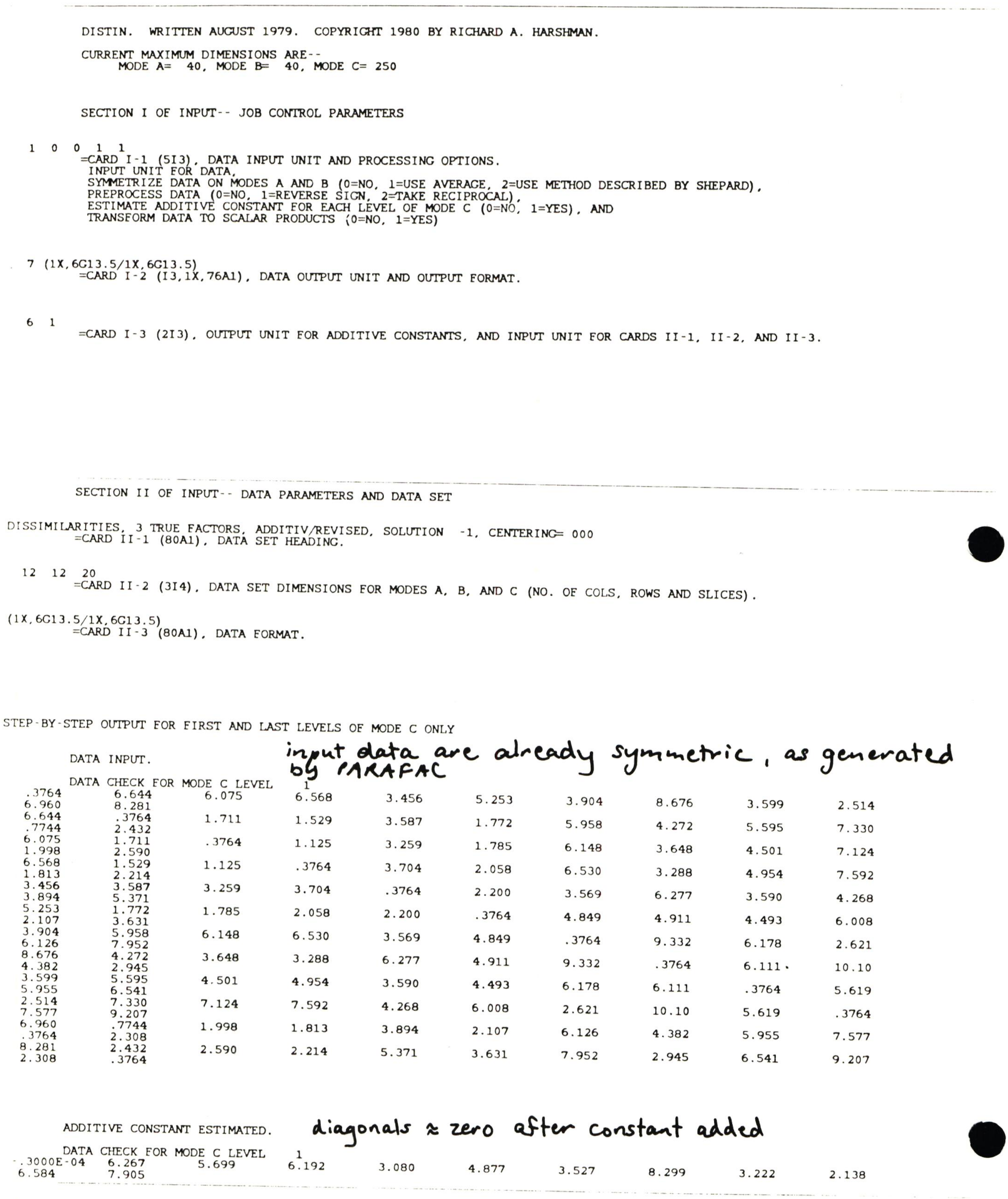
```
CARD
 I-1   1  0  0  1  1
 I-2   7 (1X,6G13.5/1X,6G13.5)
 I-3   6  1
```

(2) Suppose you have a 19x19 data matrix that you want to symmetrize for some reason. The data only are in a separate file. DISTIN input in such a case could be specified as follows:

```
CARD
  I-1   1  1  0  0  0
  I-2   7 (13F6.3/6F6.3)
  I-3   0  5
 II-1  EXAMPLE TWO-WAY DATA
 II-2   19  19   1
 II-3  (13F6.3/6F6.3)
```

Table 3.9 is the lineprinter output produced by DISTIN, given the example 1 input above. The additive constants listed here are the same as those on the PARAFAC listing when the data were generated, except for their sign. This is because DISTIN adds negative constants to cancel the effects of the positive bias that PARAFAC added when synthesizing the data. Of course, if random error had been added too, DISTIN would not be able to recover the additive constants so exactly.

Table 3.9. Example of DISTIN Lineprinter Output



```
         DISTIN.  WRITTEN AUGUST 1979.  COPYRIGHT 1980 BY RICHARD A. HARSHMAN.

         CURRENT MAXIMUM DIMENSIONS ARE--
              MODE A=  40, MODE B=  40, MODE C= 250


         SECTION I OF INPUT-- JOB CONTROL PARAMETERS


  1  0  0  1  1
         =CARD I-1 (5I3), DATA INPUT UNIT AND PROCESSING OPTIONS.
          INPUT UNIT FOR DATA,
          SYMMETRIZE DATA ON MODES A AND B (0=NO, 1=USE AVERAGE, 2=USE METHOD DESCRIBED BY SHEPARD),
          PREPROCESS DATA (0=NO, 1=REVERSE SIGN, 2=TAKE RECIPROCAL),
          ESTIMATE ADDITIVE CONSTANT FOR EACH LEVEL OF MODE C (0=NO, 1=YES), AND
          TRANSFORM DATA TO SCALAR PRODUCTS (0=NO, 1=YES)


  7 (1X,6G13.5/1X,6G13.5)
         =CARD I-2 (I3,1X,76A1), DATA OUTPUT UNIT AND OUTPUT FORMAT.


  6  1
         =CARD I-3 (2I3), OUTPUT UNIT FOR ADDITIVE CONSTANTS, AND INPUT UNIT FOR CARDS II-1, II-2, AND II-3.




        SECTION II OF INPUT-- DATA PARAMETERS AND DATA SET

DISSIMILARITIES, 3 TRUE FACTORS, ADDITIV/REVISED, SOLUTION  -1, CENTERING= 000
        =CARD II-1 (80A1), DATA SET HEADING.

 12  12  20
        =CARD II-2 (3I4), DATA SET DIMENSIONS FOR MODES A, B, AND C (NO. OF COLS, ROWS AND SLICES).

(1X,6G13.5/1X,6G13.5)
        =CARD II-3 (80A1), DATA FORMAT.



STEP-BY-STEP OUTPUT FOR FIRST AND LAST LEVELS OF MODE C ONLY

          DATA INPUT.

          DATA CHECK FOR MODE C LEVEL   1
 .3764     6.644    6.075     6.568     3.456     5.253     3.904     8.676     3.599     2.514
 6.960     8.281
 6.644     .3764    1.711     1.529     3.587     1.772     5.958     4.272     5.595     7.330
 .7744     2.432
 6.075     1.711     .3764    1.125     3.259     1.785     6.148     3.648     4.501     7.124
 1.998     2.590
 6.568     1.529    1.125      .3764    3.704     2.058     6.530     3.288     4.954     7.592
 1.813     2.214
 3.456     3.587    3.259     3.704      .3764    2.200     3.569     6.277     3.590     4.268
 3.894     5.371
 5.253     1.772    1.785     2.058     2.200      .3764    4.849     4.911     4.493     6.008
 2.107     3.631
 3.904     5.958    6.148     6.530     3.569     4.849      .3764    9.332     6.178     2.621
 6.126     7.952
 8.676     4.272    3.648     3.288     6.277     4.911     9.332      .3764    6.111     10.10
 4.382     2.945
 3.599     5.595    4.501     4.954     3.590     4.493     6.178     6.111      .3764    5.619
 5.955     6.541
 2.514     7.330    7.124     7.592     4.268     6.008     2.621     10.10     5.619      .3764
 7.577     9.207
 6.960     .7744    1.998     1.813     3.894     2.107     6.126     4.382     5.955     7.577
 .3764     2.308
 8.281     2.432    2.590     2.214     5.371     3.631     7.952     2.945     6.541     9.207
 2.308     .3764


         ADDITIVE CONSTANT ESTIMATED.

         DATA CHECK FOR MODE C LEVEL   1
-.3000E-04  6.267   5.699     6.192     3.080     4.877     3.527     8.299     3.222     2.138
 6.584      7.905
```

input data are already symmetric, as generated by PARAFAC

diagonals ≈ zero after constant added

```
 6.267      -.3000E-04  1.335       1.153       3.210       1.396       5.582       3.896       5.218       6.954
 .3980       2.056
 5.699       1.335      -.3000E-04  .7490       2.883       1.408       5.771       3.271       4.125       6.747
 1.622       2.213
 6.192       1.153       .7490      -.3000E-04  3.327       1.681       6.153       2.912       4.577       7.215
 1.437       1.837
 3.080       3.210       2.883       3.327      -.3000E-04  1.823       3.193       5.901       3.214       3.891
 3.518       4.995
 4.877       1.396       1.408       1.681       1.823      -.3000E-04  4.472       4.535       4.116       5.632
 1.731       3.255
 3.527       5.582       5.771       6.153       3.193       4.472      -.3000E-04  8.956       5.802       2.245
 5.749       7.576
 8.299       3.896       3.271       2.912       5.901       4.535       8.956      -.3000E-04  5.734       9.720
 4.006       2.568
 3.222       5.218       4.125       4.577       3.214       4.116       5.802       5.734      -.3000E-04  5.243
 5.578       6.165
 2.138       6.954       6.747       7.215       3.891       5.632       2.245       9.720       5.243      -.3000E-04
 7.201       8.830
 6.584       .3980       1.622       1.437       3.518       1.731       5.749       4.006       5.578       7.201
-.3000E-04   1.932
 7.905       2.056       2.213       1.837       4.995       3.255       7.576       2.568       6.165       8.830
 1.932      -.3000E-04
```

```
      DATA VALUES TRANSFORMED TO SCALAR PRODUCTS.

      DATA CHECK FOR MODE C LEVEL    1
 18.00      -8.497      -6.004      -8.113       5.121      -2.594       12.14      -14.64       9.494       20.24
-9.765      -15.38
-8.497       4.286       2.489       3.537      -2.145       1.467      -4.069       5.358      -5.786      -8.506
 4.974       6.892
-6.004       2.489       2.473       3.014      -2.052       .5429      -6.050       6.689      -1.584      -7.999
 2.832       5.650
-8.113       3.537       3.014       4.116      -2.611       .9433      -7.506       8.624      -2.730      -10.44
 3.936       7.233
 5.121      -2.145      -2.052      -2.611       1.732      -.4979       5.135      -5.740       1.389       6.823
-2.411      -4.744
-2.594       1.467       .5429       .9433      -.4979       .5964      -.3352       .8204      -2.486      -2.032
 1.711       1.865
 12.14      -4.069      -6.050      -7.506       5.135      -.3352       18.73      -19.93      -1.778       20.38
-4.251      -12.47
-14.64       5.358       6.689       8.624      -5.740       .8204      -19.93       21.61       .4941E-01  -22.90
 5.693       14.37
 9.494      -5.786      -1.584      -2.730       1.389      -2.486      -1.778       .4941E-01   11.37       5.471
-6.962      -6.452
 20.24      -8.506      -7.999      -10.44       6.823      -2.032       20.38      -22.90       5.471       27.06
-9.487      -18.60
-9.765       4.974       2.832       3.936      -2.411       1.711      -4.251       5.693      -6.962      -9.487
 5.821       7.908
-15.38       6.892       5.650       7.233      -4.744       1.865      -12.47       14.37      -6.452      -18.60
 7.908       13.73
```

```
      DATA INPUT.


      DATA CHECK FOR MODE C LEVEL    20
.4290E-01    7.957       5.755       8.152       4.231       6.753       5.287       9.869       3.571       2.742
 7.747       8.027
 7.957       .4290E-01   4.802       1.903       3.814       1.522       5.910       4.962       8.477       8.227
 1.405       4.748
 5.755       4.802       .4290E-01   5.506       3.846       4.687       6.166       6.996       5.213       6.471
 3.890       2.336
 8.152       1.903       5.506       .4290E-01   4.264       1.992       7.034       3.343       8.272       8.930
 3.063       5.644
 4.231       3.814       3.846       4.264       .4290E-01   2.611       3.674       6.772       5.586       4.719
 3.808       5.491
 6.753       1.522       4.687       1.992       2.611       .4290E-01   5.086       5.131       7.543       7.154
 2.338       5.317
 5.287       5.910       6.166       7.034       3.674       5.086       .4290E-01   10.06       8.030       3.644
 5.642       7.547
 9.869       4.962       6.996       3.343       6.772       5.131       10.06       .4290E-01   8.848       11.31
 5.756       6.856
 3.571       8.477       5.213       8.272       5.586       7.543       8.030       8.848       .4290E-01   6.015
 8.201       7.298
 2.742       8.227       6.471       8.930       4.719       7.154       3.644       11.31       6.015       .4290E-01
 7.854       8.511
 7.747       1.405       3.890       3.063       3.808       2.338       5.642       5.756       8.201       7.854
.4290E-01    3.629
 8.027       4.748       2.336       5.644       5.491       5.317       7.547       6.856       7.298       8.511
 3.629       .4290E-01
```

```
      ADDITIVE CONSTANT ESTIMATED.

      DATA CHECK FOR MODE C LEVEL    20
-.1498E-02   7.913       5.710       8.108       4.187       6.709       5.242       9.825       3.527       2.698
 7.703       7.983
 7.913      -.1498E-02   4.757       1.858       3.770       1.478       5.865       4.918       8.432       8.182
 1.361       4.704
 5.710       4.757      -.1498E-02   5.462       3.802       4.643       6.122       6.952       5.169       6.427
 3.845       2.291
 8.108       1.858       5.462      -.1498E-02   4.219       1.948       6.990       3.299       8.227       8.886
 3.018       5.599
 4.187       3.770       3.802       4.219      -.1498E-02   2.566       3.629       6.727       5.541       4.674
 3.764       5.447
 6.709       1.478       4.643       1.948       2.566      -.1498E-02   5.042       5.086       7.499       7.110
 2.293       5.272
 5.242       5.865       6.122       6.990       3.629       5.042      -.1498E-02   10.01       7.986       3.600
 5.598       7.503
 9.825       4.918       6.952       3.299       6.727       5.086       10.01      -.1498E-02   8.804       11.26
 5.711       6.812
 3.527       8.432       5.169       8.227       5.541       7.499       7.986       8.804      -.1498E-02   5.971
 8.157       7.254
 2.698       8.182       6.427       8.886       4.674       7.110       3.600       11.26       5.971      -.1498E-02
 7.809       8.466
 7.703       1.361       3.845       3.018       3.764       2.293       5.598       5.711       8.157       7.809
-.1498E-02   3.584
 7.983       4.704       2.291       5.599       5.447       5.272       7.503       6.812       7.254       8.466
 3.584      -.1498E-02
```

Table 3.9 (Continued)

```
          DATA VALUES TRANSFORMED TO SCALAR PRODUCTS.

          DATA CHECK FOR MODE C LEVEL   20
   23.76    -14.41     -.6848    -14.23     3.956    -7.914     8.399    -18.49     20.48     23.65
  -13.47    -11.04
  -14.41     10.04     -2.554     10.06    -1.242     6.641    -1.918     10.82    -15.71    -13.05
   8.414     2.898
  -.6848    -2.554      7.479    -4.414    -2.645    -4.328    -4.740    -2.530     5.205    -1.507
   .6642     10.05
  -14.23     10.06     -4.414     13.52    -1.299     7.576    -7.406     19.21    -12.26    -17.31
   6.524     .2443E-01
   3.956    -1.242     -2.645    -1.299     1.684     .2605     4.516    -3.893     .3130     5.325
  -1.924    -5.051
  -7.914     6.641     -4.328     7.576     .2605     5.422     .2604     7.670    -10.58    -7.157
   4.399    -2.248
   8.399    -1.918     -4.740    -7.406     4.516     .2604     20.52    -21.98    -6.801     19.19
  -1.090    -8.946
  -18.49     10.82     -2.530     19.21    -3.893     7.670    -21.98     35.79    -6.034    -30.10
   5.903     3.635
   20.48    -15.71      5.205    -12.26     .3130    -10.58    -6.801    -6.034     29.65     12.40
  -14.12    -2.544
   23.65    -13.05     -1.507    -17.31     5.325    -7.157     19.19    -30.10     12.40     30.81
  -10.77    -11.49
  -13.47     8.414      .6642     6.524    -1.924     4.399    -1.090     5.903    -14.12    -10.77
   8.636     6.834
  -11.04     2.898      10.05     .2443E-01 -5.051   -2.248    -8.946     3.635    -2.544    -11.49
   6.834     17.88
```

```
          TRANSFORMED DATA WRITTEN ON UNIT    7 IN FORMAT SUITABLE FOR INPUT TO THE PARAFAC PROGRAM
```

↳ i.e., scalar products

```
ADDITIVE CONSTANTS FOR    20 SUBJECTS WITH    12 X    12 DATA
   1  -.3764       2  -.6957       3  -.6518       4  -.3578       5  -.3834

   6  -.2555       7  -.5503       8  -.6095       9  -.2200E-02  10  -.3787

  11  -.8439      12  -.8066      13  -.6958      14  -.1380      15  -.8376

  16  -.3435      17  -.3272      18  -.6964      19  -.5139      20  -.4440E-01
```

CHAPTER 4

DATA PREPROCESSING

Sometimes the raw data may be unsuitable for PARAFAC analysis because it violates some assumptions of the model. However, this problem often can be overcome by transforming or preprocessing the data in some way before beginning the analysis (Harshman and Lundy, 1984a). Some of these transformations allow "direct fitting" of the data (i.e. the data structure is unchanged by the preprocessing) while others result in "indirect fitting" (i.e. the structure of the transformed data is different from the raw data structure).

The PARAFAC program can do two types of preprocessing: "centering" (removing means), explained in Section 4.1, and "normalization" (equalizing mean squares), explained in Section 4.2. Both permit direct fitting of the data. A special type of PARAFAC normalization, "equal-average-diagonal normalization", is separately discussed in Section 4.5 below; the descriptions of normalization and iterative preprocessing in Sections 4.2 and 4.3 do not apply to it.

Covariance computation (not done by the PARAFAC program) is an important type of data transformation which permits indirect fitting. PARAFAC analysis of covariance matrices instead of raw data is discussed in Section 4.4.

Another type of data transformation, conversion of similarities data to scalar products, is briefly mentioned in Section 4.6.

4.1 CENTERING

The purpose of centering is to make the data more consistent with the PARAFAC factor model. Centering removes constant biases or constant terms in the data which might otherwise either yield constant loadings on a factor or distort the solution. It also shifts the emphasis in certain modes from baseline values to relationships among patterns of change. PARAFAC centering involves removal of "fiber" means (row, column or tube means; see below) rather than removal of "slab" means or the overall "grand" mean. This centering method preserves the underlying factor structure of the data whereas the other types of centering distort it. Proof of this is given in a detailed discussion of centering by Harshman and Lundy (1984a, pp. 231-9).

PARAFAC can center the data on any combination of one or more modes. Of course, the simplest case is centering on only one mode. For example, PARAFAC centers the data on Mode A in the following way: for every row (where a "row" is defined as the set of values taken across all levels of Mode A while holding the levels of Modes B and C fixed), the row mean is subtracted from the data values in that row. The formula is given in Appendix B, Equation 5a; in it, the values of subscript i correspond to the levels of Mode A in the data array X, the values of j to the levels of Mode B, and the values of k to the levels of Mode C. Similarly, centering on Mode B removes column means and centering on Mode C removes tube means (where a "column" is the set of points across all levels of Mode B at fixed levels of Modes A and C, and a "tube" is the set of points across all levels of Mode C at fixed levels of Modes A and B). Equations 5b and 5c in Appendix B are the formulas used for Mode B and Mode C centering respectively.

The diagrams below illustrate what is meant by the terms "row", "column" and "tube". (You may recall from Chapter 1 that PARAFAC denotes the three ways of the data (i.e., across, down and back) as Mode A, Mode B and Mode C respectively.)

```
   ...............     ...............     ...............
  .             .:    .             .:    .X            .:
 .             . :   .             . :   .X            . :
:.............:  :  :.............:  :  :X............:  :
:XXXXXXXXXXXXX:  :  :X            :  :  :             :  :
:             :  :  :X            :  :  :             :  :
:             :  :  :X            :  :  :             :  :
:             : .   :X            : .   :             : .
:.............:.    :X............:.    :.............:.

X's denote "row"   X's denote "column"  X's denote "tube"
```

As a simple example of how PARAFAC centering works, consider the following 3X3 data matrices. The first is the original data matrix and the others show it after it has been either row-centered, or column-centered, or both row- and column-centered. Note that the entries in the double-centered matrix do not depend on the order of the centering operations (i.e., row-centering a column-centered matrix has the same result as column-centering a row-centered matrix).

3 2 4	0 -1 1	1 -1 1	0.67 -1.33 0.67
1 2 3	-1 0 1	-1 -1 0	-0.33 -0.33 0.67
2 5 2	-1 2 -1	0 2 -1	-0.33 1.67 -1.33
Original Data	Centered on only Mode A (rows)	Centered on only Mode B (columns)	Centered on both Modes A and B (rows & columns)

If Mode A and/or Mode B centering were applied to a three-way data array, the successive matrices would exhibit the same effects as those shown above. Mode C centering is a parallel procedure, but is harder to illustrate on this page.

The PARAFAC factor loadings reflect any centering that is done. That is, the loadings for each factor have a mean of zero in any mode that is centered. The level means in the Summary Statistics Table (described in Section 5.1) more indirectly indicate when centering has been done. For example, centering Mode A removes row means (across levels of Mode A, but within levels of Modes B and C). Thus, the Mode B and C level means are set to zero, but not the Mode A level means. In general, centering on one mode sets the level means of the other two modes to zero. Consequently, if two of the three modes are centered, then the level means for all three modes are zero.

4.1.1 When to Center

There is no "cookbook" method for centering, but we can give some general rules of thumb to follow for PARAFAC analyses.

When deciding whether or not to center, consider the type of data to be analysed. You may not need to center if you are sure that you have ratio scale data with no constant terms or factors. Even so, however, you may still want to so that certain variance components will be emphasized. On the other hand, if you have interval scale data, you should always center to remove constants and thus transform it to ratio scale data. This is necessary because PARAFAC requires ratio scale data.

If you choose to center, you must then decide how many modes require this transformation. The data should almost always be centered on at least one mode, and often it is better to center on two modes because this removes more unwanted constants and interactions. Although in theory, centering on three modes should be best, it usually does not work in practice because it reduces the signal-to-noise ratio too much. In other words, the factor or systematic deviations become too small compared to the random error variation, which is not affected much by the centering.

The meaning of the mode may determine whether or not you center on it. Centering across the "person" mode parallels traditional two-way factor analysis methods, but based on our experience, we would recommend <u>against</u> centering the person mode in three-way data. Also, if response biases on ratings scales (for example) are suspected, center across a mode for which the bias on each scale is constant (i.e., some mode other than the "scales" mode).

Sometimes the results of a PARAFAC analysis may indicate where centering is needed. If the analysis produces a factor which has loadings of more or less constant size and sign in one or more modes, one of the modes for which the loadings are constant should probably be centered. Such centering would remove this factor from subsequent analyses.

In addition to centering, you will usually also normalize the data (normalization is discussed below). You may have to try several different preprocessing combinations in order to get the most interpretable solution. We have sometimes found it useful to center two modes and normalize two modes, but not the same two. Especially with three-way rating scales data, we have obtained good results when we have centered the scales and the stimuli modes and have normalized the scales and person modes.

4.2 NORMALIZATION

In contrast to the centering procedure, PARAFAC normalization is not intended to make the data more appropriate for the PARAFAC model. Rather, normalization improves the solution by equating the mean square data values of "slices" (see below) of the data; large but spurious differences in these mean squares do not contradict the model, but they may adversely affect the solution in two ways:

1. During the factor estimation procedure, PARAFAC tries to minimize the total error variance. Levels with large mean squares contribute disproportionately to the total error variance, and so the factors are primarily determined by the attempt to fit the data for these levels.

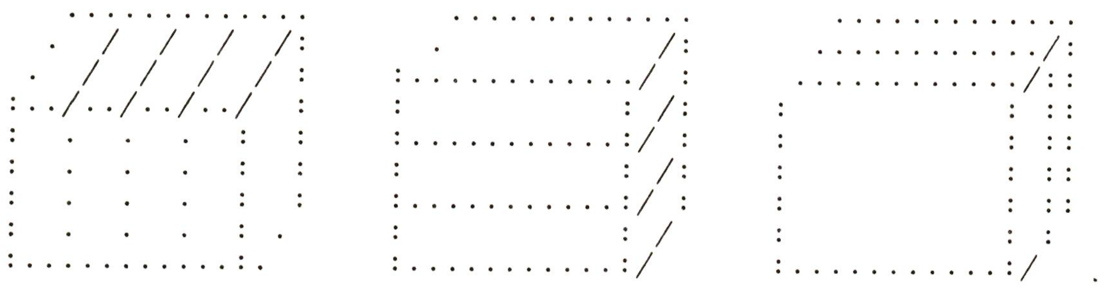
2. Levels with large mean squares require large factor loadings relative to the others, so that the large data values in those levels can be predicted. Thus, the loadings for these levels tend to be large, even though their meaning may not be central

to the interpretation of the factors on which they load highly.

PARAFAC normalization equates the influence of the data slices on the solution and also facilitates comparison of factor loadings across rows of the factor loading matrix. This preprocessing does not overcome problems that fiber-conditional differences in mean square values would cause, but our experience with PARAFAC so far suggests that such differences are not significant. See Harshman and Lundy (1984a) for more details about the rationale for PARAFAC normalization; note that they use the term "size standardization" for what is called normalization here.

As with centering, the data can be normalized on any combination of one or more modes. The simplest case is normalizing on one mode. For example, the data are normalized on Mode A as follows: in each slice (where the "slice" is defined as the matrix of points across all levels of Modes B and C for a fixed level of Mode A), the data points are all multiplied by the constant which produces a mean squared data value of 1.0 in the slice. The formula is given in Appendix B, Equation 6a. Similarly, normalizing on Mode B results in a mean squared data value of 1.0 in each Mode-A-by-Mode-C slice (Equation 6b), while Mode C normalization produces a mean square of 1.0 in each Mode-A-by-Mode-B slice (Equation 6c). The Summary Statistics Table, described in Section 5.1, verifies the normalization (but see Section 4.3 below).

The diagrams below illustrate the way the data array is "sliced" for normalization on Mode A, Mode B and Mode C respectively. In general, these slices can be referred to as "lateral", "horizontal" and "frontal" planes, respectively (e.g., Kroonenberg, 1983).



Mode A Slices (lateral)

Mode B Slices (horizontal)

Mode C Slices (frontal)

The effect of PARAFAC normalization on one matrix is shown below. You can verify for yourself that the mean squared value of the normalized matrix is (approximately) 1.0.

3	2	4	1.032	.688	1.376
1	2	3	.344	.688	1.032
2	5	2	.688	1.721	.688
Original Data			Normalized Data		

In a three-way data array, successive matrices (slices) would be similarly transformed, so that the mean squared value of each normalized matrix would be 1.0.

4.2.1 When to Normalize

Any mode for which large, meaningless mean square variations occur should be normalized. (To check the mean square values, you can input the raw data to PARAFAC with no preprocessing requests, and suppress the analysis; then refer to the Summary Statistics Table output from that run.) Variations in mean square values are meaningless if they are the result of arbitrary differences in the units of measurement across levels of a mode. For example, mean reaction time, measured in milliseconds, might vary over ten times the range of a second variable, number of correct responses. Also, even when the units of measurement are the same, mean square variations might sometimes be thought to be unimportant. For example, ratings scales might give rise to spurious differences in mean square values, depending on how they are worded and on subjects' response biases.

4.3 ITERATIVE PREPROCESSING

PARAFAC employs an iterative process when both centering and normalization on the same mode are specified or when normalization on more than one mode is requested. Each iteration involves first normalizing the mode(s) as requested, then centering the mode(s) as requested, and finally checking to see if another iteration is required. The iterative process is necessary because the normalization of any given mode is distorted by subsequent centering of the same mode or by subsequent normalization of a second mode. However, by repeating the procedures, one can more closely approximate the simultaneous centering and normalization that was requested. The convergence criterion set by PARAFAC for the iterative normalization process is accuracy within one percent; that is, for each slice in all normalized modes, the mean square value is between 0.99 and 1.01. (Because in each iteration, the program always centers after it normalizes, the centering is always exact to within computer roundoff error and thus does not need to be checked.) PARAFAC does at most 20 iterations, and if the convergence criterion is not met by then, the user is informed. The data values used in the succeeding analyses are those obtained on iteration 20. The mean square values can be checked by referring to the Summary Statistics Table listed by PARAFAC.

Another way of using PARAFAC centering and normalization to preprocess the data is what we call "one-cycle preprocessing". First, centering is done as requested and then (iterative) normalization is done as requested. This procedure differs from the preprocessing described above in that the centering is done only once before the normalization (i.e., each iteration involves only normalization). As with regular preprocessing, however, iterative normalization is performed if more than one mode is normalized, and the convergence criterion for the iterative procedure is accuracy within one percent.

One-cycle preprocessing is not an option in the current version of PARAFAC, but it can be accomplished by a two-step procedure. First, input the raw data to PARAFAC and center only (on one or more modes), suppress the analysis and save the centered data (see Cards I-2, I-4 and I-8 of the PARAFAC Input Specifications Table). Then input the centered data to PARAFAC, normalize and analyse it (or, save the normalized (and centered) data and analyse it in a separate run). An example of the PARAFAC deck set-up to do one-cycle preprocessing is given in Chapter 2.

If you obtain a solution that is hard to interpret because (you suspect) the iterative preprocessing unduly complicated the underlying data structure, you may want to try one-cycle preprocessing. As yet, we have not used it extensively, but we have obtained similar solutions with regular preprocessing and one-cycle preprocessing of rating scales data (i.e., scales and stimuli modes centered, scales and person modes normalized). The advantage of the one-cycle preprocessing is that the relationship between the raw data and the transformed data is much simpler. However, one-cycle preprocessing does not correct for the disturbance in centering subsequent normalization of the same mode causes, and so it may not be best if you want to transform slices of the data to z-scores. (The Summary Statistics Table would indicate how serious the deviation from z-scores was.)

4.4 COVARIANCE ANALYSIS

Covariance analysis is an instance of indirect fitting of the data. Direct fitting is more straightforward and hence is usually preferable, but sometimes it is advisable to analyse covariances (not correlations) instead of the raw data. PARAFAC does not compute covariances, but it should be a simple matter to find or write a program that does. So that you can use the method of Equal-Average-Diagonal normalization as recommended in Section 4.5, the covariances should be arranged as a series of matrices (like the "Mode C slices" in Section 4.2), one matrix for each occasion (or sample or condition). Hence, Modes A and B of the covariance data are symmetric. As explained in Section 4.5, the restriction of symmetry to Modes A and B simplifies the programming involved, and is not for some theoretical reason. There are several instances when you should consider analysing covariances:

1. Object variation in the data rather than system variation. (Harshman, 1970; Harshman, 1972; Harshman and Berenbaum, 1981)

2. Data consisting of several different samples, possibly of varying sizes (e.g., different numbers of people tested on several occasions).

3. Too many levels in one mode for the analysis to be practical.

Whenever "system variation" (proportional patterns of factor change) is present across the levels of each mode, PARAFAC analysis of the raw data can determine a unique orientation for the factor axes. Sometimes, however, problems of rotational indeterminacy in the PARAFAC solution do arise. If the cause is the presence of mainly "object variation" (idiosyncratic patterns of factor variation) in one mode, then these problems may be overcome by analysing covariances instead of the raw data. This is so because the PARAFAC three-way generalization of the covariance model does not assume that the raw data from which the covariances were computed contains system variation (Harshman, 1972; Harshman and Lundy, 1984b). The covariances are computed across the mode that is the source of the object variation, and so that mode "disappears" from the covariance data. Thus, analysis of the covariances gives you no information about that mode, but you can use regression techniques to estimate factor loadings for it. The analysis does, however, produce factors that reveal the same underlying structure for the other two modes as direct analysis of the raw data would have yielded (had there been system rather than object variation present in the data).

In many cases, the decision to analyse covariances rather than raw data may be somewhat arbitrary. Haan (1981) gives an instance where the raw data were composed of personality measures repeated across time. She viewed the people in the study as sources of object variation rather than of system variation, and so she computed covariances (actually, she used unstandardized covariances or cross-products) over the person mode and analysed them instead of the raw data. Sometimes, though, you may not be sure of the source or amount of object variation. You may want to begin by analysing the raw data (after appropriate centering and normalization). Provided that there is enough system variation in the data, PARAFAC may yield a unique, interpretable solution that satisfies you (any object variation will be treated as error). But a solution with factors that are all nearly constant (though not necessarily equal) in one mode indicates that insufficient system variation is present in that mode for PARAFAC to establish a unique orientation for the factor axes. (You might also notice nonuniqueness or uninterpretability of the solution under these conditions.) If you have some theoretical ground for believing that the data might contain enough object variation to cause these problems, then it is reasonable to compute and analyse covariances. If you have no reason to suspect object variation, however, error variance may be the problem and so covariance analysis may not be justified. (Note that when only a subset of the factors have constant loadings in a particular mode, the cause is probably constant terms in the data; they can be removed by centering the data on the constant mode; see Section 4.1.1 above and Harshman and Lundy, 1984a.)

One case when covariances should always be analysed occurs when each occasion involves measurements on the same variables, but for a different set of people. The sample sizes may remain constant or may vary across occasions. In one sense, this may be thought of as the most extreme case of object variation, since one cannot expect the patterns of factor variation to be proportional across occasions when different people are involved every time. In another sense, this is a violation of the PARAFAC raw score model, which requires that the same people be measured on the same variables at every occasion. Either way, analysis of covariances is required to overcome the problem.

Sometimes you may have data for which there are many levels in one mode, compared to the other modes (e.g., measures for hundreds of subjects), and to analyse it would be expensive. A way of reducing the size of the data without losing information is to compute covariances across the mode with so many levels; this eliminates that mode and hence reduces the size of the data set, but uses all the available information. (However, this approach makes an orthogonality assumption which is discussed below.) As mentioned in the discussion of object variation, the factor loadings for the mode that disappeared can be estimated using regression procedures.

Although correlations have been used in traditional two-way factor analysis, we have said that they are not appropriate for three-way analysis, whereas covariances are. The reason is that calculation of correlations separately standardizes the variance within each level of the third mode (e.g., occasions), thus destroying the proportional factor relationships across the levels. This violates a basic assumption of the PARAFAC model to a degree which may or may not be serious, depending on the data. In contrast, covariances do not involve such standardization. EAD normalization (see Section 4.5) then allows standardization of the covariances, while maintaining proportionality across levels of Mode C. The normalized covariances have an average diagonal of 1.0, compared to correlation matrices for which all diagonals are exactly 1.0.

While analysing covariances instead of the raw data is an easy way to overcome problems of object variation or very large data sets, you should keep in mind a fundamental assumption of this procedure. It is that the factors underlying the mode over which the covariances are computed (i.e., the mode that "disappears") are orthogonal (mutually independent). Even when the factors are not perfectly orthogonal in that mode, however, analysis of covariances is still reasonable, especially for initial exploration of the data. In this case, PARAFAC gives the best orthogonal approximation to the "true" (nonorthogonal) factors underlying the data. Often, such a solution is clear and interpretable.

On the other hand, if you suspect that highly correlated factors underlie the mode that "disappears", covariance analysis is not useful even for exploration. The assumptions of the model would be greatly violated, and the solution would probably be a very distorted version of the "true" factors.

4.5 EQUAL-AVERAGE-DIAGONAL NORMALIZATION

Equal-average-diagonal (EAD) normalization is applied to covariance data which are symmetric across Modes A and B. (The decision to restrict symmetry to Modes A and B was arbitrary. There is no theoretical reason that the data cannot be symmetric across Modes B and C, or across Modes A and C. Due to practical programming considerations, however, we chose to restrict symmetry to one pair of modes. Modes A and B seemed to be the best alternative, since the most common arrangement of covariance data is a series of matrices (i.e., the frontal planes are symmetric). If your data set is not in this form, you must rearrange it to make it suitable for PARAFAC.) EAD normalization equates the variance associated with each level of Mode A (and B) so that the levels can all contribute equally to the solution, but at the same time, it preserves the proportional factor relationships across the levels of Mode C. It is accomplished in two steps:

1. The mean is computed across each "tube" of diagonal elements c(i,i,k) (where tube is as defined in Section 4.1 above).

2. Each element c(i,j,k) in the array is then divided by the square root of the product of the ith and jth tube means determined in step 1. Thus, each diagonal element (where i=j) is divided by the mean value of the tube in which it is located; the average value in each tube of rescaled diagonals is then 1.0.

The exact formula for EAD normalization is given in Appendix B, Equation 7.

The diagram below illustrates how EAD normalization is applied. The individual slices are square matrices. The "X's" indicate the tube of (1,1,k) diagonal elements and the "Y's" mark the tube of (2,2,k) diagonal elements (here, k=1,2,3). The off-diagonal (1,2,k) and (2,1,k) entries are the "Q's". The "Q"-values are divided by the square root of the product of the mean X and the mean Y. Each X is divided by the mean X and each Y is divided by the mean Y. A parallel procedure would be applied to all other elements in the array to complete the EAD normalization.

```
       ............
      :XQ         :
      :QY         :
    ..:.........  :
   :XQ         :  :
   :QY         :  :
 ..:.........  :..:
:XQ         :  :
:QY         :  :
:           :..:
:           :
:           :
:...........:
```

4.5.1 When To Use Equal-Average-Diagonal Normalization

Use the EAD method to normalize covariance matrices. (Section 4.4 above is a discussion of when to use covariances.) The resultant covariance matrices are the same as would be produced if the original raw data were first centered and normalized appropriately and covariances were then computed from the transformed data. An example of the appropriate PARAFAC deck set-ups is given in Chapter 2.

For example, suppose you have covariance data of the form Variables by Variables by Occasions, and suppose that the raw data from which the covariances were computed was of the form Variables by Persons by Occasions (i.e., covariances were computed across persons, so that the person mode disappeared). EAD Normalization of the covariances equalizes the variance contributed by each variable while maintaining the pattern of factor proportionality across occasions. You can obtain the same result by centering the person mode (in general, the mode that disappears when covariances are calculated) and normalizing the variable mode in the raw data, and then computing covariances. The effect of the preprocessing is to set the variance within each variable slice of the raw data to 1.0 (assuming "variables" is Mode A, refer to the diagram of Mode A slices in Section 4.2). Thus, no level of the variable mode will contribute disproportionately to the PARAFAC solution after covariance calculation.

In combination with EAD normalization, we suggest that you select the option to standardize the factor loadings so that those in Modes A and B jointly reflect the scale of the data and those in Mode C have a mean square of 1.0 (i.e., ISTANM=4 on Card I-8; see the PARAFAC Input Specifications Table). Then you can interpret the Mode A and B loadings the same as loadings obtained via traditional two-way (orthogonal) factor analysis of a correlation matrix. That is, each factor loading represents the correlation between the variable (corresponding to that level of Mode A or B) and the factor. The Mode C loadings are the variances for the factors in the mode that disappeared during covariance computation (e.g., if that mode were "persons", the Mode C loadings would give the variances of the person loadings or "factor scores").

4.6 SCALAR PRODUCTS

PARAFAC can be used to do metric multidimensional scaling (MDS) of distance-like data, if the data are first transformed to scalar products. The scalar products are then analysed by PARAFAC in much the same way as covariances are. This procedure indirectly fits an MDS distance model (i.e., the weighted Euclidean model) to the original distance-like data by directly fitting the factor or component model to the scalar products.

The DISTIN program, described in Chapter 3, does the necessary transformations and outputs the transformed data in a format that is compatible with PARAFAC input requirements. DISTIN assumes Mode C of the input data is the "person", "condition", "occasion", etc. mode and computes scalar products over Modes A and B. Thus, the scalar product data array output from DISTIN consists of a series of matrices that are symmetric across Modes A and B (i.e., the frontal planes are symmetric). The symmetry is restricted to these two modes to simplify the programming, not for theoretical reasons.

The transformation of three-way similarities data to scalar products involves several steps. For each Mode A-by-Mode B matrix (i.e., for each subject):

1. If the data are not symmetric, symmetrize across Modes A and B.

2. If the data are proximities or similarities (where a larger value represents a stronger relationship or greater similarity) convert to dissimilarities (where a larger value represents a weaker relationship or larger "distance"). This is done by either reversing the sign or taking the reciprocal of each similarity measure. Then, estimate the additive constant and subtract it. This transforms the interval scale dissimilarities data to ratio scale distance data.

3. Square each distance value.

4. Double-center the matrix of squared distances (i.e., center on Mode A and on Mode B; see Section 2.1 above).

5. Multiply by -.5 to obtain scalar products. Torgerson (1958) explains the reason for this adjustment.

Upon inputting the scalar product data to PARAFAC, you will usually want to normalize on Mode C (see Section 4.2 above) before beginning the analysis. No other preprocessing (e.g. centering) of scalar products is required.

This section has focused on scalar products. For a more thorough discussion of multidimensional scaling, see Kruskal and Wish (1978).

CHAPTER 5

PARAFAC ANALYSIS OUTPUT

As Figure 2.1 illustrates, a PARAFAC job generates a lineprinter listing and usually also one or more disk or tape files. The listing is mostly self-explanatory. It provides a check of the parameters and data supplied to the program and shows the analysis results, while the disk (tape) files are copies of the analysis results which may be input to PARAFAC or one of the utility programs in the package.

A detailed description of both types of output is given below. Informative messages, error messages and warnings that may appear on the listing are discussed in Chapter 8.

Throughout this chapter, numerous references are made to parameter names, which are printed in upper case, and to "Card" and "Record" numbers (e.g., ISTART on Card I-5). These are all taken from the PARAFAC Input Specifications Table in Chapter 2. It is assumed that you are familiar with the information in that table.

Besides referring to this chapter when interpreting your output, you may also want to read the article by Harshman and DeSarbo (1984); they describe how they applied PARAFAC to marketing data and how they interpreted the resulting solution.

5.1 LINEPRINTER OUTPUT: VERIFICATION OF INPUT

5.1.1 Input Section I (Parameter Check)

First on the listing is an overall program heading that identifies the version of PARAFAC being used, specifies the current array limits and gives the label from the user's DIMS run (or a standard label if PARAFAC was not redimensioned). Printed next is a heading for Input Section I, followed by each Section I input record as it was read by the program, and a description of the record. Any unspecified parameters (i.e., blank on the input record) are printed as zero on the listing. In the comments following the record output, the default values assigned to these parameters by PARAFAC are listed. It is important to check this part of the listing, especially if warning or error messages are subsequently issued, to ensure that PARAFAC read the parameters as you intended (i.e. make sure you didn't place any values in the wrong column).

5.1.2 Input Section II (Data Check)

The section heading and first three data parameter cards are listed in the same way as for Section I above. For the data set, however, only the first, second and last rows of the data are listed; they are in G-format rather than the format used for input. This permits a check of data input without listing the entire data set. Use this check; otherwise, you may be unaware when the data are read improperly. For example, if PARAFAC reads only part of the file because a faulty format causes it to input extra values from each data record, no error is detected, and analysis of the faulty "data" proceeds as usual. (On the other hand, if it encounters the end of the file prematurely, the job terminates abruptly with a system message which would alert you to erroneous specification of the data dimensions or format, of problems in the data file itself.)

5.1.2.1 Missing Data. - When a missing value subscript table is input after the data set, the subscripts are listed after the data check. When special codes are used to identify missing data, the subscripts of the data cells containing these codes are printed. Make sure that the subscripts in the missing value table were read correctly, and that other cells identified here contain missing data codes. Valid data values can be erroneously identified as missing data if the data range limits or the missing data codes on Card I-4 are incorrectly specified.

5.1.2.2 Centering/Normalization. - When the user has requested that the data be centered and/or normalized on one or more modes, the first, second and last lines of the revised (i.e. after centering and/or normalizing) data.are listed to allow a check of the transformation. In some instances, a reference to iterations precedes this data check. Iterative normalization is performed if either normalization on more than one mode or centering and normalization on the same mode is requested. PARAFAC indicates the number of iterations required to meet the convergence criterion or, if the iterative procedure did not converge in twenty iterations, a message informs the user of this. (In the latter case, the values printed in the centering/normalization check and the data used in the subsequent analysis are those obtained on iteration 20.) Refer to Chapter 4 for more details on centering, normalization and the iterative procedure.

5.1.2.3 Summary Statistics Table. - Next on the listing is a table of statistics for the overall data, and for each level of each mode (i.e., for every two-way "slice" of the three-way data cube). The statistics are computed after initial estimates have replaced missing data values and after the data have been centered and/or normalized. (Equations 1a, 1b and 1c in Appendix B are the formulas used in the computations.) Check these statistics to make sure that the general properties of the data are what you expect. A comparison of the mean squares across the levels within a mode may reveal large differences. Unless these differences are meaningful, you may want to normalize the data on any modes for which substantial variation of the mean square values occurs. See Chapter 4 for a discussion of when to normalize.

On the other hand, if you requested normalization of the data, then the mean squares printed in the table allow a check of the normalization accuracy. Where the normalization was not iterative, the mean square value of every level of the normalized mode is exactly 1.0. Where an iterative procedure was necessary, and the procedure converged, the mean squares for each mode normalized may vary between 0.99 and 1.01. The closer to 1.0 they are, the more accurate is the normalization. If convergence was not reached, some of the mean squares will be outside this range.

While the mean squares in the statistics table reflect the effects of any normalization, the means do not directly show the result of centering. When only one mode is centered, the level means for that mode may be far from zero, but the level means for the other two modes are zero. If centering is done on more than one mode, however, the level means of all three modes are zero (within computer roundoff error). The reason for this is given in Chapter 4, which describes the centering process in more detail.

5.1.2.4 Symmetry Check. - When a check for symmetry in the data across Modes A and B was requested, a message about this check appears next on the listing. In addition, the location and value of any nonsymmetric points are printed, for a maximum of 50 points. If there are nonsymmetries, the program stops here.

5.1.3 Input Section III (Initial Loadings)

A section heading for Input Section III is listed only if starting loadings were supplied by the user. When variable format input was used (ISTART=2 on Card I-5), the format and the loadings for each mode are printed next, so that their input can be checked. But for a standard continuation (ISTART=1), the section heading is the only documentation for Input Section III. The loadings themselves are printed for iteration zero in the analysis part of the output. That output provides sufficient verification of their input; see Section 5.2.2.1 for more details about starting loadings.

5.1.4 Special Output

The revised data are printed next when no analyses are to be done (i.e., NSOLS=-1 on Card I-2) and IUNITD on Card I-8 is set to 6 (the standard output unit). Such output permits only a visual check of the preprocessed (or synthetic) data. Usually, you will want to write them on a disk file so that they can be analysed later.

5.2 LINEPRINTER OUTPUT: ANALYSIS DOCUMENTATION

This documentation consists of a series of factor loadings matrices which show the progression towards a solution. A similar sequence is repeated for every solution requested. The starting loadings are printed first; then reestimated loadings are listed after every NITER (from Card I-2) iterations, up to NOUTS (also from Card I-2) times. If the solution converges before NITER times NOUTS iterations have been performed, however, the loadings at the point of convergence are output.

In addition to the factor loadings, the program supplies other information with every solution. Descriptive labels and fit values precede each loadings matrix. Also, depending on the original options selected, messages that the data have been recentered and that the missing values and/or the data diagonals have been reestimated may appear between the loadings outputs. And after the final loadings for each solution, an error analysis table and matrices of factor correlations and cross-products are listed.

The analysis documentation is described in detail below, generally in the order in which it occurs on the listing. First, general information provided by the loadings matrices is discussed. Next, the loadings outputs are explained in Section 5.2.2, with specific reference to different stages of the analysis. Then, factor correlations and cross-products, error analysis, missing value and diagonal estimates, and special outputs are discussed. Informative messages which may occur along with the loadings outputs are described in Chapter 8.

5.2.1 Descriptive Information, Fit Values, Etc.

The first two lines at the top of each loadings matrix are the job title (from Card I-1) and the data title (from Record II-1). The next two lines contain the solution (SOL) number, the iteration number at which the loadings were obtained, and four fit values which measure how well the factors predict the data. The fit values generally indicate improving agreement between the predicted data and the real data as more iterations are performed. The difference values -- DIFFA, DIFFB and DIFFC -- give some indication of the stability of the current estimates of the factor loadings. The iteration number for the first loadings output in any solution is always zero, which indicates starting loadings rather than iteratively computed loadings. Subsequent loadings outputs (except for the final one when the solution converges) have an iteration number that is some multiple of NITER.

MEAN SQ ERROR is the mean squared difference between the real data values and those predicted from the factor loadings (i.e., it is the sum of the squared errors divided by the total number of points in the data set). Its value decreases as the predicted data approach the real data. Its mathematical formula is given in Appendix B, Equation 2a.

STRESS is the square root of the ratio of the sum of the squared errors to the sum of the squared real data values. Like MEAN SQ ERROR, STRESS decreases as the predicted data begin to more closely resemble the real data. Zero is the minimum value for STRESS and this only occurs if all predicted data points are identical to the real data (i.e., the fit is perfect). The formula for STRESS is given in Appendix B, Equation 2b.

R is the Pearson product-moment correlation between the real data and predicted data and RSQ is the variance accounted for (VAF) by the PARAFAC model with NFACT factors (but see Section 6.7 for how to compute the VAF in indirect fitting, e.g., when analysing covariances). If the starting loadings are random numbers, then these two values probably start out close to zero, because random loadings cannot fit the data well. Both measures increase as the fit improves, up to a maximum of 1.0, which indicates perfect fit. Generally, when R is large, MEAN SQ ERROR is small. However, if there are a few extreme outliers in the data, both a large R (fitting the very large error variance due to these errant points) and a large (undesirable) MEAN SQ ERROR are possible. Thus, you should check that the sizes of both are reasonable.

DIFFA, DIFFB and DIFFC indicate the maximum percentage change in the value of any loading in Modes A, B and C respectively from the previous iteration. (The percentage value is computed by dividing the magnitude of the change in the loading by the root mean square loading value for that factor in that mode, and then mutliplying by 100.) At iteration zero, they are all zero, because no iterations have yet been performed. During the first several iterations, large changes in the loadings occur before an initial crude fit of the data is obtained; these fluctuations are reflected by large "DIFF" values. The loadings estimates are further refined during subsequent iterations, and the differences from one iteration to the next gradually decrease. Thus, "DIFF" values for succeeding loadings outputs are smaller. The program considers the estimation process to have converged when all differences are less than the convergence criterion specified for the corresponding mode (DIFMXA, DIFMXB and DIFMXC respectively on Card I-7). (Except when ISTANM on Card I-8 equals 5, outputting the loadings at every iteration does not allow a check of DIFFA, DIFFB and DIFFC. This is because the differences are computed from changes due to reestimation of the loadings, and do not reflect changes due to the renormalization that occurs whenever the loadings are printed.)

5.2.2 Factor Loadings

Following the fit values, etc., factor weights or loadings are printed in matrix form, one matrix for each mode. Each matrix has NFACT (from Card I-2) columns and NAS, NBS or NCS (from Record II-2) rows. Column (factor) and row (level) numbers appear across the top and down the side respectively of each matrix. Any given factor (column) in one mode (matrix) corresponds to the same factor in the other two modes.

Up to ten columns may be listed across the page (unless DIMS has been used to alter this; see Chapter 3). If NFACT is greater than ten, loadings for the first ten factors are printed for all three modes, and then loadings for the remaining factors are listed.

You will use the sign and magnitude of the factor weights in the final (converged) output of each solution when interpreting the factors. More details are given below in Section 5.2.2.3.1 and in Section 6.5.

5.2.2.1 Starting Loadings (Iteration 0). - The first output for each solution consists of loadings which either have been supplied by the user or are random numbers generated by PARAFAC. These loadings represent the initial "guess" as to the form of the factors that can explain the data.

Randomly generated loadings generally do not predict the data well, and this is indicated by poor fit values: R and RSQ are near zero and STRESS is greater than one. In contrast, user-supplied loadings may predict the data better, especially if they are from a previous PARAFAC analysis. In this case, the fit measures are almost identical to those listed with the previous output (small discrepancies are due to roundoff of the loadings when they were output). DIFFA, DIFFB and DIFFC are always set to zero for the first output, since they are not defined until the iterative process begins.

If starting loadings in standard form were supplied by the user (ISTART=1 on Card I-5), their input can be verified here. (Recall that they were not printed with the Section III input check, described in Section 5.1.3.) They are identical to the final output of the PARAFAC analysis which produced them, unless ISTANM (Card I-8) has a different value than in the other job.

If the starting loadings were in nonstandard form (ISTART=2), then the output at iteration zero probably differs from the loadings listed with the Section III input check because of the standardization imposed by ISTANM. (The Section III input check involves no standardization, but unless ISTANM is 5, all other loadings outputs include standardization.)

While the loadings at iteration zero are only an initial guess and may be far from the final solution, it is sometimes helpful to know their form. Each analysis involves a different guess but (hopefully) all finally converge to the same solution. However, the starting loadings can influence the speed of convergence and whether the program reaches the true global optimum or instead gets "hung up" in a local optimum.

5.2.2.2 Intermediate Loadings. - After iteration zero, PARAFAC iteratively improves the factor loadings estimates. Every NITER (from Card I-2) iterations, the updated loadings are output. These subsequent outputs generally show improving fit values (i.e., increasing R and RSQ and decreasing MEAN SQ ERROR and STRESS) and decreasing "DIFF" values.

Intermediate loadings are sometimes useful for diagnostic purposes. Ideally, the rates of change of the fit and difference values are usually greatest for the initial iterations, and then they diminish as the solution is perfected and the correct rotation established during the final iterations. However, if there are convergence difficulties, the difference values may fluctuate substantially within or across modes, even though the fit values have stopped changing much. In this case, comparison of the intermediate loadings outputs may provide insight to the problem. For example, you may note that the estimated loadings for some factors become stable relatively early in the analysis process, while loadings for other factors continuously change with every iteration. This would indicate that some factors were not well determined, and that perhaps different starting positions and/or different dimensionalities should be tried.

5.2.2.3 Final Loadings. - The iterative reestimation procedure stops when one of the following three conditions is fulfilled (each is discussed in more detail below):

1. The convergence criterion is met. A message prior to the final loadings informs the user of this.

2. The solution has not met the convergence criterion, but the maximum number of iterations (i.e., NITER times NOUTS) has been performed. No message is issued, but the iteration number above the loadings always equals NITER times NOUTS.

3. The output is forced because of decreasing R (i.e., the iterative process has begun to decrease, rather than increase, the correlation between the actual data and data predicted from the loadings). A message prior to the final loadings informs the user of this.

Regardless of which of the above caused the iterative process to stop, the final loadings are written on a tape, disk or punched card file (see IUNITG on Card I-8) and on the listing.

5.2.2.3.1 Convergence Criterion Met. - Convergence of a solution means that reestimation of the loadings does not change them substantially from one iteration to the next. It does not necessarily mean that the solution is the global optimum one. You need to check certain aspects of it and compare it with other solutions before you know. First of all, look for a factor that has loadings of constant size and sign in two or three modes. This would suggest that you should reanalyse the data, centering on one or more modes.

Chapter 4 gives more details about when to center. Also, you should look at the factor cross-products (or correlations) to see how dependent the factors are, since highly dependent factors may indicate problems. This is explained in Section 6.1. Finally, compare the solution with solutions from 3 to 5 other starting positions. You may be able to accomplish this via a visual check, or you may have to use the CMPARE program (see Chapter 3).

Such comparisons may reveal it to be a local optimum (a solution with noticeably poorer fit values, that can be discarded), a competing solution (one of two solutions with similar fit values but different factors, that appear almost equally often; you may want to interpret both), or a stable unique solution (fit values and loadings identical for all solutions; probably the global optimum for this dimensionality). If your check shows that none of the factors are replicated across the different solutions, the analysis should be repeated with a smaller NFACT (Card I-2), since probably too many factors are being extracted. If the solutions replicate some factors but not others, however, there are two possible causes. The solutions may be selecting different subsets of dimensions underlying the data, and so the analysis should be repeated with a larger NFACT. Or, the solutions may be unable to uniquely determine some of the factors because of indistinct patterns of variation in the data; the factors may be highly correlated in this case. Reducing NFACT might help.

You will want to interpret a stable solution that represents the global optimum, so long as it's not a "degenerate" solution (see Section 6.2). The interpretive process involves both assigning descriptive labels to the factors in each mode according to the meaning of the levels at which the largest (positive and negative) loadings occur, and explaining how the factors in the different modes are related. Note that factor 1 in Mode A corresponds to factor 1 in Modes B and C, etc. Use of the PFPLOT program to get a graphical display, described in Chapter 3, is a help when interpreting the factors. In addition, where more than one solution is of interest, the CMPARE program (see Chapter 3) can be used to show how factors across the different solutions are related.

Other information supplied with the final loadings may give you a better understanding of your solution or may suggest possible problems. For example, the influence of each factor in the data is measured by its root mean squared contribution (Section 5.2.3). Cross-products and correlations measure the degree of similarity or "overlap" of the factors, and reveal problems such as "degenerate" solutions (Sections 5.2.4-5). Which parts of the data are well fit and which are poorly fit by the solution are indicated by the error analysis (Section 5.2.6). Harshman and DeSarbo (1984) present a good example of how to use both the factor loadings and the extra information when interpreting a PARAFAC solution.

5.2.2.3.2 No Convergence. - Usually when convergence is not met after NITER times NOUTS iterations, the analysis is continued by using the final loadings here as the starting position for a solution in a subsequent PARAFAC job. However, if the solution is very close to convergence and agrees with other converged solutions, it is not necessary to continue it. The solution is close to convergence when the DIFFA, DIFFB and DIFFC values for the final loadings are very close to the values specified for DIFMXA, DIFMXB and DIFMXC respectively on Card I-7. In addition, if the loadings and fit values are very similar to those of a solution which has converged from a different starting position, then it is reasonable to assume, without continuing it, that the unconverged solution is headed toward the same place.

Sometimes the solution may fail to converge because of nonunique factors, and not because of an insufficient number of iterations. If so, it may be a waste of time to continue the solution. Such nonuniqueness would be indicated by the factor cross-product and correlation tables. Refer to Section 6.1 for more details.

5.2.2.3.3 Forced Output. - Decreasing R (and hence forced output) often occurs when constraints are imposed on the factors, but even so, you can usually attain a good approximation of the constrained optimum solution. (Constrained solutions are discussed in Section 6.2.1.3.) When IRINTV is small, PARAFAC quickly detects the decrease in R, and thus the solution does not get very far beyond the optimum point. (To be very scrupulous, you could repeat the solution with IRINTV equal to 1, and the decline in R would be immediately detected.) Also, comparison of the fit values and loadings with those obtained from other starting positions may reveal strong similarities. Even when the other loadings are also the result of forced output, you can reasonably conclude that the constrained optimum has been obtained if 4 to 6 solutions (or 3 solutions in each of two split-half data sets) have more or less identical loadings. If the discrepancies between these solutions are not large enough to substantially affect the interpretation, you may take the solution with the highest fit value as the best approximation to the optimal solution. The "DIFF" values for these forced outputs may be relatively large, which indicates substantial changes in the loadings from one iteration to the next as the solution departs from the optimum one.

5.2.3 Root Mean Squared Contribution of Each Factor

A factor's contribution to an individual data point x(i,j,k) is the triple product of that factor's loadings at level i of Mode A, level j of Mode B and level k of Mode C; the factor's contribution to the average data point is the mean of its contributions to the individual data points. If the data have been centered, however, the average data value is zero and so is the average factor contribution to it; this leaves you with no real information about the factor contribution in the predicted data. To avoid this problem, the root mean squared (RMS) contribution is computed and used as a measure of the size of the factor contribution to the data.

Before they are printed, PARAFAC factors are ordered in descending size of their RMS contributions. The RMS contribution for each factor is listed below the Mode C loadings. Thus, factor 1 has the greatest influence in the predicted data, factor 2 has the second-largest influence, etc.

There are other ways to view the RMS factor contribution. It is easy to show algebraically that the RMS contribution of each factor is equal to the triple product of its RMS loading values in the three modes. With the standardization of the RMS loading value to 1.0 in two modes (when ISTANM equals 1, 2 or 3), the RMS factor contribution is in fact equal to the RMS loading in the mode that reflects the scale of the data.

When the data are centered across at least one mode (as is usually the case), other meanings can be attached to the RMS contributions:

1. The RMS contribution for each factor equals the standard deviation of its contributions to the data, and the squared RMS value is the variance of its contributions.

When the factors are mutually orthogonal (as indicated by cross-products that are all close to zero in at least one mode) in addition to the data being centered:

2. The squared RMS contribution approximates the variance in the (preprocessed) data that is accounted for by the factor.

3. The squared RMS contribution for any factor, divided by the total data variance (from the Summary Statistics Table; see 5.1.2.3), is the proportion of <u>total</u> variance accounted for by that factor.

4. The squared RMS contribution for any factor, divided by the sum of the squared RMS contributions for all the factors in the solution, is the proportion of explained variance accounted for by that factor.

5. The sum of the squared RMS contributions for all the factors in the solution is the variance accounted for by the solution (=RSQ for direct fit analyses).

Note that you cannot interpret the squared RMS contributions in terms of variance accounted for when the factors are not approximately mutually orthogonal, because they share too much overlapping variance.

If you are indirectly fitting the data (e.g., using covariances), see Section 6.7 for additional information.

5.2.4 Factor Cross-Products

Factor cross-products are computed from the final loadings and are printed in three NFACT by NFACT matrices, one matrix for each mode, after the final loadings output. Each matrix entry is the cross-product between loadings of the two factors (in the indicated mode) designated by the row and column position in the matrix. Each factor is standardized to unit length for computation of the cross-products. Appendix B, Equation 3a is the formula used; it is the same one that Harman (1960, p. 257) recommends.

The cross-products measure the dependence or similarity of pairs of factors within each mode, taking into account both the profile and elevation of the factor loadings. A cross-product with an absolute value of 0.6 or more indicates that the two factors are fairly dependent (large negative values simply mean the factor profiles and baselines are similar but opposite in sign), while a value near zero shows that they are more or less independent or orthogonal. Diagonal entries in each cross-product matrix are always 1.0, which is to be expected, since there is maximum similarity for any factor compared with itself.

Chapter 6 gives further details about using the cross-product output. For indirect fitting, see Section 6.7.

5.2.5 Factor Correlations

Like the cross-products, the factor intercorrelations are printed in three NFACT by NFACT matrices, one matrix for each mode. Each matrix entry is the Pearson product-moment correlation between loadings (in the indicated mode) of the two factors designated by its row and column position in the matrix. (See Appendix B, Equation 3b for the formula.) The correlation is essentially the cross-product computed for factors that have had the mean loading removed, and so the correlation matrix is the same as the cross-product matrix for any mode on which the data were centered (since the mean factor loading is zero for centered modes). Like the cross-products, a value near 1.0 indicates that the loadings for both exhibit a similar pattern, while a value near zero indicates that there is minimal similarity. Asterisks indicate an undefined correlation value, which occurs if one of the factors has constant loadings in the indicated mode.

See Chapter 6 for a discussion of how to use the correlations. For indirect fitting, see Section 6.7.

5.2.6 Error Analysis Table

An Error Analysis Table follows the correlation matrices at the end of the solution. The error analysis consists of the mean squared error (MSE) for each level of each mode (i.e., a mean squared error value is calculated for every possible two-way "slice" of the three-way data set). Appendix B contains the formula used.

For any mode that was normalized, you can compute the quantity (1-MSE) at each level to get the proportion of the mean squared data value at that level that was fit by the solution. Where either of the other two modes were centered, so that the level means of the normalized mode are all zero (the Summary Statistics Table confirms this), then (1-MSE) is the proportion of the variance accounted for at each level by the PARAFAC solution. For example, if the data were normalized on Mode A and centered on Mode C, then (1-MSE) for each level of Mode A is the proportion of the variance accounted for in that lateral slice (see Section 4.2) of the data.

How to use the MSE values for diagnostic .and interpretive purposes is explained in Chapter 6. For indirect fitting, see Section 6.7.

5.2.7 Missing Value Estimates

Whenever there are missing data, a table of the initial and final estimates for all the missing values is printed after the error analysis. The initial estimates are the values used for the first iteration, and the final estimates are the values obtained during the last iteration of the solution. You can check that when the estimates are reinitialized at the beginning of each solution (MISEST is 1 on Card I-3), the initial estimate for a particular missing data cell is the same for every solution in the PARAFAC run. Otherwise, the value listed as the final estimate for a given point in one solution appears in the following solution as the initial estimate for that point.

5.2.8 Diagonal Estimates

If diagonal estimation was requested (IGDIAG=1 on Card I-6), a table of initial and final diagonal values for the solution is listed next. Note that for PARAFAC version 6H the table is incorrect, since PARAFAC does not reinitialize the diagonal values at the beginning of each solution. What should be shown is that the final estimates for one solution are the initial estimates for the next solution.

The diagonal values are from the Mode A-by-Mode B matrix for each level of Mode C (i.e., the Mode C or frontal slices as shown in Section 4.2). Diagonal estimation is requested most often when the data is a set of covariance matrices. With such data, diagonal estimation is a way of dealing with the three-way analog of the communalities problem that occurs in two-way factor analysis.

5.2.9 Special Output

The revised data and/or residuals are listed if the special output parameters IUNITD and IUNITF respectively on Card I-8 have been specified as 6 (the standard output unit). This is not usually recommended, since these data are not available for subsequent analyses unless they are output to disk or tape files. The revised data are the data that have been centered and/or normalized, and/or with any missing values replaced by their final estimates; the residuals are the differences between the real data and the values predicted from the factor loadings on the final iteration.

The organization of both data listings is as described in Section II of the PARAFAC Input Specifications Table: title, dimensions and format of the data, data set, subscripts of any missing values, and the "-001" terminator code. Columns 1-40 of the title line are from Record II-1 (the title of the data being analysed); columns 41-80 indicate that the output is revised data or residuals, and give the solution number and the centering/normalization options used ("IFCEN" flags from Card I-3). The format is DATFMT from Card I-8A for the revised data and RSDFMT from Card I-8B for the residuals. When missing values were indicated in the original data, missing value subscripts are listed after the data. These subscripts match the documentary list of missing value points on the program listing, described in Section 5.1.2.1.

5.3 DISK OR TAPE FILE OUTPUT

In addition to the lineprinter listing generated by every PARAFAC job, output may be optionally directed to punched cards, or disk or tape files. This optional output occurs at the end of each solution, and may include the final loadings, the revised data and/or the residuals. (Note that the appropriate system control cards to access and save the files must be included with the PARAFAC job.)

5.3.1 Final Loadings

The final loadings are always written to a file, unless IUNITG on Card I-8 is set to -1 to suppress this output. NSOLS complete sets of loadings are written on IUNITG during a PARAFAC job. The descriptive information and the loadings in the file are identical to the final loadings output(s) on the listing, unless NFACT is greater than 6. If NFACT is 7 or more, the format of the file is as described for a "standard continuation" in Section III of the PARAFAC Input Specifications Table (Chapter 2).

This output has several important uses. The final loadings from one PARAFAC analysis may be input as the starting loadings for a subsequent one. This is useful where many iterations are required because the solution(s) is(are) slow to converge. Note that NSOLS in the continuation analysis must not exceed the number of loadings sets supplied.

The loadings may also be input to the PFPLOT and CMPARE programs (described in Chapter 3). Generally, PFPLOT is used to plot factors from converged solutions only; the plots aid in interpretation of the solution(s) obtained. CMPARE indicates the similarity amongst factors of <u>different</u> solutions just as the cross-product and correlation tables on the listing show the relationship amongst factors of the <u>same</u> solution.

And finally, another use for these loadings is as input for other types of computer analysis or transformations.

5.3.2 Revised Data

As mentioned in Section 5.2.9, the revised data are the data that have been centered and/or normalized and/or with missing values replaced by the solution's final estimates. They are output if a unit number is specified for IUNITD on Card I-8. At the end of the run, the file contains NSOLS sets of data except when analysis is suppressed by setting NSOLS to -1; then only *one* set of data with *initial* estimates for any missing data values is written. (When the file contains several data sets, they have to be separated before sets 2 through NSOLS can be analysed by PARAFAC, since it reads only one data set in each run.) The format of each data set is as described in Section 5.2.9.

The revised data may be used in several ways, depending on the transformations involved. Especially if the data set is large and/or many PARAFAC analyses are to be performed which involve preprocessing (i.e., centering and/or normalizing) the data, you may want to save the revised data. Subsequent analysis of the revised data reduces computation time by eliminating the preprocessing step in each analysis. Alternatively, PARAFAC can be used only to preprocess the data, so that the revised data are in better form for some other analysis procedure.

Where there are missing data values, the revised data can be output after a PARAFAC analysis; this data set contains sophisticated missing data estimates based on the PARAFAC solution. Such data may then be analysed by some other procedure which does not have missing data estimation capabilities of its own, or it may be used in a subsequent PARAFAC analysis to provide good initial estimates of the missing data.

Note that the revised data can both be centered and have missing values estimated. However, when such data are analysed in a subsequent job, the "IFCEN" flags (Card I-4) for that job must specify centering if PARAFAC is to recenter the data after every NITER (Card I-2) iterations and before each new solution (as it usually does when both centering and missing values are involved); otherwise, the recentering will not be done. (Of course, the initial centering done before the first solution in the subsequent job is redundant in such a case.)

5.3.3 Residuals

As noted in Section 5.2.9, the residuals are the differences between the data being analysed and the data predicted on the final iteration of the solution. They are output only if a unit number is specified for IUNITF on Card I-8. At the end of the run, the residuals file contains NSOLS sets of data, one for each solution obtained. (These data sets have to be separated before sets 2 through NSOLS can be analysed by PARAFAC, since it reads only one data set in each run.) The format of this output is as described in Section 5.2.9.

There are several applications for the residuals. One is to input them to a cluster analysis program to look at the "nonspatial" structure left after the factors have been taken out. Kettenring (1983) found meaningful patterns that shed light on the data structure when he used graphical methods, displaying the residuals in a normal probability plot and in box plots. Yet another way of dealing with residuals is to input them to other programs that test for normality of distribution, outliers, etc.

Another approach is a "hierarchical" type of PARAFAC analysis. This involves an initial PARAFAC analysis of the raw data to extract two or three "major" factors, and then a subsequent PARAFAC analysis of the residuals from the first run to extract additional "secondary" factors. This procedure may allow exploration of weak factors without the "splitting" of major ones, which sometimes occurs when too many factors are extracted at the first level of analysis. For two-way data, such a procedure always yields "secondary" factors that are orthogonal to the "major" ones. However, we have found that the "major" and "secondary" factors in three-way data are orthogonal only when the underlying dimensions are orthogonal in at least two modes and there is little or no error in the data.

CHAPTER 6

DIAGNOSTICS AND INTERPRETATION

While Chapter 5 was devoted mostly to a description of the PARAFAC output, this chapter explains how to use parts of that output for diagnostic and interpretive purposes. Diagnostic indicators are covered in Sections 6.1 and 6.4; identifying and dealing with "degenerate" solutions is described in Section 6.2. Section 6.3 explains how the cross-products and correlations can also be used during the interpretive phase of the analysis. Interpretive issues regarding the PARAFAC factor loadings and factor scores are dealt with in Sections 6.5-8, with special reference to analysis of covariances (Section 6.7) and multidimensional scaling (Section 6.8).

6.1 CROSS-PRODUCT AND CORRELATION DIAGNOSTICS

The cross-products and the correlations serve an equivalent function as diagnostic aids; as such, they can be used to quickly identify a solution that has problems. They both measure the similarity of pairs of factors; the cross-products take into account the factor profiles _and_ elevations, however, while the correlations use only the factor profiles. As noted in Chapter 5, they are identical for any mode on which the data were centered.

You can use the cross-products and correlations in virtually the same way, although you should keep the following point in mind for any modes that were not centered. While the cross-products incorporate more information about the factors than do the correlations, and thus in some sense may be a better measure of factor dependence, they sometimes may be artificially large or small because of certain combinations of factor elevations. For example, if both factors have relatively large positive baselines, as may occur in an uncentered "person" mode, the cross-product may be large even when the factor profiles are not much alike. In such a case, the cross-product gives an inflated measure of the factors' similarity. On the other hand, the baselines may sometimes "cancel each other out" so that the cross-product is small, even though the profiles are similar. Thus, you may prefer to use the correlations. We refer to "correlations" in the following discussion, but you can usually substitute "cross-products" if you wish.

Look for high correlations between factors. This means that the factors are highly dependent, and warns of possible problems. Such dependence can cause PARAFAC trouble in defining a unique set of loadings for each factor, and so the solution may not be unique or it may be very slow to converge. Or, it may "converge" before it has actually reached the global optimum (i.e., it may meet the current convergence criterion, but if it were continued with a more stringent criterion, the loadings would change considerably before it met the new one). Highly dependent factors in only one mode do not usually have such a serious effect on convergence, but may still make it difficult to get a unique solution.

When a solution with highly dependent factors is obtained, compare it with solutions obtained from several other starting positions (see CMPARE program in Chapter 3). The comparison may show one of several things:

1. While the current solution has highly dependent factors and perhaps poorer fit values, (most of) the others obtained factors that are more independent and that fit the data better. If so, the poorly behaved solution may be discarded as a local optimum that had an unlucky starting position.

2. All solutions generally agree, and so a global optimum may have been approximated. (You could further refine the solution by continuing it with smaller values specified for the convergence criteria.) However, even if the solution is a global optimum, it will be virtually uninterpretable if two or more factors are very similar in all three modes. We call such a solution "degenerate", and discuss how you can deal with it in the following section.

3. All the the solutions are different. Either they all were changing so slowly that they met the convergence criteria but in fact were not close to the global optimum, or too many factors were being extracted. Depending on the cause, you can either continue the solutions with more stringent convergence criteria or reanalyse the data at a lower dimensionality (see NFACT on Card I-2).
Another possible cause for many different solutions is that some factors do not have distinct patterns of variation across levels of one mode, and so PARAFAC cannot determine a unique orientation for

them. In such cases, changing the convergence criteria will not help and neither will reducing NFACT, with one exception. If the nonunique factors account for the smallest amount of variance, reducing NFACT by 1 or 2 might eliminate them from the solution. The effect at the lower dimensionality would then be a stable solution consisting of the larger, unique factors that remain.

The above discussion is somewhat brief, but it should give some guidance in using the cross-product and correlation output to check the solution obtained.

6.2 "DEGENERATE" SOLUTIONS

Degenerate solutions usually occur because the structure underlying the data is more complex than can be represented by the PARAFAC model (Harshman and Lundy, 1984a, p.271-80). Such structure can often be fit by the Tucker T2 or T3 models (Kroonenberg and DeLeeuw, 1980), which are more general models for three-way data than PARAFAC is. Characteristics of degenerate solutions are as follows:

1. At least two factors are highly correlated (e.g., with absolute values greater than 0.7) in three modes <u>and</u> either all three correlations are negative or only one is (so their triple product is negative). In degenerate solutions, the triple product is always negative. This contrasts with the situation where high correlations occur because too many factors are being extracted. Then, the triple product is positive (i.e., three positive correlations or two negative) as often as it is negative.

2. The same solution (with equal fit values and more or less identical factors) is consistently obtained, regardless of starting position.

3. Often, the highly correlated factors appear at low dimensionalities (e.g., when NFACT is 2 or 3), and they fit nontrivial amounts of variance. For example, if the factors are highly correlated when NFACT=2, you will often see that the RSQ fit to the data has improved by 5-10%, compared to the fit when NFACT=1. This contrasts with the situation where high correlations are because too many factors are being extracted. There, the improvement in fit from the previous dimensionality is very small, as the additional factor fits redundant or overlapping variance.

4. Probably the correlated factors are not interpretable.

6.2.1 Overcoming Degeneracies

If the solution is truly degenerate, the best way to get an interpretable PARAFAC solution is to constrain factors to be independent in one mode throughout the course of the analysis (see Section 6.2.1.3). Sometimes, though, improved preprocessing of the data or temporary independence constraints will also improve the solution.

6.2.1.1 Preprocessing. - Sometimes the degeneracy may be due in part to additive constants in the data. Factors that are more or less constant in a particular mode would be evidence of this. Usually, such a problem can be overcome by centering the data appropriately (see Chapter 4) and then repeating the analysis. Harshman and DeSarbo (1984) discuss data for which appropriate preprocessing overcame a previously uninterpretable solution.

6.2.1.2 Temporary Constraints. - Sometimes you can try temporarily constraining the factors to be orthogonal or uncorrelated in a particular mode (e.g., IORTHA=2 or 3 for temporarily uncorrelated factors in Mode A; IORTHA=5 or 6 for orthogonal factors; see Card I-6). This constraint forces the factors to be independent at first, and may overcome the apparent degeneracy if somehow the unconstrained starting position "trapped" the factors at a local optimum were they were highly correlated. If the solution is truly degenerate, however, the factors will almost immediately revert to their highly correlated form once the constraints are dropped.

6.2.1.3 Permanent Constraints. - For truly degenerate solutions, we have found that constraining the factors in one mode to remain uncorrelated until the solution converges (e.g., IORTHA=4) allows PARAFAC to discover a replicable, interpretable solution. (Orthogonality constraints can sometimes cause problems when imposed in a mode for which the data were not centered, e.g., the "person" mode.) Such "constrained solutions" have poorer fit values than degenerate solutions at the same dimensionality, further evidence that the highly correlated factors are fitting nonoverlapping variance in the data. The important thing is, however, that the factors are meaningful and can consistently be obtained from different starting positions.

Permanent constraints generally should be imposed on only one mode at a time. Such constraints on more than one mode push the solution towards a principal components axis orientation, which may be unnatural for the data. When deciding which mode to constrain, you should consider the meaning of the modes. If there is a theoretical argument for only weakly correlated factors in a particular mode, you may want to start by imposing the constraints on it. You may have to do several analyses, with the constraints on a different mode in each one, to see which gives the best result. This should not be very expensive, since a constrained analysis usually terminates quickly; see Section 5.2.2.3.3. For three-way rating scale data, we have often constrained the "scales" mode with success, but sometimes constraining another mode may also work.

6.3 FACTOR CONTRIBUTIONS (DIRECT FITTING ONLY)

Each data point can be thought to consist of the sum of NFACT factor contributions (plus error). Each factor contribution is the amount that the given factor increases or decreases the predicted score for that data point. In terms of the three-way data set, you may be concerned with the relationship of the factor contributions in rows (or columns or "tubes") of the array, in "slices" of the array, or in the the array as a whole. (See Chapter 4.1-2 for a description of "rows", etc. and "slices".) You can use the factor cross-products and correlations provided by PARAFAC to tell you something about the contribution patterns.

6.3.1 Cross-products

You will always use the cross-products for diagnostic purposes. After you have found a stable, nondegenerate solution, however, you will want to interpret it. You may then be interested in using the cross-products to compare patterns of factor contribution in the data.

As described in Section 5.2.4, the cross-products measure the similarity of the loadings patterns for pairs of factors, but they also tell us about the relationship of the "factor contributions". It happens that the cross-product computed from factor loadings in a given mode is equal to the cross-product of factor contributions to the data within that mode, for fixed levels of the other two modes. For example, the Mode A cross-product for any two factors r and s indicates both the similarity in their Mode A loadings patterns <u>and</u> the similarity in the patterns of their contributions to any row of the predicted data. Similarly, their relationships in any column and any "tube" are measured by their Mode B and Mode C cross-products, respectively.

The cross-products provided by PARAFAC can also be used to compute the cross-product of factor contributions within a slice of the data: multiply together the cross-products in the two modes comprising the slice. For example, you might be interested in the contributions of factors 1 and 2 to the data in any Mode C slice (all levels of Modes A and B at a fixed level of Mode C); you would multiply together the Mode A and B cross-products for factors 1 and 2, as provided on the PARAFAC output. Similarly, you would multiply together their Mode A and C cross-products to get the cross-product of their contributions in any Mode B slice, and you would multiply their Mode B and C cross-products to get the cross-product for any Mode A slice. Normally, you will only be interested in slicewise relationships of the factors if you've adopted the perspective that factor scores vary (see Section 6.6.2).

Finally, the cross-product of contributions for any two factors r and s over the entire predicted data is simply the triple product of their Modes A, B and C cross-products output by PARAFAC. Thus, the relationship of factor contributions in the data as a whole can be examined.

6.3.2 Correlations

In some circumstances, the factor correlations provided by PARAFAC may also be interpreted as correlations in the patterns of factor contributions to the predicted data. In the above discussion of factor contributions in rows, columns and tubes of the data, the word "cross-product" can be replaced by "correlation" for every mode that was originally centered. The PARAFAC factor correlations for uncentered modes cannot be interpreted as correlations between factor contributions, however.

Factor contribution correlations in a slice of the data are equal to the product of the cross-products as described above, so long as at least one of the two modes comprising the slice has been centered. Multiplying two PARAFAC correlations together will give you the slicewise correlation only if the data were centered on both modes. (In this case, correlations and cross-products are equal in both modes.) If neither mode was centered, you cannot compute the correlation of the factor contributions from the information at hand. As with the cross-products, you will usually only consider slicewise relationships if you assume that the factor scores change (Section 6.6.2).

6.4 ERROR ANALYSIS

Variations in fit across levels, which are revealed by the Error Analysis Table, sometimes may point to problems in the data or may help in interpreting the solution. A few very high or very low MSE values relative to the others, systematic patterns in larger versus smaller MSE values, or substantial variations in the MSE values within a single mode should all be investigated further. Of course, differences in variance across levels in the input data, as shown by the Summary Statistics Table, must be taken into account. Levels with higher variance on input usually have higher MSE values, but unexpectedly large MSE values, even considering input variances, indicate possible problems. Comparison of the MSE values is more straightforward if the variances are standardized and equalized before analysing the data. See Chapter 4 for a discussion of when such variance standardization ("normalization") is appropriate.

Suppose that each mode has one very high MSE value. Then the data point at the intersection of the levels should be checked to see if it is an extreme outlier, due perhaps to a keypunching error. Use of the PARAFAC option to check for points outside a specified range and to treat such data as missing during the analysis (see IFCODE, DLOWR and DUPPER parameters on Card I-4) is recommended to avoid such problems.

Patterns of large and small MSE values indicate that certain parts of the data cannot be predicted as well as others. For instance, large MSE values at some levels of the "stimulus" mode or for some levels of the "person" mode suggest "difficult" types of stimuli or unreliable subjects, respectively. You might want to refer back to the raw data to check these particular stimuli or subjects.

The analysis of tongue shapes (Harshman, Ladefoged and Goldstein, 1977) is an example of a useful interpretation of the variations in fit across levels indicated by the error analysis table. However, note that the values reported in the article are the MSE values divided by the number of levels in the mode, so that their sum equals the total MSE of the data; the current version of PARAFAC computes a simple MSE for each level.

6.5 MEANING OF PARAFAC FACTOR WEIGHTS

Two-way factor analysis has followed a convention whereby the factor weights in one mode reflect the scale of the data and hence are the "absolute" size of factor contributions, while the factor weights in the other mode are standardized. The factor weights that reflect the scale of the data are called "factor loadings", and the standardized factor weights are called "estimated factor scores" or "component scores" (depending on whether common or component factor analysis was performed). These factor scores are usually z-scores, since the data analysed are usually correlations. When the data consist of variables measured over people, for example, the variable weights are usually made to reflect the scale of the data (and hence are called factor loadings) and the person weights are standardized (and are called factor scores). Sometimes additional meanings are attached to the factor loadings. For example, they are often interpreted as beta weights. Or, if the factors are orthogonal, the loadings are viewed as variable-factor correlations, and the sum of the squared loadings for any given variable is the proportion of variance predicted by the factors (i.e., the "communality").

The standardization of factor weights that we use for PARAFAC analysis of three-way profile data is an extension of the two-way case described above: the root mean squared (RMS) weight for each factor in each of two modes is set to 1.0 and the weights in the other mode are rescaled in a compensatory way to retain the scale of the data. For covariance-like data (i.e., data which are symmetric across two modes), the factors in the two symmetric modes jointly reflect the scale of the data and the RMS loading value in the other mode is set to 1.0. The user can control which mode reflects the scale of the data and which modes are standardized by specifying a value for the ISTANM parameter on Card I-8; the default value of 3 causes the factor loadings in Modes A and B to be standardized while the Mode C loadings reflect the data scale. When ISTANM equals 4, the standardization described for covariance-like data is done.

Loosely speaking, the standardized PARAFAC factor weights (Mode A and B loadings when ISTANM=3) are always analogous to the factor scores in two-way factor analysis; similarly, the PARAFAC factor weights that reflect the data scale (Mode C loadings when ISTANM=3) are analogous to the two-way factor loadings. Strictly speaking, however, the standardized PARAFAC weights are truly equivalent to factor scores obtained from two-way analysis of correlations only under certain conditions where the PARAFAC loadings are also z-scores (e.g., direct fitting cases 5 and 6, and indirect fitting of covariances, discussed below). The three-way nature of the data complicates the interpretation of PARAFAC factor scores, however. The two different perspectives that are possible are discussed in Section 6.6 below.

We don't normally distinguish between PARAFAC factor "loadings" and "scores" elsewhere in this manual. Rather, we refer to all the factor weights as "factor loadings", regardless of whether they have been standardized or whether they reflect the scale of the data. To specify the loadings in a particular mode, we say "variable" loadings, "person" loadings, etc., depending on the meaning of the mode.

Several things should be kept in mind when examining PARAFAC factor loadings. The ones that reflect the scale of the data can be interpreted in the same units as the preprocessed data. Unless the preprocessing involved normalization of variances (see Chapter 4), these units will be the same as for the raw data. For example, if the data are measurements expressed in centimeters, these factor weights may be interpreted as centimeters (e.g., see Harshman, Ladefoged and Goldstein, 1977). Furthermore, these weights may be directly compared across factors and the differences expressed in the same units as the weights themselves. In contrast, the standardized weights in the other modes, and the differences between these weights, are expressed in relative units (e.g., z-score units for modes on which the data were centered).

You can always use the pattern of PARAFAC loadings to determine the meaning of the factor. In addition, you may sometimes want to assign special interpretations to the PARAFAC loadings themselves, similar to the ones applied in two-way analysis (e.g., factor-variable correlations, or variance accounted for). This is possible, given certain combinations of data preprocessing and loadings standardization that are described in the following section.

6.5.1 Special Interpretations Of PARAFAC Factor Weights

Listed below are special interpretations of PARAFAC loadings that are possible when the data and/or factor loadings meet certain conditions. We recommend that you skip this section if you are just beginning to use PARAFAC. Even if you are familiar with PARAFAC and with two-way factor analysis, you may find it easier to follow if you relate it to a specific data set as you read it. Note that some of the special interpretations are more useful for some types of data than others. For example, the idea of "size of factor contributions in a particular slice of the data" (case 2 below) may mean more if the data are ratio scale measurements such as centimeters displacement (Harshman, Ladefoged and Goldstein, 1977) rather than rating scale responses.

Direct fitting (analysis of raw or preprocessed profile data) and indirect fitting (analysis of covariances or cross-products) are considered separately in this section. The cases are ordered within these two categories from fewest conditions and most general interpretation to most restrictive conditions and most specific interpretation. In general, the meanings described are cumulative (i.e., they also apply to all successive cases in the list). The conditions are stated in terms of various input parameters

-- the IFCEN- parameters on Card I-3, the IORTH- parameters on Card I-6 and ISTANM on Card I-8 -- that have specific values, and then special interpretations are assigned to the Mode C loadings. See the note after case 6 if you are interested in interpreting the Mode A or B loadings instead.

First, let us explain two terms that are used below: (a) the expression "c(k,r)" is used to denote the row k, column r entry of the Mode C factor loading matrix that is output by PARAFAC, or, in other words, the factor r loading at level k of Mode C; and (b) "factor contributions in the data at level k of Mode C" means the contributions in the kth "Mode C slice" or "frontal slice" of the data array, as illustrated in Chapter 4.2; "factor contribution" is defined in Section 6.3 above.

<u>Category I: Direct Fitting, NFACT Factors Extracted</u>

1. When ISTANM=5

Loadings in all modes = nonstandardized regression weights (B weights) in a multiple regression equation

****

2. When ISTANM=3

(a) Each Mode C loading (i.e., each c(k,r)) = root mean square (RMS) size of factor r contributions in kth Mode C slice of the data;
(b) Squared Mode C loading = mean squared size of factor r contributions in kth Mode C slice;
(c) Mode A and B loadings same as 1 above.

****

3. When ISTANM=3
<u>and</u> IORTHA=7 or IORTHB=7 (or factors are naturally orthogonal)

(a) Interpretations 2a and 2b above apply;
(b) Squared loadings summed across row k of Mode C factor matrix = mean squared value in the kth Mode C slice of the data that is due to NFACT factors;
(c) Mean squared loading computed down column r of Mode C factor matrix (= squared RMS factor contribution; see Section 5.2.3) = mean squared value of factor r contributions in the total data set;
(d) Mode A and B loadings same as 1 above.

****

4. When ISTANM=3
and IFCENA=1 (or 3) and/or IFCENB=1 (or 3)
and IORTHA=7 or IORTHB=7 (or factors are naturally orthogonal)

(a) Absolute value of each Mode C loading = standard deviation of factor r contributions in kth Mode C slice of the data;
(b) Squared Mode C loading = variance of factor r contributions in kth Mode C slice of the data;
(c) Squared loadings summed across row k of Mode C factor matrix = variance accounted for (VAF) in kth Mode C slice by NFACT factors (i.e., the "communality");
(d) Mean squared loading down column r of Mode C factor matrix = VAF by factor r in the total data;
(e) Mode A and B loadings same as 1 above.

****

5. When ISTANM=3
and IFCENA=1 (or 3) and/or IFCENB=1 (or 3)
and IFCENC=2 (or 3)

(a) Mode C loadings = beta weights (i.e., standardized regression weights);
(b) Mode C loadings equivalent to traditional factor "loadings" obtained via 2-way factor analysis of correlations;
(c) Interpretations 3a-3c and 4a-4d above apply only if all factors happen to be approximately mutually orthogonal (i.e., cross-products close to zero) in either Mode A or B;
(d) Mode A and B loadings same as 1 above;
(e) Mode A loadings equivalent to z-score factor scores in 2-way analysis of correlations, if data were centered on Mode A (IFCENA=1 or 3);
(f) Mode B loadings equivalent to z-score factor scores if data were centered on Mode B (IFCENB=1 or 3).

****

6. When ISTANM=3
and IFCENA=1 (or 3) and/or IFCENB=1 (or 3)
and IFCENC=2 (or 3)
and IORTHA=7 or IORTHB=7 (or factors are naturally orthogonal)

(a) Each Mode C loading = simple product-moment correlation between the data values in the kth Mode C slice of the data array and the contributions of factor r in the same slice (e.g., factor-variable correlation, if levels of Mode C are variables);
(b) Interpretations 4a-4d inclusive apply, except you can replace "variance" with "proportion of variance" in the

interpretation;
(c) Mode C loadings equivalent to traditional factor "loadings" in 2-way factor analysis of correlations;
(d) Mode A and B loadings same as 1 above;
(e) Mode A and B loadings same as 5e and 5f above.

****

NOTE

Under the conditions listed above, the Mode C loadings are assigned special interpretations; this may be desirable if Mode C is the "variables" mode, for example. Putting the constraints on different combinations of modes permits special interpretation of the Mode A or the Mode B loadings instead. For example, if the constraints are changed so that the Mode A loadings reflect the scale of the data (ISTANM=1), the factors in Mode B or C are orthogonal (IORTHB=7 or IORTHC=7), and the data are centered on Modes B and/or C (IFCENB=1 (or 3) and/or IFCENC=1 (or 3)), then the Mode A loadings have the special interpretations that are described for Mode C in case 4 above.

Category II: Indirect Fitting, NFACT Factors Extracted

Preprocessing/Standardization:

Average covariance matrix is correlation matrix, mean Mode C loading on each factor is 1.0 and Mode A and B loadings jointly reflect the data scale (IFCENA=IFCENB=4; ISTANM=4)
N.B. Currently, the root mean square Mode C loading on each factor is standardized to 1.0 when ISTANM=4, and so the interpretations below do not hold. However, the loadings are proportional to what they would be if the mean loading were set to 1.0; thus the pattern of loadings can be used to interpret the factors as usual.
You can do simple calculations to rescale the PARAFAC loadings so that the interpretations below are valid. Deal with each factor separately. First, compute the mean Mode C loading for the factor; then, divide the Mode C loadings by this mean value and multiply the Mode A and B loadings by the square root of this mean.

Special Interpretations:

(a) Mode A (= Mode B) loadings = same as Mode C loadings in case 6 above (i.e., interpretations 6a-6c apply);
(b) Each mode C loading = factor score variance at the kth level of Mode C in the raw (not covariance) data array.

6.6 FACTOR SCORES (DIRECT FITTING ONLY)

With some data sets, you may wish to interpret the standardized PARAFAC loadings the way factor scores from analysis of two-way data are interpreted. As mentioned previously in this chapter, factor scores obtained from two-way factor analysis are standardized factor weights. If the data analysed are correlations (as is usually the case), the factor scores are also centered, and so they are z-scores. (If the data were fit directly, without being centered across persons, the factor scores would not have zero means but would have mean squares equal to 1.0.) In this context, the term "factor score" implies certain mathematical properties about the factor weights. However, there is a stronger empirical sense in which the term is used when intepreting certain data sets: a "factor score" is a measure of the amount of a factor (e.g., a personality trait) that is possessed by an entity (e.g., a person). "Correlation between factors" is more precisely stated as "correlation between factor scores"; thus factor correlations are defined relative to the "entity" (person) mode.

These empirical definitions of factor score and factor correlation are reasonably explicit. They can be extended to PARAFAC analysis of three-way data, but there are some complications for the direct fit case because of the additional mode in the data. You will recall that PARAFAC yields two sets of standardized loadings, for example, the Mode A and B factor loadings when ISTANM=3. Even though both possess the mathematical characteristics of factor scores, we do not simultaneously interpret both as factor scores according to the definition given above. Rather, we take one mode (e.g., people) to represent the entities for which there are factor scores; the factor scores can be viewed either as remaining constant or as changing over levels of the other mode (e.g., occasions). Herein lies the complication, because the actual factor scores .and correlations between the factors are defined differently, depending on which of the two perspectives is adopted. This complication does not occur for indirect PARAFAC1 fit of three-way data, since the model assumes factor orthogonality in the mode (usually the person mode) over which covariances are computed. (Section 6.7 has more details.)

Before discussing the two perspectives for factor scores in three-way data, let us remind you of several things. First, regardless of the factor score perspective taken, the assumption is that mostly system variation is present in the data. (Object variation is dealt with by indirect fitting, as discussed in Sections 4.4 and 6.7.) Also, remember that factor scores are standardized factor weights. Thus, the PARAFAC factor weights that you interpret as factor scores should not reflect the scale of the data. If there is a "variable" mode, it would most often be appropriate to have the factor loadings in that mode absorb the scale of your data. If there is a "person" mode, you will usually want to standardize its factor loadings. However, be aware that if you have followed our advice in Chapter 4 and have not centered the data on the person mode, then these person loadings are not z-scores. Generally, though, this will not be a problem.

6.6.1 Fixed Factor Scores

One perspective holds that factor scores are invariant. Over levels of the additional mode (e.g., occasions), the amount of the factor that is possessed by any entity (person) remains constant, but the amount that is exhibited varies; the factor scores reflect the amount of the factor that the different entities have.

This view of factor scores is appropriate for an experimental study, for example, where the test conditions are manipulated differently for each occasion, or for rating scale data. The factor scores for the experimental study are the "person" loadings as output by PARAFAC. For rating scale data, you might regard either the stimuli or the people as the entities that manifest or are sensitive to the factors. Thus, either the "stimulus" loadings or the "person" loadings could be regarded as factor scores, depending on which approach you want to take. In any case, the amount of the factor that the person (or stimulus) manifests on a particular occasion is given by the product of the person (or stimulus) weight and the occasion weight.

Factor (score) correlations are then obtained directly from the PARAFAC output: they are the factor correlations given for the mode that corresponds to the factor scores. For the experimental study discussed above, for instance, correlations in the person mode would be the ones you'd want. Or, if you choose to regard the stimulus loadings as factor scores in rating scale data, you would refer to the factor correlations in the stimulus mode. The correlations in the other modes have their usual interpretation -- they indicate the similarity or covariation of factor loading patterns and factor contribution patterns in those modes. You could also compute the "slicewise" correlations (see

Section 6.3.2), but most often you will not be concerned with them.

6.6.2 Varying Factor Scores

The other perspective holds that factor scores change. Over levels of the additional mode (e.g., occasions), the amount of the factor that is possessed by the entities changes; the factor scores reflect this change.

This view of factor scores may be taken in a developmental study, where it is reasonable to assume that the amount of a factor (e.g. responsibility trait) that a person has changes over time. In this case, the PARAFAC person loadings are not the actual factor scores, although they do indicate who has relatively more or less of the factor in general. To get the actual factor scores, you must compute the product of the "person" and "occasion" loadings (where both are standardized). This means that the factor score for a particular person (on a given occasion) is the product of his/her loading <u>and</u> the occasion loading. Each person thus has multiple scores for each factor, a different one for each occasion (Harshman and Lundy, 1984b, p. 131). This contrasts with the other perspective of factor scores where each person has only <u>one</u> score per factor. You probably will not compute all the factor scores unless you want to compare them for subsets of people or occasions, as Haan (1981) did.

Factor (score) correlations in this case are not given directly by the PARAFAC output, but they can be obtained via simple computations. Take the matrices of factor <u>CROSS-products</u> for the two modes you used when calculating the factor scores (e.g., person and occasion modes in the above example) and multiply corresponding entries together to get an NFACT by NFACT matrix of double products. As long as the data were centered on <u>at least one</u> of the two modes, the double products are the correlations between the factors; otherwise, they are cross-products. This method of computing correlations is the same as the one described in Section 6.3.2 for finding slicewise factor contribution correlations. Only the values computed for the person by occasion slice can be interpreted as correlations between the factors, however. Note that you can still use the PARAFAC cross-products and correlations to tell you about factor similarities in individual modes, even when you adopt this perspective of factor scores.

6.7 INDIRECT FITTING

Indirect fitting, with specific reference to covariances, was discussed in some detail in Sections 4.4-5; also see Harshman and Lundy (1984b, p.133-43; 202-3). It was noted that indirect fitting is done to cope with data that contains mostly object variation (e.g., Harshman and Berenbaum, 1981) or that has too many levels in one mode (e.g., people) for analysis of the raw data to be practical. A few remarks will be made here about PARAFAC output from analysis of covariances. The factor loadings, factor correlations, error analysis, etc. that are listed pertain to the covariances, but usually, you also want to know about the raw data from which the covariances were computed. We explain below how you can relate some of the output to the raw data; see also Category II in Section 6.5.1 above.

6.7.1 Factor Scores

As with indirect fitting of two-way data, you can use regression techniques to estimate factor scores for the people (i.e., for the mode that "disappeared" when covariances were computed). First, compute the generalized inverse of the factor loading matrix (i.e., the matrix of factor weights that reflect the data scale, e.g., the variable loadings) to obtain factor score coefficients. Then apply the coefficients to the uncentered raw data, separately on each occasion, so that several sets of uncentered factor scores are computed for each person. These uncentered factor scores can then be used to look at changes in factor score means (Harshman and Berenbaum, 1981).

6.7.2 Root Mean Squared Factor Contributions

You can assume that the raw data are centered (the means are removed when covariances are computed, if not before) and that the factors in the raw data are orthogonal (a basic assumption for fitting covariances via PARAFAC1). Then it is appropriate to apply to the raw (centered) data the five special meanings that are listed in Section 5.2.3, using the RMS factor contributions from the covariance analysis. For point 3, however, do not divide by the total variance. Point 5 is especially important, because it explains how to compute the variance accounted for (for direct fitting, see the RSQ value at the top of the loadings output). Harshman and Lundy (1984b, p.202-3) discuss these issues in more detail.

6.7.3 Factor Correlations

In order to fit the PARAFAC1 model to the covariances, the assumption is made that the factors are orthogonal in the mode over which the covariances were computed (usually people). This is the mode for which factor scores are estimated. Thus, in accordance with the definition given in Section 6.6 for the correlation between factors, you can say that the factors in the raw data are uncorrelated.

You also know about the factor relationships in the individual modes of the raw data. By assumption, the factors in the person mode are unrelated; hence, the interfactor correlations and cross-products for this mode would be represented by identity matrices. The PARAFAC correlations and cross-products for Mode A (and B) of the covariances also hold for the corresponding mode in the raw (centered) data. The Mode C correlations and cross-products do not directly apply to the corresponding mode in the raw data, however; they are derived from squared loadings (i.e., the Mode C loadings from the covariance analysis are the squares of the weights that would be obtained by directly fitting the raw data).

6.7.4 Error Analysis

The MSE values printed on the PARAFAC listing refer to the covariances and cannot be used to compute, for example, the proportion of variance accounted for in the raw data. However, you can use them as relative indicators of the goodness of fit in the raw data: the larger the MSE value is, the poorer the fit is for the corresponding level of the raw data, and the smaller the MSE value, the better the fit.

6.8 MULTIDIMENSIONAL SCALING

As mentioned in Section 4.6, PARAFAC can be used to indirectly fit the weighted Euclidean distance model to distance-like data (e.g., pairwise dissimilarity ratings). This is accomplished by computing scalar products from the raw data and then analysing the scalar products via PARAFAC. The theoretical basis for this is mentioned in Harshman (1972) and Harshman and Lundy (1984b, p.144-7).

Using PARAFAC to do multidimensional scaling (MDS) is like using it to analyse covariances, in that both are ways of indirectly fitting structure in the raw data. However, there are differences in the procedure. One difference is that the scalar products are usually normalized on Mode C (IFCENC=2); equal-average-diagonal normalization is not done. Another is that the Mode C factor weights reflect the scale of the data (ISTANM=3, not ISTANM=4). A third difference is that analysts are usually concerned with the fit to the scalar products and not to the raw data; hence, Section 6.7 above applies more to covariance analysis than to MDS.

Interpretation of the PARAFAC output from an MDS analysis is straightforward. The Mode A and B loadings are viewed as the stimulus projections on the dimensions ("perceptual axes"), and the Mode C loadings are the person weights (or, more precisely, the squared saliences of the different dimensions for each person). The fit of the solution to the scalar products is given by the RSQ value at the top of the final loadings output.

CHAPTER 7

DATA SYNTHESIS

The PARAFAC data synthesis capability allows you to construct artificial data with known structure. Such data are useful for testing theories, evaluating analysis procedures, comparing fit values between observed data and "null hypothesis" synthetic data, etc.

Synthetic data generated by PARAFAC may consist entirely of a systematic or "true" component, or entirely of error, or of some combination of both. The error component is composed of random error and/or additive constants. Records II-5 and II-6 of the PARAFAC input deck are used to specify characteristics of the true component, while Record II-7 is used for the error characteristics (see "PARAFAC Input for Data Synthesis"). The ISTRMS and DSIZE parameters on Record II-6 and the ERRSIZ parameter on Record II-7 can be used to specify the relative weights of the systematic component and random error component in the data (see Section 7.3).

Sections 7.1 and 7.2 describe considerations to be made when selecting values for the parameters on Records II-6 and II-7. The discussion relates the parameters to experimental conditions that may affect "real" data and explains how the various parameters influence one another during the data synthesis process. This should help you decide how to specify parameters that will simulate data with particular properties. If there is a conflict between some of the parameters that you specify, PARAFAC will resolve it either by resetting their values or by terminating the run. In either case, a message is usually output (such messages are explained in Chapter 8).

Section 7.4 describes output from a PARAFAC data synthesis run.

7.1 SYSTEMATIC COMPONENT (RECORDS II-5,6)

These parameters allow the user to specify some characteristics of the true part of the data, such as the type of data to be simulated, the sign patterns of the true factor loadings, whether or not the true factors are to be uncorrelated or orthogonal, etc. Note that none of the parameters on this record has any effect if DSIZE is set to a very small value, because the true component of the data is then essentially zero.

7.1.1 Factor Weights (ILDGIN)

Factor loadings used in the generation of the systematic part of the data can either be supplied by the user (ILDGIN=1 or 2) or generated by PARAFAC (ILDGIN=0). When ILDGIN=0, the factor weights are randomly selected from a rectangular distribution that is centered at zero and that extends over the range (-1,1) (end points excluded). Appendix C shows how the distribution is simulated.

The initial values are always normalized so that the mean squared weight for each factor on each mode is 1.0. They then may be further modified, depending on parameter specifications on Record II-6. You may find the initial normalization to be a problem, because it prevents you from generating data from factor loadings exactly as they were input (when ILDGIN=1 or 2). If so, you may want to modify the program.

7.1.2 Random Number Generator Seed (SEED2)

SEED2 is used to initiate the generation of random numbers that will be used to compute the systematic part of the synthetic data (when ILDGIN=0). The same starting seed always produces the same sequence of random numbers. Hence, if you want two different data sets to have the same systematic ("true") structure, use the same specifications for all parameters on Records II-2 through II-6. Changing DSIZE on Record II-6 will not change the <u>pattern</u> of the systematic contributions in the data, but it will alter their <u>size</u>; changing any of the other parameters could cause the true structure to be substantially altered.

7.1.3 Data Type (IDATYP)

PARAFAC can synthesize three types of data: raw score or profile data (IDATYP=0), dissimilarity or distance-like data (IDATYP=1), and cross-product- or covariance-like data (IDATYP=2). There are also some situations when you can use data type 2 to represent scalar products. Profile data is the default and the user will probably choose it most often. Dissimilarity data and covariance-like data are useful for studies of more specialized situations.

PARAFAC does the inverse of a factor analysis to generate the true part of the data. For profile and covariance-like data, PARAFAC multiplies factor loadings together according to the three-way proportional profiles factor model (Harshman and Lundy, 1984b, p.126) to obtain an array of data points exactly described by those loadings. For dissimilarity data, PARAFAC multiplies factor loadings together according to the weighted Euclidean distance model (Carroll and Chang, 1970). Because dissimilarity data are based on a different model, they are discussed separately from profile and covariance data below.

7.1.3.1 Raw Score And Covariance-like Data - Synthesizing raw score data is a straightforward procedure -- simply set IDATYP to zero -- but covariance-like data synthesis may be more complicated. Error-free covariances may be directly synthesized by setting IDATYP to 2, but covariances with error should be generated via a two-step process that involves synthesis of raw score data first. This is explained in more detail below.

Profile data allows for three distinct modes in the data array. For example, if you wished to simulate a study that involved subjects rating various characteristics of some stimuli, you would usually want to generate profile data. The three modes in this case are ratings scales, stimuli and persons.

Sometimes it is preferable to analyze covariance data rather than profile data (Section 4.4 discusses covariance analysis in more detail). In real experimental situations, covariances are obtained indirectly: profile data are collected and then covariances are computed from the profile data. There are several situations when such a transformation is done. For instance, it is not valid to analyze the raw data when the levels of a mode correspond to nonequivalent samples (e.g., when the same people are not tested on every occasion), and so covariance-like data must be computed. Or, it may not be economically feasible to analyze the raw data when one mode has "too many" levels. If you do not want to lose information by simply deleting some of the levels, you can compute covariances across the mode with the many levels. This allows use of all the information, but the resulting data array is smaller and thus more practical to analyze.

The transformation of raw score data to covariances eliminates the distinct mode over which the covariances (or cross-products) are computed, and leaves data that are symmetric across two modes. For example, suppose the raw data consist of people's scores on various personality scales in several different situations, and suppose covariances are computed across people for each situation; then the arrangement of the covariance data is scales by scales by test situation. Strictly speaking, any subsequent PARAFAC analysis of the covariances is appropriate only if the factors underlying the mode that "disappears" are orthogonal or close to orthogonality. The covariance-like data generated by PARAFAC are consistent with this orthogonality requirement.

You can use PARAFAC to directly synthesize error-free covariance type data that are symmetric across Modes A and B by specifying a value of 2 for IDATYP. If you want the covariances to contain error, however, you should use a two-step procedure that parallels the real situation -- first, generate raw score data (IDATYP=0) with error added; then compute covariances from the raw score data. When synthesizing the raw data, you may want to impose orthogonality constraints (see Section 7.1.5) on the mode over which you intend to compute the covariances. On the other hand, if there are quite a few levels (i.e., at least 10-15) in that mode, the factors will probably not be very correlated even if you do not force orthogonality. Small correlations do not invalidate the PARAFAC analysis of the covariances. In fact, this might mirror the real situation more accurately, because the real factors are probably slightly correlated anyway.

You will also use a two-step procedure if you want to study what happens when the model is violated (i.e., when PARAFAC is fit to covariances that were computed across a mode in which the factors were rather highly correlated). First, generate raw score data with only a few levels (less than 10) in one mode -- you will be more likely to get substantially correlated factors if there are not very many levels. Then if you compute covariances across the mode with high factor correlations, you will have data that do not fulfil the assumption of factor orthogonality in the mode that "disappears".

7.1.3.2 Scalar Product-like And Dissimilarity Data - Most often, scalar products are indirectly obtained from distance-like data (e.g., dissimilarity ratings). Dissimilarity data are used for multidimensional scaling (MDS) purposes. In real life, dissimilarity data usually involve people rating differences between objects or concepts of some kind. Depending on how they are collected, the raw data are often symmetric on two modes (the modes are objects by objects by people). PARAFAC generates dissimilarities type data for which the true parts are symmetric across Modes A and B; added error makes them asymmetric. The Euclidean distance model is used to simulate dissimilarities (i.e., the judged dissimilarity of two items is equivalent to the "distance" between them in the person's conceptual space, plus error). Such data should be transformed to scalar products (see the DISTIN program described in Chapter 3) before it is appropriate to use PARAFAC to analyze them. Thus, a simulated MDS analysis is a three-step procedure: generate dissimilarities using PARAFAC (IDATYP=1); then transform them to scalar products using DISTIN; and finally, use PARAFAC to analyze the scalar products.

There are a few situations for which it is permissible to directly simulate scalar product-like data (by setting IDATYP=2). The data can be error-free or they can have error added -- in contrast to covariance type data which must be error-free if directly synthesized using IDATYP=2. (Note that added error will destroy the symmetry of the data on Modes A and B; the data can be symmetrized by using a DISTIN option.) Tucker (1972), for example, treated ratings data directly like scalar products. This is generally appropriate when pairs of stimuli (e.g., adjectives) are compared on scales for which one endpoint denotes a positive relationship, the other a negative relationship (e.g., identical in meaning vs. opposite in meaning) and for which the midpoint, zero, denotes no relationship (e.g., independent meanings). Such ratings mirror the relationships that are conveyed by scalar products (i.e., positive, negative and no relationships). In contrast, dissimilarity ratings denote only varying amounts of the dissimilarity relationship, and nothing negative or opposite to it; thus it is not appropriate to input them directly to programs that assume scalar product-like input.

7.1.4 Sign Constraints (IFAPOS, IFBPOS, IFCPOS)

Whether or not constraints should be imposed on the signs of the factor loadings depends on the type of data being generated and on the situation you are trying to simulate. Sometimes it is mandatory to constrain Mode C loadings to be positive (IFCPOS=2), because of assumptions underlying the synthetic data model; sometimes it is only desirable. For example, with dissimilarity data (IDATYP=1), Mode C must be positive; otherwise, computation of distances may involve taking the square root of negative numbers. When IDATYP is 1, IFCPOS is always set to 2 before synthesis begins, regardless of what value had been specified by the user. Mode C should also be positive if covariance-like data (IDATYP=2) are to be generated. Negative values, including negative diagonals, might otherwise arise in the data. Negative diagonals never occur in real cross-product, covariance or scalar product data, since they are obtained by squaring values. Synthesis of covariance type data (IDATYP=2) is allowed to proceed even if IFCPOS is not 2, although it is theoretically questionable.

Sometimes, logical considerations imply that loadings in other modes should also be constrained to be positive. For some situations that you may simulate, you may feel that the factors have more or less effect across the levels of a particular mode, but never have a negative or opposite influence. For example, suppose that you wanted the data on one mode to represent test scores and the factors underlying the data to be viewed as types of ability. You would constrain the factors to have positive loadings if you believed that each ability contributed more or less to the overall score but that it never detracted from the score. Or, suppose one of the modes represented occasions and you felt that the factors would never reverse their effect from one occasion to the next; then you might want to constrain the factor loadings in the "occasion" mode to be positive.

You may center the "true" factor loadings in a mode to simulate data with a mean of zero across the levels of that mode. Or, you may center the loadings if you want data with an overall mean of zero and hence a mean square equal to the variance (Section 7.3 describes an instance where this would be useful). Note that centering the loadings on a particular mode (e.g., IFAPOS=3) centers the "true" part of the synthetic data, whereas centering the data on the same mode before analysing it (e.g., IFCENA=1) in effect centers both the true and error parts of the data. A practical reason for doing both types of centering with a synthetic data set is to facilitate the comparison of the true factors with the factors obtained via PARAFAC analysis of the data (i.e., it is easier to tell how similar the factors are if both sets of loadings are centered).

The default for dissimilarity data for both Modes A and B is 3 (i.e., IFAPOS=IFBPOS=3), but this centering is not enforced. If some other value is specified for IFAPOS and IFBPOS, the centroid of the configuration of stimulus points will be located at some point other than zero. This does not affect the analysis of the scalar products when recovering the true structure, however, because conversion of dissimilarities to scalar products always involves centering.

7.1.5 Dependence Constraints (IFAORT, IFBORT, IFCORT)

For many synthetic data sets, the default of "no factor dependence constraints" is best. In general, the true factors will be at most only moderately correlated for any mode with quite a few levels (e.g., at least 10 or 15). Sometimes, however, you may want to ensure that the factors are perfectly orthogonal (or uncorrelated) in one or more modes. Such a situation would arise if you were investigating factor models that assume factor orthogonality in one or more modes of the data. An example of this, discussed in Section 7.1.3.1 above, is the PARAFAC analysis of covariance type data that have been computed from raw scores. The assumption there is that the factors are orthogonal in the mode over which the covariances are computed. Another case where you would want orthogonal factors would be for the study of variance additivity properties that depend on factor independence.

On the other hand, you may want factors that are highly correlated. You cannot directly request this, but it usually is the case for any mode with only a few levels (e.g., less than 10). Highly correlated factors would be desirable if, for example, you were investigating how recovery of the true structure when the factors are highly dependent compares with recovery when the factors are independent of each other.

There are some situations where certain dependence constraints are invalid. PARAFAC checks for these and may modify or remove the offending constraint, or terminate the run. A warning or error message (explained in Chapter 8) will be issued if such action is taken. To avoid problems, use the following guidelines when specifying dependence constraints:

1. If the true part has only one underlying factor (i.e., NFGEN=1), it makes no sense to assign any dependence constraints (i.e., a value of 2 or 3 for IFAORT, IFBORT and IFCORT is not allowed).

2. If the factors are constrained to be positive in a particular mode (e.g., IFCPOS=2), they cannot simultaneously be orthogonal (e.g., do not specify IFCORT=3), but they can be uncorrelated (e.g., you can specify IFCORT=2).

3. If the factors are centered in a particular mode (e.g., IFAPOS=3), orthogonality constraints on that mode have the same effect as zero-correlation constraints (e.g., IFAORT=2 and IFAORT=3 have the

same effect).

4. If there are more factors than levels in a particular mode (e.g., if NFGEN is greater than NAS), you cannot assign dependence constraints to the factors in that mode (e.g., do <u>not</u> specify IFAORT=2 or 3).

5. If there are the same number of factors as levels in a particular mode (e.g., if NFGEN=NAS), zero-correlation constraints may cause problems (e.g., it is better not to specify IFAORT=2).

7.1.6 Root Mean Square Standardization (ISTRMS)

Set ISTRMS to 1 if you want the root mean square of the systematic component of the data to equal DSIZE. This, in combination with an appropriate ERRSIZ value, allows you to control the relative sizes of the random error and true parts of the data (explained more fully in Section 7.3). You will probably use this option most often with profile type data (IDATYP=0).

7.1.7 Factor Size Multipliers (ISZFAC)

The factor size multipliers allow you some control over the relative amounts of variance that are contributed by the different "true" factors. Version 6H of PARAFAC does not enable you to assign specific values for the multipliers, however. Rather, depending on the value of ISZFAC, the multipliers are randomly selected from different distributions. Random selection simulates the fluctuations in factor sizes that may occur in different real data sets; the distribution determines the probability that the values selected will be similar or quite different. Appendix C gives details about how the distributions are simulated.

When ISZFAC equals zero, the selection is made from a triangular distribution, which means that the values obtained are more likely to be similar and moderate in size than they are to be large or small. Hence, the factors tend to have more similar contributions, and this is probably the most natural of the three alternatives. You would probably use this in simulating naturally occurring data.

On the other hand, if you want to simulate data that have some large factors as well as factors that contribute much less to the data, you would set ISZFAC to 1. In this case, values are selected from a uniform distribution and so every value in the range of the distribution, including large and small, is equally likely. In trying to recover the true structure in such data, you would then be able to study how well all factors were recovered. In such cases the biggest factors often dominate the solutions and small factors are hard to extract. For example, the big factors may split rather than the small "true" factors being recovered. You might investigate the effects of orthogonality constraints in trying to recover all the factors in such a situation.

Setting ISZFAC to 2 (for equal factor sizes) is desirable when you want to study something that might be obscured or complicated by factors of unequal sizes. For example, with data containing error, the "elbow" of the fit-versus-dimensionality curve is most obvious when all the factors account for the same amount of variance. In contrast, where there are some large and some small factors in the data (as might happen when ISZFAC=1), the small (but real) factors may obscure the elbow of the curve. This would occur if they do not fit much more variance than small "error" factors that are subsequently extracted. If this happened with real data, split-half analyses might be necessary to confirm the existence of such small true factors.

There is a limiting feature of the current version (6H) of PARAFAC. You cannot input factor loadings (ILDGIN=1 or 2) and generate data from those factors exactly, even if you set ISZFAC to 2 (i.e., the "true" factors on the listing are not equal to the factors that were input). This is because the factors are always normalized -- whether they are randomly generated or user supplied -- so that the mean squared loading for each factor in each mode is set equal to 1.0, before they are adjusted by the size multipliers.

7.1.8 Data Size Multiplier (DSIZE)

You will usually specify a positive value for DSIZE. For raw score data (IDATYP=0), however, a positive DSIZE is not compulsory; a negative DSIZE is equivalent to reversing the signs of all the factor loadings in one mode. DSIZE must be positive for dissimilarity data (IDATYP=1) -- if not, the program takes the absolute value -- because the data represent "distances", and of course distances cannot be negative. DSIZE should also be positive for covariance type data (IDATYP=2) to be consistent with the theory underlying this kind of data (i.e., diagonals are obtained by squaring values, and so negative diagonals are impossible in real data of this type), but the program does not enforce this.

A positive value for DSIZE also seems more logical than a negative one when you want to specify the relative amounts of systematic and error variance in the data. This is explained more fully in Section 7.3 below.

DSIZE can also be used to suppress the contribution of the "true" part of the data variance. To do so, specify a very small value for it (e.g., 1.E-20) -- but not zero, since if you set it to zero, the program assigns the default value of 1.0. With a very small DSIZE, essentially 100% of the variance in the data will then be due to error.

7.2 ERROR COMPONENT (RECORD II-7)

It is possible to add random error and/or an additive constant to the data generated from the true structure. You can also have some of the data points contaminated by extra large random error, relative to the others, or else have some outliers in the data while the other points have no error at all.

You can center the true part of the data (see Section 7.1.4), but you cannot formally request that the error component be centered during data synthesis. However, the mean of the random error will usually be approximately zero anyway, if the error is one of the first four types (i.e., if IERTYP=1-4 inclusive). This is because these four error distributions are symmetric about zero and the error is randomly selected from them. On the other hand, the error mean may deviate substantially from zero if you request one or more of the following: outliers (even with IERTYP=1-4), error from the lognormal or slash distributions (IERTYP=5-7), or additive constants.

7.2.1 Random Error (IERTYP)

Four general random error distributions -- uniform, normal, lognormal and "slash" -- are simulated by PARAFAC. (For those who are interested, Appendix C provides details of the simulation procedure.) You will choose the distribution from which to select the error according to the type of data you want to generate and the empirical situation that you are trying to simulate. Brief descriptions of the distributions follow, along with general comments about when it is appropriate to use each.

1. IERTYP=1 or 2. The error is randomly selected from a simulated uniform (rectangular) distribution that has a mean of zero and a standard deviation of 1.0, and then is scaled up or down by the value of ERRSIZ. For IERTYP=2, the error component is further multiplied by the value of the "true" component to which the error will be added, so that

its size is proportional to the size of the true component.

2. IERTYP=3 or 4. The error is randomly selected from a simulated normal distribution that has a mean of zero and a standard deviation of 1.0, and then is scaled up or down by ERRSIZ. For IERTYP=4, the error component is further adjusted, as explained above for IERTYP=2.

In real data, the size of the error may be correlated with the size of the true part, or their sizes may be independent. For example, judgements of heaviness or weight are likely to have errors whose sizes (disregarding sign) are correlated substantially with the sizes of the judgements -- the greater the judged weight is, the greater the error is likely to be, either as an over- or under-estimate of the actual weight. For such data, proportional variance type error (IERTYP=2 or 4) is best. Proportional variance error is <u>not</u> appropriate for data that have no systematic variance (where DSIZE has been set to a very small value), because data that are all zero will be synthesized. Where the error size is independent of the true part, constant variance error is probably more appropriate. Such error may occur in some types of rating scale data (ignoring any end effects).

When simulating real measurements, the normal error distribution (IERTYP=3 or 4) should usually be selected. There may be times, however, when uniformly distributed error would be better. For example, if you want to simulate random ratings (i.e., rating scale data composed entirely of error), error selected from a uniform distribution (IERTYP=1 or 2) would be more appropriate. You might want to see how well PARAFAC fits such random data, compared to real rating scale data (same sized array) if you are testing the null hypothesis that there are no factors in the real data (e.g., see Harshman and DeSarbo, 1984).

3. IERTYP=5. Of all the error types, lognormal error is the only one that is not simply added to the true component. The error value is obtained in the same way as for IERTYP=3, but then it is used as an exponent for e (base for natural logarithms), which is then multiplied by the true component to give the value of the synthetic data point. Hence, the error part of the data has a kind of nonlinear proportional variance.

The lognormal distribution is positively skewed. Error selected from it can be used for data that are constrained to be positive, because it will never make a positive data value negative. This is particularly appropriate for dissimilarity data. Less error is added to small data values than to large ones; this is in accord with many psychological models which suggest that small subjective differences are judged more accurately than large ones. Also, lognormal error is consistent with the idea that the error introduced by underestimates is less than the error due to overestimates (i.e., underestimates have a lower bound because distances cannot be negative, whereas there is no upper bound for overestimates).

4. IERTYP=6 or 7. The slash distributions are equivalent to a normal distribution with a mean of zero and standard deviation of 1.0, divided by a uniform (rectangular) distribution extending over the range (0,1). Each error component is obtained by making a random selection from the simulated normal distribution and a random selection from the uniform distribution, dividing the first deviate by the second, and scaling up or down by ERRSIZ. For IERTYP=7, the error is made proportional to the size of the true part, as explained above for IERTYP=2.

You would use one of the slash distributions (IERTYP=6 or 7; see Andrews, Bickel, Hampel, Huber, Rogers and Tukey, 1972, p.68) if you are interested in testing the robustness of statistics when the data are contaminated with error that is not well-behaved. The slash distribution can contribute more extreme error than the other distributions because it is unbounded (whereas the uniform distribution covers some finite interval that depends on ERRSIZ) and because its tails are thick (compared to the normal distribution). For example, 0.5% of the error selected from it is on the average at least 100 times larger than error from the equivalent normal distribution; 0.05% of the time, the error is at least 1000 times greater.

7.2.2 Random Number Generator Seed (SEED3)

SEED3 is used to initiate the sequence of random numbers that will be used to compute the error component of the data. The same starting seed always causes random numbers to be generated in the same sequence. Therefore, by using the same specifications for the parameters on Records II-2 and II-7 in different runs, you can generate different (same-sized) data sets that have the same pattern of error contributions. If you change only ERRSIZ, the size but not the pattern of the error changes.

7.2.3 Error Size Multiplier (ERRSIZ)

You may specify any value for ERRSIZ, except when you want to control the relative amounts of error and true components in the data (using ISTRMS=1). Section 7.3 explains how DSIZE and ERRSIZ are related when ISTRMS is 1.

7.2.4 Additive Constants (ACONSZ)

The additive constant can be added to the data in one of two ways:

1. IERTYP=0 and ACONSZ is nonzero. All data points are offset by the value of ACONSZ.

2. Both IERTYP and ACONSZ are nonzero. Data points in the Mode A-by-Mode B matrix at a given level of Mode C (i.e., a Mode C frontal slice as depicted in Section 4.2) are offset by a random number that is multiplied by ACONSZ (the random number is selected from a uniform distribution that ranges from zero to one inclusive). A different random number is selected for each level of Mode C, so each Mode C slice has a different additive constant.

The error introduced by the additive constant is particularly appropriate for MDS simulations. For such simulations, you should specify a positive value of ACONSZ so that the additive constant cannot make any data points negative (since negative dissimilarities are not allowed).

The additive constant may also be used with other types of data to demonstrate, for example, the value of centering before analysis. Suppose you synthesize profile type data with no random error, and adjust each Mode C slice by a different additive constant. Analysis of the data, with and without centering, will show that the centered data set can be fit perfectly with the same number of factors as were built into it (i.e., with NFACT=NFGEN), whereas the uncentered data set requires an extra factor for perfect fit. The extra factor has constant loadings in Modes A and B. (In fact, if ISTANM=3, the Mode A and B loadings for this factor are 1.0, and the Mode C loadings are equal to the additive constants that were introduced at the synthesis stage.) This exercise shows that PARAFAC centering can eliminate certain kinds of error. It also demonstrates how the presence of a constant factor in the PARAFAC output indicates that the data need to be centered (as mentioned in Section 4.1.1).

To see how the analysis of uncentered data could be affected in a real situation, you might try adding various amounts of random error to the data as well. The extra error may somewhat obscure the distinction between the true factors and the "constant" factor in the uncentered data. The Mode A and B loadings of the extra factor will no longer be 1.0, and the Mode C loadings will not equal the additive constants. At moderate error levels, however, its Mode A and B loadings should show little enough variance that it could be identified as a constant factor.

7.2.5 Outliers (CONPRB, CONSIZ)

A data point is referred to as an outlier if its value is so large or small when compared to the other data that it appears to have come from a different distribution. Under actual experimental conditions, outlying values may occur if the measurement instrument malfunctions, the instructions are misinterpreted, or if a significant keypunching error is made. Almost without exception, outliers can be regarded as data that are composed of extreme error.

If you want to study the effects of various numbers and sizes of outliers on the recovery of the true structure, you will make use of the CONPRB and CONSIZ parameters. The random error is computed as usual, but for outliers, CONSIZ is used as an additional error size multiplier. CONPRB can assume values from zero to one inclusive, since it is a probability value. However, it should generally be set to some value less than 0.1, since an errant point cannot as validly be called an outlier if 10% of the data have values as extreme as it does. Values of 0.005 up to 0.03 for CONPRB give a realistic outlier frequency, since this means that the expected number of outliers in every 1000 data points will be from 5 to 30 respectively. (There may not be exactly 5 and 30 outliers in every 1000 points, because random selection is involved in determining which points will be treated as outliers.)

There are two ways of specifying outliers:

1. IERTYP, ERRSIZ, CONPRB and CONSIZ are all nonzero. All data points have error added as determined by IERTYP and ERRSIZ, but for the outliers, the added error is also multiplied by CONSIZ to inflate its size.

2. ERRSIZ=0; IERTYP, CONPRB and CONSIZ are all nonzero. Only the outliers have random error added, the size of which is proportional to CONSIZ. All other data points are error-free (unless ACONSZ is nonzero).

7.3 WEIGHTING OF THE TRUE AND ERROR COMPONENTS

There may be times when you want to be able to specify the relative sizes of the systematic and error contributions in the data, rather than leaving it to chance. You would need to do this if, for example, you wanted to see how recovery of the "true" structure and how fit-vs-dimensionality curves are affected by different amounts of random error in the data.

To control the mean square values of the true and random error components in the data, specify the parameters on Records II-6 and II-7 as follows:

1. Set ISTRMS=1.

2. Choose IERTYP=1, 2, 3 or 4. Do not specify outliers (CONPRB, CONSIZ) nor error due to additive constants (ACONSZ), as either of these may considerably inflate the amount of error present in the data.

3. Set DSIZE=1.0. The root mean square (RMS) of the true component always equals the value of DSIZE; hence, DSIZE squared is the mean square of the systematic part of the data.

4. Choose ERRSIZ so that the ratio of DSIZE squared (i.e., 1.0) to ERRSIZ squared is equal to the true -to- error mean square ratio that you want -- true -to- error <u>variance</u> is discussed later in this section. The expected RMS of the random error equals ERRSIZ; ERRSIZ squared is the expected mean square (EMS) of the random error in the data.

For example, for data sets with 0%, 25%, 50%, 75% and 100% error as compared to the true component, DSIZE and ERRSIZ would have the following values:

True:Error Ratio	DSIZE	ERRSIZ	EMS Error
1:0	1.0	0.0	0.0
3:1	1.0	0.577	0.33
1:1	1.0	1.0	1.0
1:3	1.0	1.732	3.0
0:1	1.E-20	1.0	1.0

The first four data sets will have the same mean square value for the true part, but different expected mean square (EMS) values for the random error (error due to additive constants is not included in the EMS error value). The fifth example is included to show how you would use DSIZE to virtually eliminate the systematic component.

In fact, the value of DSIZE is not restricted to 1.0. For IERTYP=1 or 3 (constant variance error), any value for DSIZE can be specified along with an appropriate value for ERRSIZ, so that the ratio of their squares is what you want. For example, you could set both DSIZE and ERRSIZ equal to 3.0 to obtain data that contains 50% error (i.e., true:error = 1:1). Or, for data with 75% error (true:error = 1:3), you could set DSIZE = 1.732 and ERRSIZ = 3.0.

For IERTYP=2 or 4 (proportional variance error), it is simplest just to use DSIZE=1.0 and the corresponding ERRSIZ value, as explained above. Otherwise, the relationship between DSIZE, ERRSIZ and the true and error mean square values is different, and is not explained here.

You can verify that the mean square values do indeed fulfil the desired ratio by referring to the PARAFAC output (see Sections 7.4.1.2.2-3). The actual mean squared error value may vary slightly from the EMS error value that is listed in the table above (or ERRSIZ squared), because of the random selection process used to obtain the error value and because of rounding error.

If you prefer, you can control the true-to-error variance, so long as both the true and error components have a mean of (approximately) zero. This is accomplished by doing the following things when synthesizing the data:

1. Use the same values for DSIZE and ERRSIZ as outlined above for mean square ratios.

2. Center the true part by requesting centered loadings on one or more modes (e.g., set IFAPOS=3).

3. Use IERTYP=1, 2, 3 or 4 only, and do not request outliers or additive constants. This should give error that has a mean of approximately zero, as explained in Section 7.2. (There is no parameter to directly specify centering of the error.) You can check the PARAFAC output to ascertain that the actual variance is approximately what you want.

7.4 PARAFAC SYNTHESIS OUTPUT

The output from a run that synthesizes data is very similar to the analysis output described in Chapter 5, except for some extra information that is specific to the synthesis procedure. The information below is discussed in the order in which it appears on the lineprinter listing, but only what pertains specifically to data synthesis is described in detail. You are referred to the appropriate sections of Chapter 5 for more discussion.

7.4.1 Lineprinter Output: Verification Of Input

The listing consists of two general sections, one that verifies the program input and one that documents the analysis (if done) as it proceeds. Most of the changes occur in Input Section II.

7.4.1.1 Input Section I (Parameter Check) - See Section 5.1.1 for a complete description.

7.4.1.2 Input Section II (Data Check) - The differences in this section reflect the modified input that is required when the data are to be generated by PARAFAC, instead of being supplied by the user. Following the section heading are the input records for data synthesis, and a description of parameters, formats, and any default values assigned by PARAFAC. Then summary statistics, factor loadings, and matrices of cross-products and correlations are listed. They are described below.

7.4.1.2.1 Loadings Used To Generate Data - A reference is made to the factor loadings that will be used to generate the data. The actual wording of the message varies, depending on the value assigned to the ILDGIN parameter on Record II-5. For ILDGIN=0, random loadings are used; for ILDGIN=1, PARAFAC- format loadings are input, and the descriptive information that precedes the loadings matrices (see Section 5.2.1) is listed next. In both cases, the loadings are not listed here. For ILDGIN=2, however, loadings are in non-PARAFAC format, and they are listed so that you can check the input.

7.4.1.2.2 Statistics For Synthetic Data Components - These statistics are not to be confused with the summary statistics described in Section 5.1.2.3, that are computed after centering, normalization, and missing value estimation. Rather, these statistics are computed separately for the true and error components before centering, etc.

Two tables comprise these statistics. The first is a set of means, variances and mean squares for the various components of the data. This part of the output is particularly useful if you are interested in the ratio of true-to-error variance or mean square (see Section 7.3). You can also check the mean of the error components here (you will recall that you can't request that the error component be centered). The variance and mean square will be equal, of course, for any component that has a mean of zero. If the true part was suppressed by specifying a very small DSIZE value, then the statistics associated with it are all zero.

The second table of statistics--a set of cross-product, covariance, cosine and correlation values--are computed from the true and error components of each data point. As in the table above it, the "total error" here is the random error plus the additive constant error; hence, both columns of values are equal if no additive constants were specified. These statistics give some indication of the relationship between the data components. Usually the correlation between the random error and true components is close to zero. Even proportional variance error (e.g., IERTYP=2 or 4) is usually uncorrelated with the true component -- their absolute values would be correlated, however.

Note that for lognormal error (IERTYP=5), the statistics are not computed, but are printed as zero. This will be rectified in a future version of the program.

7.4.1.2.3 Factor Loadings - Factor loadings are listed next. They are the ones used to generate the systematic structure of the synthetic data. Even if the systematic part is essentially zero because you specified a very small value for DSIZE, the loadings will be output here; however, the mode that reflects the scale of the data will have loadings that are approximately zero.

The loadings listed here will not be the same as were input (when ILDGIN=1 or 2). This is due to several things: version 6H of PARAFAC always normalizes the mean squared factor loadings in all three modes, then adjusts to meet any positivity or orthogonality constraints requested, and finally multiplies by the factor size multiplier and data size multiplier. Hence, even loadings in PARAFAC format on input (ILDGIN=1) are modified somewhat before being output here. As we've said, this is sometimes a limitation, since this means you cannot generate a systematic data component from factors exactly as they were input (unless the program is modified).

The descriptive information and fit values printed above the factor loadings apply to the data synthesis procedure. The fit values give the relationship of the systematic or "true" structure of the data to the total data (true plus error components). Hence, if DSIZE is very small, so that the true component is essentially zero, MSE and STRESS will be large, while R and RSQ will be approximately zero. Or, if you didn't request that error be added, perfect fit will be indicated by R and RSQ values of

1.0, and MSE and STRESS values of zero.

If you wanted a certain true-to-error ratio in the data (Section 7.3), you can check RSQ and MSE to make sure that this was obtained. For example, if you wanted 75% error (and hence 25% systematic structure), you may have specified DSIZE=1.0 and ERRSIZ=1.732. The error component will have an expected mean value of zero and an expected variance value of 3.0 (i.e., ERRSIZ squared). Due to random selection from the error distribution, however, the mean and variance of the error will not be exactly equal to the expected values, although they should be close (check the statistics table). Hence, while RSQ should be close to 0.25 and MSE should be close to 3.0 (ERRSIZ squared), they will likely not be exactly equal to these values. (Note that RSQ may deviate substantially from what you think it should be if the data include additive constants, but you will see that the mean is not close to zero.)

The "Root mean squared contribution for each factor" is a measure of the "size" of the factors. Each value depends in part on the factor size multipliers, selected according to the value of ISZFAC (Section 7.1.7). If you had specified ISZFAC=0, you would expect to see that the factors had similar RMS constributions, although by chance one could be much larger or smaller than the others. On the other hand, if ISZFAC=2, the factors all have exactly equal RMS contributions. See Section 5.2.3 for a general discussion of RMS factor contributions.

7.4.1.2.4 Factor Cross-products And Correlations - Cross-products and correlations are computed between the loadings of the factors that are used to generate the systematic component of the data. They reflect any orthogonality or zero-correlation constraints that you may have requested via IFAORT, etc. (Section 7.1.5). Even if you didn't request such constraints, you may want to see how correlated the "true" factors are, especially if you are studying violations of factor models (Section 7.1.3.1).

7.4.1.2.5 Data Check, Etc. - Section 5.1.2 has a complete description of the information that appears next: a data check, missing data subscript table, centering/normalization check, summary statistics table, and a symmetry check.

7.4.1.3 Input Section III (Initial Loadings) - Unless analysis of the data is suppressed (NSOLS=-1), information about the starting loadings is listed next; see Section 5.1.3.

7.4.1.4 Special Output - The (centered and/or standardized) synthetic data will appear as the last thing on the listing if NSOLS=-1 and IUNITD=6 (the standard output unit).

7.4.2 Lineprinter Output: Analysis Documentation

Analysis of data generated by PARAFAC proceeds in exactly the same way as analysis of user-supplied data, and the documentation is also the same. A complete description is given in Section 5.2.

7.4.3 Disk Or Tape File Output

In general, this output is as described in Section 4.3, but there is additional output resulting from the data synthesis process.

7.4.3.1 Final Loadings - Even if no analysis is performed (NSOLS=-1), the loadings used to generate the systematic part of the data can be saved on disk, just as the final loadings from an analysis are. If both data synthesis and analysis are performed in the same run, then (NSOLS+1) complete sets of loadings will be written to IUNITG, the first being the "true" loadings used to generate the data and the following NSOLS sets being the loadings obtained from the analyses. Before using them to continue the analyses, you would therefore have to separate the first set of loadings from the others.

7.4.3.2 Revised Data And Residuals - The description in Sections 5.3.2 and 5.3.3 applies here. When analysis is suppressed, one set of data is written to IUNITD (i.e., the data as they were generated, and then centered and/or normalized according to the specifications on Card I-3, with a missing value table appended if missing values were specified), but no residuals are output, even if you request them by mistake.

If the run involved both data synthesis and analysis, then IUNITF has NSOLS sets of residuals, as usual, while

IUNITD has (NSOLS+1) sets of data. The data sets on IUNITD will all be identical unless there are missing values specified. Then, the first data set will have initial estimates of the missing values (i.e., before analysis), while the others will have estimates based on the final loadings from each solution.

CHAPTER 8

PARAFAC MESSAGES

Messages output by PARAFAC fall into three categories: analysis, warning and error. Analysis messages are part of the normal output for a PARAFAC solution; they provide information about the course of the analysis. Warning messages signal that the course of the data synthesis and/or analysis was changed when PARAFAC reset a parameter value to remedy a potential problem. Error messages point out a problem which caused program execution to halt prematurely. The messages are discussed below by category, analysis messages first, warnings next, and errors last.

8.1 ANALYSIS MESSAGES

Depending on the data preprocessing and analysis options selected by the user, PARAFAC outputs informative messages as well as the factor loadings matrices during the course of an analysis. This additional information allows the user to follow the program procedures more closely. The messages are listed below in the same general order that they are output for a PARAFAC solution, along with an explanation of each.

8.1.1 Initial Messages

The following messages are printed before the first loadings output (i.e. iteration 0) of the solution:

BEGINNING OF SOLUTION n

This is always the first message. The number n also appears in the descriptive information of all subsequent loadings outputs for the solution.

THE STARTING LOADINGS ARE RANDOM NUMBERS. THE INITIAL SEED FOR THE RANDOM NUMBER GENERATOR IS dd

This message occurs if ISTART on Card I-5 is zero. It is useful to have the seed value dd if you want to repeat the solution from the same starting point (e.g. if the solution terminated prematurely because computer time limits were exceeded by the job).

MISSING VALUE STARTING ESTIMATES ARE THE SAME AS FOR SOLUTION n-1

This occurs for solutions 2 through NSOLS (Card I-2), when MISEST (Card I-4) is zero (where n-1 is the solution immediately preceding the current one).

MISSING VALUE STARTING ESTIMATES ARE THE SAME AS FOR SOLUTION 1

This is printed for solutions 2 through NSOLS, when MISEST is 1.

DATA IS RECENTERED BEFORE ITERATION 1

This is output for solutions 2 through NSOLS after the missing value estimates have been set up, when at least one of the "IFCEN" flags on Card I-4 has a value of 1 or 3. Renormalization is not performed.

8.1.2 Intermediate Messages

Other messages appear after the starting and/or intermediate loadings outputs. They are listed below under the headings of "Centering", "Check of R", "Constraints" and "Missing Values and Diagonals".

8.1.2.1 Centering. -

DATA IS RECENTERED AFTER ITERATION i

When there are missing data and at least one of the "IFCEN" flags on Card I-4 is 1 or 3, the data is recentered after every NITER (Card I-2) iterations. This adjusts for any distortion of the centering which may result from the reestimation of missing values, and occurs after every loadings output except the final one.

8.1.2.2 Check Of R. -

iTH ITER., R= r1, AVRINC= r2, RACCEL= r3, DIFFA= aa, DIFFB= bb, DIFFC= cc

This is the result of checking for a decreasing R value, but other information is also provided that allows one to follow the change in the fit and "DIFF" values as the iteration process converges. Unless IRINTV on Card I-6 is set to -1 to suppress this, the above message is output every IRINTV iterations. Thus, it may occur several times between successive loadings outputs. R is the correlation at the ith iteration between the data being analysed and the data predicted by the model. AVRINC is the difference in R between the current iteration and IRINTV iterations before the current one (when a similar message was output), divided by IRINTV (i.e., it is the average increase in R). RACCEL is the average acceleration of R over the last IRINTV iterations. DIFFA, DIFFB and DIFFC give the maximum percentage change in the value of loadings in Modes A, B and C respectively from the previous iteration (i.e., not averaged across the previous IRINTV iterations). A negative AVRINC signals decreasing fit. This normally happens only when orthogonality or zero-correlation constraints have been imposed on the factors. The smaller IRINTV is, the fewer iterations are performed before the program detects this problem and takes action. RACCEL may be positive for initial iterations when the fit is rapidly improving, but the rate of change of R quickly levels off during the analysis, and RACCEL then becomes negative.

NONINCREASING R IS PRESUMABLY DUE TO DATA RECENTERING, NO CORRECTIVE ACTION TAKEN.

When the AVRINC value in the previous message is negative, the program checks for possible causes. This message is output next if there are missing data and "IFCEN" flags on Card I-4 have been set to 1 or 3; in this case, periodic recentering of the data is performed, which might temporarily reduce the fit of the PARAFAC solution to the data. After printing this, the iterative process resumes.

FIT NOT INCREASING, TEMPORARY DEPENDENCE CONSTRAINTS DROPPED.

When a negative AVRINC value cannot be explained by data recentering, PARAFAC next checks to see if IORTHA, IORTHB and/or IORTHC are set to 2, 3, 5 or 6 (i.e. for temporary constraints). All "IORTH" flags which have any of these values are reset to 1, even though the normal conditions for dropping the constraints have not yet been met. After this message, the iterative process resumes.

FIT NOT INCREASING, OVERRELAXATION SUPPRESSED.

If neither recentering nor temporary dependence constraints are found, the program then checks whether the negative AVRINC occurred when overrelaxation was being performed. After the first 10 iterations of any solution, new loadings are extrapolated 27.5 percent beyond the current best regression estimates in the direction of change from the previous iteration. This process is called overrelaxation and is used to accelerate convergence, but sometimes it may push the solution too far and the fit may worsen. PARAFAC continues the analysis after printing the above message, but does not extrapolate beyond the best loadings estimates on subsequent iterations.

FIT NOT INCREASING, MISSING VALUE RE-ESTIMATION INITIATED.
When neither recentering, temporary dependence constraints nor overrelaxation can explain the negative AVRINC value, the iterative procedure terminates unless there are missing values that have not yet been reestimated. Normally, missing value reestimation is not begun until iteration 11. However, if a negative AVRINC is obtained before this and there are missing data, the program begins to reestimate the missing values from the current iteration on. The iterative process resumes after output of this message.

FIT NOT INCREASING, FURTHER ITERATION SUPPRESSED. FORCED OUTPUT AT ITERATION i
This happens either when none of the procedures described above are applicable to the current analysis, or when all the appropriate ones have been done and AVRINC is still negative after another IRINTV iterations.
The solution at iteration i is output in the same way that final loadings for any solution are output after the convergence criterion is met or NITER times NOUTS iterations have been performed. The program then proceeds to the next solution in the usual way. Forced output is explained further in Section 5.2.2.3.3.

8.1.2.3 <u>Constraints.</u> -

DEPENDENCE CONSTRAINTS IN EFFECT FOR--
MODE A
MODE B
MODE C
This is printed after the loadings outputs for which each of IORTHA, IORTHB and IORTHC (Card I-6) have a value greater than 1. Any mode is omitted from the above list if its "IORTH" flag value is 1.

NO DEPENDENCE CONSTRAINTS IN EFFECT
This is printed after loadings outputs for which IORTHA, IORTHB and IORTHC all are 1. The message appears after every output if no constraints were imposed at the beginning of the analysis. On the other hand, if temporary constraints were requested, it is not output until all constraints have been dropped (see message described below).

CONSTRAINT THAT MODE A FACTORS BE INDEPENDENT WAS DROPPED BEFORE ITERATION i. DIFFA= aa PERCENT
(Similar messages are output where applicable for Mode B and/or Mode C.)

This message appears only once. This signals that the temporary constraint placed on Mode A factors at the beginning of the solution has been dropped (i.e., the value of IORTHA has been reset to 1). Where IORTHA was originally 2 or 5 (i.e. weak constraints), the aa value is approximately 50 times the value of DIFMXA (Card I-7); where IORTHA was 3 or 6 (i.e. moderate constraints), the aa value is about 10 times the value of DIFMXA.

8.1.2.4 Missing Values And Diagonals. -

MISSING VALUES ARE RE-ESTIMATED ON EVERY ITERATION AFTER ITERATION i

Generally, the value of i is 10. When there are missing values and the starting loadings are random numbers (ISTART is zero on Card I-5), this follows the starting loadings output as long as NITER on Card I-2 is greater than 10. Missing values are not usually reestimated for ten iterations because the loadings do not predict the data well at first and thus would not yield good estimates of the missing values. However, when the solution converges before iteration 10, missing value reestimation is begun immediately. In this case, the loadings are listed when convergence is initially reached (the iteration value is not usually a multiple of NITER, as for normal intermediate loadings output) but no convergence message is issued. Then when the solution converges with reestimated missing values, a convergence message and final loadings are output as usual.

MISSING VALUES ARE RE-ESTIMATED ON EVERY ITERATION

When there are missing values and the starting loadings are random numbers, this follows every intermediate loadings output after iteration 10.

When the starting loadings are supplied by the user (ISTART is 1 or 2), missing values are reestimated from iteration 1, and this message occurs after every loadings output except the final one.

DIAGONAL CELLS ARE RE-ESTIMATED ON EVERY ITERATION AFTER ITERATION i

Generally, the value of i is 10. When IGDIAG is 1 (Card I-6) and the starting loadings are random numbers, this follows the starting loadings output, as long as NITER is greater than 10. Diagonals are not usually re-estimated during the first ten iterations for the same reason given above for the missing values.

DIAGONAL CELLS ARE RE-ESTIMATED ON EVERY ITERATION

When IGDIAG is 1 and the starting loadings are random numbers, this follows every intermediate loadings output

after iteration 10.
When starting loadings are user-supplied, diagonals are reestimated from iteration 1, and this message occurs after every loadings output except the final one.

8.1.3 Terminating Messages

The next message is printed immediately before the final loadings of the solution (if applicable):

```
CONVERGENCE CRITERION MET ON ITERATION  i
MODE A MAXIMUM CHANGE=  aa  PERCENT
MODE B MAXIMUM CHANGE=  bb  PERCENT
MODE C MAXIMUM CHANGE=  cc  PERCENT
```

No iterations are performed after output of this message because the solution has "converged". In other words, the solution is stable enough that, from one iteration to the next, reestimation of the loadings did not cause any of them to change more than the arbitrary amounts (i.e., convergence criteria) set by the user on Card I-7. For each mode, the change in a loading for any given factor is expressed as a percentage of the root mean squared loading of that factor; hence the reference to "percent" in the above message. This allows larger absolute fluctuations in the mode which reflects the scale of the data. The aa, bb and cc values are less than or equal to the values specified for DIFMXA, DIFMXB and DIFMXC respectively, and are the DIFFA, DIFFB and DIFFC values respectively listed at the head of the final loadings output.

Finally, when ISTART equals zero, the following message appears at the end of the listing:

```
THE SEED FOR THE RANDOM NUMBER GENERATOR AT END OF EXECUTION
IS  dd
```

dd would have been used as the seed for the random number generator for the next set of loadings, had one more solution been requested.

8.2 WARNING MESSAGES

Warning messages inform the user when the program has detected some nonstandard or inconsistent conditions and has reset a parameter so that the data synthesis or analysis could proceed. The messages provide only a limited amount of information, and so an explanation of the problem associated with each message is given below. The explanation assumes that you have verified the input to ensure that the warning is not due simply to the misplacement of values on the input records. The warnings are listed below in alphabetical order by the first word of the message.

DATA IS ALL ZERO, SO FIT VALUES NOT COMPUTED.

Either the data were read as zeroes, and so you should check your input format, or the data were synthesized as zeroes, and so you should check DSIZE, IERTYP and/or ERRSIZ. For example, synthetic data equal to zero will result if you specify a very small DSIZE (to minimize the systematic component in the data) but forget to request that nonproportional variance error be added.

DATA SIZE MULTIPLIER RESET TO 0.0 TO AVOID EXPONENT UNDERFLOW PROBLEMS.

Certain combinations of options for data synthesis cause some of the computations involving an extremely small DSIZE (from Record II-6) to result in a value too small to be represented precisely by the computer (i.e., exponent underflow). While this usually does not cause program execution to cease, a message output to the lineprinter warns the user of this occurrence (depending on the computer system). This would be repeated many times during synthesis of the data and would produce much unnecessary output. To avoid this, the program resets DSIZE when the cube of DSIZE equals zero (due to exponent underflow). Since the small DSIZE value was originally specified to minimize the contribution of the "true" component of the data, and this is accomplished when DSIZE is reset to zero, synthesis of the data is not affected.

DATA VALUES FOR ALL POINTS (i, j, K), K=1, n WERE MISSING, SO THE MEAN COULD NOT BE COMPUTED. ZERO WAS USED AS THE ESTIMATE.

You have used code(s) to indicate missing data values (IFCODE=1 on Card I-4). Since no data point in the indicated "tube" contained legitimate data values, a mean could not be computed to use as the starting etimate for missing data in the tube, and zero was used instead. This does not alter the analysis, except perhaps to slow it down if zero is a poor estimate.

DEFAULT VALUE FOR IORTHC RESET TO 1 BECAUSE NCS IS NOT GREATER THAN NFACT.

You did not specify a value for IORTHC on input, so the program set it to its default value 2 (i.e., weak zero-correlation constraints); the parameter check for Card I-6 on the listing verifies this. However, because NFACT is at least as big as NCS, imposing the default constraint may cause problems because one factor is forced to have constant loadings in Mode C in order to fulfil the constraint. This further constrains the solution, and worsening fit over several iterations may occur. Thus, the program resets IORTHC to 1 to eliminate the constraint and avoid these problems.

DEPENDENCE OPTION FOR MODE C RESET TO 2 (ZERO CORRELATION) TO ELIMINATE POSSIBILITY OF NEGATIVE DISTANCES.

Resetting IFCORT (Record II-6) resolves the conflict between the positivity constraint implied because the data is distance-like (IDATYP=1), and the bipolarity (i.e. both positive and negative loadings) of factors 2, 3, etc. necessary for factor orthogonality.

DIAGONALS ARE NOT DEFINED IF NAS DOES NOT EQUAL NBS (MATRICES NOT SQUARE). OPTION TO ESTIMATE (IGNORE) DIAGONALS IS SUPPRESSED.

Your NAS is not equal to your NBS (see Record II-2). The IGDIAG parameter (Card I-6) is reset to zero and the diagonal entries are treated as legitimate data values, instead of missing values, in the analysis.

FLAG TO STANDARDIZE SCALE OF OUTPUT LOADINGS (ISTANM) RESET TO 3.

This message may be printed before output of the "true" loadings in a data synthesis run. Resetting ISTANM only affects how the loadings are standardized prior to output; it is usually done to eliminate the possiblility of dividing by zero during the standardization process. For raw profile or covariance-like data (i.e., IDATYP=0 or 2), ISTANM is reset if the Mode C loadings are very small (which might occur if the DSIZE value were very small). For distance-like data (i.e., IDATYP=1), however, ISTANM is reset if it was 1 or 2 originally, regardless of the Mode C loadings; the symmetry of such data across Modes A and B makes it less appropriate to have one or the other reflect the data scale.

MODE WITH CONSTANT LOADINGS CANNOT BE CONSTRAINED TO BE UNCORRELATED OR ORTHOGONAL. IORTHA RESET TO 1.
(Similar messages are output for Mode B and/or Mode C when IFHLDB and IORTHB and/or IFHLDC and IORTHC respectively have the same values as described here.)

You set all the starting loadings in Mode A to 1.0 (IFHLDA=2 on Card I-6) and also requested constraints on the factors in Mode A. It is impossible to fulfil the constraints with factor loadings that have no variation, so no constraints are applied to the factors in Mode A during the analysis.

NEGATIVE MODE C LOADINGS DURING SYNTHESIS OF TYPE 2 DATA IMPLY NEGATIVE SQUARED VALUES--INCONSISTENT WITH THE BASIC ANALYSIS MODEL.

In the multiplication process used to produce cross-product, covariance, or scalar product data from actual raw data, the Mode C loadings of the raw data are squared. Consequently, Mode C loadings for Type 2 data would never be negative. Except for outputting the above message, however, no action is taken by the program and execution continues as usual. (It is assumed that the user is simulating such "impossible" data for test purposes.)

NO ANALYSIS PERFORMED BECAUSE DATA IS ALL ZERO.
See the explanation for the warning message DATA IS ALL ZERO, SO FIT VALUES NOT COMPUTED.

aa PERCENT OF THE DATA IS MISSING ACCORDING TO THE SUBSCRIPT TABLE AND/OR THE MISSING DATA CODES INPUT. THEREFORE, THE SOLUTION(S) OBTAINED MAY BE UNSTABLE.
This message is output only if more than ten percent of the data is to be treated as missing. Note that diagonal entries are also treated as missing values when IGDIAG equals one (Card I-6). However, aa does not include the diagonal entries unless they are entered in the missing data subscript table or contain a missing data code.

POSITIVE DATA SIZE MULTIPLIER TO BE USED BECAUSE NEGATIVE DISTANCES ARE IMPOSSIBLE.
The absolute value of the specified DSIZE (on Record II-6) is used because distance-like data (IDATYP=1) must be positive. Other than this change, synthesis of the data proceeds normally.

PREDICTED DATA IS ALL ZERO, SO FIT VALUES NOT COMPUTED.
This message appears before output of the "true" loadings during a data synthesis run, if you minimized the systematic component in the data by specifying a very small DSIZE, for example. In this case, it is impossible to compute fit values, and so the fit values that are printed above the "true" loadings are all set to zero. Otherwise, the run proceeds as usual.
This message also appears before loadings output during data analysis, if user-supplied starting loadings were read as zeroes because of incorrect format specifications. In this case, the analysis will probably proceed, but with no change in the loadings; you will have to correct the loadings input format and rerun the job.

RESIDUALS MATRIX IS NOT DEFINED, SINCE NO ANALYSIS HAS BEEN ATTEMPTED. OUTPUT OF RESIDUALS SUPPRESSED.
You have requested that no analysis be done (NSOLS=-1 on Card I-2) but you have specified an output unit for the residuals (IUNITF on Card I-8). The program terminates execution in the normal way, but does not try to output residuals.

SIGN OPTION FOR MODE C RESET TO 2 (POSITIVITY CONSTRAINT) BECAUSE NEGATIVE DISTANCES ARE IMPOSSIBLE.
Distance-like data (IDATYP=1 on Record II-6) must be positive. However, with no positivity constraint on Mode C loadings, they may be negative and computation of distances would then involve negative square roots.

Thus, IFCPOS on Record II-6 is reset to constrain Mode C loadings to be positive.

SINCE NFACT IS EQUAL TO NAS THE REQUESTED ZERO-CORRELATION CONSTRAINT FOR MODE A MAY CAUSE PROBLEMS.
(Similar messages are output if these conditions occur for Mode B and/or Mode C.)

No parameters are reset and the analysis proceeds as usual. The problems may arise when interpreting the solution obtained, since the constraint forces one factor to have constant loadings in Mode A. (The Mode A correlations between the constant factor and the other factors are then undefined.) In practice, this happens only when forced output has occurred and IORTHA is 4. Otherwise (i.e., when IORTHA is 2 or 3), the constraints are always dropped before the solution converges and before forced output occurs. With no constraints, the factor does not remain constant in Mode A, and so the interpretation problem is avoided.
(The Mode A correlation tables following the loadings may list correlation values close to zero, instead of asterisks indicating undefined values, for a factor that is constant in Mode A. This happens if the factor loadings actually differ when represented to many decimal places by the computer, since these values are used in computing the correlations, even though the loadings are equal when rounded off to four significant digits.)

SINCE NFGEN IS EQUAL TO NAS THE REQUESTED ZERO-CORRELATION CONSTRAINT FOR MODE A MAY CAUSE PROBLEMS.
(Similar messages are output if these conditions occur for Mode B and/or Mode C.)

No parameters are reset and data synthesis proceeds normally. However, the constraint will force constant loadings (i.e., identical values for all levels) on one factor in Mode A. This, in conjunction with a request that the loadings be centered (i.e., IFAPOS=3 on Record II-6) would result in all loadings on this factor being zero. Thus, this factor would not contribute to the true part of the data at all and NFGEN would actually be one less than the number specified on Record II-4.

YOU ASKED FOR POSITIVE MODE C LOADINGS (IFCPOS=2), BUT SPECIFIED A NEGATIVE DATA SIZE VALUE.

This is a questionable combination on theoretical grounds if IDATYP on Record II-6 is 1 or 2, and so the message is issued to alert you to a possible error in specifying the parameter values. If IDATYP is 1 (for distance-like data), another message will inform you that a positive data size multiplier is used. More details are included in the explanation of that message. For IDATYP equal to 2 (scalar product data), the specified combination results in negative values, including those on the diagonal; these negative diagonals are inconsistent with real cross-product, covariance or scalar product data diagonals, which result from the squaring of values and are therefore nonnegative. No parameters are reset, and data synthesis and analysis proceeds normally, however.

8.3 ERROR MESSAGES

PARAFAC outputs an error message when it detects an error or inconsistency and halts execution rather than arbitrarily resetting parameters to get around the problem. The messages are listed below in alphabetical order by their first words (excluding the word "error" unless it is part of the sentence), along with an explanation of each. The explanations assume that the error is not due simply to misplacement of values on the input records.

DIMENSIONS OF MISSING DATA ARRAY EXCEEDED AT THE DATA POINT (i, j, k). ARRAY CURRENTLY DIMENSIONED FOR nn MISSING VALUES. POINTS FOLLOWING IN THE DATA ARRAY WERE NOT CHECKED FOR MISSING DATA CODES.

There are two possible causes for this message. One is that you have made an error in specifying missing data code values or data range limits on Card I-4. Thus the program is treating many valid data values as missing, and the missing data arrays are too small to accommodate them all. Correcting the code values and/or limits should rectify this problem. The other cause is that the missing data arrays actually are too small for all the missing data, and the DIMS program must be used to increase the array space for missing data. Use of DIMS is explained in Chapter 3.

EQUAL-AVERAGE-DIAGONAL NORMALIZATION CANNOT BE PERFORMED IF NAS DOES NOT EQUAL NBS.

The data must consist of square matrices (i.e., NAS=NBS). Change the values of IFCENA and IFCENB on Card I-4 to 3 or less.

ERROR IN CENTERING/NORMALIZATION REQUESTS.

This output is followed by one or more messages which describe the specific error made. These other messages are not included here since they are self-explanatory. Refer to the description of Card I-4 in the PARAFAC Input Specifications Table for legitimate values of IFCENA, IFCENB and IFCENC.

ERROR IN SPECIFIED BOUNDS FOR THE DATA RANGE.

The upper limit specified is less than the lower limit (DUPPER and DLOWR respectively on Card I-4). Change their values so that DUPPER is greater than DLOWR.

INVALID MISSING DATA SUBSCRIPTS FOR LINE nn IN MISSING DATA LIST. SUBSCRIPTS ARE READ AS-- i j k

This occurs either if i, j or k is less than or equal to zero, or if i, j or k is greater than NAS, NBS or NCS (as specified on Record II-2) respectively. Correct the erroneous subscript(s) at the indicated line in the missing value subscripts table.

NAS= n1 AND NBS= n2, BUT FOR DATA TYPE 1, NAS MUST EQUAL NBS.

(A similar message may appear for data type 2.)

For real dissimilarity data (type 1) and real cross-product data (type 2), the two "ways" of the data, which are Mode A and Mode B here, are assumed to refer to identical entities (e.g., objects, questionnaire items, etc.). Thus the two ways of the data have an equal number of levels or, NAS equals NBS. Therefore, it makes no sense to synthesize such data when NAS does not equal NBS. Either change the value of IDATYP on Record II-6 to zero, or set NAS equal to NBS on Record II-2.

(You can simulate data for "unfolding" analysis (i.e., data consisting of dissimilarities between one set of N things and a different set of M things) by first synthesizing the full (N+M) by (N+M) matrix and then extracting an N by M subpart for analysis.)

SINCE CONPRB IS A PROBABILITY VALUE, IT MUST BE IN THE RANGE ZERO TO ONE INCLUSIVE.

Change the value of CONPRB on Record II-7 to something in the indicated range.

SINCE NFACT IS GREATER THAN NAS, THE REQUESTED FACTOR DEPENDENCE CONSTRAINT FOR MODE A IS IMPOSSIBLE TO ACCOMPLISH.

(A similar message appears for Mode B and/or Mode C if NFACT exceeds NBS and/or NCS respectively.)

Reset the value of IORTHA (IORTHB, IORTHC) on Card I-6 to 1, so that no constraints are imposed on the factors in the indicated mode. Also, reset IORTHA (IORTHB, IORTHC) on Card -6 to 1 if NFACT on Card I-2 is greater than NAS (NBS, NCS).

SINCE NFGEN IS GREATER THAN NAS, THE REQUESTED FACTOR DEPENDENCE CONSTRAINT FOR MODE A IS IMPOSSIBLE TO ACCOMPLISH.

(A similar message appears for Mode B and/or Mode C if NFGEN exceeds NBS and/or NCS respectively.)

Reset the value of IFAORT (IFBORT, IFCORT) on Record II-6 to 1, so that no constraints are imposed on the factors in the indicated mode.

SYMMETRY FAILURE--POINT (i, j, k)= aa BUT POINT (j, i, k)= bb

Note that the symmetry check is done after missing data values are estimated and the requested centering and/or normalization performed. Usually, however, symmetry failures are not detected for symmetric input data, even after these transformations, as long as IFCENA equals IFCENB (Card I-4). The program allows a difference of up to approximately 0.0005 percent of the data values to compensate for differences introduced due to roundoff during calculations. Thus, if a symmetry failure is detected, check the input data to see if it really is symmetric, and/or check that IFCENA equals IFCENB.

The following messages indicate that the DIMS program should be used to increase the appropriate array sizes (see Chapter 3):

TASK SIZE PARAMETERS CALL FOR n1 FACTORS, BUT ARRAYS ONLY ALLOW FOR n2.
TASK SIZE PARAMETERS CALL FOR n1 LEVELS OF MODE A BUT ARRAY SIZES ALLOW ONLY n2.
(A similar message is printed for Mode B and/or Mode C if n1 exceeds n2 in those modes.)
TOO MANY ENTRIES IN THE MISSING DATA SUBSCRIPT TABLE. MISSING DATA ARRAY CURRENTLY DIMENSIONED FOR n1 MISSING VALUES.

APPENDIX A

MATRIX NOTATION

While it is assumed that users are familiar with common terms associated with matrix notation and two-way factor analysis, some terms are reviewed here. A basic concept in two-way factor analysis is a matrix. A data *matrix* is a tabular arrangement of data values into rows and columns. A *square* matrix has the same number of rows as columns, while a *rectangular* matrix has either more rows than columns, or vice versa. The size (and shape) of a matrix is denoted by the phrase "n by m" or "n X m", where n is the total number of rows and m is the total number of columns. For example, a 3x4 matrix is rectangular and consists of 3 rows and 4 columns of data values; a 3x3 matrix is square and has 3 rows and 3 columns. (By convention, the row number is designated first; as you will see, PARAFAC notation reverses the row and column order.)

A *cell* of the matrix is the intersection of a row and column (i.e., the location of a data entry in the matrix). The total number of cells in a matrix is the product of n and m, the number of rows and columns. Thus the 3x4 matrix mentioned above has 12 cells. A specific cell is denoted by the ordered pair (x,y), where x is the row and y is the column that intersect at the cell (x is not greater than n and y is not greater than m). The diagram below shows the location of the (2,3) cell in a 3x4 matrix:

```
:--:--:--:--:
:  :  :  :  :
:--:--:--:--:
:  :  : X:  :    X indicates (2,3) cell
:--:--:--:--:
:  :  :  :  :
:--:--:--:--:
```

Diagonal and off-diagonal cells are mentioned with respect to a square matrix, but these terms are meaningless in a rectangular matrix. *Diagonal* cells are those for which the row and column numbers are equal in the ordered pairs that specify them. For example, (1,1) is the first diagonal cell and (n,n) the nth diagonal cell. Together, all the diagonal cells compose the matrix diagonal. Off-diagonal cells are all cells above and below the diagonal. The following diagram indicates the diagonal of a 3x3 matrix:

```
:--:--:--:
: X:  :  :
:--:--:--:
:  : X:  :   X's denote diagonal cells
:--:--:--:
:  :  : X:
:--:--:--:
```

A symmetric matrix is a (square) matrix for which each off-diagonal (x,y) entry has the same value as each (y,x) entry. For example, in the symmetric 3x3 matrix below, the (1,2) and (2,1) cells contain equal values, as do the (2,3) and (3,2) cells and the (1,3) and (3,1) cells (there is no restriction on the diagonal values):

```
:--:--:--:
:  : 1: 8:
:--:--:--:
: 1:  : 4:
:--:--:--:
: 8: 4:  :
:--:--:--:
```

APPENDIX B

Formulae used by PARAFAC

1. Summary Statistics

a) Level Means

$$\bar{x}_{i..} = \left[\sum_{k}^{NCS} \sum_{j}^{NBS} x_{ijk} \right] / (NBS * NCS)$$

where $\bar{x}_{i..}$ is the mean for Mode A level i
x_{ijk} is the real data value
NBS is the number of levels in Mode B
NCS is the number of levels in Mode C

Parallel formulae are used for the level means of Modes B and C.

b) Level Mean Squares

$$ms_{i..} = \left[\sum_{k}^{NCS} \sum_{j}^{NBS} (x_{ijk})^2 \right] / (NBS * NCS)$$

where $ms_{i..}$ is the mean square for Mode A level i
x_{ijk}, NBS and NCS are as defined above

Parallel formulae are used for the level mean squares of Modes B and C.

c) Level Variances

$$v_{i..} = \left[\sum_{k}^{NCS} \sum_{j}^{NBS} (x_{ijk} - \bar{x}_{i..})^2 \right] / (NBS * NCS)$$

where $v_{i..}$ is the variance for Mode A level i
x_{ijk}, $\bar{x}_{i..}$, NBS and NCS are as defined above in formula 1a.

Parallel formulae are used for the level variances of Modes B and C.

2. Fit Values

a) MEAN SQUARE ERROR (for entire data set)

$$\left[\sum_{k}^{NCS} \sum_{j}^{NBS} \sum_{i}^{NAS} (x_{ijk} - \hat{x}_{ijk})^2 \right] / (NAS * NBS * NCS)$$

where NAS is the number of levels in Mode A
NBS is the number of levels in Mode B

NCS is the number of levels in Mode C
x_{ijk} is the real data value
$\hat{x}_{ijk}$ is the corresponding estimated data value.

$\hat{x}_{ijk}$ is computed from the factor loadings according to the basic proportional profiles equation.

$$\hat{x}_{ijk} = \sum_{r}^{NFACT} a_{ir} b_{jr} c_{kr}$$

where $NFACT$ is the number of factors extracted
a_{ir} is the loading of Mode A level i on factor r
b_{jr} is the loading of Mode B level j on factor r
c_{kr} is the loading of Mode C level k on factor r

b) STRESS

$$\sqrt{\left[\sum_{k}^{NCS} \sum_{j}^{NBS} \sum_{i}^{NAS} (x_{ijk} - \hat{x}_{ijk})^2\right] \Big/ \left[\sum_{k}^{NCS} \sum_{j}^{NBS} \sum_{i}^{NAS} (x_{ijk})^2\right]}$$

where NAS, NBS, NCS, x_{ijk} and $\hat{x}_{ijk}$ are as defined above for MEAN SQUARE ERROR.

3. Factor Relationships

a) Factor Cross-Products

$$C_{rs} = \left[\sum_{i}^{NAS} a_{ir} a_{is}\right] \Big/ \left[\sqrt{\sum_{i}^{NAS} a_{ir}^2 \sum_{i}^{NAS} a_{is}^2}\right]$$

where C_{rs} is the cross-product between factors r and s on the Mode A
a_{ir} is the loading of Mode A level i on factor r
a_{is} is the loading of Mode A level i on factor s
NAS is the number of levels in Mode A.

The formula for the cross-product between factors r and s on Mode B can be defined by replacing "NAS" with "NBS", "a_{ir}" and "a_{is}" with "b_{ir}" and "b_{is}", and "Mode A" with "Mode B" in the above formulae and/or definitions.

Similarly, the formula for the cross-product between factors r and s on Mode C can be defined by replacing "NAS" and "Mode A" with "NCS" and "Mode C", etc.

b) Factor Correlations

$$r_{st} = \left[\sum_{i}^{NAS} (a_{is} - \bar{a}_{.s})(a_{it} - \bar{a}_{.t})\right] \Big/ \left[\sqrt{\sum_{i}^{NAS} (a_{is} - \bar{a}_{.s})^2 \sum_{i}^{NAS} (a_{it} - \bar{a}_{.t})^2}\right]$$

where r_{st} is the correlation between factors s and t on Mode A
a_{is} is the loading of Mode A level i on factor s
$a_{.s}$ is the mean Mode A loading on factor s
a_{it} is the loading of Mode A level i on factor t
$a_{.t}$ is the mean Mode A loading on factor t

NAS is the number of levels in Mode A.

Parallel formulae are used to compute the correlations between factor s and t on Modes B and C.

4. Error Analysis — Level MEAN SQUARE ERRORS

$$mse_{i..} = \left[\sum_{k}^{NCS}\sum_{j}^{NBS}(x_{ijk}-\hat{x}_{ijk})^2\right] / (NBS * NCS)$$

where $mse_{i..}$ is the mean square error for Mode A level i
x_{ijk} is the real data value
$\hat{x}_{ijk}$ is the estimated data value (see formula 2a above for the estimation equation)
NBS is the number of levels in Mode B
NCS is the number of levels in Mode C

Parallel formulae are used for the level mean squares of Modes B and C.

5. Centering

a) Mode A Centering

$$\overset{*}{x}_{ijk} = x_{ijk} - \bar{x}_{.jk}$$

where $\bar{x}_{.jk} = \dfrac{1}{NAS}\sum_{i}^{NAS} x_{ijk}$

b) Mode B Centering

$$\dot{x}_{ijk} = x_{ijk} - \bar{x}_{i.k}$$

where $x_{i.k} = \dfrac{1}{NBS}\sum_{j}^{NBS} x_{ijk}$

c) Mode C Centering

$$\overset{o}{x}_{ijk} = x_{ijk} - \bar{x}_{ij.}$$

where $\bar{x}_{ij.} = \dfrac{1}{NCS}\sum_{k}^{NCS} x_{ijk}$

If more than one mode is centered, the centering operations are applied to the data sequentially, each step taking as input the output from previous centering operation.

6. "Normalization" (Variance or Mean-Square Standardization)

a) Mode A Normalization

$$\overset{*}{x}_{ijk} = x_{ijk}/\sqrt{ms_i}$$

where $ms_i = \sum_{j}^{NBS}\sum_{k}^{NCS}(x_{ijk})^2/NBS * NCS$

b) Mode B Normalization

$$\overset{*}{x}_{ijk} = x_{ijk}/\sqrt{ms_j}$$

where $ms_j = \sum_i^{NAS} \sum_k^{NCS} (x_{ijk})^2/NAS * NCS$

c) Mode C Normalization

$$\overset{*}{x}_{ijk} = x_{ijk}/\sqrt{ms_k}$$

where $ms_k = \sum_i^{NAS} \sum_j^{NBS} (x_{ijk})^2/NBS * NCS$

If more than one mode is normalized, (or if normalization is combined with centering) the operations are performed in sequence and the sequence is repeated iteratively until all requested relations are satisfied within ± .1%. A final centering stage insures that all centering relations will be satisfied "exactly" (i.e. within machine roundoff error).

7. Equal-Average-Diagonal Normalization

$$\overset{*}{c}_{ijk} = c_{ijk}/[(\bar{c}_{ii\cdot})^{1/2} * (\bar{c}_{jj\cdot})^{1/2}]$$

where $\overset{*}{c}_{ijk}$ is the rescaled covariance value

c_{ijk} is the unscaled covariance

$\bar{c}_{ii\cdot} = \frac{1}{p}\sum_k^{NCS} c_{iik}$ and $\bar{c}_{jj\cdot} = \frac{1}{p}\sum_k^{NCS} c_{jjk}$

NCS is the number of levels in Mode C

APPENDIX C

RANDOM NUMBER GENERATOR

The code for the random number generator used by PARAFAC was adapted from Schrage (1979). The generator is machine independent so long as the machine can represent 31-bit integers exactly, either in single or double precision format. It is a full cycle congruential generator; in the cycle, every integer from 1 to 2**31-2 = 2147483646 is generated once, and only once, in a random sequence. The first integer in the sequence, INT(1), is set to some arbitrary integer value in the range 0<INT<2147483647. Subsequent integers are obtained via the recursion

INT(i+1) = A*INT(i) mod P,
where A=7**5=16807 and P=2**31-1=2147483647.

The generator also returns a random fraction in the range (0,1). PARAFAC calls the generator to approximate a random selection from a uniform or rectangular distribution in the range (0,1).

See Schrage (1979) for a discussion of its statistical properties and a comparison with other generators.

C.1 SIMULATING OTHER DISTRIBUTIONS

The random number generator described above is the basis of the various simulated distributions that are used to produce random factor loadings, factor size multipliers and error components. The expressions used to simulate the distributions are given below. Variables appearing in the expressions are defined first.

a(i,r) =loading for factor r at ith level of Mode A
b(j,r) =loading for factor r at jth level of Mode B
c(k,r) =loading for factor r at kth level of Mode C
X(i,j,k) = $\sum_{r}$ a(i,r)*b(j,r)*c(k,r)
=systematic component of the data value

R =uniform random number in the range (0,1); returned after one call to the generator
SUM2R =sum of two uniform random numbers in the range (0,1); requires 2 calls to the generator
SUM18R =sum of 18 uniform random numbers in the range (0,1); requires 18 calls to the generator
ESIZE =ERRSIZ for data points that are not outliers
=ERRSIZ*CONSIZ for outliers (see Record II-7)

C.1.1 Random Factor Loadings (ISTART=0 or ILDGIN=0)

Loadings are randomly selected from a simulated uniform distribution in the range (-1,1). The distribution is simulated via

(R*2.) - 1.

Subsequent normalization or rescaling of the factors before output may change these initial values.

C.1.2 Factor Size Multipliers

ISZFAC=1: Each multiplier is randomly selected from a simulated triangular distribution in the range (0.1,1.9), with a mean of 1.0. The distribution is simulated via

(SUM2R*0.9) + 0.1

ISZFAC=2: Each multiplier is randomly selected from a simulated rectangular distribution in the range (0.1,1.9), with a mean of 1.0. The distribution is simulated via

(R*1.8) + 0.1

C.1.3 Random Error Component

IERTYP=1: The error is randomly selected from a simulated uniform distribution with a mean of zero and a standard deviation of 1.0, and then scaled up or down by ESIZE. The distribution is simulated via

$(R-0.5)*\sqrt{12.} * ESIZE$

IERTYP=2: Same as IERTYP=1, except the error component is also scaled by the value of the systematic component. The distribution is simulated via

$(R-0.5)*\sqrt{12.} * ESIZE * X(i,j,k)$

IERTYP=3: The error is randomly selected from a simulated normal distribution with a mean of zero and standard deviation of 1.0, and then scaled by ESIZE. The distribution is simulated via

((SUM18R-9.0)/$\sqrt{1.5}$) * ESIZE

IERTYP=4: Same as IERTYP=3, except the error component is also scaled by the value of the systematic component. The distribution is simulated via

((SUM18R-9.0)/$\sqrt{1.5}$) * ESIZE * X(i,j,k)

IERTYP=5: The computation used to obtain the value of a data point contaminated with lognormal error is:

SIGN(1.0,X(i,j,k))*|X(i,j,k)|*EXP(((SUM18R-9.0)/$\sqrt{1.5}$)*ESIZE)

(SIGN is an intrinsic Fortran function, and EXP is a basic external Fortran function.)

The error deviate that serves as the exponent is obtained in the same way as described for IERTYP=3.

IERTYP=6: The error is randomly selected from a simulated "slash" distribution, equivalent to a normal distribution with a mean of zero and standard deviation of 1.0 (cf. IERTYP=3) that is divided by a uniform distribution in the range (0,2). The error value is then scaled by ESIZE. The distribution is simulated via

(((SUM18R-9.0)/$\sqrt{1.5}$) / (R*2.)) * ESIZE

IERTYP=7: Same as IERTYP=6, except the error component is also scaled by the value of the systematic component. The distribution is simulated via

(((SUM18R-9.0)/$\sqrt{1.5}$) / (R*2.)) * ESIZE * X(i,j,k)

APPENDIX D

PARAFAC PACKAGE INSTALLATION (VERSION 6H)

The following information is provided to minimize the effort necessary to get the PARAFAC programs running on your system. Sections D.1 and D.2 list the characteristics and contents of the magnetic tape, and Section D.3 discusses general aspects of program installation (e.g., user access to files). Section D.4 describes general and specific program characteristics (e.g., I/O units and maximum values), and supplies estimates of computer core requirements. Instructions for modifying these features are given in Section D.5; Cyber users must always make the changes noted in Section D.5.3. Additional modifications (e.g., for array sizes and width of lineprinter output) are explained in Section D.6.

D.1 TAPE SPECIFICATIONS

1. 9-track
2. IBM EBCDIC characters
3. Unlabelled
4. Density=1600 bpi
5. Blocksize=160
6. Logical record length=80
7. Odd parity

D.2 TAPE CONTENTS

There are 11 files altogether. Files 2 and 7-11 inclusive contain Fortran source code for programs in the PARAFAC package, while the other files are for testing tape reading and program installation.

1. (37 records) -Descriptive information about the program package. It is included to check your tape reading accuracy (see Appendix E for a copy of its contents). Note that for this file only, column 1 of each record is blank.

2. (3929 records) -Source code for synthesis version of PARAFAC program (i.e., program to generate as well as analyse data).
3. (17 records) -Input parameter file to be used in testing PARAFAC synthesis program (i.e., file 2).
4. (147 records) -Data file that should be output if PARAFAC executes correctly using file 3 as input (may be small differences due to round-off).
5. (128 records) -Loadings file that should be output if PARAFAC executes correctly using file 3 as input (may be small differences due to round-off).
6. (8 records) -Input parameters to be used (with file 4 as input data) in testing PARAFAC synthesis and non-synthesis versions (i.e., files 2 and 7 respectively).
7. (3264 records) -Source code for non-synthesis version of PARAFAC (i.e., program to analyse data but not synthesize it).
8. (187 records) -Source code for DIMS (i.e., program to redimension arrays in PARAFAC).
9. (1214 records) -Source code for PFPLOT (i.e., program to plot factor loadings).
10. (780 records) -Source code for CMPARE (i.e., program to compare factors from different solutions).
11. (564 records) -Source code for DISTIN (i.e., program to transform data before input to PARAFAC for analysis).

D.3 PROGRAM INSTALLATION

System installation of tape files 2 and 7-11 is discussed below. Hereafter, the files are referred to as PARAFAC(S), PARAFAC(NS), DIMS, PFPLOT, CMPARE and DISTIN respectively. PARAFAC is used whenever a comment applies to <u>both</u> the PARAFAC(S) and PARAFAC(NS) files.

As noted above, PARAFAC(S) can both synthesize and analyse data, while PARAFAC(NS) can only analyse data. PARAFAC(NS) requires less core memory to load and it may be possible to use it should PARAFAC(S) exceed the core limits of a small computer system. (Core requirements are given in Section D.4.3.) Otherwise, it is only necessary to install PARAFAC(S).

As specified in the multi-user lease agreements, user acess to the source code of all programs, except for the main program of PARAFAC(S) and PARAFAC(NS), is prohibited. Thus, all programs should be compiled and stored so that users only have access to the compiled files. (This does not apply for single-user agreements.)

When installing PARAFAC under a multi-user agreement, the main program must first be separated from the subroutines. This can be accomplished by running DIMS, which creates a file containing only the source for the main program (DIMS parameter values for this are given in the footnote below). Or, you can split the PARAFAC program at line 1150 (line numbers are discussed in D.4.1.2 below) and put everything up to and including line 1150 (i.e., the main program) in one file, and all subsequent code (i.e., the subroutines) in a second file. The main source code and the compiled subroutines should then be made accessible to all users. Users must run DIMS to get their own redimensioned copy of the main program, and then compile and link it to the other compiled subroutines to run a PARAFAC job.

Those under single-user agreements may use the procedure noted above for PARAFAC installation. Or, if they do not want a separate main created whenever they run DIMS, they may remove the comment line numbered 1140 in the PARAFAC program (i.e., C DECK END THIS CARD MUST BE INCLUDED IF PROGRAM DIMS IS TO BE RUN). Then when DIMS is run, the entire PARAFAC source (i.e., redimensioned main and all subroutines) are written to a new file. The new source may then be compiled and run without linkage to any other subroutine files. This method may be simpler for those who are less familiar with computer techniques, but it requires more disk storage space.

D.4 PROGRAM CHARACTERISTICS

D.4.1 General Features

Installation of the programs on most systems should require little effort. The source code is portable and well documented by comments, and the lines of the code are numbered. All program variables are single-precision, except the seed for the random number generator in PARAFAC, and all arithmetic computations are also single-precision.

To create a separate PARAFAC main identical to the one on the tape, the two DIMS parameter cards should contain:

```
STANDARD PROGRAM
   18   18   35   10   50
```

D.4.1.1 Portability. - All programs except PFPLOT conform to PFORT specifications. PFORT is a portable subset of American National Standard Fortran X3.9-1966 (published by American Standards Association, Inc., 10 East 40th Street, New York, N. Y. 10016). The minor deviations of the PFPLOT code from PFORT (i.e., alphanumeric variables contain 2 characters rather than 1 as specified by PFORT) should cause no problems on most systems. However, if your system restricts the contents of an alphanumeric variable to 1 character, PFPLOT cannot be installed (since it cannot easily be modified to work with 1 character per variable).

D.4.1.2 Line Numbers. - The records of each program are consecutively numbered in steps of 10, with the line number right-justified in columns 76-80 of the line (leading zeroes suppressed). For example, the first record of PARAFAC(S) contains 10 in columns 79-80 and the last contains 39290 in columns 76-80. Line numbers are referred to below in the instructions for altering program features.

D.4.1.3 Note For Cyber Users Only. - "PROGRAM" statements are not part of the executable code in any of the programs. Such statements must be included for the programs to run. See Section D.5.3 below for instructions.

D.4.2 Specific Features

Listed below are specific program requirements. Depending on your system, some may have to be modified. Instructions for doing so are provided in the next section.

1. Input and Output	Unit Number	Program
Standard input (e.g., cards)	5	all
Standard output (e.g., lineprinter)	6	all
Default for data output	7	PARAFAC, DISTIN
"Scratch" file for I/O during program execution	1	CMPARE only
Data input	1	DIMS only
Data output	2	DIMS only

2. Maximum Values	Program
Floating point: 10.E30	PARAFAC, DISTIN
Integer: (2**31)-1 (i.e., 2147483647)	PARAFAC only

D.4.3 Core Requirements

Estimates of the computer core necessary to load each program are shown below (1K=1024 words; estimates are based on compilation by a CDC Cyber 170). Not included in the "total" and "program only" estimates is the space required by I/O buffers, which is system-dependent. The "array only" estimates are based on standard arrays (i.e., with no changes made); if PARAFAC arrays have been redimensioned via DIMS, a new estimate can be computed from the information printed on the DIMS listing.

If the total core requirement for any program exceeds the capacity of your system, it may be reduced by decreasing the size of some arrays. Instructions for doing so are provided in Section D.6.1.

Program	Total	Program Only	Arrays Only
PARAFAC(S)	41K	27K	14K
PARAFAC(NS)	34K	20K	14K
DIMS	11K	11K	---
PFPLOT	32K	10K	22K
CMPARE	46K	13K	33K
DISTIN	14K	11K	3K

D.5 PROGRAM MODIFICATIONS

In the instructions below, the line to be altered in the source code is identified by the number contained in columns 76-80 of the line (e.g., line 10 means the line <u>numbered</u> 10, not the tenth line of the file).

D.5.1 I/O Changes

1. <u>Standard Input and Output</u>

Consecutive assignment statements (i.e., ISTDIN = 5; ISTDOU = 6) in the source code of each program set the standard input and output units to 5 and 6 respectively. To reset the standard input unit, change 5 in the first statement; to reset the standard output unit, change 6 in the second statement. The appropriate line numbers for these changes are listed below; the first number applies to the statement for the input unit and the number in parentheses refers to the statement for the output unit):

DIMS -- Line 490 (500)
PFPLOT -- Line 2090 (2100)
CMPARE -- Line 1580 (1590)

DISTIN -- Line 970 (980)
PARAFAC -- Either run DIMS, specifying the desired numbers in the parameter file (recommended procedure) or modify line 790 (800)

2. Default Diskfile Output
PARAFAC -- Either run DIMS (recommended), or change 7 in the assignment statement on line 810 (ISTDLD = 7).
DISTIN -- Change 7 in the assignment statement on line 990 (ISTDTD = 7).

3. CMPARE "Scratch" I/O File
Change 1 in the assignment statement on line 1860 (IUNIT = 1).

4. DIMS Data Input and Output
Input unit -- Change 1 in the assignment statement on line 510 (IUNITA = 1).
Output unit -- Change 2 in the assignment statement on line 520 (IUNITB = 2).

D.5.2 Maximum Values

1. Floating Point

The maximum floating point value is set in an assignment statement (VLARGE = 10.E30). If your system cannot represent 10.E30, change this value to the maximum allowed on your system.

PARAFAC -- Line 930
DISTIN -- Line 1020

2. Integer

If your system does not provide for a minimum 32-bit representation of integers, it cannot represent the value (2**31)-1 as an integer (i.e., 2147483647). In this case, you must use the double-precision floating point version of the PARAFAC subroutine that generates random numbers (all computations in it involve double-precision arithmetic). This version will work on any system that can represent (2**31)-1 exactly in double-precision (i.e., any system with a minimum 16-bit representation of integers).

Implementing the double-precision floating point version involves two steps. The procedure is the same for both PARAFAC(S) and PARAFAC(NS), but the line numbers are different. The numbers in parentheses in the instructions below apply to PARAFAC(NS).

1. Lines 38520-38840 (31870-32190) inclusive:
Either delete from the source code or replace all blanks in column 1 by "C" (i.e., change all executable statements to comments). This step

eliminates the integer version of the random number generator.

2. Lines 39020-39270 (32370-32620) inclusive: Replace the characters "CX" in columns 1-2 with 2 blanks (but do not alter lines with "C" in column 1 and blank in column 2). This procedure transforms the code for the double-precision floating point version from comments to executable statements.

D.5.3 Cyber Users Only

A PROGRAM statement with "C" in column 1 has been included near the beginning of each program. You can either replace the "C" in column 1 of that line (and its continuation lines, if any) with a blank to make the statement executable, or you can insert your own PROGRAM statement at the beginning of the program. Note that if you change the I/O units (described above), you must also modify the PROGRAM statement provided in the source code if you want to use it. In any case, the user should be informed of which units are permissible to use. Currently, PROGRAM statements are as follows:

Program	Line(s)	I/O Units Specified
PARAFAC	130,140	1-9 inclusive
DIMS	10	1, 2, 5 and 6
PFPLOT	10	1, 5 and 6
CMPARE	10-40	1-28 inclusive
DISTIN	10,20	1-7 inclusive

Using the PROGRAM statements provided (i.e., without modification) will add approximately 1K per I/O unit to the core memory requirements of the programs, assuming the default buffer size is 1K. For example, the 9 units specified in PARAFAC will increase the total PARAFAC core to load, given in Section D.4.3, by roughly 9K.

D.6 MISCELLANEOUS MODIFICATIONS

D.6.1 Array Dimensions

Except for PARAFAC, changes in array size should generally be necessary only if the current core memory requirements exceed the computer capacity; the arrays of the other programs are large so that they can accommodate most data without modification. The DIMS program enables each user to redimension the PARAFAC arrays for himself and thus no changes are necessary during installation of PARAFAC. For the other programs, however, any changes must be made during program installation, since user access to the source code is not allowed (unless the lease is a single-user agreement). Listed below are the program limits, along with instructions for changing them if necessary.

PARAFAC. Both versions of PARAFAC can accommodate data sets up to 18x18x35 in size (number of levels in Modes A, B and C respectively), with up to 50 missing values, and can extract up to 10 factors. The arrays thus require about 14K single-precision words of memory. To modify any of these limits, use DIMS. The DIMS lineprinter output tells you how much space the newly dimensioned arrays require.

PFPLOT. As dimensioned, the arrays of PFPLOT require about 22K single-precision words of memory. Two types of limits must be adhered to:

1. Mode level limits -- The maximum number of levels in each mode is 250.
2. Product limits -- The product is the number of levels per mode times the number of factors. For both Modes A and B, the maximum allowed product is 1500; for Mode C, the maximum allowed is 3500. For example, up to 14 factors can be plotted if the data set has 100 levels in each of Modes A and B and 250 levels in Mode C, but only 6 factors can be accommodated if there are 250 levels in each mode.

One or more of the following changes may be made if required:

1a). Modifying the mode level limits
Mode A -- Change 250 in MXNAS = 250 (line 1990) and in CH1(250) (line 1680).
Mode B -- Change 250 in MXNBS = 250 (line 2000) and in CH2(250) (line 1680).
Mode C -- Change 250 in MXNCS = 250 (line 2010) and in CH3(250) (line 1680).

Note: If the limits for any mode are increased beyond 250, also delete lines 5470, 5480 and 5580-5610 inclusive (6 lines altogether). In this case, you should also make the modifications listed in 1b) below. Otherwise, you will have to specify labels for all levels of the mode(s), as PFPLOT only assigns defaults for a maximum of 250 levels.

1b). Change 250 in CHCOD1(250) (lines 1710, 4490, 7010 and 8890).
Insert character codes (labels) for CHCOD1(251), CHCOD1(252), etc., starting at line 5350.

2. Modifying the product limits
Mode A -- Change 1500 in MAXA = 1500 (line 2020) and

in A(1500) (line 1600).
Mode B -- Change 1500 in MAXB = 1500 (line 2030) and in B(1500) (line 1660).
Mode C -- Change 3500 in MAXC = 3500 (line 2040) and in C(3500) (line 1660).

CMPARE. As dimensioned, the arrays of CMPARE require about 33K single-precision words of storage. Three types of limits must be adhered to:

1. Mode level limits -- The maximum number of levels in each mode is 250.

2. Factor limits -- The maximum number of factors in the merged set is 75.

3. Product limits -- The maximum product in each mode is 8750. This is the product of the number of levels in the mode and the largest number of factors in any one loadings set. The merged set will usually have the most factors, unless the input sets are large and only a few factors are selected for merging.

One or more of the following changes may be made if required:

1. Modifying the mode level limits
Mode A -- Change 250 in MXNAS = 250 (line 1460).
Mode B -- Change 250 in MXNBS = 250 (line 1470).
Mode C -- Change 250 in MXNCS = 250 (line 1480).

2. Modifying the factor limit
Change 75 in MXFACT = 75 (line 1490) and in R(75,75) (line 1390).

3. Modifying the product limits
Mode A -- Change 8750 in MAXA = 8750 (line 1500) and in A(8750,1) (line 1380).
Mode B --Change 8750 in MAXB = 8750 (line 1510) and in B(8750,1) (line 1380).
Mode C -- Change 8750 in MAXC = 8750 (line 1520) and in C(8750,1) (line 1380).

DISTIN. As dimensioned, the DISTIN arrays require about 3K single-precision words of storage. The limits are as follows:

1. The maximum number of levels in each of Modes A and B is 40.
2. The maximum number of levels in Mode C is 250.
(i.e., data up to 40x40x250 can be accommodated without redimensioning DISTIN)

One or more of the following changes may be made if required:

Mode A -- Change 40 in MXNAS = 40 (line 930). Also, change 1600 (= MXNAS * MXNBS) in DIS(1600,1) (line 870) to the new product (= new MXNAS * MXNBS).
Mode B -- Change 40 in MXNBS = 40 (line 940). Also, change 1600 (= MXNAS * MXNBS) in DIS(1600,1) (line 870) to the new product (= MXNAS * new MXNBS).
Mode C -- Change 250 in MXNCS = 250 (line 950) and in ADDCON(250) (line 870).

D.6.2 Width Of Tabular Lineprinter Output

If the programs are to be used with, say, 80-column CRTs or 72-column teletypes, printer wraparound will make PARAFAC and CMPARE tables and PFPLOT plots hard to read. (DIMS and DISTIN lineprinter output is not obscured by wraparound.) This problem can be overcome by making minor changes in the source code.

PARAFAC. Currently, PARAFAC prints tables up to 124 characters across. The user can alter this width by using DIMS, and so no change in the code need be made during installation.

PFPLOT. To use the option for two-way plots, you must have a printer capable of printing 130 characters per line; otherwise, printer wraparound makes the plot unreadable. No simple revisions to the code will overcome this limitation.

Currently, one-way plots require 12 columns, plus 15 columns per factor, up to at most 117 columns (i.e., a maximum of 7 factors). A few changes in the source code will reduce the maximum width required. For example, for output to an 80-column CRT, you will want to change the maximum number of factors from 7 to 4 (i.e., the maximum required is then 72 characters across). To do so:

1. Replace 7 by 4 on lines 9280, 9320 and 9330.
2. Replace -6 by -3 (i.e., -4+1) on line 9290.
3. Replace 7(A2,... by 4(A2,... on line 10530.

CMPARE. Tables require 3 columns, plus 6 columns per factor, up to at most 111 columns (i.e., a maximum of 18 factors). To change this requires only that the 10 on line 1560 of the code be changed to a new value. The maximum width of the output is reduced (increased) 12 columns per unit that the new value is less (greater) than 10.

For example, to reduce the output width for an 80-column CRT, you would substitute 7 for 10 on line 1560. This reduces the maximum output width by 36 columns to 75, which is within the capacity of an 80-column printer.

APPENDIX E

CONTENTS OF TAPE FILE 1

```
ABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789 =+-*/(),.$"
= EQUALS SIGN
+ PLUS SIGN
- MINUS SIGN
* ASTRISK
/ SLASH
( LEFT PARENTHESIS
) RIGHT PARENTHESIS
, COMMA
. DECIMAL POINT
$ CURRENCY SYMBOL
" QUOTATION MARKS

LISTED ABOVE IS THE FORTRAN CHARACTER SET USED IN
THE FORTRAN SOURCE CODE ON THIS TAPE.  EXCEPT FOR
THE QUOTATION MARKS, IT IS RESTRICTED TO THE
STANDARD CHARACTER SET AS DEFINED BY THE 1966
FORTRAN STANDARD (SEE BELOW).  WHILE WE HAVE
EMPLOYED QUOTATION MARKS IN COMMENT LINES
TO IMPROVE CLARITY, THEY ARE USED NOWHERE ELSE.
SINCE COMMENT LINES ARE NOT COMPILED, THE USE OF
QUOTATION MARKS IN COMMENTS SHOULD NOT AFFECT
PORTABILITY.

THIS IS THE FIRST FILE OF THE PARAFAC ANALYSIS PACKAGE
COPYRIGHT 1980 BY RICHARD A. HARSHMAN
P.A.P. IS A PROPRIATARY SOFTWARE PACKAGE, AND MAY NOT BE REPRODUCED
OR DISTRIBUTED WITHOUT PRIOR WRITTEN PERMISSION OF
RICHARD A. HARSHMAN, OR HIS DESIGNATED REPRESENTATIVE OR AGENT.
THE SYNTAX AND CONVENTIONS OF THE PROGRAMS IN THE
PARAFAC ANALYSIS PACKAGE CONFORM TO THE X3.9-1966 FORTRAN STANDARD.
SEE AMERICAN STANDARD FORTRAN, X3.9-1966
PUBLISHED BY AMERICAN STANDARDS ASSOCIATION
10 EAST 40TH STREET
NEW YORK, N. Y. 10016

NOTE--IN THIS FILE ONLY, ALL CARDS HAVE A BLANK IN COLUMN 1
```

REFERENCES

American Standard FORTRAN. (1966). New York: American Standards Association.

Andrews, D. F., Bickel, P. J., Hampel, F. R., Huber, P. J., Rogers, W. H., & Tukey, J. W. (1972). Robust estimates of location: Survey and advances. Princeton: N. J.: Princeton University Press.

Carroll, J. D., & Arabie, P. (1980). Multidimensional scaling. Annual Review of Psychology, 31, 607-649.

Carroll, J. D., & Chang, J. J. (1970). Analysis of individual differences in multidimensional scaling via an N-way generalization of Eckart-Young decomposition. Psychometrika, 35, 283-319.

Cattell, R. B. (1944). "Parallel Proportional Profiles" and other principles for determining the choice of factors by rotation. Psychometrika, 9, 267-283.

Comrey, A. L. (1973). A first course in factor analysis. New York: Academic Press.

Green, P. E., & Carroll, J. D. (1976). Mathematical tools for applied multivariate analysis. New York: Academic Press.

Haan, N. (1981). Common dimensions of personality development: Early adolescence to middle life. In D. H. Eichorn, J. A. Clausen, N. Haan, M. P. Honzik, & P. H. Mussen (Eds.), Present and past in middle life (pp. 117-151). New York: Academic Press.

Harman, H. H. (1960). Modern factor analysis. Chicago: The University of Chicago Press.

Harshman, R. A. (1970). Foundations of the PARAFAC procedure: Models and conditions for an "explanatory" multi-modal factor analysis. UCLA Working Papers in Phonetics, 16, 86 pp. Reprinted by University Microfilms International, Ann Arbor, Michigan, Order No. 10,085.

--------. (1972). PARAFAC2: Mathematical and technical notes. UCLA Working Papers in Phonetics, 22, 30-47. Reprinted by University Microfilms International, Ann Arbor, Michigan, Order No. 10,085.

--------. (1976). PARAFAC: Methods of three-way factor analysis and multidimensional scaling according to the principle of proportional profiles. PhD thesis, UCLA. Published by University Microfilms International, Ann Arbor, Michigan, Order No. 76-25,210.

Harshman, R. A., & Berenbaum, S. A. (1981). Basic concepts underlying the PARAFAC-CANDECOMP three-way factor analysis model and its application to longitudinal data. In D. H. Eichorn, J. A. Clausen, N. Haan, M. P. Honzik, & P. H. Mussen (Eds.), *Present and past in middle life* (pp. 435-459). New York: Academic Press.

Harshman, R. A., & DeSarbo, W. S. (1984). An application of PARAFAC to a small sample problem, demonstrating preprocessing, orthogonality constraints, and split-half diagnostic techniques. In H. G. Law, C. W. Snyder, Jr., J. A. Hattie, & R. P. McDonald (Eds.), *Research methods for multimode data analysis* (pp. 602-642). New York: Praeger.

Harshman, R. A., & Lundy, M. E. (1984a). Data preprocessing and the extended PARAFAC model. In H. G. Law, C. W. Snyder, Jr., J. A. Hattie, & R. P. McDonald (Eds.), *Research methods for multimode data analysis* (pp. 216-284). New York: Praeger.

--------. (1984b). The PARAFAC model for three-way factor analysis and multidimensional scaling. In H. G. Law, C. W. Snyder, Jr., J. A. Hattie, & R. P. McDonald (Eds.), *Research methods for multimode data analysis* (pp. 122-215). New York: Praeger.

Harshman, R. A., Ladefoged, P., & Goldstein, L. (1977). Factor analysis of tongue shapes. *Journal of the Acoustical Society of America*, *62*, 693-707.

Kettenring, J. (1983). A case study in data analysis. In R. Gnanadesikan (Ed.), *Proceedings of Symposia in Applied Mathematics, vol. 28, Statistical data analysis* (pp. 105-139). Providence, R. I.: American Mathematical Society.

Kim, J-O., & Mueller, C. W. (1978). *Introduction to factor analysis: What it is and how to do it*. Beverly Hills: Sage.

Kroonenberg, P. M. (1983). *Three-mode principal component analysis: Theory and applications*. Leiden, The Netherlands: DSWO Press.

Kroonenberg, P. M., & DeLeeuw, J. (1980). Principal component analysis of three-mode data by means of alternating least squares algorithms. *Psychometrika*, *45*, 69-97.

Kruskal, J. B., & Wish, M. (1978). *Multidimensional scaling*. Beverly Hills: Sage.

Law, H. G., Snyder, C. W., Jr., Hattie, J. A., & McDonald, R. P. (1984). *Research methods for multimode data* analysis. New York: Praeger.

Lingoes, J. C., & Borg, I. (1978). A direct approach to individual differences scaling using increasingly complex transformations. Psychometrika, 43, 491-519.

Ramsay, J. O. (1977). Maximum likelihood estimation in multidimensional scaling. Psychometrika, 42, 241-266.

--------. (1978). Confidence regions for multidimensional scaling analysis. Psychometrika, 43, 145-160.

Sands, R., & Young, F. W. (1980). Component models for three-way data: An alternating least squares algorithm with optimal scaling features. Psychometrika, 45, 39-67.

Schrage, L. (1979). A more portable Fortran random number generator. ACM Transactions on Mathematical Software, 5, 132-138.

Sentis, K. P., Harshman, R. A., & Stangor, C. (1983). PARAFAC three-way factor analysis of dichotomous data: A Monte Carlo study. Unpublished manuscript.

Shepard, R. N. (1972). Psychological representation of speech sounds. In E. E. David, Jr., & P. B. Denes (Eds.), Human communication: A unified view. New York: McGraw-Hill.

Takane, Y., Young, F. W., & DeLeeuw, J. (1977). Nonmetric individual differences multidimensional scaling: An alternating least squares method with optimal scaling features. Psychometrika, 42, 7-67.

Torgerson, W. S. (1958). Theory and methods of scaling. New York: Wiley.

Tucker, L. R. (1964). The extension of factor-analysis to three-dimensional matrices. In N. Frederiksen (Ed.), Contributions to mathematical psychology. New York: Holt, Rinehart & Winston.

--------. (1966). Some mathematical notes on three-mode factor analysis. Psychometrika, 31, 279-311.

--------. (1972). Relations between multidimensional scaling and three-mode factor analysis. Psychometrika, 37, 3-27.

Weeks, D. G., & Bentler, P. M. (1979). A comparison of linear and monotone multidimensional scaling models. Psychological Bulletin, 86, 349-354.

INDEX