Paper 245

Using Macros to Construct Overlaid Contour Plots For Response Surface Optimization

David J. Meade, Scott Lacey
Advanced Micro Devices, 5204 E. Ben White Blvd., MS 608, Austin, TX 78741

ABSTRACT

Contour plots are frequently used to visually illustrate the relationship between two experimental factors and one response variable. However, in practice researchers often need to know how experimental factors affect many responses simultaneously. By using SAS® macros in conjunction with PROC G3GRID, PROC GCONTOUR, PROC IML, and PROC SQL contour plots can be overlaid one on top of another. The overlaid contour plot can then be used to determine how a set of responses is affected by other experimental factors. Our set of SAS macros will allow the user to fit unique models for each response, construct separate contour plots for each model, and then overlay those plots. SAS/AF®, and SCL are also used to provide a friendly graphical user interface. This paper is intended for SAS users who are familiar with linear modeling techniques.

INTRODUCTION

When analyzing data researchers frequently need to find a set of operating conditions that will optimize several responses simultaneously. After identifying an optimal set of conditions, the next challenge is to visually display the results in a manner that managers or other researchers will understand the relationships that exists between experimental factors and the responses. A common practice is to construct two-dimensional contour plots for each response.

For example, consider a process where three experimental factors are manipulated in order to determine an optimal set of conditions under which the process will be run in the future. These three factors are Time, Temperature, and Power. The researcher fits the regression models shown below.

$$\text{DELTA\_A} = \beta_0 + \beta_1 TIME + \beta_2 TEMP + \beta_{11} TIME^2 + \beta_{22} TEMP^2$$
$$\text{DELTA\_B} = \beta_0 + \beta_1 TIME + \beta_2 TEMP + \beta_{12} TIME \times TEMP + \beta_3 POWER + \beta_{23} TEMP \times POWER + \beta_{33} POWER^2$$
$$\text{DELTA\_C} = \beta_0 + \beta_1 TIME + \beta_2 TEMP$$

PROC G3GRID and PROC GCONTOUR can then be used to construct individual contour plots for each response, see Figures 1-3.

Figure 1: DELTA_A vs. Time,Temp

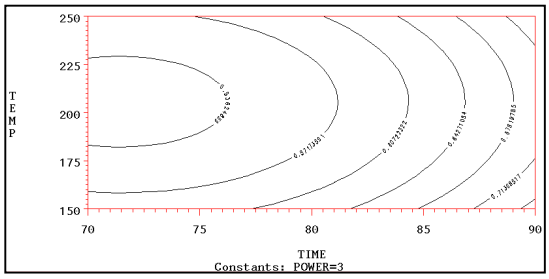


Figure 2: DELTA_B vs. Time,Temp

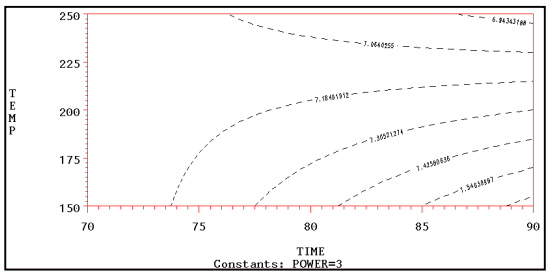
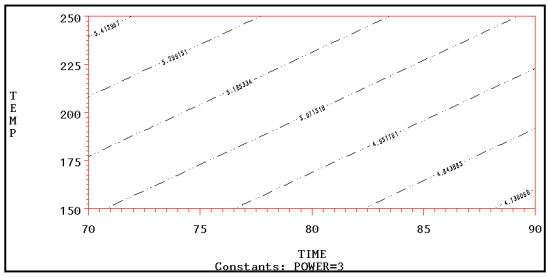
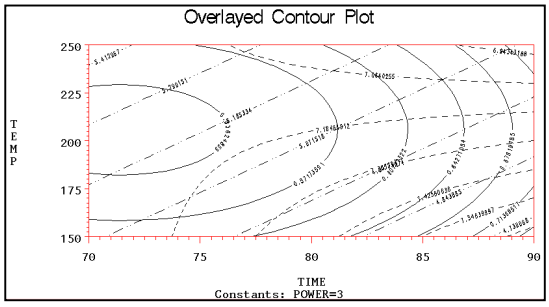


Figure 3: DELTA_C vs. Time,Temp



These plots clearly illustrate how Time and Temperature affect each of the responses individually. However, it is still difficult to visualize the effect of Time or Temperature upon all three of the response variables simultaneously. An approach that is commonly used in such cases is to overlay the individual contour plots so that all of response surfaces are illustrated on the same graph. The overlaid contour plot for the example just shown is illustrated in Figure 4.

Figure 4: Overlaid Contour Plot of DELTA_A,DELTA_B, and DELTA_C



As stated previously, the advantage of overlaying the contour plots is that you can visualize what is happening to all of the responses simultaneously for any given set of conditions. In the remaining pages, we will present a set of macros, which allow the user to overlay up to five contour plots at the same time. The macro itself uses a variety of SAS procedures including PROC IML, PROC G3GRID, PROC

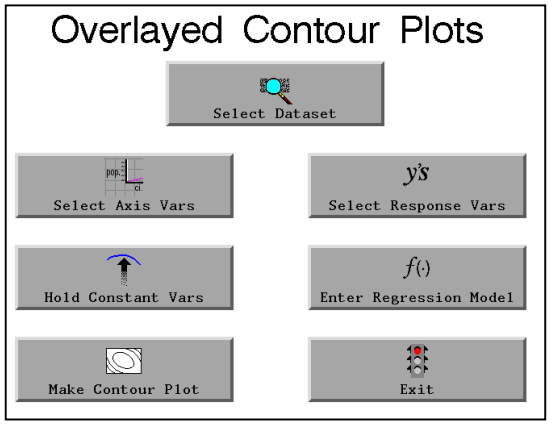
GCONTOUR, PROC GREPLAY, and PROC SQL. A graphical user interface using SAS/AF® and SCL is also included.

SAS MACRO FOR OVERLAID CONTOUR PLOTS

Constructing overlaid boxplots is a four-step process. First, the user must identify the dataset to be used as well as select which variables will be used as the X and Y axis. Second, the user provides information regarding which, if any, variables must be held constant while those used for the X and Y axis are manipulated. Third, the user selects up to five responses to model. Fourth, the user specifies which model is to be used for each of the selected responses. The main menu is set up so that the user can easily step through each of the necessary steps.

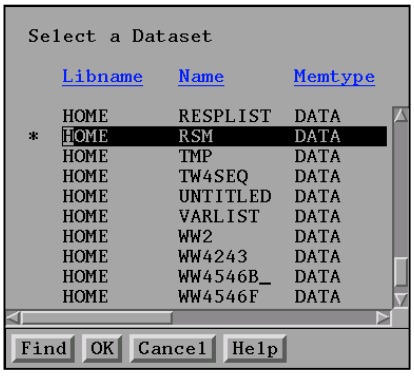
The Main Menu: Figure 5 illustrates the main menu system. When a button is clicked the appropriate piece of SCL code from Appendix 1 is exectuted.

Figure 5: Main Menu



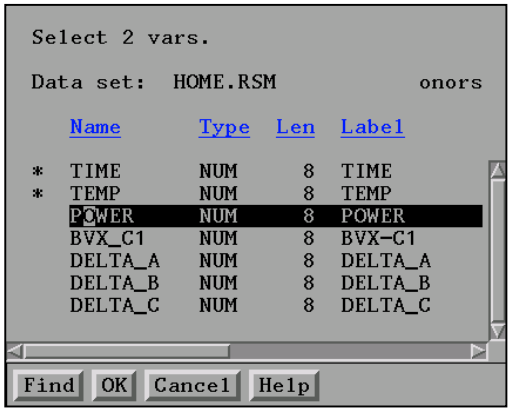
Button 1(Select Data Set): When the user clicks the first button they are taken to the window illustrated in Figure 6. Here they can select the library and data set to be used. In this example we have selected the "RSM" data set located in the "HOME" library.

Figure 6: Select a Data Set



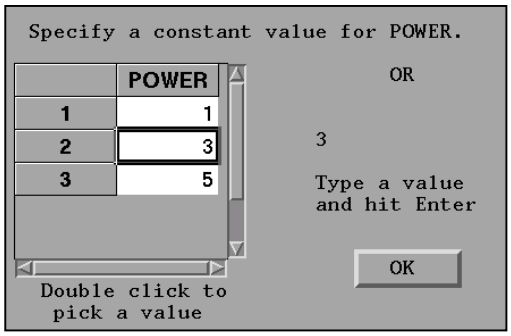
Button 2(Select Axis Vars): When the user clicks the button labeled "Select Axis Vars" they are taken to the window illustrated in Figure 8. Here they select the variables to be plotted on the X and Y axis.

Figure 7: Select Axis Variables



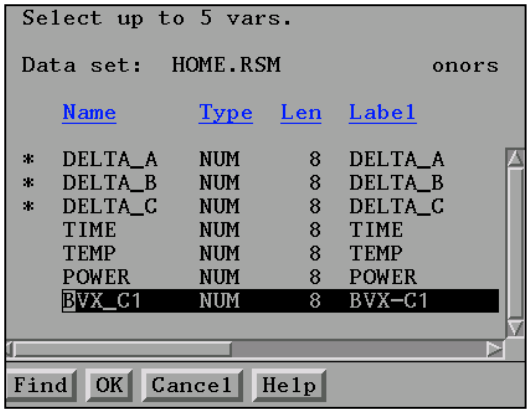
Button 3 (Hold Constant Vars): The next step is to identify any variables that will be held constant while those selected in step 1 are manipulated. In this example, Power is being held constant at 3. See illustration in Figure 8.

Figure 8: Specify Constant Value



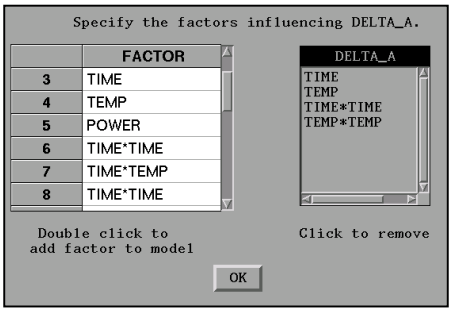
Button 4(Select Response Vars): Next, we select the response variables to be used in the analysis. In this example, DELTA_A, DELTA_B, AND DELTA_C have been selected as the response variables of interest, see Figure9.

Figure 9: Select Response Variables



Button 5(Enter Regression Model): The next step is to define a model for each response. This set of models will be used to generate the contour plots once the main program is executed. Simply select the terms you want to include in each model. Figure 10 illustrates how a model is defined for the response DELTA_A. Models for the other response variables are fit in a similar manner.

Figure 10: Define model for DELTA_A



Button 6(Make Contour Plot): The final step is to execute the code that generates the overlaid contour plot. Simply click the "Make Contour Plot" button. The final graph is shown in Figure 12.

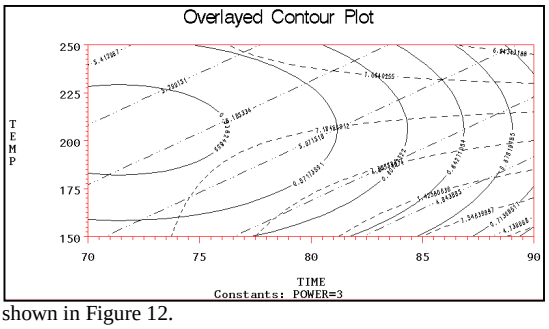


Figure 12: Overlaid Contour Plot

CONCLUSION

Contour plots are frequently used to visually illustrate the relationship between two experimental factors and one response variable. However, in practice researchers often need to know how experimental factors affect many responses simultaneously. By using SAS macros in conjunction with PROC REG, PROC GCONTOUR, PROC IML and PROC SQL contour plots can be overlaid one on top of another. The overlaid contour plot can then be used to determine how a set of responses is affected by other experimental factors.

APPENDIX 1 (Code for variable input and graph construction)

```
length command templib tempdata libname dataname
tmpname1 tmpname2 $8 string $17 footstr $200;
length deplid indeplid constlid cvallid templid i j
k q n p xct rc 8;
init:
control allcmds;
deplid=makelist();
indeplid=makelist(0,'global');
constlid=makelist();
cvallid=makelist();
reglid=makelist();
templid=makelist(); rc=curlist(templid);
return;
main:
command=lastcmd(); call nextcmd();
if command>'' then put 'Last command is ' command;
select(command);
when('PICKDATA') link pickdata;
when('INDEPVAR') link indepvar;
when('DEPVARS') link depvars;
when('CONSTVAR') link constvar;
when('REGMODEL') link regmodel;
when('CONTPLOT') link contplot;
when('EXIT') _status_='H';
otherwise;
end;
return;
term:
return;
pickdata:
/* need to detect a new dataset in order to clear
other settings, also */
/* save name in case user wants to cancel and keep
current setting     */
templib=libname; tempdata=dataname;
libname=liblist('*','*','Select a Library','Y');
if libname='' then do;
if tempdata>'' then libname=tempdata;
else _msg_='A library must be selected.';
return;
end;
dataname=dirlist(libname,'DATA VIEW',1,'N','*','*',
  'Select a Dataset','Y');
if dataname='' then do;
if libname=templib and tempdata>'' then
dataname=tempdata;
else _msg_='A dataset must be selected.';
end;
/* clear settings when new dataset picked */
if libname^=templib or dataname^=tempdata then do;
rc=clearlist(indeplid); rc=clearlist(deplid);
rc=clearlist(constlid); rc=clearlist(cvallid);
rc=clearlist(reglid,'Y');
end;
return;
indepvar:
if libname='' or dataname='' then do;
_msg_='Please select a dataset first.';
return;
end;
dsid=open(libname||'.'||dataname,'I'); tempdata='';
tempdata=varlist(dsid,'N',2,'Select 2
vars.','Y','Y');
rc=close(dsid);
if tempdata>'' then
indeplid=copylist(templid,'N',indeplid);
return;
depvars:
if libname='' or dataname='' then do;
_msg_='Please select a dataset first.';
return;
end;
dsid=open(libname||'.'||dataname,'I'); tempdata='';
tempdata=varlist(dsid,'N',5,'Select up to 5
vars.','Y','Y');
rc=close(dsid);
if tempdata>'' then
deplid=copylist(templid,'N',deplid);
return;
constvar:
if libname='' or dataname='' then do;
_msg_='Please select a dataset first.';
return;
end;
dsid=open(libname||'.'||dataname,'I');
tempdata='';
tempdata=varlist(dsid,'N',99,'Select vars to hold
constant.','Y','Y');
rc=close(dsid);
if tempdata='' then return;
constlid=copylist(templid,'N',constlid);
rc=clearlist(cvallid);
j=listlen(constlid);
do i=1 to j;
tempdata=getitemc(constlid,i);
call
display('constval.frame',libname,dataname,tempdata,
k);
if k=. then do;
_msg_='Constant settings are not complete.';
rc=clearlist(cvallid);
return;
end;
cvallid=insertn(cvallid,k,-1);
end;
return;
regmodel:
j=listlen(deplid);
```

```
if j=0 then do; _msg_='Please select reponse vars
first.';
return;
end;
do i=1 to j;
tempdata=getitemc(deplid,i);
k=listlen(reglid);
if i>k then do;
k=makelist();
rc=insertl(reglid,k,-1);
end;
else k=getiteml(reglid,i);
put 'regmodel before call: ' indeplid=;
call putlist(indeplid,'regmodel again',3);
call
display('regmodel.frame',libname,dataname,tempdata
   ,indeplid,constlid,k);
put 'coutour:regmodel after call';
call putlist(k,'factor list',3);
end;
return;
contplot:
/*need list to translate factor names into new
names */
rc=clearlist(templid);
/* submit macro variable assignments */
string=compress(libname||'.'||dataname);
submit; %let dataset=&string; endsubmit;
/* independent vars */
xct=listlen(indeplid);
if xct^=2 then do;
_msg_='Please pick 2 independent vars.';
return;
end;
do i=1 to 2;
tempdata=getitemc(indeplid,i);
tmpname1=compress('x'||put(i,1.0));
tmpname2=trim(tmpname1)||'hold';
rc=insertc(templid,tempdata,-1,tmpname1);
submit;
%let &tmpname1=&tempdata;
%let
&tmpname2=0;
endsubmit;
if i=1 then submit;
%let xaxis=&tempdata;
endsubmit;
else submit;
%let yaxis=&tempdata;
endsubmit;
end;
/* constant vars */
j=listlen(constlid); footstr='';
do i=1 to j;
xct=xct+1;
tempdata=getitemc(constlid,i);
k=getitemn(cvallid,i);
if i=1 then footstr='Constants:';
footstr=trim(footstr)||' '||trim(tempdata)||'='
  ||compress(put(k,best10.));
tmpname1=compress('x'||put(xct,5.0));
tmpname2=trim(tmpname1)||'hold';
rc=insertc(templid,tempdata,-1,tmpname1);
submit;
%let &tmpname1=&tempdata;
%let &tmpname2=&k;
endsubmit;
end;
submit;
%let numfact=&xct;
%let footn=&footstr;
endsubmit;
/* fill out any remaining factors to 9 */
j=xct+1;
do i=j to 9;
tmpname1=compress('x'||put(i,5.0));
submit;
%let &tmpname1=xxx;
endsubmit;
end;
/* response vars */
j=listlen(deplid);
submit;
%let numresp=&j;
endsubmit;
do i=1 to j;
tempdata=getitemc(deplid,i);
tmpname1=compress('r'||put(i,5.0));
tmpname2=trim(tmpname1)||'modl';
submit;
%let &tmpname1=&tempdata;
%let &tmpname2= endsubmit;
k=getiteml(reglid,i);
q=listlen(k);
do n=1 to q;
string=getitemc(k,n);
do p=1 to xct;
tmpname1=getitemc(templid,p);
tmpname2=nameitem(templid,p);
string=tranwrd(string,tmpname1,tmpname2);
end;
string=compress(string,'*');
submit;
&string  endsubmit;
end;
submit;
endsubmit;
end;
j=j+1;
do i=j to 5;
tmpname1=compress('r'||put(i,5.0));
submit; %let &tmpname1=rrr; endsubmit;
end;
*j=preview ('browse');
link plotcode;
return;
plotcode:
submit continue;
/* Insert following code */
endsubmit;
return;
/* End of SCL Code */
/* This is macro code which can be used as plain
SAS program */
/* If running stand alone then use following macro
vars:    %let x1=xxxx;   %let x1hold=0;
         %let x2=xxxx;   %let x2hold=0;
         %let x3=xxxx;   %let x3hold=0;
         %let x4=xxxx;   %let x4hold=0;
         %let x5=xxxx;   %let x5hold=0;

         %let xaxis=xxxx;
         %let yaxis=xxxx;

         %let r1=rrr;
         %let r2=rrr;
         %let r3=rrr;
         %let r4=rrr;
         %let r5=rrr; */
%let titl =Overlayed Contour Plot;
/* set values to min and max for now
%let xlow = 60;     %let xhi = 100; */
proc summary data=&dataset noprint min max;
var &xaxis;
output out=work._summary(keep=min max) min=min
max=max; run;
data _null_; set work._summary;
call symput('xlow',compress(put(min,best10.)));
call symput('xhi',compress(put(max,best10.)));
run;
/* set values to min and max for now
%let ylow = 100;     %let yhi = 260; */
proc summary data=&dataset noprint min max;
var &yaxis;
output out=work._summary(keep=min max) min=min
max=max; run;
data _null_; set work._summary;
call symput('ylow',compress(put(min,best10.)));
call symput('yhi',compress(put(max,best10.)));
run;
%let x= x;
%let hold = hold;
%let under = _;
%let r=r;
%let m=modl;
data dset; set &dataset;
run;
%macro deldot;
%do mi=1 %to &numfact;
data dset; set work.dset;
if &&x&mi<>.;
run;
%end;
%mend deldot;
%deldot;
%macro deldot2;
%do mi=1 %to &numresp;
data dset; set work.dset;
if &&r&mi<>.;
run;
%end;
%mend deldot2;
%deldot2;
%macro makecol;
data work.temp;
set work.dset;
run;
data work.temp; set work.temp;
%do ni=1 %to &numfact;
%do mi=&ni %to &numfact;
&x&ni&x&mi=&&x&ni*&&x&mi;
&x&ni=&&x&ni;
%end;
%end;
run;
%mend makecol;
%makecol;
data work.tempval;
set work.dset end=eof;
output;
```

```
if eof then do;
&r1=.; &r2=.; &r3=.; &r4=.; &r5=.;
do &xaxis=&xlow to &xhi by ((&xhi-&xlow)/30);
do &yaxis=&ylow to &yhi by ((&yhi-&ylow)/30);
output;
end;
end;
end;
run;
%macro makecl2a;
data work.tempval; set work.tempval;
%do ni=1 %to &numfact;
%do mi=&ni %to &numfact;
&x&ni&x&mi=&&x&ni*&&x&mi;
&x&ni=&&x&ni;
_&x&ni=&&x&ni&hold;
%end;
%end;
run;
%mend makecl2a;
%makecl2a;
%macro makecol2;
data work.tempval; set work.tempval;
%do mi=1 %to &numfact;
if &&x&mi&hold<>0 then &x&mi=&&x&mi&hold;
%end;
data work.tempval; set work.tempval;
%do ni=1 %to &numfact;
%do mi=&ni %to &numfact;
&x&ni&x&mi=&x&ni*&x&mi;
%end;
%end;
run;
%mend makecol2;
%makecol2;
%let y = y;
%let outfil = outpred;
%macro calcpred;
%do mi = 1 %to &numresp;
proc iml;
use work.temp;
read all var{&&r&mi&m} into x;
read all var{&&r&mi} into y;
n=nrow(x);
x=j(n,1)||x;
b=inv(x'*x)*(x'*y);
print y b;
use work.tempval;
read all var{&&r&mi&m} into x2;
n1=nrow(x2);
x2=j(n1,1)||x2;
&y&mi=x2*b;
create &outfil&mi var{&y&mi};
append;
run;
%end;
%mend calcpred;
%calcpred;
data tempval; set work.tempval;
order=ceil(_n_/1);
run;
data final; set work.tempval;
run;
%macro mrg;
%do mi=1 %to &numresp;
data &outfil&mi; set work.&outfil&mi;
order=ceil(_n_/1);
run;
data final;
merge final &outfil&mi;
by order;
run;
data final; set work.final;
&&r&mi=&y&mi;
run;
%end;
%mend mrg;
%mrg;
goptions reset = all;
goptions nodisplay cback=white device=xcolor
rotate=landscape
csymbol=black gsfname=graphout
gsfmode=append htext=1 htitle=1.5;
%let ofset1 = -28;
%let ofset2 = -10;
%let ofset3 = 8;
%let ofset4 = 10;
%let ofset5 = 28;
%let ofset=ofset;
%let line1 = 1;
%let line2 = 20;
%let line3 = 42;
%let line4 = 8;
%let line5 = 33;
%let l = line;
%let color1 = blue;
%let color2 = red;
%let color3 = green;
%let color4 = black;
%let color5= magenta;
%let color = color;
%macro makecont;
%let delflag=n;
data _null_;
set sashelp.vscatlg(where=(libname='WORK' and
   memname='GSEG'));
call symput('delflag','Y'); run;
%if "&delflag"="Y" %then %do;
proc catalog c=work.gseg kill;
run;
quit;
%end;
%do mi = 1 %to &numresp;
legend across = 1
position=(bottom outside center)
offset=(&&&ofset&mi,0)
shape=line(2) value=(height=.5)
label=("&&r&mi");
title "&titl";
footnote "&footn";
symbol1 h=2 pct line = &&&l&mi color=&&&color&mi;
symbol2 h=2 pct line = &&&l&mi color =&&&color&mi;
symbol3 h=2 pct line = &&&l&mi color=&&&color&mi;
symbol4 h=2 pct line = &&&l&mi color=&&&color&mi;
symbol5 h=2 pct line = &&&l&mi color =&&&color&mi;
symbol6 h=2 pct line = &&&l&mi color =&&&color&mi;
symbol7 h=2 pct line = &&&l&mi color =&&&color&mi;
symbol8 h=2 pct line = &&&l&mi color=&&&color&mi;
symbol9 h=2 pct line = &&&l&mi color=&&&color&mi;
symbol10 h=2 pct line = &&&l&mi color=&&&color&mi;
symbol11 h=2 pct line = &&&l&mi color=&&&color&mi;
proc g3grid data=work.final out=work.agrid;
grid &yaxis*&xaxis=&&&r&mi/ naxis1=30 naxis2 =30 ;
run;
proc gcontour data=work.agrid;
plot &yaxis*&xaxis=&&&r&mi/
nolegend autolabel=(tolangle=30 check=20);
run;
quit;
%end;
%mend makecont;
%makecont;
goptions display;
proc greplay nofs igout=work.gseg tc=sashelp.templt
template=whole;
treplay 1:1 1:2 1:3 1:4 1:5;
run;
quit;
```

APPENDIX 2 (Code for input of variables held constant)

```
entry libname dataname sasname $8 k 8;
init:
control always;
call notify('TEXTLAB','_set_text_',
  'Specify a constant value for
'||trim(sasname)||'.');
submit continue;
proc sort data=&libname.&dataname(keep=&sasname)
nodupkey
out=work._const_; by &sasname; run;
endsubmit;
call
notify('DATATAB','_set_dataset_','WORK._CONST_');
return;
main:
return;
term:
return;
datatab:
call
notify('DATATAB','_get_column_value_',sasname,k);
call notify('TEXTENT','_set_value_',k);
put 'Datatab: ' k=;
return;
textent:
call notify('TEXTENT','_get_value_',k);
put 'Textent:' k=;
return;
```

APPENDIX 3 (Code for definition of regression models)

```
entry libname dataname sasname $8 indeplid constlid
   listid 8;
length factor $17 tempname $8 i j k rc boxlid 8;
init:
call notify('TEXTLAB1','_set_text_',
'Specify the factors influencing
   '||trim(sasname)||'.');
call notify('LISTBOX','_set_title_',sasname);
/* get list id for the list of items in the list
box */
call notify('LISTBOX','_get_value_',i);
boxlid=getiteml(i,3);
rc=clearlist(listid);
j=listlen(indeplid);
```

```
put 'regmodel init: ' indeplid= j=;
submit;
data work._indep_; length name $8 number 8;
endsubmit;
do i=1 to j;
tempname=getitemc(indeplid,i);
submit;
name="&tempname";
number=&i;
output;
endsubmit;
end;
k=j;
j=listlen(constlid);
do i=1 to j;
k=k+1;
tempname=getitemc(constlid,i);
submit;
name="&tempname";
number=&k;
output;
endsubmit;
end;
submit continue; run;
data work._factor_(keep=factor);
set work._indep_;
length factor $17; factor=name; run;
proc sql;
create table work._quadra_ as
select a.name, b.name as name2
from work._indep_ a, work._indep_ b
where a.number <= b.number;
quit;
data work._quadra_(keep=factor);
set work._quadra_; length factor $17;
factor=compress(name||'*'||name2); run;
proc append data=work._quadra_ base=work._factor_;
run;
endsubmit;
call
notify('DATATAB','_set_dataset_','WORK._FACTOR_');
return;
main:
return;
term:
rc=copylist(boxlid,'N',listid);
return;
datatab:
call
notify('DATATAB','_get_column_text_','FACTOR',facto
r);
rc=searchc(boxlid,factor);
if rc=0 then call notify('LISTBOX','_add_',factor,-
1);
return;
listbox:
call notify('LISTBOX','_selected_',1,rc);
if rc>0 then call notify('LISTBOX','_delete_',rc);
return;
```

ACKNOWLEDGMENTS

SAS AND SAS/AF are registered trademarks or trademarks of SAS Institute Inc. in the USA and other countries. ® indicates USA registration.

CONTACT INFORMATION

David J. Meade
5204 E. Ben White Blvd.
Mail Stop 613
Austin, TX 78741
USA
Email: david.meade@amd.com

Scott Lacey
5204 E. Ben White Blvd.
Mail Stop 613
Austin, TX 78741
USA
Email: scott.lacey@amd.com