PROGRAM DOCUMENTATION FOR SAS PLS PROGRAM I

General

This document describes the source code for the first program described in the article titled: "SAS® PLS for analysis of spectroscopic data" published in the *Journal of Near Infrared Spectroscopy* Volume 11, page 414–430 (2003), by James B. Reeves, III and Stephen R. Delwiche. Users are welcome to use or alter the program as they see fit, but are asked to reference the article in question to give proper credit to the developers of this program. As outlined in the article, this program is designed to read two data sets: the first data set consists of a GRAMS multifile (Galactic Industries, Inc, Salem, NH) and contains the spectral information only; the second file consists of an ASCII file in the form of the old GRAMS "CFL" file without the header information and contains the file identifications and analyte values. The data in the analyte file must be in the same order as the spectral files in the multifile. Proper sequencing of the data is the responsibility of the operator and no checking is done. The essential function of the program is to create a file containing pre-treated and original spectral data which can then be used as input data to SAS® PLS. The operator should check the article in question for information on the capabilities of SAS® PLS. The function of this document is only to document the source code only.

The basic structure of the program is to start with an input data set, and to sequentially add data that has been pre-treated in various fashions. As implemented, the newly created pre-treated data is added to the prior data set, and thus the new data set has both the original data and the new pre-treated data. An example is:

1. Original data consisting of X spectra = "WORKFILE"
2. WORKFILE saved as a temporary file.
3. Data in temporary file pre-treated to give a set of mean centered and a set of mean centered and variance scaled spectra.
4. Temporary file appended to original WORKFILE to give a new WORKFILE containing:
    a. Set of original spectra (n = X)
    b. Set of MC spectra (n = X)
    c. Set of MC and VS spectra (n = X).

Along with the file id's and analyte values.

5. Second pre-treatment is applied to new WORKFILE, i.e. 1st and 2nd derivatives, by saving original data and creating a new temporary file to which all derivatives are applied. If five 1st and five 2nd derivatives were used, the resulting temporary file would have:

a. Five sets of spectra with five different 1st derivatives applied to original spectra.

b. Five sets of spectra with five different 1st derivatives applied to MC spectra.

c. Five sets of spectra with five different 1st derivative applied to MC+ VS spectra.

d. Five sets each of spectra as above, but with 2nd derivatives used.

6. The temporary file from step 5 is appended to the WORKFILE as it existed prior to step 5 which contained 3 sets of X spectra (original, MC and MC+VS) resulting in a file containing 33 different sets of X spectra plus the file ids and analyte values.

7. By use of a by statement, PLS is ran using each of the 33 different sets of spectra.

Notes on SAS® arrays, control loops and macro language

This program makes extensive use of control statements (DO loops, IF then, etc.), arrays and the SAS® macro language. Some points to note are included below. SAS® operates on three levels within the program presented and alteration or addition of code to the program needs to recognize this, as the implementation of otherwise valid structures at the wrong place will result in program failure. Although it is not the objective of this documentation or the J. NIRS article to present an instruction manual for SAS®, since many people are familiar with SAS®, but few have probably used SAS® arrays, and even fewer the macro language, a few points of importance are included. Note: SAS® code is listed in bold, italics

Arrays and variables:

ARRAY X(10) VARW1-VARW10; This statement links an array X to variables named VARW1, VARW2, ... VARW10. Two points are: First, the VARW variables are not an array and can not be addressed as such in a DO loop such as:

```
DO Z = 1 TO 10;
    VARW(Z) = VARW(Z) * 2;
```

Second, the linkage of the array and the variable set in the first statement has the effect of transferring the values of the variables into the array elements, but more importantly the two are linked such that any change in the value of the array elements also changes the values of the variables linked to the array. In essence, the array and variables address the same memory location, but in different ways. For this reason, if the original values are needed, means must be undertaken to retain them. For example, if

VARW1-VARW10 had the values 1 to 10 respectively, the code:

```
DATA NEW; SET OLD;
ARRAY X(10) VARW1-VARW10;
DO Z = 1 TO 10;
   X(Z) = X(Z) * 2;
END;
```

would result in ***VARW1-VARW10*** having values of 2, 4, 6, 8, ... 20, respectively.

Another important point is the presence of two types of control statements in the program. While SAS® can be programmed much in the manner of a programming language such as C, it makes extensive use of modules called “PROCS”. However, SAS® does not allow the use of PROCS within control loops such as that presented above, i.e. the code below will result in an error.

```
DO Z = 1 TO 10;
   PROC PRINT;
END;
```

Also, the use of an array as a variable or even a list of variables within a PROC to control what is done with each iteration of a loop is illegal. To accomplish these two objectives, the SAS® Macro language must be used. The statement:

```
%LET CHEMS = ANALYTES1-ANALYTES5;
```

creates a macro variable ***CHEMS*** with five variables assigned to it. Either of the two can be used in code.

The statement:

```
%DO Z = 1 %TO 10;
   PROC PRINT;
      VAR ANALYTES&Z
%END;
```

creates a macro DO loop in which PROCS can be placed and macro variables can be treated as an array

as shown. For more information on the construction and use of SAS® macro variables and control statements the user should consult the texts listed at the end of this document. This brief introduction is presented only so that the user is not lead astray by alterations made to the program. SAS® and this program in essence operates on three different levels: the basic programming level using data statements and basic control statements, at the level of a “High Level” user friendly program using PROCS, and at the Macro language level which both adds flexibility to the basic programming level and also bridges the basic and high level areas. While very useful, improper mixing of the three levels will result in program failure.

Note on code documentation: The reader is assumed to know basic SAS®, and thus obvious code such as sorting of files is not always documented unless the objective or particular configuration used is not felt to be obvious.

SOURCE CODE DOCUMENTATION BY LINE

Line	**Function of Code**
8	Location and name of multifile defined
12	Location and name of analyte data file defined
13	Location and name of file to store PLS results: also printed to SAS output screen
15	Name of first data file given: Reduces computation time for determining the number of factors in PLS macro, otherwise program will iterate through each file (Value same for all files).
16	Defines the number of spectral data points in input files
17	Defines the number of files read
18	Defines the number of analytes in input file
20	Sets parameter for determining whether spectral data points are to be skipped, averaged or all used in computations.
21	Determines number of data points to skip or average, needs to be set to 1 if all data points to be used.
22	Defines the first spectral data point in a file to use, if set to > 1, spectral values at the start of the file are set to missing “.”.
23	Defines the last spectral data point in a file to use, if set to < MAXINDEX, then values at

the end of the file are set to missing ".".

Note: The values for STARTAT and ENDAT reference spectral data points and not wavelength values. Thus a file with values from 400 to 2498 nm with data collected every 2 nm has a total of (2498 - 400)/2 + 1 data points with the first being 1 and the last 1050 unless otherwise designated by the two parameters in question. To use only wavelengths from 1100 to 2498 nm, STARTAT would be set to 351 and ENDAT to 1050 assuming the multifile started at wavelength 400. The program as written does not include any references to the actual wavelengths of the input data.

25 Determines whether Savitsky-Golay derivatives are to be performed on the data.

26 Determines whether "Multiplicative Scatter Correction" is to be performed.

27 Determines whether Standard Normal Variate (SNV) and Detrend (DET) pre-treatments will be performed. If set to "YES", two sets of data are produced, one with SNV alone and one with SNV and DET.

28 Determines if DET without first performing SNV is to performed, code = DTO in output file. If set to "YES" along with "SNVDETYESNO", three data sets result: SNV only, SNV with DET and DET only.

29 Sets which option is to be used for mean centering (MC) and variance scaling (VS). If set to "PRO" MC is done using the MACRO included in the program. Note that under this condition, spectra are MC, and MC and VS prior to the PLS. If set to "SAS", MC and VS are done using the SAS® program options and may be done as is done using the MC Macro or with each iteration of the one-out cross validation (see J. NIRS article for further details on this option). Also, if set to "SAS", the pre-treatment is applied to all the spectral data, thus unlike the "PRO" option where a new set of MC data is added to the existing set, with the "SAS" version, there is no non-MC data available for the PLS.

31 Location of Tables for generating Savitsky-Golay derivatives

32-36 Parameters for determination of which Savitsky-Golay derivatives are generated, the options are described in Table 1 of the J. NIRS article. Basically, the Table settings determine the type of derivative and polynomial degree used, while the BEGINSG and ENDSG determine the number of points used which must be odd numbers.

39-45 Determine the gap values to used for generating 1st and 2nd gap derivatives and whether an even or odd number of data points is used as the gap.

49-59 These are Macro variables which allow passage of data within the program and do not change except as automatically defined by the number of analytes, spectra and spectral data points used as defined by other Macro variables.

MULTIFILE Macro

62-133 Macro for reading of GRAMS type multifile. Other means of data input can be used as long as a file containing only the spectra in sequential file order results, i.e. Spectra1 (Values1 to X), Spectra2 (Values1 to X),...SpectraN (Values1 to X). No other information, i.e. File identifier or wavelengths can be in the file unless other program code is changed.

66-68 Temporary data file and arrays for storage created.
69 Input file read as specified by Macro variable "MULTIFILE" (line 8).
70-72 Checks that input file is the proper type (NEW STYLE of GRAMS multifile)
Reads each spectrum from the input file.

Note this section of code was designed by Dr. Delwiche and is documented in the second program included in this discussion. Additional details on this section can be found there.

110-121 Constructs an ASCII data file from the GRAMS binary format, see the Galactic Web site at www.GALACTIC.COM for further information on the binary format used.

127 Determines number of decimal places for spectral data

128 Checks the number of spectra read with number of files expected (NUMFILES) and aborts program if different.

129 Checks the number of data points per spectra as input against expected (MAXINDEX) and aborts program if different.

130 Keeps only the absorbance values, eliminating extraneous variables

GETCHEM Macro

138-158 Macro for importing analyte data file, inputs file id and analytes as specified on line 52.

144-146 Checks that the number of samples read equals number expected (NUMFILES)

148-150 Creates the combined data file (WORKFILE) containing the sample ID, analyte values and spectra.

152-155 Removes the unneeded files containing the spectra and analyte data. Through out the program similar "DROP _ALL_" code is used to remove unneeded data files and to free memory space.

OVERHEAD Macro

161-196 "OVERHEAD" macro which adds additional variables to "WORKFILE" to record pre-treatments implemented and to set spectral data points not used to missing ".". In all four new variables are added and set to initial values (In bold):

- FORM = "**STR**" (No scatter correction), MSC (Multiplicative Scatter corrected), SNV (Standard Normal Variate correction), DET (SNV + Detrend).
- MCVS = "**ORG**" (Original data, but MC or MC + VS done by Macro within program), "**SAS**" (All spectra MC and VS using SAS® options, ("**NON**" (no MC or VS at all).
- DERIVAT = "**NON**" (Original non-derivatized data), "1ST" (1st derivative, gap type), "2ND" (2nd derivative, gap type), "SG1-SG10" (Savitsky-Golay derivatives using Tables 1 to 10 respectively).
- GAPS = "**0**" (No gap), If DERIVAT = "1ST or 2ND", then GAPS values correspond to number of data points used (*NOT nm, etc*.), if "SG1 to "SG10" then GAPS values of 1 to 11 correspond to data points in order 5, 7, 9, 11...25 used in computations.

186-188 Set unwanted data points at beginning of spectra to missing.

190-192 Set unwanted data points at end of spectra to missing.

194 Drops the variable I so that WORKFILE does not contain any variables not needed in future file manipulations, such variables can prevent merging or appending of files.

SAVITSKY-GOLAY Derivative Macro

199-337 Macro which creates Savitsky-Golay derivatives as defined by lines 25 and 31 to 36. Also runs the "DERIVATIVE" macro which follows in the program. As written, gap derivatives as described at lines 38 to 45 are always ran if the "GAPSGDERIVATIVE" macro is active, but Savitsky-Golay derivatives are optional.

201 Determines if running of Savitsky-Golay derivatives has been selected.

203	Creates copy of WORKFILE for manipulation. This is a common operation through out this program for both the WORKFILE and other files and will not be further documented.
204	Creates output data file (new WORKFILE) containing starting input data
205	Assigns “SGFILE” to data file “SAVGOL”
207-212	Inputs data from “SGFILE” needed on for Savitsky-Golay algorithm into matrix of 14 rows and 11 columns.
215-311	Creates loop to read Tables specified in lines 31 to 36, all Tables may be read, or Tables skipped by changing values for “TABLEBG, TABLEEND and TABLEBY” variables. Maximum value of TABLEEND is 10.
217-218	Creates a duplicate (“CHOSENTABLE”) of data file “SAVGOL” and deletes all data but that specified by loop control variable “TABLEINDEX” in line 215.
220-221	Transposes “CHOSENTABLE” and places results in file T_CHOSENTABLE
223-298	Performs actual derivatization using data spacing specified by variables “BEGINSG and ENDSG”. Note that “BY” must be 2 as implemented here.
224-227	Convolutes the data point spacing chosen with the values in “T_CHOSENTABLE”, the reader should check the references provided in the J NIRS paper for information on how this creates the Savitsky-Golay derivatives.
229-231	Sets the value of sign as required for the derivative in question.
233-234	Selects which of the Table’s eleven convolution windows is utilized.
236-239	Creates a combined data set containing the input spectral data and the information in the “CONVOLUTE” data set, this is then used in lines 242 to 274 to create the actual derivatized spectra.
242	A new data set (FINSAVITSKY) is created to store each derivative. variation as created.
243-260	Initialization of variables
262-274	Derivatization performed as specified.
280-289	Sets value for variable “IDERIVAT” according to TABLE used.
290	Assigns data point value used to IGAPS as specified previously.
291-292	Assigns proper values to first and last point with values in derivatized spectra.
294-297	Eliminates invalid derivatives created using specified “TABLE” (2, 5, or 7) and five data points) by keeping all other combinations. Also, removes unwanted variables from file by keeping only those specified.
301-304	Removes unwanted variables when “TABLE” equals values other than 2, 5, or 7).
306-309	Appends data file (OUTSAVITSKY) containing newly derivatized data to file containing original data and derivatives formed in previous loops of “TABLEINDEX”.
313-315	Removes original data from final data set (“SAVITSKYDERIVATIVES”. Note the original data must be present in order to be able to have a file to append the first file containing the derivatives to. Failure to remover the original data will result in duplicate spectra when the gap derivative macro “DERIVATIVES”(lines 331-531 is ran). Also, as implemented, the “DERIVATIVES” macro can be ran without running the SGDERIVATIVES, but not the reverse.
335	Runs gap derivative macro.

DERIVATIVES macro for creating gap derivatives

First set of line numbers correspond to 1st derivative, 2nd in Italics perform same function for creation of 2nd derivatives.

340-532 Macro for creating 1st and 2nd degree gap derivatives.
342-421 *423-503* Code for creating 1st and 2nd derivatives, respectively

343-348 *424-429* Initialization of variables
350-352 *431-433* Sets value of variable "DERIVAT" to 1st for each new data set to be created (Each form of derivative).
354-357 *435-438* As above for variables "IFORM and "IMCVS".
359-364 *440-445* Variable initialization and creation of temporary arrays needed for processing of data.
366-370 *447-451* Loads input data into 2-D array [DER]: Number of 1st derivatives (I) by number of spectral data points (J).

374-404 *458-485* Creates derivatives specified, as written gaps are incremented by a factor of two (i.e., 1, 2, 4, ...) with each iteration of the "DO UNTIL" loop until a maximum gap value of "ENDGAP1ST" is reached. With odd gaps the values used are 1 + two times the previous gap -1 (i.e., 1, 3, 5...).

377-382 *458-463* Determines gap value to use in iteration based on last gap.
384 *465* Stores gap value for later output
386-388 *467-469* Creation of 1st derivative spectra
390-396 *471-477* Sets values at beginning and end of spectra eliminated by derivative process to missing.
398-399 *479-480* Records values of first and last valid spectral data points for each derivative created.
401-403 *482-484* Moves derivatized data to array "DER" which stores all the derivatized data.
407-412 *488-493* Moves values of variables listed from storage array into permanent variables for output to data file "FIRST".
414-416 *495-497* Moves specific derivative values into "TEMPABS" which also assigns them to variables "ABS1 to ABS&MAXINDEX".
418-419 *499-500* Eliminates unneeded variables and stores those listed into data file "FIRST"

505-506 Append data files containing the 1st and 2nd derivatives to the original input file.
508-510 If Savitsky-Golay derivatives were previously created, then the data file containing that derivatized data is appending to the data file created by the "DERIVATIVES" macro as outlined above.

514-523 Sets data points eliminated by derivative processes to zero, otherwise original spectral values still occupy those file locations.

SKIPAVER macro

535-613 Macro for averaging or skipping n data points.

538-539 Initialize storage arrays for input and altered data
541-551 Code for skipping spectral data points
542 Determines number of data points in new spectrum
543 First data point set as first point in input file
545-547 Stores data points to be kept for remaining data points
549 Determines position of last data point in new spectrum
550 Sets position of first data point in new spectrum

553-568 Code for averaging spectral data points
554 Determines number of data points in new spectrum
555 Initializes "Counter" variable used to determine position of data points being averaged
557 Determines position of first data point at end of spectrum which was not averaged due to insufficient data points remaining to meet averaging criteria, i.e., 4 points, etc.

558-559	Initialize storage array “ASDATA” to zeros.
561-564	Every n data points summed
566	Sums of data points divided by n to create averages
567	Sets value of “IFIRST” variable to 1
570-580	If “REMAINDER” is > 0, then the last data points are averaged by this code to create the last new data point in the spectral file.
582	Sets value of “ILAST” to number of averaged data points
585-589	Sets data points eliminated by derivative processes to zero, otherwise original spectral values still occupy those file locations.
594-603	Sets empty data points to zero.
605	Resets value of “MAXINDEX” to data points remaining after averaging or skipping of data points

MSCSNV (Scatter correction) macro

616-882	Macro for scatter correction procedures
616-679	Code for multiplicative scatter correction
682-722	Code for standard normal variate correction
725-771	Code for detrend after SNV application
773-819	Code for detrend without running SNV first

Multiplicative scatter correction

618-619	Variables for storing mean values
621	Setting determines if MSC to be carried out
623	Data file “FORMSC” created for computation of scatter corrected values
626-631	Mean values computed and stored in file “MNSPEC” for each data pre-treatment
633-637	File containing mean values for all variables transposed and mean values for peaks output in file “AVERSPEC” which is then sorted by pre-treatment.
640-644	Above operation performed on all spectral data (non-mean spectra)
646-649	The computed mean spectra and the average spectrum are merged together into file “MORGSPEC” and again sorted.
651-653	For each pre-treatment, each spectrum is regressed against the average spectrum to determine the “SLOPE” and “INTERCEPT” values needed to adjust the values for each spectrum to the mean of all the spectra.
656-661	Removal of unneeded data and renaming of variables containing the slope and intercept values for each spectrum along with sorting.
663-667	Data sets merged for final computations and sorted
669-674	Scatter correction performed using computed slope and intercept values.
676	Elimination of unneeded variables - All corrected data in file “FINALMSC”.
682	Setting determines if SNV or SNV+DET to be performed
684-687	Creation of working data file, storage arrays and initialization of “MNSPEC” variable
689-691	Sums all the spectral values for a spectrum and computes the average.
695-697	Scales all spectra by subtracting the average spectral values from each spectral data value.
699-701	Computes squared spectral values from the scaled spectrum
703	“MNSUM” initialized
705-709	Computes the variance of each spectra by computing mean of the squared spectra as done above for the non-squared spectra and then computes the square root of that value divided by the number of wavelengths - 1.
711-713	Re-scales each mean scaled spectrum by dividing each spectral data point by the variance of the spectrum.
715-720	Storage of SNV corrected data, etc.

725	Setting determines if DET is to be applied to SNV corrected data
727-732	Loads SNV corrected data into a workfile, sets FORM variable and sorts data set
734-736	Creation of Transposed file (“TRANSSNV”) from SNV corrected data file.
738-742	The wavelength values produced by the transpose program vary from 1 to (Number of wavelengths per spectra)×(Number of Spectra). This section changes those values to sets of 1 to (Number of wavelengths per spectra). N = value for data point position as defined above. X is the number of the spectra in question. I is the new wavelength number and ranges from 1 to the number of wavelengths per spectra.
744-746	Computation of polynomial equation coefficients for each spectra by regressing each spectra against wavelength positions. Computes a 2^{nd} degree polynomial equation $Y = a\times X + b\times X^2$ where X is the data point number (i.e., 1 ,2, 3, 4), and Y is the absorbance or spectral value at point X.
748-754	Renaming of variables and elimination of unneeded variables prior to merging of data set containing the regression coefficients with the spectral data to be detrended.
756-760	Initialization of file to contain detrended spectra
762-765	Detrend algorithm, subtracts polynomial computed spectra from data spectra at each point.
767-769	Elimination of unneeded variables from detrend file and sorting.
773-819	Applies DET to original, non-SNV pre-treated data (line 775), code is otherwise identical to the above code for application to SNV corrected data.
822-839	Creates a combined data file containing the original pre-scatter corrected data and scatter corrected data based on the switch settings. Resulting file may in addition to the original data contain various combinations of the following corrections: MSC, SNV, SNV+DET (DET) and DET without SNV (DTO). Application of SNV or DET to MSC corrected data is not implemented.

MEAN CENTERING AND VARIANCE SCALING

882-970	Macro code for MC and MC with VS
884-889	New storage arrays, etc.
891-892	Creation and sorting of data file for computations
894-900	Computation and storage in file “MCVARSCALED” of means and SD values for all data variables for each set of pre-treated spectra followed by sorting of output file.
902-904	Code for merging of “MCVARSCALED” file with original data file (“MCVAR”).
906-910	File created (“MEANCENTERED”) and arrays initialized for mean centering.
912-914	Spectra mean centered
916-918	Analyte values mean centered
920	Setting of IMCVS variable
922-928	File created (“VARSCALED”) and arrays initialized for variance scaling of mean centered data.
930-932	Spectra data variance scaled after mean centering.
934-936	Analyte data variance scaled after mean centering.
938	Setting of IMCVS variable
939-940	Data saved and unneeded variables eliminated for variance scaled data.
942-944	Variance data saved for re-scaling of data and also for computation of number of factors using the GRAMS F-test procedure, this data is needed for the F-test computation even if MC is done using the SAS® program option as opposed to this macro.

946 Setting determines if MC is done by this program (Is the data kept).

948-950 Data saved and unneeded variables eliminated for MC and VS scaled data.

953-954 Appends files containing MC and VS data to WORKFILE.

PREPARE FOR PLS Macro

973-998 Because SAS® eliminates all observation with missing values from the PLS calibration, all missing values are set to zero. File is also sorted by order desired for PLS.

PLS Macro

990-1198 Macro for execution of PLS.

992 Macro DO loop control based on number of analytes.

993-995 Opens a file "PRESS_RESULTS" for storage and retention of various variables creating internally by SAS® during the execution of the PLS PROC.

997-1019 Determines how to execute the PLS PROC based on the type of MC and VS scaling desired: PRO = by Macro previously discusses; SAS = using options in PLS PROC; or none at all. Note that under the "SAS" option there are other settings which can be used, i.e. MC alone or VS alone, or done as in the Macro in this program (all data pre-treatment prior to PLS as opposed to being before with each one-out cross validation). Further information can be found in the J NIRS article.

998-1002
1006-1010
1014-1018 PLS performed for each set of pre-treated spectra (BY STATEMENT). According to method of MC selected, otherwise process is the same.

1001
1009
1017 Data output includes the predicted values for the analytes and XSCOREs as variables T1 to Tn (see accompanying J NIRS article). Only difference in three sections is how MC is performed.

1027-1030 Computes the correlation between the predicted and actual analyte values based on the PLS results for each set of spectra. Note the correlation value is labeled with the name of the analyte in question ("CHEMANALYTESX"). This label is used to label the output column containing the final r-square for the model.

1032-1036 Computes r-square for models and removes unneeded variables from the resulting file.

Note: PLS creates a new variable for each allowable factor called an XSCORE assigned the names T1 to T15 if a maximum of 15 factors allowed (Default setting, see accompanying article for details). For the final model, all factors included have a value and those not are set to missing "." The next section of code uses this information to determine the number of factors in the model.

1038-1046 Determine number of factors in optimal model. Determines the number of factors, uses only the data for the first sample since the it is the same for all samples in a data set.

1038-1042 Data set creation, variable initialization.

1044-1048 Determination of number of factors in model

1049 Removes unneeded data from file

1063-1066 Writes the results in "SASWAY" to an output file designed by the macro variable "OUTPUTFILE" set at the start of the program.

Note: All results are output in the order of the goodness of fit of the model, i.e., from lowest r-square to the highest.

1082-1190	Code for selecting model based on F-test used in GRAMS software using data stored in file “PRESS_RESULTS” created during PLS execution
1086-1090	Compute the mean press value for the model “ROOTMEANPRESS” and output in file “GETTHEMIN”
1092-1094	Computes the “MINPRESSVALUE” from the “ROOTMEANPRESS” and places it in a new file.
1100-1106	Computes the PRESS values for each of the possible models for 1 to maximum number of factors and deletes the model if the number of factors was zero.
1112-1114	File “GRAMSLIKETABLE” created.
1115	F-ratio computed and added to “GRAMSLIKETABLE”.
1116	F-test performed on F-ratios.
1117-1118	Determines all acceptable models based on F-test results and stores the number of factors for each such model in variable “MINREC”. Value of 0.75 is that suggested and used by GRAMS, other values can be substituted to obtain models with either fewer or more factors on average.
1126-1127	Sets MINREC to missing if number of factors selected was zero.
1132-1136	Determines the model with the minimum number of factors (lowest value for MINREC) by using the MEAN PROC and stores the results in the file “THEMINREC”.
1142-1144	Files “GRAMSLIKETABLE” and “THEMINREC” merged to give file “GRAMSTABLE”.
1145-1148	F-test and ratio results set to missing for models with factors > best model based on these tests.
1161-1166	Corrects variance scaled PRESSVALUE when variance scaling pre-treatment used so that all PRESSVALUEs are computed the same and can be compared across pre-treatments (Also, used when SAS VS used).
1177-1184	Unwanted data eliminated, file sorted and output in order of high to low PRESSVALUE models.
1186-1190	Results of model selection using F-test output appended to previous output file.
1201-1202	Diverts log output to designated file, otherwise program execution can be halted when log window becomes filled during execution of program using large numbers of spectral pre-treatments.
1204-end	Order of execution and which macros to execute.

SOURCES FOR INFORMATION ON USE OF SAS® MACRO LANGUAGE AND OUTPUT DELIVERY SYSTEM (ODS).

1. M. M. Burlew. SAS® Macro Programming Made Easy. SAS® Institute Inc., Cary, NC (2001).

2. A. Carpenter. Carpenter’s Complete Guide to the SAS® Macro Language., SAS® Institute Inc., Cary, NC (2001). Cary, NC (1998).

3. L. E. Haworth. Output Delivery System: The Basics. SAS® Institute Inc., Cary, NC (2001).

Also, see *Journal of Near Infrared Spectroscoscopy* article for references on various pre-treatments implemented here [J.B. Reeves, III and S.R. Delwiche, *J. Near Infrared Spectrosc.* **11,** 414–430 (2003)].

.