MULTIVARIATE LINEAR MODELS POWER, VERSION 2.03

Jacqueline L. Johnson, Keith E. Muller, James C. Slaughter, Matthew J. Gurka, Matthew J. Gribbin and Sean L. Simpson

The program may be downloaded at no cost from *http://www.bios.unc.edu/~muller*

The POWERLIB203 program is given freely by the authors and may be distributed by others at no charge. Please cite appropriately in all publications. Use these modules at your own risk. The software comes with no warranty, and the authors cannot be held liable for any consequences resulting from its use.

1. OVERVIEW
1.1 Features

This software computes statistical power for a wide variety of univariate, multivariate, and repeated measures linear models with Gaussian errors and fixed predictors. The generality of the program stems from using matrix notation and the matrix syntax of the SAS/IML environment. Please refer to the IML manual for the rules of creating matrices and other language syntax. Although the most general cases require approximations, the program uses exact methods when they are available. The program computes power for the three most popular multivariate test statistics: Hotelling-Lawley trace (HLT), Wilks' likelihood ratio (WLK), and the Pillai Bartlett trace (PBT). In this manual, the three tests and related methods are collectively referred to as the *MULTIREP* approach. The program also computes power for the four commonly used tests for the "univariate" approach to repeated measures: uncorrected (UN), Huynh-Feldt (HF), Geisser-Greenhouse (GG), and Box (BOX) conservative. The four tests and associated methods will be collectively referred to as the *UNIREP* approach.

1.2 Model and Hypothesis Behind the Power Calculations

Please refer to Muller, LaVange, Ramey and Ramey (1992) for a review of the theory behind the power methods implemented. For complete details of specific methods, please see Muller and Peterson (1984), Muller and Barton (1989), Kim, Gribbin, Muller and Taylor (2005) and Muller, Edwards, Simpson and Taylor (2005). Muller and Benignus provided a brief introduction to the most basic ideas of power in the context of toxicology, while O'Brien and Muller (1993) provided a lengthy tutorial in linear models power.

For N independent sampling units, p responses, and q predictors, the General Linear Multivariate Model (GLMM) may be stated as:

$$Y = XB + E,$$

with Y ($N \times p$) containing information on random responses such as repeated measures, X ($N \times q$) a fixed design matrix containing predictors, B ($q \times p$) containing unknown parameters, and E containing random errors. Rows of Y, X and E correspond to independent sampling units (such as subjects), columns of Y, B and E to level of the multivariate response (often time), and columns of X and rows of B to predictors. With $r = \mathrm{rank}(X)$, the methods used in

this program assume $N > r$, $\text{row}_i(\boldsymbol{E}) \sim N_p(\boldsymbol{0}, \boldsymbol{\Sigma})$, $i \in \{1, 2, \ldots, N\}$, all rows independent, and no missing data in $\boldsymbol{Y}$ or $\boldsymbol{X}$.

The corresponding General Linear Hypothesis (GLH) involves $\boldsymbol{\Theta} = \boldsymbol{CBU}$, with $\boldsymbol{C}$ an $a \times q$ matrix of known constants defining "between subject" contrasts, and $\boldsymbol{U}$ a $p \times b$ matrix of known constants defining "within subject" contrasts. The power program requires $\text{rank}(\boldsymbol{C}) = a \leq q$ and $\text{rank}(\boldsymbol{U}) = b \leq p$. The GLH is

$$H_0 : \boldsymbol{\Theta} = \boldsymbol{\Theta}_0$$
$$H_1 : \boldsymbol{\Theta} \neq \boldsymbol{\Theta}_0 \, .$$

Most often $\boldsymbol{\Theta}_0 = \boldsymbol{0}$. In the context of data analysis, values in $\boldsymbol{B}$ and $\boldsymbol{\Sigma}$ are estimated. In contrast, in the context of power analysis $\boldsymbol{B}$ and $\boldsymbol{\Sigma}$ are assumed known constants. Timm (2001), among others, provides a detailed discussion of basic theory and practice for GLMM's, as well as many related models.

1.3 The Simplest Program: One Power Value

The program assumes that the design matrix, $\boldsymbol{X}$, is specified in terms of an *essence* matrix, $\text{Es}(\boldsymbol{X})$, and a replication factor (REPN in the program): $\boldsymbol{X} = \text{Es}(\boldsymbol{X}) \otimes \mathbf{1}_{\text{REPN}}$, with $\boldsymbol{G} \otimes \boldsymbol{H} = \{g_{ij} \cdot \boldsymbol{H}\}$. The essence matrix (Helms, 1988) contains one and only one copy of each unique row of the original matrix. For example, cell mean coding for an independent groups t-test with 10 observations per group has, with $\mathbf{1}_{10}$ a 10×1 vector of 1's and $\mathbf{0}_{10}$ a 10×1 vector of 0's,

$$\boldsymbol{X} = \begin{bmatrix} \mathbf{1}_{10} & \mathbf{0}_{10} \\ \mathbf{0}_{10} & \mathbf{1}_{10} \end{bmatrix}$$
$$= \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \otimes \mathbf{1}_{10}$$

and

$$\text{Es}(\boldsymbol{X}) = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} = \boldsymbol{I}_2 \, .$$

Computing power for a GLH with fixed predictors requires knowing only seven variables: $\boldsymbol{\Sigma}$, $\boldsymbol{X}$, $\boldsymbol{B}$, $\boldsymbol{C}$, $\boldsymbol{U}$, α and $\boldsymbol{\Theta}_0$. The power program assumes default values of $\boldsymbol{U} = \boldsymbol{I}_p$ (which corresponds to a MANOVA hypothesis with multivariate responses), $\boldsymbol{\Theta}_0 = \boldsymbol{0}$, and $\alpha = 0.05$. The user must always specify $\boldsymbol{B}$, $\boldsymbol{C}$, $\boldsymbol{\Sigma}$, as well as $\boldsymbol{X}$ in terms of $\text{Es}(\boldsymbol{X})$ and a replication factor. The following code is the complete code needed to compute power for an independent groups t-test with ten observations per group and cell mean cell coding. See Muller and Fetterman (2002, chapters 12-14) for an introduction to ANOVA coding in one and two way designs.

```
PROC IML SYMSIZE=2000;
%INCLUDE "..\IML\POWERLIB203.IML"/NOSOURCE2;

ESSENCEX = I(2)  ;
REPN     = { 10 };
BETA     = {0 1}`;
SIGMA    = {1}; *=variance, because here covariance matrix is 1x1 ;
C        = {1 -1};
```

```
RUN POWER;
```

The program prints the following output
```
    ALPHA     SIGSCAL    BETASCAL    TOTAL_N     POWER
     0.05           1           1         20     0.562
```

Please note the following general principles, as just illustrated.

1) The first two program statements are always required. They initialize IML, ask for extra symbol space, and make the power modules available for use.

2) The matrices ESSENCEX, BETA, SIGMA, and C are also always required and specify inputs to the power program. REPN is needed here because it has a default value of 1.

3) The RUN POWER statement is also always required.

4) For univariate models, SIGMA is the *variance*, not the standard deviation, because $\Sigma = \sigma^2$ is 1×1. This potential source of confusion from the substantial advantages of treating the univariate case as a special case of the multivariate, which requires the same variable name (SIGMA) for both settings.

1.4 Unequal Cell Sizes

There are various ways to allow for unequal cell sizes. As an example, consider

$$X = \begin{bmatrix} \mathbf{1}_{10} & \mathbf{0}_{10} & \mathbf{0}_{10} \\ \mathbf{0}_{15} & \mathbf{1}_{15} & \mathbf{0}_{15} \\ \mathbf{0}_{20} & \mathbf{0}_{20} & \mathbf{1}_{20} \end{bmatrix}.$$

One way to specify this is:
```
ESSENCEX = {1 0 0, 1 0 0,
            0 1 0, 0 1 0, 0 1 0,
            0 0 1, 0 0 1, 0 0 1, 0 0 1};
REPN = 5;
```

If in doubt about how to use ESSENCEX and REPN together, the user can always specify ESSENCEX as the entire X matrix and set REPN = 1. For example, the X matrix immediately above could be given using:
```
ONE = {1 0 0}; TWO = {0 1 0}; THREE = {0 0 1};
ESSENCEX = REPEAT(ONE, 10, 1) // REPEAT(TWO, 15, 1) // REPEAT(THREE, 20, 1);
REPN = 1;
```

1.5 Producing Power for a Range of Scenarios

The program makes it easy to compute power for ranges of values of sample size, test size, mean differences, variance, and correlation level among response variables. The following code generalizes the preceding example. The inputs ask for 2 values of REPN, 10 values of BETASCAL, 3 of SIGSCAL, and 3 of ALPHA. BETASCAL is multiplied times BETA, and could be called THETASCAL because it also multiplies Θ by the same amount. In the t-test example, $\Theta = \mu_1 - \mu_2$. Hence the BETASCAL statement leads to power being computed for $\mu_1 - \mu_2 = 0.0$, $\mu_1 - \mu_2 = 0.1$, $\mu_1 - \mu_2 = 0.2, \ldots, \mu_1 - \mu_2 = 0.9$. Similarly, SIGSCAL values are multiplied times Σ. Hence the SIGSCAL statement leads to power for $\sigma^2 = 0.5$, $\sigma^2 = 1.0$, and $\sigma^2 = 2.0$. Including the three values in ALPHA causes power to be computed for $\alpha = 0.005$, $\alpha = 0.01$, and $\alpha = 0.05$. Each combination of inputs leads to a power value, for a total of $2 \cdot 10 \cdot 3 \cdot 3 = 180$ power values.

3

```
PROC IML SYMSIZE=2000;
%INCLUDE "..\IML\POWERLIB203.IML"/NOSOURCE2;

BETA     = {0 1}`;
C        = {1 -1};
SIGMA    = {1}; *=variance, because covariance "matrix" is 1x1 ;
ESSENCEX = I(2);

REPN     = {10 20};      * <=> REPN=DO(10,20,10);
BETASCAL =DO(0,.9,.1);
SIGSCAL  ={.5 1 2};      * <=> SIGSCAL={.5,1,2};
ALPHA    ={.005 .01 .05};

RUN POWER;
```

Please note the following general principles, as just illustrated.

5) Some programs inputs, including REPN, ALPHA, SIGSCAL, BETASCAL, and RHOSCAL, may be assigned lists of values.

6) Such lists may be matrices which are $1 \times n$ (a matrix with one row) or $n \times 1$ (a vector).

7) The program produces power for all factorial combinations of inputs.

1.6 Choosing Options

The user has control of a large number of options that are specified by including their names in either OPT_ON or OPT_OFF, which are character matrices, and must have either only one row or only one column. For example, to allow the design matrix to be less than full rank, and to exclude SIGSCAL from the output, insert the following statements before RUN POWER;.
```
OPT_ON={LTFR};
OPT_OFF={SIGSCAL};
```

Specify the options to turn on as elements of OPT_ON, and those to turn off as elements of OPT_OFF. The default set of options corresponds to executing the statements:
```
OPT_ON ={COLLAPSE HLT PBT WLK ALPHA SIGSCAL BETASCAL
           TOTAL_N MAXRHOSQ ESSENCEX BETA SIGMA RHO C U CBETAU WARN};
OPT_OFF = {POWERCASE RHOSCAL UN HF BOX LTFR FRACREPN ORTHU N_EST RANK_EST
           FMETHOD UCDF UMETHOD MMETHOD RANKX RANKC RANKU NOPRINT DS};
```

The options fall into four groups: computations, power confidence limits, output produced, and output contents. We describe each group separately below. In addition, input matrices that have related effects are also described at the same time.

Please note the following general principles, as just illustrated.

8) Options and input matrices fall into four groups: computations, power confidence limits, output produced, and output contents.

9) Assigning values to the character matrix OPT_ON or OPT_OFF causes the option named to be turned off or turned on. Each matrix must have only one row or have only one column.

10) Order does not matter, nor does upper or lower case.

1.7 Power Computations (Options and Inputs)

Table 1 lists the options that control which power values are computed. Listing one of the names in the table in the character matrix OPT_ON causes them to be computed and included in the output, while listing one in OPT_OFF skips the computation and excludes the value from the

output. If the COLLAPSE option is on (as is default), then if $b = 1$ (and $a \geq 1$) all MULTIREP and UNIREP powers are combined into one output column labeled POWER. If the collapse option is on with $b > 1$ and $a = 1$ then all MULTIREP variable powers are combined into one column labeled POWER_MULT.

User utility modules UPOLY1, UPOLY2, and UPOLY3 (documented in a later section) may be used to create orthonormal contrast matrices for 1-3 factors within or between.

Table 1 Options to Control Which Tests Have Power Computed

Name	Description	Default ON?
HLT	Compute power for Hotelling-Lawley Trace	✓
PBT	Compute power for Pillai-Bartlett Trace	✓
WLK	Compute power for Wilks' Lambda	✓
UN	Compute power for UNIREP uncorrected	
HF	Compute power for UNIREP Huynh-Feldt	
GG	Compute power for UNIREP Geisser-Greenhouse	✓
BOX	Compute power for UNIREP Box	
LTFR	Allow use of less than full rank X	
FRACREPN	Allow use of fractional REPN values	
ORTHU	Allow use of non-orthonormal U	

Table 2 lists all input matrices that, together with the options, help control power computations. Most users will only ever need to use ROUND and MMETHOD. The TOLERANCE variable is included to provide a user with sophistication in computing methods some flexibility when working with large models. The MMETHOD, UMETHOD and UCDF variables serve two purposes: 1) allow reproducing results from earlier version 1 of this software and other related software, and 2) allow easy access to the O'Brien and Shieh (1992) variation of the MULTIREP power approximations.

Table 2 Input Matrices that Control Power Approximations

Name	Description	Default Value
ROUND	round _HOLDPOWER values to # decimal places	3
TOLERANCE	value judged as numeric zero	1×10^{-12}
MMETHOD	MULTIREP approximations	$\begin{bmatrix} 4 & 2 & 2 \end{bmatrix}'$
UMETHOD	UNIREP E$\tilde{\epsilon}$ and E$\hat{\epsilon}$ approximations	$\begin{bmatrix} 2 & 2 \end{bmatrix}'$
UCDF	UNIREP CDF approximations	$\begin{bmatrix} 2 & 2 & 2 & 2 \end{bmatrix}'$

The input matrices ROUND and TOLERANCE control the rounding of output values and the threshold for judging whether a numerical value is judged to be zero. In addition, the input matrices MMETHOD, UMETHOD, and UCDF may be assigned values by the user to control the methods of approximation for MULTIREP and UNIREP tests.

Approximations are used for all tests whenever $s = \min(a, b) > 1$, and for the UNIREP tests whenever $b > 1$ (whether or not $a = 1$). All MULTIREP tests' powers (and test sizes) coincide whenever $s = 1$, while MULTIREP and UNIREP powers (and test sizes) all coincide if $b = 1$.

1.8 Power Confidence Limits (Options and Inputs)

The CLTYPE, ALPHA_CL, ALPHA_CU, RANK_EST, and N_EST input matrices control whether confidence limits for all test powers are computed. Such confidence intervals are needed when using an estimate of Σ from a previous study to plan a new study, or estimates of Σ and B from a previous study to plan a new study. In either case, the randomness of the inputs create randomness in resulting power values. Confidence intervals about power values, and confidence regions about power curves, greatly help in planning by accounting for the uncertainty due to using estimates. The methods of Taylor and Muller (1995, 1996) for univariate confidence intervals are exact whenever the power calculation is exact. Exact and approximate confidence intervals are now available for the MULTIREP and UNIREP tests, respectively; see working papers Gribbin and Muller (2006) and Park and Muller (2006). For confidence limits for multivariate tests, we assume the user has given the REML estimate for Σ. Note that the current version of the program does not allow compensating for the bias of truncation discussed Taylor and Muller (1996) and Muller and Pasour (1997).

CLTYPE takes the following values:

$< 1 \;\Rightarrow\;$ No confidence limits are calculated

$= 1 \;\Rightarrow\;$ Confidence limits for B known and Σ estimated are calculated

$= 2 \;\Rightarrow\;$ Confidence limits for B estimated and Σ estimated are calculated

If CLTYPE ≥ 1, then RANK_EST and N_EST are required inputs.

A set of options listed in Table 5 control whether power confidence limit inputs and results are included in the output matrix and file.

1.9 Output Produced (Options and Inputs)

Output from the software is stored in four matrices accessible to the user.

Table 3 Matrices Containing Output

Matrix Name	Contains
_HOLDPOWER	Results of all power calculations
_HOLDPOWERLBL	Column Labels of _HOLDPOWER
_POWERWARN	Counts of certain power calculation difficulties
_POWERWARNLBL	Labels for the rows of _POWERWARN

In addition, if requested by the user, the program writes the matrix _HOLDPOWER to a SAS file, with variable names _HOLDPOWERLBL. Printing and creation of an output file are controlled by the options listed in Table 4.

The user input matrix DSNAME gives control of the name of the SAS file that is created, if any. To create a file, the user must specify the DS option by including it in OPT_ON. The user can name the data file by defining the matrix DSNAME= { libref membername };. For example, if DSNAME = {IN1 MYDATA}, the output file will be called IN1.MYDATA. Here IN1 refers to a library defined by a LIBNAME or DD statement. When using a library other than WORK as the default, define DSNAME = {libref membername defaultlib};.

If DSNAME is not defined (and the DS option is also selected) or if "membername" already exists in the library specified by "libref", a default file name is used. The default file names are numbered and of the form PWRDT### (where ### is a number). The program scans the library for the largest numbered data file and assigns the next number to the new data file. The maximum ### is 999. If PWRDT999 exists no more data files can be created. Note that the program uses the name _PWRDTMP as an intermediate file. If this file already exists in the specified library no files can be created.

The software will not write over existing files. To continually write to the same file with multiple runs of the power software, the user must consciously delete the existing file. To delete a file IN1.MYDATA, for example, execute the statement CALL DELETE (IN, MYDATA); prior to executing the software.

Table 4 Options Controlling Printing and File Output

Name	Description	Default ON?
ESSENCEX	Print ESSENCEX matrix	✓
RANKX	Print the rank of the X matrix	
BETA	Print B matrix	✓
SIGMA	Print Σ matrix	✓
RHO	Print RHO matrix	✓
C	Print C matrix	✓
RANKC	Print the rank of the C matrix	
U	Print U matrix	✓
RANKU	Print the rank of the U matrix	
THETA0	Print Θ_0 matrix	if THETA0 $\neq$ 0
CBETAU	Print CBU matrix	✓
WARN	Print power program warnings	✓
NOPRINT	Suppress all printed output	
DS	Write _HOLDPOWER to a SAS file with variable names _HOLDPOWERLBL	

1.10 Output Contents (Options and Inputs)

In general, requesting any power to be computed automatically causes it to be included in the output matrix (and file if requested). However, the COLLAPSE option reduces the number of actual power columns included in special cases. Associated bookkeeping variables, such as total sample size, are either included or not by default, with the user being able to decide. Furthermore, including the name of a power value (such as HLT) in OPT_OFF will eliminate the power value and related information from the output matrix.

Asking the program to compute confidence limits for power values due to using an estimate of Σ (CLTYPE=1;) or estimates of both Σ and B (CLTYPE=2;) adds columns to the output matrix (which may in turn be deleted by adding the variable name to OPT_OFF).

Table 5 Options Controlling Contents of Output Matrix _HOLDPOWER

Name	Description	Default ON?
COLLAPSE	$s = 1$ powers reduce to 1 column	✓
	If $b = 1$ collapse MULTIREP and UNIREP	
	If $b > 1$ and $a = 1$ collapse MULTIREP	
POWERCASE	Include row #	
ALPHA	Include ALPHA	✓
SIGSCAL	Include SIGSCAL	✓
RHOSCAL	Include RHOSCAL	
BETASCAL	Include BETASCAL	✓
TOTAL_N	Include TOTAL_N	✓
MAXRHOSQ	Include max canonical correlation	✓
CLTYPE	Include CLTYPE	if CLTYPE ≥ 1
N_EST	Include N_EST	
RANK_EST	Include RANK_EST	
ALPHA_CL	Include ALPHA_CL	if CLTYPE ≥ 1
ALPHA_CU	Include ALPHA_CU	if CLTYPE ≥ 1
FMETHOD	Include FMETHOD	
MMETHOD	Include MMETHOD	
UMETHOD	Include UMETHOD	
UCDF	Include UCDF	

After all requested power calculations have been performed, power values, any confidence limits requested, and certain inputs are copied to the matrix _HOLDPOWER. _HOLDPOWERLBL contains the labels for the columns of _HOLDPOWER. The elements of _HOLDPOWERLBL are also the variable names in the SAS file created (if one is requested) with the DS option). Tables 6a, 6b, 6c and 6d contain all possible labels for columns of _HOLDPOWER. The particular set of columns present vary with power computation and print options the user has selected explicitly or by default.

If requested by the user, the _SASDS module creates a file from the matrix _HOLDPOWER. Variable labels in the file are the same as the column labels for the matrix _HOLDPOWER, namely the elements of the global output matrix _HOLDPOWERLBL.

Table 6a _HOLDPOWER Column Labels ⇔ Output File Variable Names

Column Label ⇔ Variable Name	Description
ALPHA	test size, α value for power calculation
ALPHA_CL	α value for power lower confidence limit
ALPHA_CU	α value for power upper confidence limit
BETASCAL	Multiplier for B that gave this power
EPSILON	Population Value of ϵ for UNIREP test power
EXEPS_GG	Approximate E$\hat{\epsilon}$ for the GG test
EXEPS_HF	Approximate E$\tilde{\epsilon}$ for the HF test

Table 6b _HOLDPOWER Column Labels ⇔ Output File Variable Names
Method Used To Calculate F Probability.

Power Lower CL Method	Power Method	Power Upper CL Method
FMETHOD_L	FMETHOD	FMETHOD_U
FMETHOD_MULT_L	FMETHOD_MULT	FMETHOD_MULT_U
FMETHOD_HLT_L	FMETHOD_HLT	FMETHOD_HLT_U
FMETHOD_PBT_L	FMETHOD_PBT	FMETHOD_PBT_U
FMETHOD_WLK_L	FMETHOD_WLK	FMETHOD_WLK_U
FMETHOD_UN_L	FMETHOD_UN	FMETHOD_UN_U
FMETHOD_HF_L	FMETHOD_HF	FMETHOD_HF_U
FMETHOD_GG_L	FMETHOD_GG	FMETHOD_GG_U
FMETHOD_BOX_L	FMETHOD_BOX	FMETHOD_BOX_U

Table 6c _HOLDPOWER Column Labels ⇔ Output File Variable Names
Method Used To Calculate F Probability.

Power Lower Conf. Limit	Power	Power Upper Conf. Limit
POWER_L	POWER	
POWER_MULT_L	POWER_MULT	POWER_MULT_U
POWER_HLT_L	POWER_HLT	POWER_HLT_U
POWER_PBT_L	POWER_PBT	POWER_PBT_U
POWER_WLK_L	POWER_WLK	POWER_WLK_U
POWER_UN_L	POWER_UN	POWER_UN_U
POWER_HF_L	POWER_HF	POWER_HF_U
POWER_GG_L	POWER_GG	POWER_GG_U
POWER_BOX_L	POWER_BOX	POWER_BOX_U

Table 6d _HOLDPOWER Column Labels ⇔ Output File Variable Names

Column Label ⇔ Variable Name	Description
MAXRHOSQ	Maximum canonical correlation
N_EST	# of obs. which gave Σ and/or B estimates
POWERCASE	Row number of _HOLDPOWER matrix
RANK_EST	rank(X) in analysis for Σ and/or B estimates
RHOSCAL	Multiplier for correlations from Σ
SIGSCAL	Multiplier for Σ
TOTAL_N	Total sample size
UCDF_UN	CDF approximation used for UN power
UCDF_HF	CDF approximation used for HF power
UCDF_GG	CDF approximation used for GG power
UCDF_BOX	CDF approximation used for BOX power
UMETHOD_HF	Method used for calculating $E(\tilde{\epsilon})$
UMETHOD_GG	Method used for calculating $E(\hat{\epsilon})$

1.11 Summary of Inputs Controlling The Program

Twenty-four global matrices, along with values in OPT_ON and OPT_OFF, are the only inputs to the program. The input matrices are listed in the following table, grouped by application. OPT_ON and OPT_OFF play a special role in controlling the program.

Table 7 All Possible Input Matrices By Category

Design & Hypothesis	Scenario Variations	Output & Methods Desired	Confidence Limits	Numeric	Approximation
ESSENCEX	REPN	OPT_ON	CLTYPE	ROUND	MMETHOD
BETA	BETASCAL	OPT_OFF	ALPHA_CL	TOLERANCE	UMETHOD
C	ALPHA		ALPHA_CU		UCDF
U	SIGSCAL	DSNAME	N_EST		
THETA0	RHOSCAL		RANK_EST		
SIGMA					

Table 8 All Possible Input Matrices

Name	Description	Size	Default
ALPHA	type I error rates	1 row or col	0.05
BETASCAL	List of multipliers for BETA	1 row or col	1
REPN	List specifying # of times to duplicate each row of ESSENCEX	1 row or col	1
RHOSCAL	List of multipliers for correlation matrix RHO, created from SIGMA	1 row or col	1
SIGSCAL	List of multipliers for SIGMA	1 row or col	1
THETA0	Matrix $\boldsymbol{\Theta}_0$ in GLH	$a \times b$	$\mathbf{0}$
U	Matrix $\boldsymbol{U}$ in GLH	$p \times b, b \leq p$	$\boldsymbol{I}_p$
MMETHOD	List specifying HLT, PBT, WLK power approximation	$3 \times 1, 1 \times 3,$ 1×1	$\begin{bmatrix} 4 & 2 & 2 \end{bmatrix}'$
UCDF	List specifying CDF approximation for UNIREP (UN, HF, GG, BOX)	$4 \times 1, 1 \times 4$ 1×1	$\begin{bmatrix} 2 & 2 & 2 & 2 \end{bmatrix}'$
UMETHOD	List specifying method for approximate E$\tilde{\epsilon}$ and E$\hat{\epsilon}$	$2 \times 1, 1 \times 2$ 1×1	$\begin{bmatrix} 2 & 2 \end{bmatrix}'$
CLTYPE	type of confidence interval to include in power calculations	1×1	-1
ALPHA_CL	Scalar specifying lower tail confidence limit probability	1×1	0.025
ALPHA_CU	Scalar specifying upper tail confidence limit probability	1×1	0.025
N_EST	# of observations in the analysis that yielded Σ and/or B estimates	1×1	None
RANK_EST	Scalar giving design matrix rank in the analysis that yielded Σ and B estimates	1×1	None
ROUND	Scalar specifying how many decimal places to round power values	1×1	3
TOLERANCE	Scalar specifying what the software considers numeric zero	1×1	1×10^{-12}
DSNAME	specifies output SAS file name and location	$1 \times 2, 2 \times 1,$ $1 \times 3, 3 \times 1$	{WORK DODFAULT WORK}
OPT_ON	user options to turn on	1 row or col	See later table
OPT_OFF	user options to turn off	1 row or col	See later table

1.12 Calling the Program from a User Written Module

The software can also be run by using a CALL statement. This allows using the power software within other modules. Note that the POWERLIB inputs matrices must all be specified to be global variables. The following incomplete code illustrates how to define a user module that calls the power software.

```
START user_defined_module ( parm1, parm2, ... parmn)
        GLOBAL (ESSENCEX, SIGMA, BETA, C, U, THETA0,
                 REPN, BETASCAL, SIGSCAL, RHOSCAL, ALPHA,
                 ROUND, TOLERANCE, TOLERANC, UCDF, UMETHOD, MMETHOD,
                 LIMCASE, CLTYPE, N_EST, RANK_EST, ALPHA_CL, ALPHA_CU,
                 DSNAME, OPT_ON, OPT_OFF);

*some of your code here;
CALL POWER(_HOLDPOWER, _HOLDPOWERLBL, _POWERWARN, _POWERWARNLBL);
*some more of your code here;

FINISH user_defined_module;
```

2. USER UTILITIES

Four modules are included which are useful for creating U contrast matrices. UMEAN, which generates a $p \times 1$ U matrix which computes the average response, $p^{-1}[1 \quad 1 \quad \ldots \quad 1]'$. For example, it could create the matrix $[1/5 \quad 1/5 \quad 1/5 \quad 1/5 \quad 1/5]'$. UPOLY1, UPOLY2, and UPOLY3 each generate U contrast matrices with orthogonal polynomial coding for one, two, or three repeated factors, respectively, via the SAS ORPOL function.

UMEAN is a function module with one input, p, which tells the size of the matrix to create. For example, to create a 4×1 matrix $U = [1/4 \quad 1/4 \quad 1/4 \quad 1/4]'$, you would execute the statement U = UMEAN(4);. For another value of p, simply say U = UMEAN(p);.

UPOLY1 takes two inputs, VALUES and NAME. Here VALUES is a $k \times 1$ vector or $1 \times k$ matrix which gives the k levels of the single repeated factor. NAME is a 1×1 character matrix describing the repeated factor. The module outputs two matrices U and ULBL. Columns of U contains up to level $k - 1$ polynomial contrasts; ULBL contains labels for the order of the polynomial contrast each column represents. Call the UPOLY1 module using the syntax.
```
CALL UPOLY1 (VALUES, NAME, U, ULBL);
```

As an example, to create a U matrix with orthogonal polynomial contrasts for four levels, 1, 10, 100, and 1000, of the factor DOSE, execute the following code.
```
LEVELS = {1, 10, 100, 1000};
FACTOR = {DOSE};
CALL UPOLY1 (LEVELS, FACTOR, U, ULBL);
PRINT U [COLNAME = ULBL];
```

The columns of U (as described by ULBL) are the linear, quadratic, and cubic polynomial trends for DOSE. This produces the matrix

$$
U = \begin{bmatrix} & DOSE1 & DOSE2 & DOSE3 \\ -0.330 & 0.442 & -0.667 \\ -0.320 & 0.314 & 0.741 \\ -0.212 & -0.836 & -0.074 \\ 0.862 & 0.080 & 0.001 \end{bmatrix}
$$

UPOLY2, and UPOLY3, generate U contrast matrices with orthogonal polynomial coding for two and three repeated factors, respectively. Modules UPOLY2 and UPOLY3 work the same way as the UPOLY1 module except they require 4 or 6 input matrices, respectively, and produce 6 or 12 matrices, respectively, due to the adding one or two more factors. Call the modules using the following syntax.
```
CALL UPOLY2 (VALUES1,NAME1,  VALUES2,NAME2,
                 U1,U1LBL,        U2,U2LBL,    U12,U12LBL );
```

```
CALL UPOLY3 (VALUES1,NAME1,  VALUES2,NAME2,  VALUES3,NAME3,
              U1,U1LBL,       U2,U2LBL,       U3,U3LBL,
              U12,U12LBL,     U13,U12LBL,     U23,U23LBL,   U123,U123LBL );
```

Here VALUES1, VALUES2, and VALUES3 give the levels for factors 1, 2, and 3, respectively, and NAME1, NAME2, NAME3 describe factors 1, 2, and 3, respectively.

Also, U1, U2, and U3 give the main effect contrasts for factors 1, 2, and 3, respectively and U12, U13, and U23 give the two way interaction contrasts for factor 1 with 2, 1 with 3, and 2 with 3, respectively. U123 gives the three way interaction contrasts for factor 1 with 2 and 3.

U1LBL, U2LBL, U3LBL, U12LBL, U13LBL, U23LBL, and U123LBL give column labels for matrices U1, U2, U3, U12, U13, U23, and U123, respectively.

The following example code gives all orthogonal polynomial trends for three factors, AGE, DRUG, and TIME, with levels {2, 4, 6}, {1, 2, 3}, and {10, 30, 60}.

```
LEVELS1 = {2, 4, 6};
NAME1 = "AGE";
LEVELS2 = {1, 2, 3};
NAME2 = "DRUG";
LEVELS3 = {10, 30, 60};
NAME3 = "TIME";
CALL UPOLY3 (LEVELS1,NAME1, LEVELS2,NAME2, LEVELS3,NAME3,
               U1,U1LBL,       U2,U2LBL,       U3,U3LBL,
             U12,U12LBL,  U13,U13LBL,  U23,U23LBL,   U123,U123LBL);
PRINT U1 [COLNAME = U1LBL];
PRINT U2 [COLNAME = U2LBL];
PRINT U3 [COLNAME = U3LBL];
PRINT U12 [COLNAME = U12LBL];
PRINT U13 [COLNAME = U13LBL];
PRINT U23 [COLNAME = U23LBL];
PRINT U123 [COLNAME = U123LBL];
```

Again, numbers in the column labels describe the degree of the polynomial trend corresponding to that column. Labels with one variable indicate a main effect; labels with two variable names indicate a two-way interaction; labels with three variables indicate a three-way interaction. To create a U orthogonal polynomial trends matrix including all trends for main effects, two-way interactions, and the three-way interaction, execute the following statements.

```
U = U1 || U2 || U3 || U12 || U13 || U23 || U123;
LBL = U1LBL || U2LBL || U3LBL || U12LBL || U13LBL || U23LBL || U123LBL;
```

3. CHOICE OF APPROXIMATION METHODS
3.1 Overview

Naturally, the default approximation methods have been chosen, given the current state of knowledge, to be the best available. Hence most users will never need to change the approximation methods from the default. The information here is included to allow comparing to POWERLIB 1 and other program output, and to allow for future developments.

All powers are approximated by noncentral F probabilities. For the MULTIREP tests, the Muller and Peterson (1984, MP) approach requires specifying approximate degrees of freedom, which implies a critical value via the FINV() function, and an approximate noncentrality. By default, the program uses two moment approximations (Rao, 1951; McKeon, 1974; Muller, 1998) for the null distributions, which imply degrees of freedom and critical values. Optionally, older and less accurate one moment methods are also available Pillai (1954, 1955, 1959). By default, the MULTIREP tests use the Muller and Peterson (1984, MP) noncentrality approximations. Optionally, each MULTIREP noncentrality may be multiplied by N/[N-

rank(X)] as recommended by O'Brien and Shieh (1992, OS). Using the OS multiplier gives slightly larger approximate powers. Especially for HLT, the original Muller and Peterson noncentralities can be somewhat conservative in small samples. Elements of MMETHOD correspond to choosing methods as follows.

3.2 Choosing MULTIREP Approximations With MMETHOD

MMETHOD[1] $\Rightarrow$ Choices for Hotelling-Lawley Trace

= 1 $\Rightarrow$ Pillai (1954, 55) one moment null approximation

= 2 $\Rightarrow$ McKeon (1974) two moment null approximation

= 3 $\Rightarrow$ Pillai (1959) one moment null approximation + OS noncen multiplier

= 4 $\Rightarrow$ McKeon (1974) two moment null approximation + OS noncen multiplier

MMETHOD[2] $\Rightarrow$ Choices for Pillai-Bartlett Trace

= 1 $\Rightarrow$ Pillai (1954, 55) one moment null approximation

= 2 $\Rightarrow$ Muller (1998) two moment null approximation

= 3 $\Rightarrow$ Pillai (1959) one moment null approximation + OS noncen multiplier

= 4 $\Rightarrow$ Muller (1998) two moment null approximation + OS noncen multiplier

MMETHOD[3] $\Rightarrow$ Choices for Wilks' Lambda

= 1 $\Rightarrow$ Rao (1951) two moment null approximation

= 2 $\Rightarrow$ Rao (1951) two moment null approximation

= 3 $\Rightarrow$ Rao (1951) two moment null approximation + OS noncen multiplier ON

= 4 $\Rightarrow$ Rao (1951) two moment null approximation + OS noncen multiplier ON

The duplication of settings for Wilks' test is merely for programming convenience.

The default setting is MMETHOD={4,2,2};. Using the O'Brien and Shieh multiplier for all three multivariate tests may be chosen by setting MMETHOD={4,4,4}; or MMETHOD=4; because MMETHOD can be specified as a 1×1 matrix.

By default, version 1 of POWERLIB used the Pillai null approximations for HLT and PBT, and did not use the OS multiplier, which corresponds to having MMETHOD={1,1,1} or MMETHOD={1,1,2}.

3.3 Choosing UNIREP Approximations With UCDF and UMETHOD

Through the UCDF values, the software allows choosing one of four approximations for the distribution of the UNIREP test statistic under the alternative.

UCDF[1] $\Rightarrow$ Choice for Uncorrected test

UCDF[2] $\Rightarrow$ Choice for Huynh-Feldt test

UCDF[3] $\Rightarrow$ Choice for Geisser-Greenhouse test

UCDF[4] $\Rightarrow$ Choice for Box test

UCDF[J] = 1 $\Rightarrow$ Muller and Barton (1989) one moment approximation

= 2 $\Rightarrow$ Muller, Edwards, and Taylor (2004) two moment approximation

= 3 $\Rightarrow$ exact via Davies' method (This may fail; if it does a missing value is returned.)

= 4 $\Rightarrow$ exact via Davies' Method, except if it fails, use Muller Edwards Taylor (2004) approximation.

The default is UCDF = {2, 2, 2, 2}; with all four UNIREP tests using the Muller, Edwards, Taylor (2004) approximation for the distribution the test statistic under the alternative. Exact results may be achieved, at the cost of computing time, for the uncorrected (UN) and Box tests by specifying UCDF = {3, 2, 2, 3};. Version 1 of POWERLIB used the equivalent of UCDF={1,1,1,1}; (the previous version did not support the Box test).

The UMETHOD list specifies whether to use the Muller and Barton (1989) or Muller, Edwards, and Taylor (2004) approximations for $E(\tilde{\epsilon})$ and $E(\hat{\epsilon})$. In turn, this option implies the approximate critical value used for the GG and HF tests.

UMETHOD[1] = 1 $\Rightarrow$ Muller and Barton (1989) approximation for $E(\tilde{\epsilon})$

UMETHOD[2] = 1 $\Rightarrow$ Muller and Barton (1989) approximation for $E(\hat{\epsilon})$

UMETHOD[1] = 2 $\Rightarrow$ Muller, Edwards, and Taylor (2004) approximation for $E(\tilde{\epsilon})$

UMETHOD[2] = 2 $\Rightarrow$ Muller, Edwards, and Taylor (2004) approximation for $E(\hat{\epsilon})$

The default is UMETHOD={2,2};. Version 1 of POWERLIB used the equivalent of UMETHOD={1, 1};.

4. LIMITATIONS AND KNOWN DIFFICULTIES

1) The user must avoid using any of the following matrix names except as input to the software: ESSENCEX, SIGMA, BETA, C, U, THETA0, REPN, BETASCAL, SIGSCAL, RHOSCAL, ALPHA, ROUND, TOLERANCE, TOLERANC, UCDF, UMETHOD, MMETHOD, LIMCASE, CLTYPE, N_EST, RANK_EST, ALPHA_CL, ALPHA_CU, DSNAME, OPT_ON, OPT_OFF.

2) The user will lose any values stored in the following output matrix names: _HOLDPOWER, _HOLDPOWERLBL, _HOLDWARN, _HOLDWARNLBL.

3) The code includes matrix names longer than 8 characters, a naming convention added in version 7. Hence the modules will not run in SAS versions 6.12 or earlier.

4) The software has not been tested with SAS version 7. The software has been tested using SAS version 8.2 for both Windows and UNIX operating systems.

5) Currently known bugs are as follows.

5.1) The SAS functions FNONCT and CNONCT will return missing values for extreme conditions, such as very high power. In turn, this will generate an error message and cause IML to stop. Reducing the range of input values will avoid the problem.

5.2) DSNAME conflicts can occur with multiple invocations of the power software in a single IML session, but not across sessions. The user may be misled by conflicting messages from IML and the SAS system about the existence or nonexistence of particular files.

5.3) Currently, a few options that can be reset by other options with the occurrence of special cases are not reset. The new value would carry over to a second invocation of the software. A simple solution is to repeat the OPT_ON and OPT_OFF matrices before the second RUN POWER. Possible examples involve THETA0, COLLAPSE, and CLTYPE.

6) The user familiar with IML may wish to reset the default values of the options. We urge retaining a copy of the original statements as a backup. Defaults are set in section E.2 of the IML module _POWER.

7) This software does not allow for compensation of the bias in confidence intervals due to truncation studied by Taylor and Muller (1996) and Muller and Pasour (1997).

5. HELP FOR OLD FRIENDS

5.1 Incompatibilities With Previous Versions of POWERLIB and POWERLIM

We list here changes affecting the user familiar with POWERLIB 1.0 or POWERLIM. See the previous sections of this manual for details about any matrices or modules named here.

1) UNIREP power variable names have been changed to allow a consistent naming convention.

2) The following matrices have been renamed.

Table 9 Matrices Renamed

Old Name	New Name
_HOLDPOW	_HOLDPOWER
_HOLNM	_HOLDPOWERLBL
TOLERANC	TOLERANCE

3) The following options, which are controlled by including them in OPT_ON or OPT_OFF, have been renamed or implemented in a different way.

Table 10 Options Renamed

Old Name	New Name
CASE	POWERCASE
UNIHF	HF
UNIGG	GG
UNI	UN
NONCEN_N	Incorporated in MMETHOD

4) The defaults have been changed to include HLT and PBT as default "on" and RHOSCAL and POWERCASE as default "off."

5) The MAXRHOSQ now always reports the population value. Earlier versions reported the value used in power approximation, which was smaller by the factor $[N - \text{rank}(\boldsymbol{X})]/N$ if $s > 1$ and NONCEN_N was off.

6) Version 1 of POWERLIB and POWERLIM defaults used the equivalent of

MMETHOD={1,1,1}; UMETHOD={1, 1}; UCDF = {1,1,1,1};.

5.1 Incompatibilities With Previous Versions of POWERLIM Alone

7) The confidence limit variable names have been changed to allow a consistent naming convention.

8) The input variable LIMCASE no longer exists. The variable CLTYPE serves the same function.

9) CLTYPE=1 corresponds to estimated Σ and fixed $\boldsymbol{B}$, and CLTYPE=1 $\Leftrightarrow$ LIMCASE=2 in POWERLIM,

10) CLTYPE=2 corresponds to estimated Σ and estimated $\boldsymbol{B}$, and CLTYPE=2 $\Leftrightarrow$ LIMCASE=3 in POWERLIM.

11) In POWERLIM, choosing LIMCASE=0 (fixed Σ and fixed $\boldsymbol{B}$) produced zero width confidence intervals (three columns with the same values). In the new program, choosing CLTYPE=0, or any other number less than 1.0, will have the same effect as CLTYPE being a zero by zero matrix (not defined). A single column of power values, without confidence limits, will be produced.

12) No analog of LIMCASE=2 (estimated $\boldsymbol{B}$ and fixed Σ) is supported. The decision to drop LIMCASE=2 was based upon a philosophical position that a Z test rather than an F should be used if the variance is really known.

5.2 New Features

Version 2 has many improvements: 1) more accurate approximations for the UNIREP tests, 2) exact confidence intervals for MULTIREP tests and approximate confidence intervals for UNIREP tests, 3) support of approximate and exact methods for the UNIREP Box test, 4) more stable and accurate F probability calculations, 5) more accurate critical value approximations for HLT and PBT, 6) more detailed warning information about computing accuracy, 7) additional options and user modules to simplify use for orthogonal polynomial contrasts, 8) support for less than full rank design matrices, 9) support for fractional sample size replication factors, 10) additional new output variables, and 11) documentation of a module interface that allows conveniently calling the power software from other IML programs.

The matrices for confidence limits are new to POWERLIB, although similar variables were introduced in POWERLIM. The matrices are CLTYPE, ALPHA_CL, ALPHA_CU, N_EST, and RANK_EST.

New optional input matrices are MMETHOD, UMETHOD, UCDF.

New global output matrices are _POWERWARN and _POWERWARNLBL.

New options specified in OPT_ON and OPT_OFF are {BOX, ESSENCEX, FMETHOD, FRACREPN, LTFR, MMETHOD, ORTHU, UCDF, UMETHOD, UN}.

New modules are included to help the user with constructing trends and orthogonal polynomial contrasts include UMEAN, UPOLY1, UPOLY2 and UPOLY3.

6. EXAMPLES

The code and output listings for all the following power programs are found in the "EXAMPLES" folder included in the ZIP file available for download on the website. They can be run in an interactive or a batch environment and do not require any preexisting files.

Only one change IS needed for the user to run them. The location of the IML code must be specified correctly in the statement
%INCLUDE "your location here"/NOSOURCE2;
The programs that produce plots will likely require changing the FILENAME statement, as well as a few GOPTIONS, such as DEVICE, to tailor the output to the particular computer. The DISPLAY (in GOPTIONS) may also need to be controlled.

EXAMPLEA01.SAS

The following program calculates power for a two-sample t-test. The hypothesis to be tested is that the two group means are equal.

```
TITLE1 "EXAMPLEA01.SAS--Two sample T-test";

*****************************************************************;
* Perform power calculations for a two sample T test,          ;
* replicating the results in "Increasing scientific power with ;
* statistical power", by K.E. Muller and V.A. Benignus,        ;
* Neurotoxicology and Teratology, vol 14, May-June, 1992       ;
* The code reports power for a limited number of predicted     ;
* differences in means, compared to the number of values       ;
* needed for plotting.  Code for plot is in ExampleA02.sas     ;
*****************************************************************;

PROC IML SYMSIZE=1000 WORKSIZE=2000;
%INCLUDE "..\IML\POWERLIB203.IML"/NOSOURCE2;

* Define inputs to power program;

SIGMA = {1};
SIGSCAL= {0.32 1.00 2.05};

ESSENCEX = I(2);
REPN = { 10 };

BETA = {0 1}`;
BETASCAL = DO(0,2.5,0.5);
C = {1 -1};

OPT_OFF= {C U};

RUN POWER;
```

EXAMPLEA02.SAS

The following program performs the same power calculations for an independent t-test as in EXAMPLEA01.SAS, but utilizes graphs to display power curves for a set of values of the variance and of the hypothesized difference in group means. This particular program produces two graphs, as shown below:
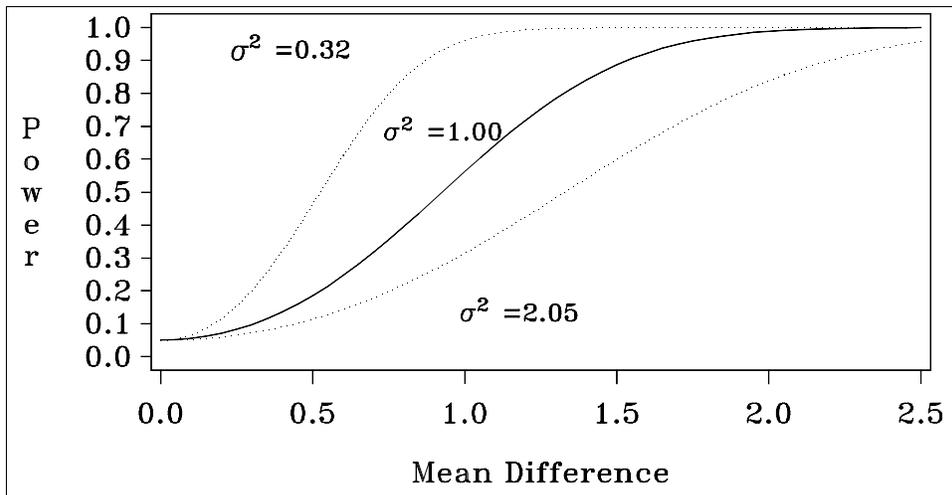

```
*TITLE1 "EXAMPLEA02.SAS--Two sample T-test Plot";
```

18

```
FILENAME OUT01 ".\EXAMPLEA02.CGM";

*****************************************************************;
* Performs power calculations for a two sample T test,        ;
* replicating the plots in "Increasing scientific power with  ;
* statistical power", by K.E. Muller and V.A. Benignus,       ;
* Neurotoxicology and Teratology, vol 14, May-June, 1992      ;
*****************************************************************;

PROC IML SYMSIZE=1000 WORKSIZE=2000;
%INCLUDE "..\IML\POWERLIB203.IML"/NOSOURCE2;

* Define inputs to power program;

SIGMA = {1};
SIGSCAL= {0.32 1.00 2.05};

ESSENCEX = I(2);
REPN = { 10 };

BETA = {0 1}`;
BETASCAL = DO(0, 2.5, 0.05);
C = {1 -1};

OPT_ON = {DS NOPRINT};

RUN POWER;

*The DS option creates a SAS file with the power calculation results.;
*Since no name for the file was specified, WORK.PWRDT1 is created.;

QUIT;

PROC CONTENTS DATA=PWRDT1;
RUN;

* Create Plots *;

PROC SORT DATA=PWRDT1
          OUT=ONE;
   BY BETASCAL SIGSCAL;

* create file for power curves of varying VARIANCE *;

PROC TRANSPOSE DATA=ONE OUT=TWO PREFIX=SIGPWR;
VAR POWER;
BY BETASCAL;

* create ANNOTATE dataset and assign symbols for labeling plots *;

DATA LABELS (KEEP= X Y XSYS YSYS TEXT STYLE SIZE);
LENGTH   TEXT $ 5  STYLE $ 8;
XSYS="2"; YSYS="2";

X=.26;  Y=.95; TEXT="s";     STYLE="cgreek"; *SIZE=1.0; OUTPUT;
X=.31;  Y=.97; TEXT="2";     STYLE="TRIPLEX"; SIZE=.75; OUTPUT;
X=.50;  Y=.95; TEXT="=0.32"; STYLE="TRIPLEX"; SIZE=1.0; OUTPUT;
X=.76;  Y=.70; TEXT="s";     STYLE="cgreek";  SIZE=1.0; OUTPUT;
X=.81;  Y=.72; TEXT="2";     STYLE="TRIPLEX"; SIZE=.75; OUTPUT;
X=1.00; Y=.70; TEXT="=1.00"; STYLE="TRIPLEX"; SIZE=1.0; OUTPUT;
```

```
X=1.01; Y=.15; TEXT="s";     STYLE="cgreek";  SIZE=1.0; OUTPUT;
X=1.06; Y=.17; TEXT="2";     STYLE="TRIPLEX"; SIZE=.75; OUTPUT;
X=1.25; Y=.15; TEXT="=2.05"; STYLE="TRIPLEX"; SIZE=1.0; OUTPUT;
RUN;

* The plot will be saved to this file for future inclusion in a document.*;
GOPTIONS GSFNAME=OUT01 DEVICE=CGMOF97P
CBACK=WHITE COLORS=(BLACK) HORIGIN=0IN VORIGIN=0IN
HSIZE=5IN VSIZE=3IN HTEXT=12PT FTEXT=TRIPLEX;

SYMBOL1 I=JOIN V=NONE L=34 W=1.0;
SYMBOL2 I=JOIN V=NONE L=1  W=1.0;
SYMBOL3 I=JOIN V=NONE L=34 W=1.0;
AXIS1 ORDER=(0 TO 1 BY .1)  W=1.5 MINOR=NONE MAJOR=(W=1.5)
      LABEL=(ANGLE=-90 ROTATE=90);
AXIS2 ORDER=(0 TO 2.5 BY .5) W=1.5 MINOR=NONE MAJOR=(W=1.5);

* The plot overlays power curves for three different variances *;
PROC GPLOT DATA=TWO;
PLOT SIGPWR1*BETASCAL=1
     SIGPWR2*BETASCAL=2
     SIGPWR3*BETASCAL=3/OVERLAY VAXIS=AXIS1 HAXIS=AXIS2 ANNOTATE=LABELS;
LABEL SIGPWR1="Power"  SIGPWR2="Power"  SIGPWR3="Power"
     BETASCAL="Mean Difference";
RUN;
QUIT;
```

EXAMPLEB01.SAS

The following program performs power calculations for a simple paired t-test using a general linear hypothesis in a multivariate setting.

```
TITLE1 "EXAMPLEB01.SAS--Paired T-test";

PROC IML SYMSIZE=2000;
%INCLUDE "..\IML\POWERLIB203.IML"/NOSOURCE2;

* Define matrices *;

ESSENCEX = I(1);
SIGMA = {2 1, 1 2};
BETA = {0 1};
C = {1};
U = {1 -1}`;

SIGSCAL= {1};
BETASCAL = DO(0,2.5,0.5);
REPN = { 10 };
OPT_ON = {COLLAPSE};
OPT_OFF= {C U};

RUN POWER;
QUIT;
```

EXAMPLEB02.SAS

This program performs power calculations for a paired t-test, equivalent to the results of EXAMPLEB01.SAS. However, the code uses difference scores to test the null hypothesis of no difference between group means.

```
TITLE1 "EXAMPLEB02.SAS--Paired T-test using Difference Scores";
* Equivalent to EXAMPLEB01.SAS ;
```

```
PROC IML SYMSIZE=2000;
%INCLUDE "..\IML\POWERLIB203.IML"/NOSOURCE2;

* Define matrices *;

ESSENCEX = I(1);
SIGMA = {2};                 * Variance of Difference of Means *;
BETA = {1};
C = {1};
U = {1};

SIGSCAL= {1};
BETASCAL = DO(0,2.5,0.5);
REPN = { 10 };
OPT_ON = {COLLAPSE};
OPT_OFF= {C U};

RUN POWER;
QUIT;
```

EXAMPLEC01.SAS

This program performs a more complicated set of power calculations for the test of the hypothesis of no time by treatment interaction.

```
TITLE1 "EXAMPLEC01.SAS--power for Time by Treatment Interaction";
TITLE2 "Compares power of tests when SIGMA most favorable to UN";

PROC IML SYMSIZE=1000 WORKSIZE=2000;
%INCLUDE "..\IML\POWERLIB203.IML"/NOSOURCE2;

OPT_OFF={ALPHA};
OPT_ON = {ORTHU UN HF GG BOX  HLT PBT WLK MMETHOD  UMETHOD MMETHOD};
* Specifying the option ORTHU in OPT_ON allows the program to provide;
* an orthonormal U matrix if one is not given by the user;
* This is the case for the following code;

P=3;
Q=4;
C=J(Q-1,1,1)||(-I(Q-1));
U=( J(P-1,1,1)||(-I(P-1)) )`;
ALPHA=.01;

VARIANCE=1;  RHO=0.4;
SIGMA=VARIANCE#(I(P)#(1-RHO) + J(P,P,RHO)); *Compound symmetry;
SIGSCAL={1, 2};

ESSENCEX=I(Q); REPN={5,10};
BETA=J(Q,P,0);
BETA[1,1]=1;
BETASCAL=DO(0, 2.0 , 0.50);

MMETHOD={4,4,4};    *two moment null approximations + OS noncen multiplier ON;

UCDF = {4,2,2,4};  * UN and Box (4) exact via Davies' algorithm (1980), as in;
                   * Muller Edwards Taylor (2003). If exact fails then ;
                   * switch to approximation 2, MET 2003;
                   * HF and GG: (2) Muller, Edwards, and Taylor (2004) approx;

RUN POWER;
QUIT;
```

EXAMPLED01.SAS

This program produces a very useful 2-dimensional graph when performing a power analysis. The program utilizes the power confidence limit calculations available in the software, replicating the figure seen in Taylor and Muller (1995).
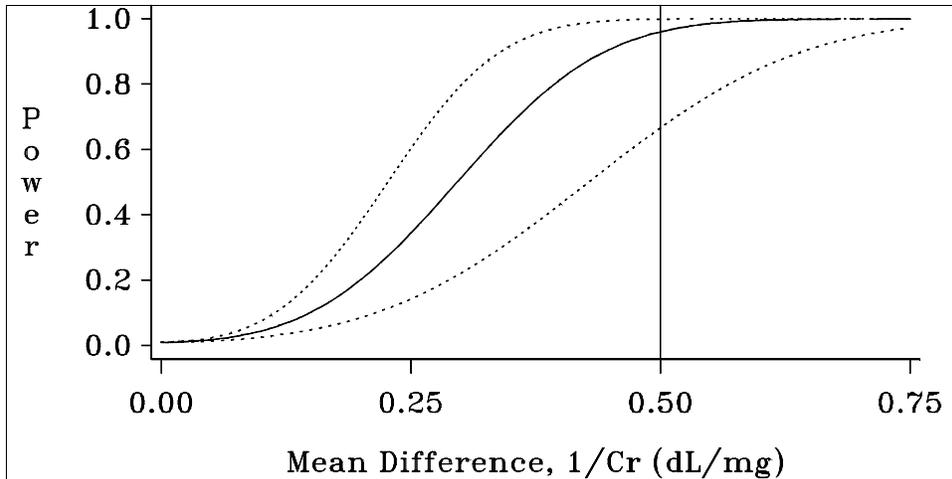

```
*TITLE1 "EXAMPLED01.SAS--Power Confidence Limit due to varhat";
*TITLE2 "Figure 1 Taylor and Muller, 1995, Amer Statistician, 49, p43-47";

FILENAME OUT01 ".\EXAMPLED01.CGM";

PROC IML WORKSIZE=1000 SYMSIZE=2000;
%INCLUDE "..\IML\POWERLIB203.IML"/NOSOURCE2;

OPT_OFF = {ALPHA};        *Turn options off;
OPT_ON  = {NOPRINT DS}; *Turn options on;

*Model is Y = XB + E;
*Specify essence of X matrix, the basic design;
ESSENCEX=I(2);  *Balanced two group t test, cell mean coding;
REPN={12};

*Hypothesis, as usual in ANOVA, is grand mean invariant so subtract;
BETA = {0 1}`;
BETASCAL=DO(0,.75,.01);
*Gives power for Beta={0  0}` , {0  .01}` , . . . {0  .75}`;

*Hypothesis is Ho: CBU=Theta0 ;
ALPHA={.01};  *Test size for hypothesis test;
C    ={1 -1}; * U=I and Theta0=0 are defaults; *This is t-test;

SIGMA  = {.068}; *Multivariate notation=Variance in univariate;
SIGSCAL= {1}   ; *Set of covariance matrix multipliers;

*Statements to create confidence limits;
CLTYPE=1;
N_EST=21;          *# Obs for variance estimate;
RANK_EST=1;        *# model df for study giving variance estimate;
ALPHA_CL=.025;   *Lower confidence limit tail size;
ALPHA_CU=.025;   *Upper confidence limit tail size;
RUN POWER; *IML and power code end here.  Power program creates file PWRDT1;
************************************************************;

*Remainder of code creates plot;
```

```
GOPTIONS GSFNAME=OUT01 DEVICE=CGMOF97P
CBACK=WHITE COLORS=(BLACK) HORIGIN=0IN VORIGIN=0IN
HSIZE=5IN VSIZE=3IN HTEXT=12PT FTEXT=TRIPLEX;

SYMBOL1 I=SPLINE V=NONE L=34 W=2;
SYMBOL2 I=SPLINE V=NONE L= 1 W=2;
SYMBOL3 I=SPLINE V=NONE L=34 W=2;
AXIS1 ORDER=(0 TO 1 BY .2)     W=3 MINOR=NONE MAJOR=(W=2)
      LABEL=(ANGLE=-90 ROTATE=90);
AXIS2 ORDER=(0 TO .75 BY .25) W=3 MINOR=NONE MAJOR=(W=2)
      LABEL=("Mean Difference, 1/Cr (dL/mg)");

PROC GPLOT DATA=PWRDT1;
PLOT (POWER_L POWER POWER_U)*BETASCAL/ OVERLAY NOFRAME HREF=.5
       VZERO VAXIS=AXIS1 HZERO HAXIS=AXIS2 NOLEGEND;
LABEL POWER_L="Power"  POWER  ="Power"  POWER_U="Power"
      SIGSCAL="Variance Multiplier";
```

EXAMPLEE01.SAS

This program produces a 3-dimensional graph that illustrate power tradeoffs among total sample size and the hypothesized difference between the two group means. The analysis in this particular example is for an independent groups t-test.
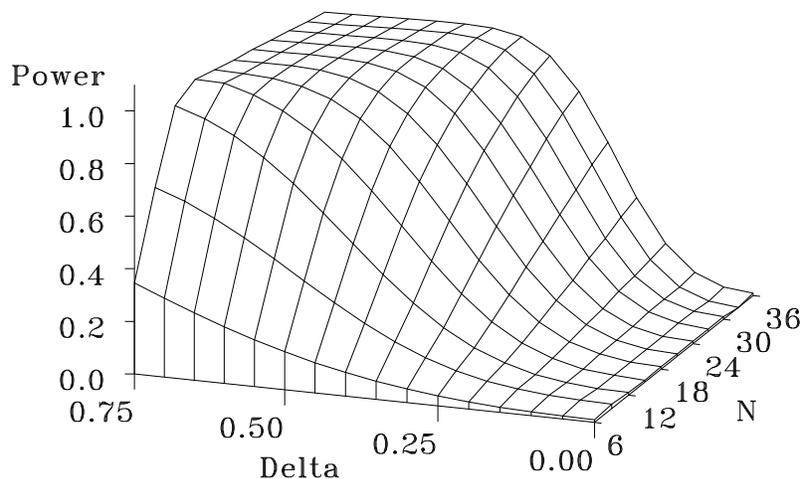

```
*TITLE1 "EXAMPLEE01.SAS--Power as a fct of Sample Size and Mean Dif";

FILENAME OUT01 ".\EXAMPLEE01.CGM";

PROC IML WORKSIZE=1000 SYMSIZE=2000;
%INCLUDE "..\IML\POWERLIB203.IML"/NOSOURCE2;

OPT_OFF = {WARN ALPHA};
OPT_ON={NOPRINT DS};
ROUND=6;
ALPHA={.01};
SIGMA = {.068};  SIGSCAL=  {1} ;
BETA = { 0    1}`;
C = {-1 1};
ESSENCEX=I(2);
REPN=DO(3,18,3);
BETASCAL=DO(0,.75,.05);
RUN POWER;
```

```
%MACRO GRAPH(TILT,ROTATE);
TITLE1;
PROC G3D DATA=ONE GOUT=OUT01;
PLOT BETASCAL*TOTAL_N=POWER/
     ZMIN=0 ZMAX=1.0 ZTICKNUM=6   YTICKNUM=4   XTICKNUM=6   SIDE;
LABEL TOTAL_N ="N"    BETASCAL="Delta"      POWER   ="Power" ;
RUN;
%MEND;

PROC G3GRID DATA=PWRDT1 OUT=ONE;
     GRID  BETASCAL*TOTAL_N=POWER /SPLINE NAXIS1=16 NAXIS2=11 ;

GOPTIONS GSFNAME=OUT01 DEVICE=CGMOF97P
CBACK=WHITE COLORS=(BLACK) HORIGIN=0IN VORIGIN=0IN
HSIZE=5IN VSIZE=3IN HTEXT=12PT FTEXT=TRIPLEX;

%GRAPH(90, 300);
```

EXAMPLEF01.SAS

This program demonstrates the use of the UPOLY1 module when performing power calculations for a time by treatment interaction.

```
TITLE1 "EXAMPLEF01.SAS--Illustrate use of the UPOLY1 module";

PROC IML WORKSIZE=1000 SYMSIZE=2000;
%INCLUDE "..\IML\POWERLIB203.IML"/NOSOURCE2;

OPT_ON ={GG HF HLT PBT TOTAL_N};
OPT_OFF={WARN WLK UN COLLAPSE ALPHA SIGSCAL BETASCAL};

ALPHA=.05;
VARIANCE=1.5;  RHO=0.25;
SIGMA=VARIANCE#(I(5)#(1-RHO) + J(5,5,RHO)); *Assume CS;

ESSENCEX = I(2);
REPN={10,20,40};

BETASCAL=1;
BETA = {0 0 0 0 1,
        1 0 0 0 0};
C = {1 -1};

TIMES ={2 4 6 8 10};
RUN UPOLY1(TIMES  ,"Time", USCORE , SCORENM );
U = USCORE;

RUN POWER;
QUIT;
```

EXAMPLEG01.SAS

This program demonstrates the direct creation of three way contrast matrices of two types, orthonormal polynomials, and pair-wise differences to a reference level. As long as cell mean coding is used for a factorial design (including the special case of a one-way design), the approach taken to create U matrices may be applied to C matrices, and vice versa, with the obvious change of transposing matrices. Although we recommend using UPOLY1-UPOLY3, this example is intended to provide a basis for creating contrasts for more unusual designs.

```
TITLE1 "EXAMPLEG01.SAS--direct programming of 2 types of contrasts";
TITLE2 "factorial design, repeated measures: A, B, C between, D, E, F within";
```

```
PROC IML SYMSIZE=1000 WORKSIZE=2000;
RESET FUZZ NOAUTONAME FW=6 LINESIZE=80;
%INCLUDE "..\IML\POWERLIB203.IML"/NOSOURCE2;

ALPHA=.05;

*Choose dimensions of design;
GA=3; *=# groups for between factor A;
GB=3; *=# groups for between factor B;
GC=3; *=# groups for between factor C;
TD=3; *=#Times for within factor D;
TE=3; *=#Times for within factor E;
TF=3; *=#Times for within factor F;


P=TD#TE#TF;
Q=GA#GB#GC;
ESSENCEX= I(Q);
BETA=J(Q,P,0);
BETA[1,1]=1;

SIGMA=DIAG(DO(1,P,1));*Variances are 1,2,3,...p;

*Get orthonormal submatrices for U matrices;
POLYD=ORPOL(1:TD);  UD1=POLYD[,2:NCOL(POLYD)]`;
POLYE=ORPOL(1:TE);  UE1=POLYE[,2:NCOL(POLYE)]`;
POLYF=ORPOL(1:TF);  UF1=POLYF[,2:NCOL(POLYF)]`;

*U matrix for Main effect D;
UD=(      UD1    @ J(1,TE,1   )    @ J(1,TF,1   ))` ;

*U matrix for Main effect E;
UE=(J(1,TD,1   ) @      UE1        @ J(1,TF,1   ) )`;

*U matrix for Main effect F;
UF=(J(1,TD,1   ) @ J(1,TE,1    )    @      UF1    )`;

*U matrix for DxE interaction;
UDE=HDIR(UD,UE);

*U matrix for DxExF interaction;
UDEF=HDIR(UDE,UF);

*Get submatrices for between factors;
CA1=J(GA-1,1,-1)||I(GA-1);
CB1=J(GB-1,1,-1)||I(GB-1);
CC1=J(GC-1,1,-1)||I(GC-1);

*Main effect A;
CA=   CA1    @ J(1,GB,1) @ J(1,GC,1) ;

*Main effect B;
CB=J(1,GA,1) @   CB1     @ J(1,GC,1) ;

*Main effect C[FORMAT=2.];
CC=J(1,GA,1) @ J(1,GB,1) @   CC1     ;

*AxB interaction;
CAB=(HDIR(CA`,CB`))`;

*AxBxC interaction;
```

25

```
CABC=(HDIR(CAB`,CC`))`;

BETASCAL={9 18 27};
ROUND=4;
OPT_ON={NOPRINT  GG HF UN  PBT HLT WLK};
OPT_OFF={WARN SIGSCAL ALPHA};
BUG=" ";

C=CA;
U=UD;
  DO REPN=2 TO 12 BY 2;
  RUN POWER;
  HOLDA=HOLDA//_HOLDPOWER;
  END;
PRINT / "AxD";
PRINT HOLDA[COLNAME=_HOLDPOWERLBL ROWNAME=BUG];

C=CAB;
U=UDE;
  DO REPN=2 TO 12 BY 2;
  RUN POWER;
  HOLDABDE=HOLDABDE//_HOLDPOWER;
  END;
PRINT / "AxB x DxE Interaction";
PRINT HOLDABDE[COLNAME=_HOLDPOWERLBL ROWNAME=BUG];

C=CABC;
U=UDEF;
  DO REPN=2 TO 12 BY 2;
  RUN POWER;
  HABCDEF=HABCDEF//_HOLDPOWER;
  END;
PRINT / "AxBxC x DxExF Interaction";
PRINT HABCDEF[COLNAME=_HOLDPOWERLBL ROWNAME=BUG];
```

EXAMPLEG02.SAS

Program EXAMPLEG01.SAS illustrates using UPOLY3 to provide the same results as in EXAMPLEG01.SAS.

```
TITLE1 "EXAMPLEG02.SAS--UPOLY3 for contrasts";
TITLE2 "factorial design, repeated measures: A, B, C between, D, E, F within";

PROC IML SYMSIZE=1000 WORKSIZE=2000;
RESET FUZZ NOAUTONAME FW=6 LINESIZE=80;
%INCLUDE "..\IML\POWERLIB203.IML"/NOSOURCE2;

ALPHA=.05;

*Choose dimensions of design;
GA=3; *=# groups for between factor A;
GB=3; *=# groups for between factor B;
GC=3; *=# groups for between factor C;
TD=3; *=#Times for within factor D;
TE=3; *=#Times for within factor E;
TF=3; *=#Times for within factor F;

P=TD#TE#TF;
Q=GA#GB#GC;
ESSENCEX= I(Q);
BETA=J(Q,P,0);
BETA[1,1]=1;
```

26

```
SIGMA=DIAG(DO(1,P,1));*Variances are 1,2,3,...p;

*Get orthonormal U matrices;
CALL UPOLY3 ( (1:TD),"D",  (1:TE),"E",  (1:TF),"F",
                         UD,UDLBL,     UE,UELBL,     UF,UFLBL,
             UDE,UDELBL,  UDF,UDFLBL,  UEF,UEFLBL,  UDEF,UDEFLBL );

*Get orthonormal C matrices;
CALL UPOLY3 ((1:GA),"A" , (1:GB),"B" , (1:GC),"C",
                         U1,CALBL,    U2,CBLBL,      U3,CCLBL,
             U12,CABLBL, U13,CACLBL, U23,CBCLBL,  U123,CABCLBL);

BETASCAL={9 18 27};
ROUND=4;
OPT_ON={NOPRINT  GG HF UN  PBT HLT WLK};
OPT_OFF={WARN SIGSCAL ALPHA};
BUG=" ";

C=U1`;
U=UD;
  DO REPN=2 TO 12 BY 2;
  RUN POWER;
  HOLDA=HOLDA//_HOLDPOWER;
  END;
PRINT / "AxD";
PRINT HOLDA[COLNAME=_HOLDPOWERLBL ROWNAME=BUG];

C=U12`;
U=UDE;
  DO REPN=2 TO 12 BY 2;
  RUN POWER;
  HOLDABDE=HOLDABDE//_HOLDPOWER;
  END;
PRINT / "AxB x DxE Interaction";
PRINT HOLDABDE[COLNAME=_HOLDPOWERLBL ROWNAME=BUG];

C=U123`;
U=UDEF;
  DO REPN=2 TO 12 BY 2;
  RUN POWER;
  HABCDEF=HABCDEF//_HOLDPOWER;
  END;
PRINT / "AxBxC x DxExF Interaction";
PRINT HABCDEF[COLNAME=_HOLDPOWERLBL ROWNAME=BUG];
```

EXAMPLEH01.SAS

This program illustrates using UPOLY2 and for two within and no between subject factor. It uses SIGSCAL and combines results from multiple runs of the power module. The results reproduce, except for some rounding differences, the predicted GG and HF powers in Table III in Coffey and Muller (2003), which used version 1 of POWERLIB, based on Muller and Barton (1989) methods. The example program also produces predicted HF and GG powers using the methods from Muller Edwards and Taylor (2003), which are the new default. The new methods are far more accurate, especially for very small (near $1/b$) or very large (near 1.0) values of ϵ.

```
TITLE1 "EXAMPLEH01.SAS-two within factors, UPOLY2, no between, vary SIGMA";
TITLE2 "From Coffey C.S. and Muller K.E. (2003) Properties of ";
TITLE3 "internal pilots with the univariate approach to repeated measures";
TITLE4 "Statistics in Medicine, 22(15), in press.";
```

```
PROC IML SYMSIZE=1000 WORKSIZE=2000;
%INCLUDE "..\IML\POWERLIB203.IML"/NOSOURCE2;

ALPHA=.04;
OPT_ON={NOPRINT  GG HF BOX TOTAL_N  UCDF UMETHOD};
OPT_OFF={WARN  ALPHA  BETASCAL  HLT PBT WLK };
ROUND=2;

BETASCAL=1;
THETA={.25}#{.5 1 -1 .5}; * =Theta(cr) from 1st sentence *after*
                          *  equation 7, Coffey and Muller (2003);

*Following from Table II in Coffey and Muller (2003);
VARSTARE={.47960 .01000 .01000 .01000}; *epsilon ~ .28;
VARSTARF={.34555 .06123 .05561 .04721}; *epsilon ~ .50;
VARSTARG={.23555 .17123 .05561 .04721}; *epsilon ~ .72;
VARSTARH={.12740 .12740 .12740 .12740}; *epsilon = 1;
VARSTAR=VARSTARE//VARSTARF//VARSTARG//VARSTARH;

SIGSCAL={0.50 1.00 2.00}; * <=> gamma in Coffey and Muller (2003);

*log base 2 spacing Clip (2,4,16) and Region(2,8,32);
*Get orthonormal U matrices;
RUN UPOLY2({1 2 4},"A", {1 3 5},"B",
           UA,NMA,    UB,NMB,  UAB ,NMAB);
U=UAB;
C=1;

ESSENCEX={1};
REPN={20};

  DO IVAR=1 TO 4 BY 1;
  SIGSTAR=DIAG(VARSTAR[IVAR,*]);

  SIGMA=U*SIGSTAR*U`;  *1st paragraph in section 2.4, Coffey and Muller 2003;
  BETA=THETA*U`;       *1st paragraph in section 2.4, Coffey and Muller 2003;

    DO VERSION=1 TO 2 BY 1;  *POWERLIB version;
    UCDF=J(4,1,VERSION);
    UMETHOD=J(2,1,VERSION);
    RUN POWER;
    HOLDALL=HOLDALL//_HOLDPOWER;
    END;
  END;

CREATE ONE VAR _HOLDPOWERLBL;
APPEND FROM HOLDALL;


PROC SORT DATA=ONE OUT=TWO;
BY UCDF_GG UMETHOD_GG SIGSCAL EPSILON;

PROC PRINT DATA=TWO UNIFORM NOOBS;
BY UCDF_GG UMETHOD_GG  UCDF_HF UMETHOD_HF UCDF_BOX TOTAL_N;
PAGEBY UCDF_GG;
TITLE5 "All data in file";

PROC PRINT DATA=TWO(RENAME=(SIGSCAL=GAMMA)) UNIFORM NOOBS;
VAR EPSILON GAMMA  POWER_GG POWER_HF ;
BY UCDF_GG UMETHOD_GG  UCDF_HF UMETHOD_HF UCDF_BOX TOTAL_N;
```

```
PAGEBY UCDF_GG;
TITLE5 "Version 2 far more accurate for Table III, Coffey and Muller (2003)";
```

EXAMPLEI01.SAS

This program utilizes a dataset that contains cerebral vessel tortuosity measures for subjects in four regions of the brain. A set of power calculations is performed for the test of the hypothesis of no gender by region interaction with five age groups.

```
*TITLE1 "EXAMPLEI01.SAS--Power for Region by Gender interaction with 5 age
groups";

LIBNAME IN01 "..\DATA\";

PROC IML SYMSIZE=2000 WORKSIZE=2000;
%INCLUDE "..\IML\POWERLIB203.IML"/NOSOURCE2;

OPT_ON = {GG HF BOX UN TOTAL_N ORTHU NOPRINT};
OPT_OFF={WARN};

*Importing covariance matrix of data*;

USE IN01.P0104;
READ ALL VAR {ANT LEFT POST RIGHT} WHERE(_TYPE_ = "COV") INTO INSIGMA;

*Rounding and viewing covariance matrix*;

RNM={ANT LEFT POST RIGHT};
ALPHA = .05/6;
SIGMA = ROUND(INSIGMA,.0001);
PRINT SIGMA[FORMAT=11.5 COLNAME=RNM];

*Define input matrices*;

ESSENCEX = I(10);
REPN ={10};

P = 4;
Q = 2;

*Pattern of means for Gender by Region*;

BETARG= J(2,4, 3.2) + {.30}#({ -1  0 1  0,
                              -1  0 1  0}) ;
PRINT BETARG[COLNAME=RNM];
BETASCAL={1};

C = {1 -1} @ J(1,5,1);

GROUP={F1 F2 F3 F4 F5  M1 M2 M3 M4 M5};
REGION = {1,2,3,4};
RUN UPOLY1(REGION,"REGION",U1,REGU);
U=U1;

DO DELTA=0 TO .10 BY .02;

*Creation of Beta matrix based on varying Gender differences;

  BETARGD=BETARG + (J(2,2,0)||(DELTA//(-DELTA))||J(2,1,0));
  BETA= BETARGD @ J(5,1,1) ;          *Final Beta matrix with age groups added;
  PRINT  / DELTA BETA[COLNAME=RNM ROWNAME=GROUP];
  RUN POWER;
```

```
   HOLDALL=HOLDALL//_HOLDPOWER;
END;

PRINT HOLDALL[COLNAME=_HOLDPOWERLBL];          *Power calculation results*;

QUIT;
```

EXAMPLEJ01.SAS

The following program uses the tortuosity data to produce a 2-dimensional graph with Geisser-Greenhouse power curves for sample sizes of 20, 40, and 80 for the gender by region interaction.
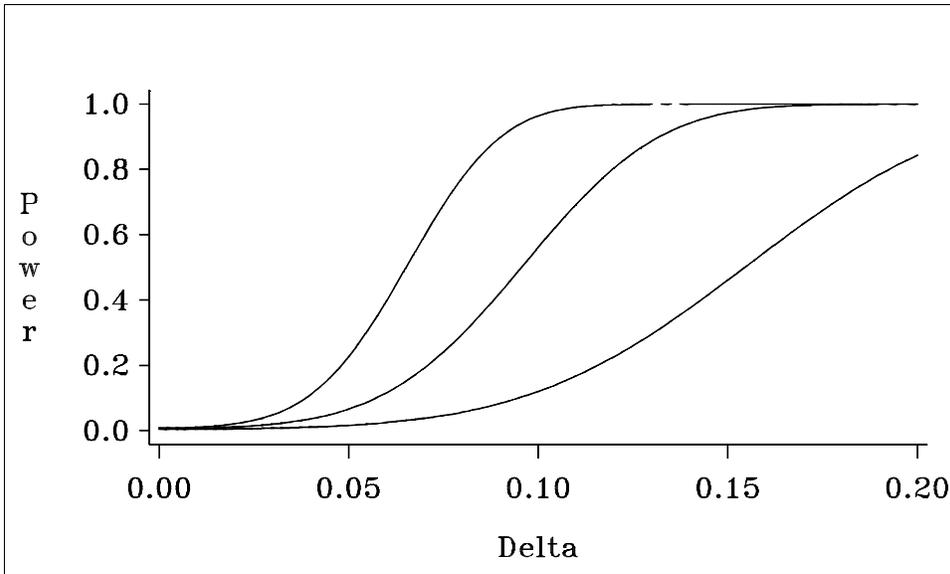

```
*TITLE1 "EXAMPLEJ01.SAS--Region by Gender power plot for N=20,40,80";

LIBNAME IN01 "..\DATA\";
FILENAME OUT01 ".\EXAMPLEJ01.CGM";

PROC IML SYMSIZE=2000 WORKSIZE=2000;
%INCLUDE "..\IML\POWERLIB203.IML"/NOSOURCE2;

OPT_ON = {GG HF BOX UN TOTAL_N ORTHU NOPRINT};
OPT_OFF={WARN ALPHA};

USE IN01.P0104;
READ ALL VAR {ANT LEFT POST RIGHT} WHERE(_TYPE_ = "COV") INTO INSIGMA;

ALPHA = .05/6;
SIGMA = ROUND(INSIGMA,.0001);

ESSENCEX = I(10);
REPN = {2 4 8};

P = 4;
Q = 2;
BETARG= J(2,4, 3.2) + {.30}#({ -1  0 1  0,
                               -1  0 1  0}) ;
BETASCAL={1};

C = {1 -1} @ J(1,5,1);

REGION = {1,2,3,4};
```

30

```
RUN UPOLY1(REGION,"REGION",U1,REGU);
U=U1;

DO DELTA=0 TO .20 BY .0025;
   BETARGD=BETARG + (J(2,2,0)||(DELTA//(-DELTA))||J(2,1,0));
   BETA= BETARGD @ J(5,1,1) ;
   RUN POWER;
   HOLDALL=HOLDALL//( _HOLDPOWER||J(NROW(_HOLDPOWER),1,DELTA) );
END;

*Creation of dataset from power calculations*;

NAME={"SIGSC" "BETASC" "N" "P_MULT" "EPS" "P_UN" "EXEPS_HF" "P_HF" "EXEPS_GG"
"P_GG" "P_BOX" "DELTA"};
CREATE ONE FROM HOLDALL [COLNAME = NAME];
APPEND FROM HOLDALL;
CLOSE ONE;

PROC SORT DATA=ONE OUT=TWO;
BY N DELTA;

DATA THREE;
     MERGE TWO( WHERE=(N=20) RENAME=(P_GG=P20) )
           TWO( WHERE=(N=40) RENAME=(P_GG=P40) )
           TWO( WHERE=(N=80) RENAME=(P_GG=P80) );

*Power curves for N=20,40,80*;

GOPTIONS GSFNAME=OUT01 DEVICE=CGMOF97P
CBACK=WHITE COLORS=(BLACK) HORIGIN=0IN VORIGIN=0IN
HSIZE=5IN VSIZE=3IN HTEXT=12PT FTEXT=TRIPLEX;

SYMBOL1 I=SPLINE V=NONE L=1 W=2 R=3;

AXIS1 ORDER=(0 TO 1.0 BY .20)    W=3 MINOR=NONE MAJOR=(W=2)
      LABEL=(ANGLE=-90 ROTATE=90);
AXIS2 LABEL=("Delta") ORDER=(0 TO .20 BY .05) W=3 MINOR=NONE MAJOR=(W=2);

PROC GPLOT DATA=THREE;
PLOT (P20 P40 P80)*DELTA/ OVERLAY NOFRAME
      VZERO VAXIS=AXIS1 HZERO HAXIS=AXIS2 NOLEGEND;
LABEL P20 ="Power" P40 ="Power" P80 ="Power";

RUN; QUIT; RUN;
```

EXAMPLEK01.SAS

The following program uses the tortuosity data to produce a set of 3-dimensional graphs that illustrate the tradeoffs among sample size, expected size of the effect, and Geisser-Greenhouse power for the gender by region interaction.
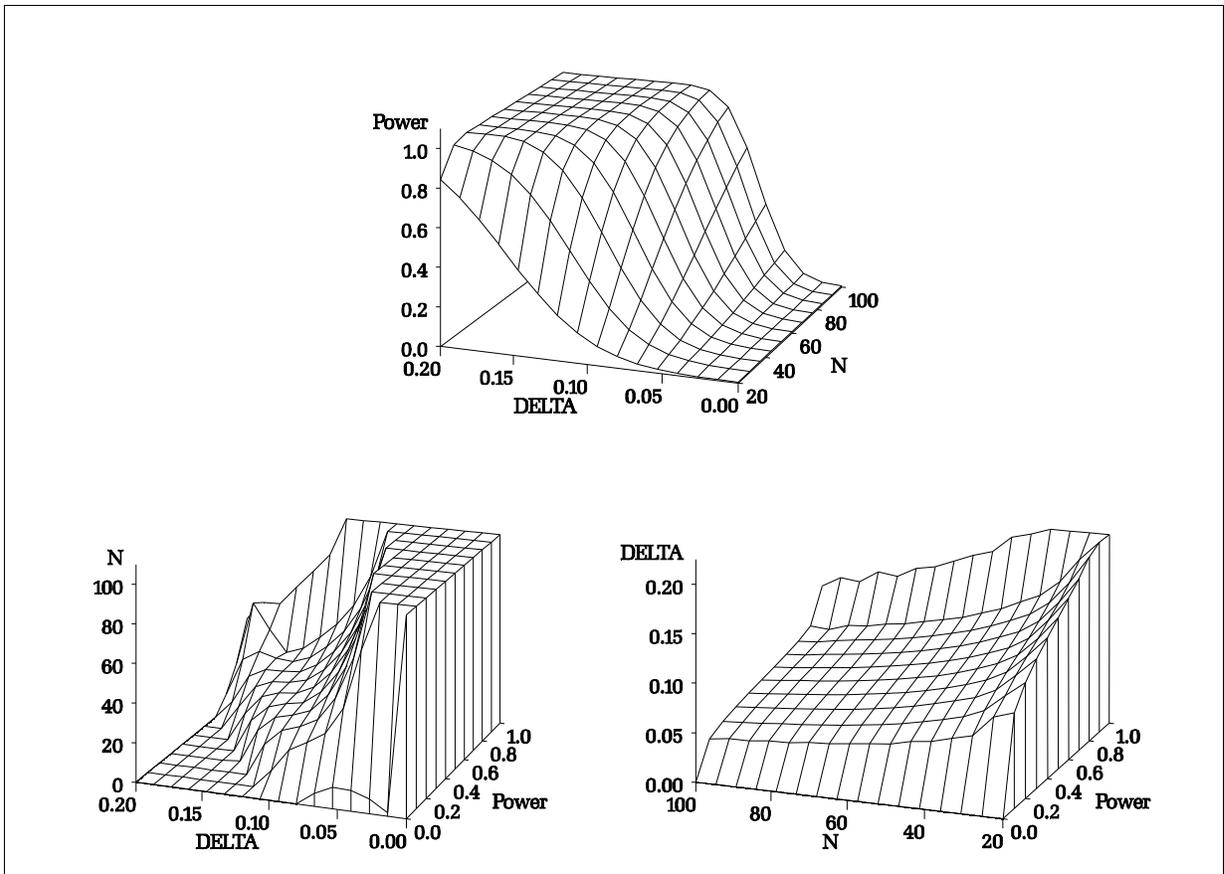
```
*TITLE1 "EXAMPLEK01.SAS--Region by Gender 3D power plots";

LIBNAME INOUT01 "..\DATA\";
FILENAME OUT01 ".\EXAMPLEK01.CGM";

PROC IML SYMSIZE=2000 WORKSIZE=2000;
%INCLUDE "..\IML\POWERLIB203.IML"/NOSOURCE2;

OPT_ON = {GG HF BOX UN TOTAL_N ORTHU NOPRINT};
OPT_OFF={WARN ALPHA};

USE INOUT01.P0104;
READ ALL VAR {ANT LEFT POST RIGHT} WHERE(_TYPE_ = "COV") INTO INSIGMA;

ALPHA = .05/6;
SIGMA = ROUND(INSIGMA,.0001);

ESSENCEX = I(10);
REPN = DO(1,10, 1);

P = 4;
Q = 2;
BETARG= J(2,4, 3.2) + {.30}#({ -1  0 1  0,
                               -1  0 1  0}) ;

BETASCAL={1};

C = {1 -1} @ J(1,5,1);
```

```
REGION = {1,2,3,4};
RUN UPOLY1(REGION,"REGION",U1,REGU);
U=U1;

DO DELTA=0 TO .20 BY .0008;
   BETARGD=BETARG + (J(2,2,0)||(DELTA//(-DELTA))||J(2,1,0));
   BETA= BETARGD @ J(5,1,1) ;
   RUN POWER;
   HOLDALL=HOLDALL//( _HOLDPOWER||J(NROW(_HOLDPOWER),1,DELTA) );
END;

NAME={"SIGSC" "BETASC" "N" "P_MULT" "EPS" "P_UN" "EXEPS_HF" "P_HF" "EXEPS_GG"
"P_GG" "P_BOX" "DELTA"};
CREATE ONE FROM HOLDALL [COLNAME = NAME];
APPEND FROM HOLDALL;
CLOSE ONE;

DATA TWO;
N=100; DELTA=0; P_GG=0;        *Create 'dummy' entry for graph*;
RUN;

DATA INOUT01.EXK01;     *Dataset also used in EXAMPLEK02.SAS*;
SET ONE TWO;
KEEP N P_GG DELTA;
RUN;

*3D Power curves*;

%MACRO GRAPH(TILT,ROTATE);
PROC G3D DATA = THREE GOUT = FOUR;
      PLOT DELTA*N = P_GG/
            ZMIN=0 ZMAX=1.0 ZTICKNUM=6 YTICKNUM=5 XTICKNUM=5;
      LABEL P_GG = "Power";
RUN;
%MEND;

PROC G3GRID DATA = INOUT01.EXK01 OUT = THREE;
      GRID DELTA*N = P_GG/SPLINE NAXIS1=16 NAXIS2=11;

GOPTIONS GSFNAME=OUT01 DEVICE=CGMOF97P
CBACK=WHITE COLORS=(BLACK) HORIGIN=0IN VORIGIN=0IN
HSIZE=7IN VSIZE=5IN HTEXT=16PT FTEXT=ZAPF;

%GRAPH(90,300);
RUN;QUIT;RUN;


%MACRO GRAPH(TILT,ROTATE);
PROC G3D DATA = THREE GOUT = FOUR;
      PLOT DELTA*P_GG = N/
            ZMIN=0 ZMAX=100 ZTICKNUM=6 YTICKNUM=5 XTICKNUM=6 SIDE;
      LABEL P_GG = "Power";
RUN;
%MEND;

PROC G3GRID DATA = INOUT01.EXK01 OUT = THREE;
      GRID DELTA*P_GG = N/SPLINE NAXIS1=16 NAXIS2=11;

GOPTIONS GSFNAME=OUT01 DEVICE=CGMOF97P
CBACK=WHITE COLORS=(BLACK) HORIGIN=0IN VORIGIN=0IN
HSIZE=7IN VSIZE=5IN HTEXT=16PT FTEXT=ZAPF;
```

```
%GRAPH(90,300);
RUN;QUIT;RUN;


%MACRO GRAPH(TILT,ROTATE);
PROC G3D DATA = THREE GOUT = FOUR;
      PLOT N*P_GG = DELTA/
            ZMIN=0 ZMAX=0.2 ZTICKNUM=5 YTICKNUM=5 XTICKNUM=6 SIDE;
      LABEL P_GG = "Power";
RUN;
%MEND;

PROC G3GRID DATA = INOUT01.EXK01 OUT = THREE;
      GRID N*P_GG = DELTA/SPLINE NAXIS1=16 NAXIS2=11;

GOPTIONS GSFNAME=OUT01 DEVICE=CGMOF97P
CBACK=WHITE COLORS=(BLACK) HORIGIN=0IN VORIGIN=0IN
HSIZE=7IN VSIZE=5IN HTEXT=16PT FTEXT=ZAPF;

%GRAPH(90,300);
RUN;QUIT;RUN;

PROC GREPLAY IGOUT=FOUR TC=TEMPCAT NOFS;
TDEF TEMP2BY2 DES="2 rows, 2 columns (3 plots)"
 1/LLX= 25 ULX= 25 LRX= 75 URX= 75    LLY=50 ULY=100 LRY=50 URY=100
 2/LLX= 0 ULX= 0 LRX= 50 URX= 50    LLY=0 ULY=50 LRY=0 URY=50
 3/LLX= 50 ULX= 50 LRX= 100 URX= 100    LLY=0 ULY=50 LRY=0 URY=50;
TEMPLATE TEMP2BY2;
TREPLAY 1:G3D  2:G3D1 3:G3D2;
RUN;QUIT; RUN;
```

EXAMPLEK02.SAS

This program produces a plot of the hypothesized gender by region interaction with a sample size of 100 and an approximate Geisser-Greenhouse power of 0.90.
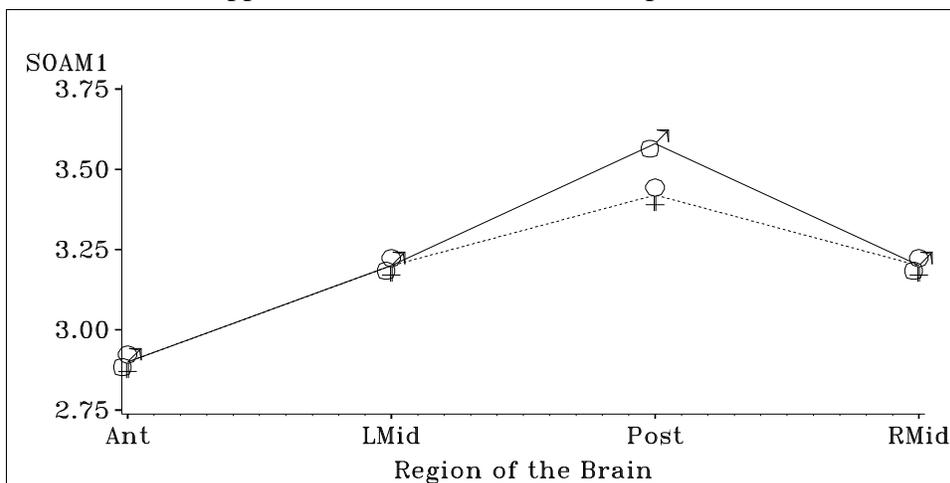

```
*TITLE1 "EXAMPLEK02.SAS--Region by Gender power, Beta Plot for 90% power";

LIBNAME IN01 "..\DATA\";
FILENAME OUT01 ".\EXAMPLEK02.CGM";

*Extracting data with GG power >= 0.89*;
```

```
DATA ONE;
SET IN01.EXK01;
IF N = 100 AND P_GG >= 0.89;
KEEP N P_GG DELTA;
RUN;


PROC IML SYMSIZE=2000 WORKSIZE=2000;

USE ONE;
READ ALL VAR {DELTA} INTO INDELTA;


BETARG= J(2,4, 3.2) + {.30}#({ -1  0 1  0,
                               -1  0 1  0});


*Creating dataset for Beta plot*;


RESET PRINT;
DELTA = INDELTA[1,1];    *Value of delta for which power is approx. 90%*;
BETARGD=(BETARG + (J(2,2,0)||(DELTA//(-DELTA))||J(2,1,0)))`;
HOLD1= BETARGD[,1]//BETARGD[,2];
HOLD2=((DO(1,4,1))`) // ((DO(1,4,1))`) ;
HOLD3=J(4,1,1)//J(4,1,2);
HOLD=HOLD1||HOLD2||HOLD3;
HNAMES={"SOAM1" "IREGION" "GENDER"};
CREATE TWO FROM HOLD [COLNAME = HNAMES];
APPEND FROM HOLD;
CLOSE TWO;
QUIT;


GOPTIONS GSFMODE=REPLACE GACCESS=SASGASTD GSFNAME=OUT01
         CBACK=WHITE COLORS=(BLACK) DEVICE=CGMOF97P
         HTITLE=.1666 IN HTEXT=.1666 IN HBY=.1666 IN
         HPOS=80 VPOS=60 HSIZE=6.0 IN VSIZE=3.0 IN
         FTEXT=TRIPLEX FTITLE=TRIPLEX FBY=TRIPLEX;

AXIS1 LABEL=("SOAM1") ORDER=(2.75 TO 3.75 BY 0.25) MAJOR=(W=3) MINOR=NONE;
AXIS2 LABEL=("Region of the Brain") MAJOR=(W=3) VALUE=("Ant" "LMid" "Post"
"RMid");

SYMBOL1 I=JOIN L=1 VALUE=e H=6 FONT=SPECIAL;
SYMBOL2 I=JOIN L=2 VALUE=c H=6  FONT=SPECIAL;

PROC GPLOT DATA=TWO;
  PLOT SOAM1*IREGION=GENDER / NOLEGEND NOFRAME VAXIS=AXIS1 HAXIS=AXIS2;
RUN;
QUIT;
```

EXAMPLEL01.SAS

Program EXAMPLEL01.SAS reproduces results from EXAMPLED01.SAS making use of a One-Sided Confidence Limit (Lower Bound Alpha = 0.05, Upper Bound Alpha = 0) rather than a Two-Sided CL. The program utilizes the power confidence limit calculations available in the software, replicating the figure seen in Taylor and Muller (1995).
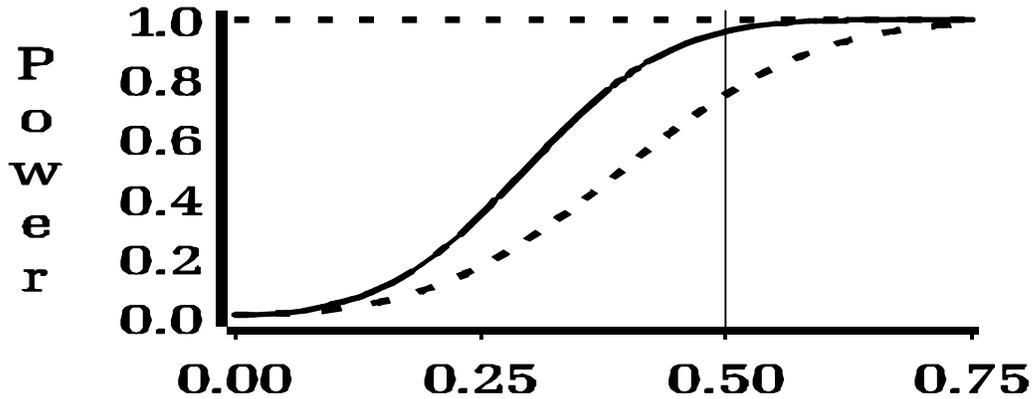

```
*TITLE1 "EXAMPLEL01.SAS--one sided Power Confidence Limit due to varhat";

/* Test whether specifying
      ALPHA_CL=.05;   *Lower confidence limit tail size;
      ALPHA_CU= 0;    *Upper confidence limit tail size;
   works correctly  */

FILENAME OUT01 ".\&PROG..EMF";

PROC IML WORKSIZE=1000 SYMSIZE=2000;
%INCLUDE "..\IML\POWERLIB203.IML" / NOSOURCE;

OPT_OFF = {ALPHA WARN  SIGSCAL TOTAL_N  CLTYPE  ALPHA_CL ALPHA_CU};
OPT_ON  = { DS};

ESSENCEX=I(2);  *Balanced two group t test, cell mean coding;
REPN={12};

BETA = {0 1}`;
BETASCAL=DO(0,.75,.005);

ALPHA={.01};  *Test size for hypothesis test;
C    ={1 -1}; * U=I and Theta0=0 are defaults; *This is t-test;

SIGMA  = {.068}; *Multivariate notation=Variance in univariate;
SIGSCAL= {1}   ;

*Statements to create confidence limits;
CLTYPE=1;
N_EST=24;         *# Obs for variance estimate;
RANK_EST=2;       *# model df for study giving variance estimate;
ALPHA_CL=0.05;    *Lower confidence limit tail size;
ALPHA_CU= 0;      *Upper confidence limit tail size;
RUN POWER;
************************************************************;
```

```
*Remainder of code creates plot;
GOPTIONS GSFNAME=OUT01 DEVICE=emf
CBACK=WHITE COLORS=(BLACK) HORIGIN=0IN VORIGIN=0IN
HSIZE=6IN VSIZE=4IN HTEXT=24PT FTEXT=ZAPF;

SYMBOL1 I=SPLINE V=NONE L=34 W=4;
SYMBOL2 I=SPLINE V=NONE L= 1 W=4;
SYMBOL3 I=SPLINE V=NONE L=34 W=4;
AXIS1 ORDER=(0 TO 1 BY .2)    W=6 MINOR=NONE MAJOR=(W=2)
      LABEL=(ANGLE=-90 ROTATE=90);
AXIS2 ORDER=(0 TO .75 BY .25) W=6 MINOR=NONE MAJOR=(W=2)
      LABEL=NONE;
*      LABEL=("Mean Difference, 1/Cr (dL/mg)");

PROC GPLOT DATA=PWRDT1;
PLOT (POWER_L POWER POWER_U)*BETASCAL/ OVERLAY NOFRAME HREF=.5
        VZERO VAXIS=AXIS1 HZERO HAXIS=AXIS2 NOLEGEND;
LABEL POWER_L="Power"  POWER  ="Power"  POWER_U="Power";
run; quit;
```

EXAMPLEL02.SAS

   Program EXAMPLEL02.SAS reproduces results from EXAMPLED01.SAS making use of a
One-Sided Confidence Limit (Lower Bound Alpha = 0, Upper Bound Alpha = 0.05) rather than
a Two-Sided CL. The program utilizes the power confidence limit calculations available in the
software, replicating the figure seen in Taylor and Muller (1995).
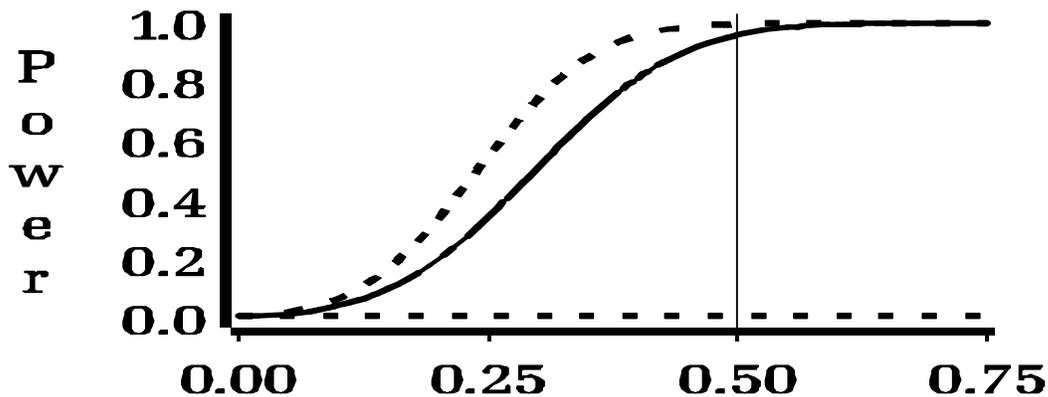

```
*TITLE1 "EXAMPLEL02.SAS--one sided Power Confidence Limit due to varhat";

/* Test whether specifying
     ALPHA_CL=  0;   *Lower confidence limit tail size;
     ALPHA_CU=.05;   *Upper confidence limit tail size;
   works correctly  */

FILENAME OUT01 ".\&PROG..EMF";

PROC IML WORKSIZE=1000 SYMSIZE=2000;
%INCLUDE "..\IML\POWERLIB203.IML" / NOSOURCE;

OPT_OFF = {ALPHA WARN  SIGSCAL TOTAL_N  CLTYPE  ALPHA_CL ALPHA_CU};
OPT_ON  = { DS};

ESSENCEX=I(2);  *Balanced two group t test, cell mean coding;
REPN={12};
```

```
BETA = {0 1}`;
BETASCAL=DO(0,.75,.005);

ALPHA={.01};  *Test size for hypothesis test;
C    ={1 -1}; * U=I and Theta0=0 are defaults; *This is t-test;

SIGMA  = {.068}; *Multivariate notation=Variance in univariate;
SIGSCAL= {1}   ;

*Statements to create confidence limits;
CLTYPE=1;
N_EST=24;        *# Obs for variance estimate;
RANK_EST=2;      *# model df for study giving variance estimate;
ALPHA_CL=0;    *Lower confidence limit tail size;
ALPHA_CU=0.05;   *Upper confidence limit tail size;
RUN POWER;
*********************************************************;

*Remainder of code creates plot;
GOPTIONS GSFNAME=OUT01 DEVICE=emf
CBACK=WHITE COLORS=(BLACK) HORIGIN=0IN VORIGIN=0IN
HSIZE=6IN VSIZE=4IN HTEXT=24PT FTEXT=ZAPF;

SYMBOL1 I=SPLINE V=NONE L=34 W=4;
SYMBOL2 I=SPLINE V=NONE L= 1 W=4;
SYMBOL3 I=SPLINE V=NONE L=34 W=4;
AXIS1 ORDER=(0 TO 1 BY .2)    W=6 MINOR=NONE MAJOR=(W=2)
      LABEL=(ANGLE=-90 ROTATE=90);
AXIS2 ORDER=(0 TO .75 BY .25) W=6 MINOR=NONE MAJOR=(W=2)
      LABEL=NONE;
*      LABEL=("Mean Difference, 1/Cr (dL/mg)");

PROC GPLOT DATA=PWRDT1;
PLOT (POWER_L POWER POWER_U)*BETASCAL/ OVERLAY NOFRAME HREF=.5
      VZERO VAXIS=AXIS1 HZERO HAXIS=AXIS2 NOLEGEND;
LABEL POWER_L="Power"  POWER ="Power"  POWER_U="Power";
run; quit;
```

7. AUTHORSHIP AND ACKNOWLEDGMENTS

   Over the past twenty years, this power software has evolved through multiple versions. Bercedis Peterson (now at Duke University) and Keith Muller wrote POWERLIB for PROC MATRIX in 1984. In 1992, Lynette Keyes (now at Rho, Inc.) and Keith Muller updated this to POWERLIB 1 for SAS/IML. Keyes also provided the first version of this manual, which was in turn based on the PROC MATRIX manual. Doug Taylor (now at Family Health International) created code for confidence limits calculations in 2001, which Keith Muller then used to create POWERLIM. Finally, in 2003, Jacqueline Johnson (UNC) and Keith Muller wrote the current version, POWERLIB 2, with additions as described in "Help for Old Friends". Additionally, James (Chris) Slaughter (UNC) greatly improved the _PROBF module, and Matthew Gurka (UNC) conducted extensive tests. Finally, this manual was significantly revised from the previous version.

BIBLIOGRAPHY

Arnold, S. F. (1981) *The Theory of Linear Models and Multivariate Analysis*. New York: Wiley.

Coffey C. S. and Muller K. E. (2003) Properties of internal pilots with the univariate approach to repeated measures, *Statistics in Medicine*, **22**, 2469-2485.

Davies, R. B. (1980) Algorithm AS 155: The distribution of a linear combination of chi-square random variables. *Applied Statistics*, **29**, 323-333.

Glueck D. H. and Muller K. E. (2003) Adjusting power for a baseline covariate in a linear model, *Statistics in Medicine*, **22**, 2535-2551.

Helms, R. W. (1988) Definitions of linear model parameters and hypotheses as functions of E(Y). *Communications in Statistics: Theory and Method*, **17**, 2715-2723.

Johnson, N. L. and Kotz, S. and Balakrishnan N. (1995) *Continuous Univariate Distributions - 2* (2nd edition). New York: Wiley.

Kim, H., Gribbin, M. J., Muller, K. E. and Taylor, D. J. (2005) Analytic, computational and approximate forms for ratios of noncentral and central gaussian quadratic forms, *in review*.

McKeon, J. J. (1974) F approximations to the distribution of Hotelling's generalized T_0^2 statistic, *Biometrika*, **61**, 381-383.

Muller, K. E. (1998) A new F approximation for the Pillai-Bartlett trace under H_0, *Journal of Computational and Graphical Statistics*, **7**, 131-137.

Muller, K. E., and Barton, C. N., (1989) Approximate power for repeated-measures ANOVA lacking sphericity, *Journal of the American Statistical Association*, **84**, 549-555.

Muller K. E. and Benignus V. A. (1992) Increasing scientific power with statistical power, *Neurotoxicology and Teratology*, **14**, 211-219.

Muller, K. E., Edwards, L. J., Simpson S. L. and Taylor, D. J. (2005) Accurate test size and power in small samples of repeated measures without missing data, *in review*.

Muller K. E., and Fetterman B. A. (2002) *Regression and ANOVA: An Integrated Approach Using SAS® Software*. Cary, NC: SAS Institute.

Muller, K. E., LaVange, L. M., Ramey, S. L., and Ramey, C. T. (1992) Power calculations for general linear multivariate models including repeated measures applications, *Journal of the American Statistical Association*, **87**, 1209-1226.

Muller K. E. and Pasour V. B. (1997) Bias in linear model power and sample size due to estimating variance, *Communications in Statistics: Theory and Methods*, **26**, 839-851.

O'Brien R. G. and Muller K. E. (1993) A unified approach to statistical power for t-tests to multivariate models, Chapter 8, *Applied Analysis of Variance in Behavioral Sciences*, L. K. Edwards, ed., 297-344. New York: Marcel Dekker.

O'Brien, R. G. and Shieh, G. (1992) Pragmatic, unifying algorithm gives power probabilities for common tests of the multivariate general linear hypothesis, *Unpublished manuscript*, University of Florida.

Pillai, K. C. S. (1954) On some distribution problems in multivariate analysis, *Mimeo Series*, **88**, Institute of Statistics, University of North Carolina, Chapel Hill.

Pillai, K. C. S. (1955) Some new test criteria in multivariate analysis, *Annals of Mathematical Statistics*, **26**, 117-121.

Pillai, K. C. S., and Samson, P. (1959) On Hotelling's generalization of T^2, *Biometrika*, **46**, 160-168.

Rao, C. R. (1951) An asymptotic expansion of the distribution of Wilks' criterion, *Bulletin of the Institute of International Statistics*, **33c**, 177-180.

Taylor, D. J. and Muller, K. E. (1995) Computing confidence bounds for power and sample size of the general linear univariate model, *American Statistician*, **49**, 43-47.

Taylor, D. J. and Muller, K. E. (1996) Bias in linear model power and sample size calculation due to estimating noncentrality, *Communications in Statistics: Theory and Methods*, **25**, 1595-1610.

Timm, N. H. (2001) *Applied Multivariate Analysis*. New York: Springer-Verlag.