

GENDEX

FFD

CUT

RAT

NOA

**GENDEX FOR
FRACTIONAL REPLICATION,
RESPONSE SURFACE DESIGNS,
AND ORTHOGONAL ARRAYS**

Walter T. Federer

and

Sophonie Nshinyakobeje

Abstract

A previous Technical Report, BU-1433-M, described the installation of the GENDEX package and demonstrated the use of five modules to construct experiment designs. This report describes the use of the other four modules of the GENDEX package which may be used for constructing fractional replicates, FRD, of a complete factorial treatment design and for constructing response surface, RSD, treatment designs (FFD), blocking FRD's and RSD's (CUT), constructing FRD's and RSD's to account for trends (RAT), and constructing orthogonal arrays (NOA).

The constructions are optimal or near optimal with emphasis being placed on orthogonal or near-orthogonal treatment designs or orthogonal arrays. Resolution III, main effect, Resolution IV, main effect and sums of two-factor interactions, and Resolution V, main effects and two-factor interactions designs may be constructed using the FFD module of this package. Two-level and three-level factors may be used. The number of factors in the FFD module is limited to 15 and the number of runs (treatments or combinations) is limited to 100. For the CUT module, the number of factors m must be less than or equal to 21 and the number of runs n must be less than or equal to 64; the number of blocks is $b < n/2$ and b is a multiple of n . For the RAT module, $m_0 \leq 2$ trend variables, $m \leq 21$ factors including the m_0 trend variables, and $n \leq 64$ runs may be used. For the NOA module, m and n must be less than or equal to 64 and two level orthogonal arrays may be constructed.

Introduction

User's notes for the four modules FFD (fractional factorial design), CUT(blocking), RAT (robust against trends), and NOA (near orthogonal array) are available with the GENDEX software package. This note is intended to supplement some of the material available with the package and to complement the discussion presented in an earlier Technical Report, BU-1433-M, by the authors. Computer commands for various examples are given. A description of the output is also presented.

FFD Module For FFD's

The FFD module is a program for constructing orthogonal or near-orthogonal two and three level fractional replicates, FRD, of a complete factorial. An existing design may be augmented with additional factors. The constraints imposed on the number of 2-level factors, m , and the number of combinations or runs, n , are that $1 \leq m \leq 15$ and $3 \leq n \leq 100$ for Resolution III treatment designs. (Note: User's Note states 128 but using this number results in an error message.) For Resolution V treatment designs, $m \leq 7$. Use of m greater than 7 results in the error message "invalid floating point operation". For Resolution IV treatment designs, $m < 8$. Let

p denote the number of parameters,
 m_0 denote the number of factors in an existing design,
 m_3 denote the number of 3-level factors,
 n_1 denote the number of runs, combinations, at the high level, +1, or the low level, -1,
 -5 denote that a Resolution V design is desired,
 -4 denote that a Resolution IV design is desired,
 -p denote that the inverse of the $X'X$ matrix is to be obtained,
 +1 and -1 denote the two levels of a 2-level factor,
 +1, 0, and -1 denote the three levels of a 3-level factor,
 the default for $m_3 = 0$ be zero,
 the default for $m_0 = 0$ be zero, and
 the default for n_1 be $n/3$.

Constructing 2-level designs - The first illustration below is based on the FFD module. Suppose we want to construct a fractional replicate with three factors each with two levels. Let $m = 3$ for this case. A main effect design, Resolution III, with $n = 4$ runs is obtained with the GENDEX command:

Command: FFD m=3 n=4 -p tries=20

The option -p is used to obtain the inverse of the $X'X$ matrix.

To obtain a copy and printed output of this command, use the command

Command: COPY FFD.OUT PRN (or PRINT FFD.OUT for printed copy only)

The printed output from this command is given below:

NOTE: A design with $m=3$, $p=4$ and $n=4$

```
rand_seed try iter      f          det      std. det
741060029  1    1    0.0  2.560E+0002  1.000E+0000
```

Factor levels

```
X
  1    1    1
 -1    1   -1
 -1   -1    1
  1   -1   -1
```

X'X

```

4   0   0   0
   4   0   0
     4   0
       4

```

inv(X'X)

```

0.2500  0.0000  0.0000 -0.0000
         0.2500  0.0000 -0.0000
                0.2500 -0.0000
                        0.2500

```

NOTE: FFD used 0.12 seconds.

Rand_seed is the random number used to start the iteration process in obtaining the fractional replicate. Unless specified, the random seed is set by the system clock. Try gives the number of attempts, tries, used to obtain the given output. Here the FFD was obtained on the first try. Iter is the number of iterations used to obtain the FRD. The letter f denotes how close to orthogonality the construction is. The program automatically stops when $f = 0$. When $f = 0$ as for the above example, an orthogonal FFD is obtained. When orthogonality is achieved, $f = 0$ and the $X'X$ matrix has the form

$$\begin{array}{ccc}
 n & \mathbf{0}'_r & -\mathbf{b}\mathbf{1}'_m \\
 \mathbf{0}_r & \text{diag}(b, \dots, c) & \mathbf{0}_{r \times m} \\
 \mathbf{b}\mathbf{1}_m & \mathbf{0}_{m \times r} & \mathbf{c}\mathbf{J}_m + d\mathbf{I}_m
 \end{array}$$

where \mathbf{J} is a matrix of ones, \mathbf{I} is the identity matrix, $\mathbf{0}$ is a matrix of zeros, $r = m + m(m-1)/2$, and b and c are scalars on the diagonal. If $c = b^2/n$, the design is said to be orthogonal. If option -4 is used, f is followed by an asterisk when the design is a Resolution IV design. Thus, the smaller the value of f , the better the design. The determinant of $X'X$ is given under det. The next item, std. det, is the standardized form of the determinant, which is one for orthogonal designs with n on the main diagonal, and is the determinant of $X'X$ divided by n^p . The best design X with the factor levels as columns is given as is the $X'X$ matrix. The -p command gives the inverse matrix of $X'X$, i.e., $\text{inv}(X'X)$. This inverse matrix times an estimate of the error variance results in the variance-covariance matrix for the $p = 4$ parameter estimates, the over-all mean, factor A effect, factor B effect, and factor C effect. The effect for factor B is the sum of the first two observations minus the sum of the last two divided by $n = 4$.

For the second example, it is desired to have a Resolution V design for a FRD with $m = 5$ factors with two levels each. This will require $n = 16$ runs. The fifth factor is to be added to an existing 2^4 factorial treatment design. We start with a Resolution V design which is a 2^4 factorial obtained with the first command below. The next step is to make the outfile from this command as the infile for the next command. Then a command is used which states that there are to be $m = 5$ factors, four of which are to come from the infile, i.e. $m_0=4$. Use the TYPE command to display the output on the computer screen.

```

Commands:   FFD m=4 n=16 -5 tries=20

            COPY FFD.OUT=FFD.IN

            FFD m=5 m0=4 n=16 -5 tries=20

            TYPE FFD.OUT (displays output on screen)

            COPY FFD.OUT PRN (or PRINT FFD.OUT)

```

The output from the above commands results in the following:

NOTE: A design with $m=5$, $p=16$ and $n=16$

```
rand seed try iter      f      det      std. det
137877041  1    4    0.0  1.845E+0019  1.000E+0000
```

Factor levels

```
 1  -1  1  1  1
 1  -1 -1 -1  1
 1  1  -1  1  1
-1 -1  1  1 -1
-1  1 -1  1 -1
-1 -1 -1 -1 -1
-1 -1  1 -1  1
 1  1  1 -1  1
 1  1 -1 -1 -1
 1 -1  1 -1 -1
-1  1  1  1  1
 1  1  1  1 -1
 1 -1 -1  1 -1
-1 -1 -1  1  1
-1  1  1 -1 -1
-1  1 -1 -1  1
```

X'X

```
16  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
   16  0  0  0  0  0  0  0  0  0  0  0  0  0  0
      16  0  0  0  0  0  0  0  0  0  0  0  0  0
         16  0  0  0  0  0  0  0  0  0  0  0  0
            16  0  0  0  0  0  0  0  0  0  0  0
               16  0  0  0  0  0  0  0  0  0  0
                  16  0  0  0  0  0  0  0  0  0
                     16  0  0  0  0  0  0  0
                        16  0  0  0  0  0
                           16  0  0  0
                              16  0  0
                                 16  0
                                    16
```

NOTE: FFD used 0.33 seconds.

The constructed FRD is an orthogonal design, and therefore $f = 0$.

Constructing 3-level designs

The first example consists of eight factors with three levels each, i.e., $m_3 = 8$. The number of runs required to provide a saturated main effect design, i.e., Resolution III, is $n = 17$. The command is:

```
Command:      FFD m3=8 n=17
              PRINT FFD.OUT
```

These two commands give the following output:

NOTE: A design with $m=8$, $p=9$ and $n=17$

rand_seed	try	iter	f	det	std. det
597436310	1	20	7.0	1.583E+0009	1.334E-0002
995046084	3	13	6.0	1.600E+0009	1.349E-0002
282897014	5	15	3.0	1.650E+0009	1.391E-0002
1489479157	46	21	2.0	1.666E+0009	1.405E-0002
84185623	315	20	1.0	1.683E+0009	1.419E-0002

Factor levels

0	0	0	1	-1	0	-1	1
-1	0	-1	0	0	0	0	0
1	-1	0	0	0	0	1	1
-1	1	1	-1	0	1	0	0
-1	-1	1	-1	0	-1	-1	1
0	0	0	0	1	1	0	1
-1	-1	0	1	1	1	1	0
0	0	1	1	-1	1	0	-1
1	0	0	0	-1	0	0	0
1	0	1	1	0	-1	0	0
-1	0	-1	1	0	-1	-1	-1
1	1	-1	0	1	1	-1	0
0	-1	-1	-1	-1	0	1	-1
1	-1	0	-1	1	0	-1	-1
0	1	1	0	1	-1	1	-1
0	1	0	-1	-1	0	0	0
0	1	-1	0	0	-1	1	1

X'X

17	0	0	0	0	0	0	0	0
	10	0	0	0	0	0	0	0
		10	0	0	0	0	0	0
			10	0	0	0	0	0
				10	0	0	0	0
					10	1	0	0
						10	0	0
							10	0
								10

NOTE: FFD used 846.27 seconds.

Note that no tries were specified and the program ran until it was stopped when f was one. A near-orthogonal design with only a single non-diagonal, non-zero value in the $X'X$ matrix was produced. On another run for this example, the program ran for about 20 minutes and produced a similar design on try number 4,003.

For the second 3-level factorial example, a saturated Resolution V design is produced by the following command:

Command: FFD m3=3 n=19 -5 tries=30

This command produced a Resolution V design for the three factors and their two-factor interactions. The following output was obtained:

NOTE: A design with $m=3$, $p=10$ and $n=19$

```

rand_seed try iter      f      det      std. det
1063202655  1    6    8.5  1.174E+0009  1.916E-0004

```

Factor levels

```

  0  -1  0
  1   0 -1
 -1  0 -1
  0  0  1
  0  0  0
  1  0  0
  1 -1  1
  0  1 -1
  0 -1  0
 -1 -1 -1
  1  1  1
 -1  1  0
  1 -1 -1
 -1  1  1
  0  1 -1
 -1  0  0
 -1 -1  1
  0  0  1
  1  1  0

```

X'X

```

19  0  0  0  0  0  0  12  12  12
   12  0  0  0  0  0  0  0  0  0
      12  0  0  0  0  0  0  0  0
         12  0  0  0  0  0  0  0
            8  2  0  0  0  0
               8  0  0  0  0
                  8  2  0  0
                     12  8  8
                        12  8
                           12

```

NOTE: FFD used 20.98 seconds.

If the program had been allowed more than 30 tries, it is possible that a better design with a smaller f value could have been obtained. Alternatively, a fresh start may also have produced a better design.

Constructing mixed 2-level and 3-level designs

In this first example we want to construct an FRD for obtaining main effects estimates for two three level factors and two two level factors as well as their two-factor interactions. To do this, use the following command:

Command: FFD $m=4$ $m3=2$ $n=20$ -5 tries=30

A Resolution V design for this FRD is obtained. The combinations X and the $X'X$ matrix for the constructed FRD are given below:

NOTE: A design with m=4, p=13 and n=20

rand_seed	try	iter	f	det	std. det
530345479	1	9	105.4	2.210E+0013	2.697E-0004
1030233982	4	9	101.4	2.111E+0013	2.577E-0004
2002873402	6	15	64.6	4.832E+0013	5.898E-0004

Factor levels

0	1	-1	1
1	-1	-1	-1
1	1	1	1
1	0	1	-1
0	0	1	1
1	0	-1	1
0	-1	1	-1
-1	0	-1	1
0	0	-1	-1
1	1	-1	-1
-1	0	1	-1
1	-1	1	1
-1	-1	-1	-1
-1	1	1	1
0	0	-1	1
-1	1	-1	-1
0	1	1	-1
0	0	1	-1
0	-1	-1	1
-1	-1	1	1

X'X

20	0	0	0	0	0	0	0	0	0	0	12	12
	12	0	0	0	0	0	0	0	0	0	0	0
		12	0	0	0	0	0	0	0	0	0	0
			20	0	0	0	0	0	0	0	0	0
				20	0	0	0	0	0	0	0	0
					8	0	0	0	0	0	0	0
						12	4	0	0	0	0	0
							12	0	0	0	0	0
								12	4	0	0	0
									12	0	0	0
										20	4	4
											12	8
												12

NOTE: FFD used 29.31 seconds.

The value of f is not near zero and perhaps another run or more tries will result in a design with a smaller f value.

For a second example, suppose we wish to have a saturated main effect plan for six three level factors, m3 = 3, and six two level factors. To do this, use the following command:

Command: FFD m=12 m3=6 n=19 tries=30

A saturated Resolution III main effect plan is constructed for six 3-level factors and six 2-level factors from the above command. To obtain a print-out, use the following command:

Command: PRINT FFD.OUT

The following design X and the X'X matrix were obtained:

NOTE: A design with m=12, p=13 and n=19

rand_seed	try	iter	f	det	std. det
2062499656	1	40	85.0	1.921E+0015	4.567E-0002
1942280010	2	52	54.0	2.140E+0015	5.088E-0002
1273629480	15	39	49.0	2.135E+0015	5.077E-0002

Factor levels

0	0	0	1	1	-1	-1	-1	-1	1	-1	1
1	1	0	1	0	0	-1	1	1	-1	-1	1
-1	0	1	1	1	1	1	1	1	-1	1	-1
-1	1	-1	-1	0	-1	1	1	-1	-1	1	1
-1	1	1	-1	0	-1	-1	-1	1	-1	-1	-1
0	-1	1	0	1	0	-1	1	-1	-1	-1	1
0	0	-1	-1	1	0	-1	1	-1	1	1	-1
1	-1	1	0	1	-1	1	-1	-1	-1	1	-1
0	0	-1	1	-1	0	1	1	-1	-1	-1	-1
-1	-1	0	0	-1	1	-1	1	-1	1	-1	-1
0	-1	-1	0	-1	-1	-1	-1	1	-1	1	-1
-1	-1	0	0	0	1	-1	-1	1	1	1	1
1	0	0	0	-1	1	-1	-1	-1	-1	1	1
-1	1	1	1	-1	0	1	-1	-1	1	1	1
1	1	0	0	0	0	-1	1	1	1	1	-1
1	1	0	-1	0	1	1	-1	-1	1	-1	-1
0	0	-1	-1	1	1	1	-1	1	-1	-1	1
1	-1	1	-1	-1	-1	1	1	1	1	-1	1
0	0	-1	1	0	0	1	-1	1	1	-1	-1

X'X

19	0	0	0	0	0	0	-1	-1	-1	-1	-1	-1
	12	0	0	-1	0	-1	0	0	0	0	-2	0
		12	-1	0	0	0	2	0	0	0	0	0
			12	1	1	-1	0	0	0	0	0	2
				12	-1	1	0	0	0	0	0	0
					12	0	0	0	0	-2	0	0
						12	0	0	0	2	0	0
							19	-1	-1	-1	-1	-1
								19	-1	-1	-1	-1
									19	-1	-1	-1
										19	-1	-1
											19	-1
												19

NOTE: FFD used 100.11 seconds

It is possible that another start would have produced a design with a smaller f value than f = 49.0 as obtained in the above output.

A number of examples using the FFD module are given in the User's Note. They are listed below:

- Example 1: $m=5$ $n=15$ -5
 Example 2: $m=5$ $m0=4$ $n=16$ -5 (see above)
 Example 3: $m=7$ $n=8$ $-p$
 Example 4: $m=7$ $m0=3$ $n=8$ (four factors added to a 2^3 factorial)
 Example 5: $m=6$ $n=16$ $tries=10$ -4
 Example 6: $m=4$ $n=12$ $tries=10$ -5
 Example 7: $m=6$ $n=24$ $tries=10$ -4
 Example 8: $m=3$ $m3=1$ $n=12$ -5
 Example 9: $m=3$ $m3=2$ $n=18$ -5
 Example 10: $m=4$ $m3=1$ $n=24$ -5
 Example 11: $m=5$ $m3=1$ $n=24$ -4

FFD Module For RSD's

A response surface design, RSD, is a 3-level fractional factorial with unequal numbers of levels. To construct a RSD, the number of levels of the $+1$ and -1 levels are set by specifying the value of $n1$. The 0 level is the center point of a factor, and there are $n - 2(n1)$ combinations or runs of the factors all at the zero level. For the first RSD example, let $m3 = 2$ factors with the high and low levels being repeated $n1 = 2$ times each, with $n - 2(n1) = 5$ center points, and with $n = 9$ runs. The command for constructing this RSD is:

Command: FFD $m3=2$ $n=9$ $n1=2$ -5

The " -5 " was used so that a 2^2 factorial at the $+1$ and -1 points would result. The output from the above command is obtained from the command:

Command: PRINT FFD.OUT

The resulting output is:

NOTE: A design with $m=2$, $p=6$ and $n=9$

rand_seed	try	iter	f	det	std. det
1968734817	1	1	4.9R	0.000E+0000	0.000E+0000

Factor levels

0	0
1	1
0	0
0	0
-1	1
1	-1
0	0
0	0
-1	-1

X'X

9	0	0	0	4	4
	4	0	0	0	0
		4	0	0	0
			4	0	0
				4	4
					4

NOTE: FFD used 0.11 seconds.

This RSD consists of 2^2 factorial for the +1 and -1 points and with $n - 2(n1) = 5$ center points with coordinates (0,0).

Examples of RSD's presented in the FFD Module User's Note are:

Example 12: $m=3$ $n=15$ $n1=4$ -5

Example 13: $m=4$ $n=27$ $n1=6$ -5

Example 14: $m=5$ $n=46$ $n1=8$

Example 15: $m=6$ $n=54$ $n1=12$ tries=100 -5

Example 16: $m=7$ $n=62$ $n1=12$ -5

Example 17: $m=3$ $n=20$ -5

CUT Module for Blocking FFD's and RSD's

The CUT module is a program for blocking fractional factorials and response surface designs. The approach to blocking a design is to find a suitable unblocked design and then to allocate the runs to the blocks in an optimal manner as described in the User's Note included with the GENDEX package. Some options associated with the CUT module are

- f denotes that the full instead of the main effect model is used,
- i adds the two-factor interaction terms to the main effect terms,
- m uses the full model but places more emphasis on orthogonality between input variables from the input file and the block variables, and
- c maximizes the number of columns being orthogonally blocked.

The constraints are that $m \leq 21$, $n \leq 64$, and $b < n/2$ and b is a multiple of n .

The first example is the one in User's Note for a 2^3 factorial blocked into $b = 2$ blocks. The factorial, fractional factorial, or response surface treatment design to be blocked needs to be created first and used as an infile for the CUT program. To create the fractional factorial design, use the command:

Command: FFD $m=3$ $n=8$ -5

Then the FFD.OUT needs to be edited to remove everything but the listing of the runs or combinations. This is done with the command:

Command: EDIT FFD.OUT

After editing, select SAVE and then EXIT from the FILE menu. To construct the blocked design for $b = 2$ blocks, use the command:

Command: CUT IN=FFD.OUT $m=3$ $n=8$ $b=2$ -f

To observe the blocked design on the computer screen, use the command:

Command: TYPE CUT.OUT

To obtain a saved and/or printed copy of CUT.OUT, use

Command: COPY CUT.OUT PRN (or PRINT CUT.OUT)

The output from these commands is given below.

NOTE: A blocked design with $m=3$, $p=8$, $n=8$ and $b=2$

```

rand_seed  try iter      f      det      BF
609829460   1   2    0.00  4.1943E+0006  1.0000

```

Factor levels

Block 1

```

  1  -1  1
-1  1  1
-1 -1 -1
  1  1 -1

```

Block 2

```

  1  -1 -1
-1 -1  1
-1  1 -1
  1  1  1

```

X'X

```

  4  0  0  0  0  0  0  0
    4  0  0  0  0  0  0
      8  0  0  0  0  0
        8  0  0  0  0
          8  0  0
            8  0
              8

```

NOTE: CUT used 0.15 seconds.

This design completely confounds the three-factor interaction ABC with blocks. The block means are associated with the first two columns of the X'X matrix and the last six columns relate to the three main effects and their three two-factor interactions.

For a second example, construct a Resolution V design for $m = 5$ factors in 16 runs with $b = 4$ blocks. To do this, use the following commands:

```

Commands:  FFD m=5 n=16 -5
           EDIT FFD.OUT (EDIT, SAVE, and EXIT)
           CUT IN=FFD.OUT m=5 n=16 b=4 -m -c tries=20
           COPY CUT.OUT PRN (or PRINT CUT.OUT)

```

The output from these commands is given below:

NOTE: A blocked design with $m=5$, $p=19$, $n=16$ and $b=4$

```

rand_seed  try iter      f      det      BF
395407380   1   5 192.00*  0.0000E+0000  0.0000

```

Factor levels

Block 1

```

  1  1 -1  1  1
-1  1  1 -1 -1
-1 -1 -1  1  1
  1 -1  1 -1 -1

```

Block 2

```

-1  1  -1  -1  1
 1  -1  -1  -1  1
 1  1  1  1  -1
-1  -1  1  1  -1

```

Block 3

```

 1  1  1  -1  1
-1  1  -1  1  -1
-1  -1  1  -1  1
 1  -1  -1  1  -1

```

Block 4

```

 1  -1  1  1  1
-1  1  1  1  1
-1  -1  -1  -1  -1
 1  1  -1  -1  -1

```

X'X

```

 4  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  -4  -4  4
    4  0  0  0  0  0  0  0  0  0  0  0  0  0  0  4  -4  -4
      4  0  0  0  0  0  0  0  0  0  0  0  0  0  0  -4  4  -4
        4  0  0  0  0  0  0  0  0  0  0  0  0  0  4  4  4
          16  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
            16  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
              16  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
                16  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
                  16  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
                    16  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
                      16  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
                        16  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
                          16  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
                            16  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
                              16  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
                                16  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
                                  16  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
                                    16  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
                                      16  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
                                        16  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
                                          16  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
                                            16  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
                                              16  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
                                                16  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
                                                  16  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
                                                    16  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0

```

NOTE: CUT used 3.51 seconds.

The three two-factor interactions AB, AE, and BE are completely confounded with the four blocks. The above is Example 6 in the User's Note for this module.

Five examples of using the CUT module for blocking RSD's are given in the FFD Module User's Note. These are Examples 13-17. In the User's Note for the CUT Module, the following examples are presented (b is the number of blocks):

Example 1: $m=3$ $n=8$ $b=2$ -f (divide a 2^3 factorial into two blocks)

Example 2: $m=3$ $n=8$ $b=4$ (divide a 2^3 factorial into four blocks)

Example 3: $m=4$ $n=16$ $b=4$ -m (divide a 2^4 factorial into four blocks such that main effects are orthogonal to blocks)

Example 4: $m=4$ $n=16$ $b=4$ -m -c tries=10 (divide 2^4 factorial into four blocks such that main effects, columns, are orthogonal to blocks and remaining columns orthogonal to blocks maximized)

Example 5: $m=5$ $n=16$ $b=2$ -m -c tries=10 (divide a half fraction of a 2^5 into two blocks)

Example 6: $m=5$ $n=16$ $b=4$ -m -c tries=10 (divide a half fraction of a 2^5 into four blocks)

Example 7: $m=6$ $n=16$ $b=4$ -m -c tries=10 (divide a quarter fraction of a 2^6 into four blocks)

- Example 8: $m=4$ $n=18$ $b=3$ -m tries=100 (divide a 2^4 with 18 runs into three blocks)
 Example 9: $m=4$ $n=18$ $b=3$ -i tries=100 (divide a 2^4 with two center points into three blocks)
 Example 10: $m=2$ $n=18$ $b=3$ -f (divide a 3^2 factorial with 18 runs into three blocks)
 Example 11: $m=3$ $n=27$ $b=3$ -f (divide a 3^3 factorial into three blocks)
 Example 12: $m=2$ $n=9$ $b=3$ -f tries=10 (divide a 3^2 factorial into three blocks)
 Example 13: $m=3$ $n=20$ $b=5$ -m tries=10 (divide a 3^3 factorial in 20 runs into five blocks)
 Example 14: $m=3$ $n=6$ $b=2$ -i (divide six distinct binary blends into two blocks)
 Example 15: $m=4$ $n=24$ $b=2$ -i (divide 24 distinct 4-component blends into two blocks)
 Example 16: $m=4$ $n=16$ $b=2$ -i (divide 16 binary blends into two blocks)
 Example 17: $m=4$ $n=24$ $b=2$ (divide 24 blends distinct 3-component blends into two blocks)
 Example 18: $m=4$ $n=24$ $b=2$ (divide 12 distinct binary blends and 12 distinct 3-component blends into two blocks)
 Example 19: $m=6$ $n=12$ $b=4$ tries=10 (divide an orthogonal array of one 3-level factor and four 2-level factors into four blocks)
 Example 20: $m=9$ $n=12$ $b=3$ (divide 12 runs of an orthogonal array for nine 2-level factors into three blocks)

RAT Module For Trend-Free FRD's and RSD's

RAT, robust against trend, (an up-dated version of TFD, trend-free design) is a module for constructing FRD's and RSD's which are trend-free or nearly so. The constraints imposed are that $m_0 \leq 2$, $m \leq 21$, and $n \leq 64$. The number m_0 is the number of trend variables and $m - m_0$ is the number of 2-level factor variables to be included. Three-level factors cannot be used in the commands. For the trend values, the orthogonal polynomial coefficients are divided by the largest value in the set of coefficients. This results in the largest coefficient being one. As before, n is the number of runs. The following options are available:

- f denotes use of full factorial rather than a min-effect model,
- i denotes that two-factor interactions are added to main-effect terms,
- m denotes use of full model but orthogonality between input variables in the input file and trend variables are more important than between added variables and trend variables, and
- p denotes that the inverse of $X'X$ is to be printed.

For the first example, suppose that a curvilinear trend is suspected, that the runs are to be done sequentially, and that $m = 4$ factors are to be investigated. The commands for creating this design are:

```
Command:      RAT m0=2 m=4 n=8 tries=20
              TYPE RAT.OUT (to see output on screen)
              COPY RAT.OUT PRN
```

The output from these commands is:

NOTE: A design with $m_0=2$, $m=4$, $p=7$ and $n=8$

rand_seed	try	iter	f	det	TF
1288877260	1	3	2.45	2.4649E+0003	0.9189
853434993	4	1	1.80	2.5862E+0003	0.9278

Factor levels (columns 3-6)

-1.0000	1.0000	0	0	0	0
-0.7143	0.1429	-1	1	1	1
-0.4286	-0.4286	-1	-1	1	-1
-0.1429	-0.7143	0	0	-1	-1
0.1429	-0.7143	0	0	1	1
0.4286	-0.4286	-1	1	0	0
0.7143	0.1429	0	0	1	-1
1.0000	1.0000	0	0	0	0

X'X

3.4286	0.0000	0.0000	0.7143	0.1429	-0.1429	-0.7143
	3.4286	0.0000	0.7143	0.1429	-0.1429	0.4286
		8.0000	-3.0000	1.0000	3.0000	-1.0000
			3.0000	-1.0000	-2.0000	0.0000
				3.0000	0.0000	2.0000
					5.0000	1.0000
						5.0000

NOTE: RAT used 0.66 seconds.

It would appear that the factors are all at three levels as the numbers +1, 0, and 1 appear in the design for each factor. Only the linear effect of each factor can be assessed. It would require $1 + 2 + 4(2) = 11$ runs to have enough degrees of freedom to estimate linear and quadratic effects of each factor plus the linear and quadratic trends.

For a second illustration, suppose that a Resolution V FRD with $m=3$ factors is to be conducted in $n = 8$ runs. The runs are to be conducted sequentially, and it is suspected that a linear time trend exists. The commands for creating this RAT design are given below.

Command: RAT m0=1 m=3 n=8 -i
 COPY RAT.OUT PRN

The printed output from these commands is:

NOTE: A design with $m0=1$, $m=3$, $p=8$ and $n=8$

rand_seed	try	iter	f	det	TF
1540086805	1	2	6.53	5.4783E+0006	0.9619

Factor levels (columns 2-4)

-1.0000	1	1	1
-0.7143	1	-1	-1
-0.4286	-1	-1	1
-0.1429	-1	1	-1
0.1429	-1	1	1
0.4286	-1	-1	-1
0.7143	1	-1	1
1.0000	1	1	-1

X'X

```

3.4286 -0.0000  0.0000 -0.0000 -1.1429  0.0000  0.0000 -2.2857
      8.0000  0.0000  0.0000  0.0000  0.0000  0.0000  0.0000  0.0000
          8.0000  0.0000  0.0000  0.0000  0.0000  0.0000  0.0000  0.0000
              8.0000  0.0000  0.0000  0.0000  0.0000  0.0000  0.0000  0.0000
                  8.0000  0.0000  0.0000  0.0000  0.0000  0.0000  0.0000  0.0000
                      8.0000  0.0000  0.0000  0.0000  0.0000  0.0000  0.0000  0.0000
                          8.0000  0.0000  0.0000  0.0000  0.0000  0.0000  0.0000  0.0000
                              8.0000  0.0000  0.0000  0.0000  0.0000  0.0000  0.0000  0.0000
                                  8.0000  0.0000  0.0000  0.0000  0.0000  0.0000  0.0000  0.0000
                                      8.0000  0.0000  0.0000  0.0000  0.0000  0.0000  0.0000  0.0000

```

NOTE: RAT used 34.75 seconds.

Using the option -f in place of -i results in the above design.

A User's Note for the TFD Module was obtained but one for the RAT Module was not. RAT is the new version of TFD. The examples given in the TFD Module User's Note are:

Example 1: m0=2 m=6 n=16 -f seed=123 tries=10 (2^4 main effects and interactions are orthogonal or near-orthogonal to linear, L+, and quadratic, Q+, trends)

Example 2: m0=2 m=6 n=16 -m seed=1071228089 (2^4 main effects and interactions are L+ and Q+ with more emphasis on main effects)

Example 3: m0=2 m=7 n=16 (five 2-level factors main effects are L+ and Q+)

Example 4: m0=1 m=12 n=16 (11 2-level factor main effects are L+)

Example 5: m0=2 m=7 n=12 -s tries=10 (five 2-level factor main effects are L+ and Q+)

Example 6: m0=1 m=13 n=18 (six 3-level factor main effects are L+)

Example 7: m0=1 m=4 n=15 -f -s (2^3 factorial effects are L+)

Example 8: m0=2 m=5 n=15 -f -s tries=10 (2^3 factorial effects are L+ and Q+)

Example 9: m0=2 m=5 n=15 -f (main effects and interactions of a 2^3 factorial are L+)

Example 10: m0=2 m=5 n=15 =f -s tries=10 (main effects and interactions of a 2^4 factorial are L= and Q+)

Example 11: m0=1 m=4 n=15 -f (Box-Behnken design for three factors which is L+)

Example 12: m0=1 m=4 n=15 -f (Box-Behnken design for four factors which is L+)

NOA Module For Orthogonal Arrays

Instead of constructing FRD's and RSD's, an orthogonal array might be desired. Suppose an orthogonal array of m = 7 two level factors in n = 8 run is desired. It is known that one exists since a Hadamard matrix of order eight exists. This means that average = ave will be zero, D-efficiency = D-eff will be one, and A-efficiency = A-eff will be one. The commands to construct this design are:

```

Command:      NOA m=7 n=8
              COPY NOA.OUT PRN

```

The printed output from these commands is:

```

NOTE: Construct a design with n=8 and m=7
  rand_seed  try iter   ave   max  freq  D-eff  A-eff
1337704761    1   4 0.00*    0   21  1.000  1.000

```

X

```

  1  -1  -1   1  -1   1  -1
  1   1   1   1   1  -1  -1
 -1   1   1  -1  -1   1  -1
 -1  -1  -1  -1   1  -1  -1
  1   1  -1  -1   1   1   1
 -1   1  -1   1  -1  -1   1
  1  -1   1  -1  -1  -1   1
 -1  -1   1   1   1   1   1

```

X'X

```

8  0  0  0  0  0  0
   8  0  0  0  0  0
      8  0  0  0  0
         8  0  0  0
            8  0  0
               8  0
                  8

```

NOTE: NOA used 0.04 seconds.

Adding a column of ones to the above X matrix results in a Hadamard matrix.

The number of factors may be more than the number of runs. To illustrate this, we construct an orthogonal array for $m = 6$ factors in $n = 4$ runs by using the following commands:

Command: NOA m=6 n=4

COPY NOA.OUT PRN

The printed output for these commands is:

NOTE: Construct a design with $n=4$ and $m=6$

```

rand_seed  try iter  ave  max  freq  D-eff  A-eff
1646369324  1   1  3.20*  4    3  0.000  0.000

```

X

```

1  1  -1  -1  1  -1
-1  1  1  -1  -1  1
1  -1  1  1  -1  -1
-1  -1  -1  1  1  1

```

X'X

```

4  0  0  0  0  -4
   4  0  -4  0  0
      4  0  -4  0
         4  0  0
            4  0
               4

```

NOTE: NOA used 8256.06 seconds.

For the third example, suppose that it is desired to construct a near-orthogonal array for eight sets of +1's and -1's which are near-orthogonal to a linear trend in $n = 10$ runs. Note that this could also be considered as a fractional replicate with trend over five time periods with two runs for each time period. A trend matrix needs to be constructed first. Here we consider only a linear trend of time but quadratic and higher-order trends may be included if desired. To create the NOA.IN matrix of m_0 columns, here $m_0 = 1$, we use the command EDIT. A blank screen appears and the desired matrix is typed in and SAVED AS NOA.IN. The commands are:

Commands: EDIT (Type in the following numbers in one column: -2, -2, -1, -1, 0, 0, 1, 1, 2, and 2.)
SAVE AS NOA.IN, EXIT

NOA m=9 m0=1 n=10
 COPY NOA.OUT PRN (or PRINT NOA.OUT)

The following output was obtained from the above commands. Note that the NOA.IN is in the first column of the X (combinations). This produces a near-orthogonal array for eight 2-level factors which are nearly orthogonal to a linear trend. In order to limit the computing time, it would be desirable to specify the number of tries in the NOA command. The printed output obtained is:

NOTE: Construct a design with n=10 and m=9

rand_seed	try	iter	ave	max	freq	D-eff	A-eff
1855515815	1	10	3.33	2	30	0.860	0.719
290143482	2	8	3.11	2	28	0.873	0.745

X

-2	-1	1	-1	1	1	1	-1	1
-2	1	-1	-1	-1	1	-1	1	-1
-1	-1	-1	1	-1	-1	-1	-1	1
-1	1	1	1	1	-1	-1	1	-1
0	-1	1	1	-1	-1	1	-1	-1
0	1	-1	-1	1	-1	1	1	1
1	1	1	1	-1	1	1	-1	1
1	-1	-1	1	1	1	1	1	-1
2	1	-1	-1	1	1	-1	-1	-1
2	-1	1	-1	-1	-1	-1	1	1

X'X

20	0	0	0	0	0	0	0	0
	10	-2	-2	2	2	-2	2	-2
		10	2	-2	-2	2	-2	2
			10	-2	-2	2	-2	-2
				10	2	2	2	-2
					10	2	-2	-2
						10	-2	2
							10	-2
								10

NOTE: The first 1 columns are read from file NOA.IN.

NOTE: NOA used 2123.71 seconds.

The following examples are presented in the User's Note for the NOA Module:

Example 1: n=6 m=10 tries=10 (ten 2-level factors)

Example 2: n=12 m=16 tries=30 -t

Example 3: n=12 m=24 tries=10 -m -t

Example 4: n=12 m=11 tries=10

Example 5: n=13 m=12 seed=529264271 tries=1

Example 6: n=12 m=11 m0=8 tries=10 -p

Example 7: n=12 m=11 m0=5 tries=10

Example 8: n=16 m=13 m0=5 tries=10

Example 9: n=20 m=12 m0=10

Example 10: n=16 m=13 m0=5 tries=10

Example 11: n=16 m=7 m0=2 seed=1777107967

Example 12: n=16 m=12 m0=8

Example 13: n=12 m=5 m0=1

Example 14: $n=20$ $m=8$ $m_0=1$ seed=1446201341 (This random seed was found after thousands of tries)

Comment

The four GENDEX modules investigated, i.e., FFD, CUT, RAT, and NOA, are very useful for constructing optimal or near-optimal FRD's and RSD's. Two and three level factors may be used, although the use of three level factors is limited to a much greater extent than two level factors, e.g., for the NOA module. A desirable goal would be to expand GENDEX to include wider applicability for three level factors and to expand to higher level factors. To illustrate, an orthogonal array for $m_3 = 11$ with $m_0 = 8$, say, is not possible. Orthogonal arrays with 4, 5, 6, etc. elements would be desirable. Although some higher levels can be used with an infile, their use with the outfile is desired.

Literature Cited

Federer, W. T., N. Nshinyabakobeje, and N-K Nguyen (1998). GENDEX for constructing experiment designs. Technical Report BU-1433-M in the Technical Report Series of the Department of Biometrics, Cornell University, Ithaca, NY