Appendix II: The SAS Excel Link Program

```
/*
ExcelLinkNov2006.SAS.
*/
%macro ExcelSolve;
%macro ReadData;
option errors=0 nocenter;%global acts n sheet;%let acts=;%let n=;
%Do p=1 %to 2;
  %let sheet=%sysfunc(byte(&p+64));%put sheet=&sheet;%global actsall&sheet;
  filename model1 dde "excel|&sheet.!r1c6:r1c220";
  data _null_;infile model1 dlm='09'x notab dsd missover lrecl=4000;
    array va{200} $ ;input va{*};length actsall $3000.;
    do ii=1 to dim(va);
    if upcase(va{ii})='_TYPE_' then do;N1=ii-1;leave;end;
    else actsall=trim(left(actsall))||' '||trim(left(va{ii}));end;
    call symputx("actsall&sheet",actsall);call symputx("N&sheet",N1);run;
%put actsall&sheet=&&actsall&sheet;%put n&sheet=&&n&sheet;

%Let StartRow=;
filename model1a dde "excel|&sheet.!r1c1:r200c1";
data _null_;infile model1a dlm='09'x notab dsd missover lrecl=4000;
  informat startrow $char13.;input startrow;
  if UPCASE(startrow)="CONSTRAINT" then call
symputx('StartRow',put(_n_+1,f2.0));run;
%put startrow=&startrow;

filename model1b dde "excel|&sheet.!r&StartRow.c1:r500c250";
data &sheet free&sheet;infile model1b dlm='09'x notab dsd missover lrecl=7000;
  informat name $char23. units $char20. slabel  $char20.  Default $char20.
activity $34. _type_ $char4. _type_&sheet $char4.;
   input name $ units $  slabel $ default $  activity $ &&actsall&sheet
           %if &p=1 %then _type_ nada _rhs_;
           %if &p>1 %then _type_&sheet nada _rhs_&sheet;;
  _error_=0;
  name=trim(left(upcase(name)));Order1=_n_;
```

```
if upcase(lag(name))='GROWER NON-SEED COSTS' then stop;
if (upcase(name)="SNFYUS"|upcase(name)="SNCFUS"|upcase(name)="LNFYUS"|upcase(n
ame)="LNCFUS")
   then do;output Free&sheet;f+1;end;
else output &sheet;
  run;

%put actsall&sheet;
%let Acts=&Acts &&actsall&sheet;%let N=%eval(&N+&&n&sheet);
proc sort data=&sheet;by name;run;

%if&p>1%then%do;
proc sort data=A;by name;run;
data A;length _type_ $8.; merge A(IN=A) &sheet(IN=&sheet);by name;
  if &Sheet then do;
    _rhs_=max(put(_rhs_,f30.10),put(_rhs_&sheet,f30.10));
    _type_=trim(left(_type_&sheet));
    end;
  _type_=trim(left(_type_));
  if ~index(upcase(_type_),"FR");*Eliminate free constraints;
  if index(upcase(_type_),"LE") then _type_="LE";
  if index(upcase(_type_),"PA") then _type_="PARMS";*new;
  if index(upcase(_type_),"EQ") then _type_="EQ";
  if index(upcase(_type_),"GE") then _type_="GE";
  if index(upcase(_type_),"QU") then _type_="QUAD";
  if index(upcase(_type_),"MA") then _type_="MAX";
  if index(upcase(_type_),"MI") then _type_="MIN";
  if index(upcase(_type_),"FI") then _type_="FIXED";
  drop _rhs_&sheet _type_&sheet;
    sheets="      ";s=0;
    if B then sheets="B"||trim(left(sheets));
    if A then sheets="A"||trim(left(sheets));
    sheets=trim(left(sheets));
      if sheets="AB" then s=1;
      if sheets="A" then s=2;
      if sheets="B" then s=3;
      Output A;
  run;
%end;%end;

proc sort data=a out=a2;by s order1;run;
*This section writes out the entire tableau in Model sheet;
filename qpout dde "excel|model!r1c1:r6002c250" lrecl=4000;
data _null_;set A2;
  file qpout;
```

```
Name_=trim(left(translate(trim(left(name)),"_"," ")));
  format _type_ $4.  _rhs_ d9.;
  if _n_=1 then put "Name sheets &acts _type_ _rhs_ Name";
  format _type_ $char8.;
   put Name_ sheets &acts _type_ _rhs_ Name_;
  run;
%mend ReadData;
%ReadData;

%put acts=&acts n=&n;

%let BaseOrScen=Base;
data a;set a(drop=s sheets) end=end ;LENGTH _NAME_ $8 ;
if ~index(upcase(_type_),"FR");
ARRAY ACTS(*) &ACTS;ARRAY ACTS2{&N} _TEMPORARY_;
*********************************************;
*Set up Hession matrix;
   IF upcase(_type_)='QUAD' then do;
     DO I=1 TO DIM(ACTS);ACTS2{I}=ACTS{I};ACTS{I}=0;END;
     DO I=1 TO DIM(ACTS);
     ACTS{I}=ACTS2{I};
       CALL VNAME(ACTS{I},_NAME_);
       _RHS_=.;OUTPUT;ACTS{I}=0;
   END;END;
*********************************************;
*Create separate constraints for fixed coefficients;
Else IF upcase(_type_)='FIXED' then do;
   DO I=1 TO DIM(ACTS);ACTS2{I}=ACTS{I};ACTS{I}=.;END;
   DO I=1 TO DIM(ACTS);
   IF ACTS2{I}~=. THEN DO;ACTS{I}=1;_RHS_=ACTS2{I};_TYPE_="EQ";*CALL
SOUND(8500,1);
      *PUT 'FIXED ' I=  acts{i}= _rhs_=;
      OUTPUT;ACTS{I}=.;_RHS_=.;
   END;END;END;
*********************************************;
ELSE if upcase(_type_)='MAX' then do;_type_='LINEAR';output;*Required for NLP;
   if index(upcase(name),"SCE") then call symputx("BaseOrScen","Scenario");
   end;
*********************************************;
ELSE output;
*********************************************;
IF END THEN DO;
  do i=1 TO DIM(ACTS);ACTS{i}=0;_TYPE_='LOWERBD';_RHS_=.;name='LOWERBD';END;
  output;*Restrict to nonnegative quadrant;
```

```
  /*_type_="PARMS";OUTPUT;*/*Starting solution of null vector;
  end;
run;

option ps=max ls=max;
filename nlplog dde "excel|LOG!r1c1:r500c25" notab lrecl=4000;
data _null_;file nlplog; do r=1 to 500;put " ";end;run;
filename nlpout dde "excel|OUTPUT!r1c1:r500c25" notab lrecl=4000;
data _null_;file nlpout; do r=1 to 500;put " ";end;run;
option notes;
proc printto print=nlpout log=nlplog;run;*allows log to be read;
************************************************************;
*This is the quadratic optimization section.  Parameters have
been set to Proc NLP (nonlinear programming) standards with
iteration restrictions and convergence criteria;

PROC NLP INQUAD=A OUTEST=QPOUT short maxiter=2000 absgconv=1e-10 20;
*maxtime=60 noeignum;* out=out1 outder=2;* nomiss;*PALL;*options;
MAX;
PARMS &ACTS;*list of activity names (parameters);
RUN;
************************************************************;
%put SYSERR=&syserr  -A nonzero syserr code indicates NLP execution problems;
proc printto;run;
option nonotes;
%let crit=;

data log_;
  infile nlplog dlm='09'x notab dsd missover lrecl=4000;
  informat row $char125.;input row;
  if index(row,"convergence criterion satisfied.")then do;
    call symputx('crit',"Convergence criterion satisfied.");
    call sound(3000,5);put 'It worked. ' row;*stop;
    end;
  if ~index(row,'randomly')& ~index(row,'printto')& ~index(row,'put s');run;
data _null_;set log_;file nlplog;put row;
run;
%put crit=&crit;
filename Esystem dde "Excel|system";
%if &crit=%then%do;
  data _null;  file Esystem;
```

```
    put '[ALERT("Convergence criteria may not be satisfied for this NLP optimi-
zation.", 3)]';
  run;
  %let crit=Problem with convergence;
%end;

filename nlplog2 dde "excel|LOG!r1c1:r3c25" notab lrecl=4000;
data _null_;file nlplog2;
  d=date();t=time();
  put "U.S. Structural Change Model.  &BaseOrScen intercepts";
  put "Optimized " d date8. "  "  t timeampm10.;
  put "&crit";
  run;
filename nlpout2 dde "excel|OUTPUT!r1c1:r2c25" notab lrecl=4000;
data _null_;file nlpout2;
  d=date();t=time();
  put "U.S. Structural Change Model. &BaseOrScen intercepts" d date8. "  "  t
timeampm10.;
  put "&crit";
  run;

%macro WriteOut;
%Do p=1 %to 2;
%let sheet=%sysfunc(byte(&p+64));
filename Solution dde "excel|&sheet.!r16c6:r19c220";

data qpout;set qpout(WHERE=((UPCASE(_TYPE_)='PARMS'| UPCASE(_TYPE_)='GRAD'|
        UPCASE(_TYPE_)='UPPERBD'| UPCASE(_TYPE_)='LOWERBD') & _iter_~=0));
file solution lrecl=4000;
put &&actsall&sheet _type_ "09"x _rhs_;
run;
************************************;
%if &BaseOrScen=Base %then filename Sol1 dde "excel|&sheet.!r3c6:r3c200";;
%if &BaseOrScen=Scenario %then filename Sol1 dde "excel|&sheet.!r2c6:r2c200";;
data _null_;set qpout(WHERE=(UPCASE(_TYPE_)='PARMS' & _iter_~=0));
file sol1 lrecl=4000;
put &&actsall&sheet _type_ "09"x _rhs_;
run;

%if &BaseOrScen=Base %then filename Grad1 dde "excel|&sheet.!r8c6:r8c200";;
```

```
%if &BaseOrScen=Scenario %then filename Grad1 dde "excel|&sheet.!r7c6:r7c200";;
data _null_;set qpout(WHERE=(UPCASE(_TYPE_)='GRAD' & _iter_~=0));
file grad1 lrecl=4000;
put &&actsall&sheet _type_ "09"x _rhs_;
run;
************************************;
%end;
%mend writeout;
%writeout;

DATA QPOUT1;LENGTH NAMES $14; SET QPOUT(WHERE=(UPCASE(_TYPE_)='PARMS'));
  ARRAY ABC{*} _NUMERIC_;
  DO I = 1 TO DIM(ABC);
  CALL VNAME(ABC{I},NAMES);ESTIMATE=ABC{I};
  OUTPUT;
  END;
FORMAT ESTIMATE F13.4;
KEEP ESTIMATE NAMES;
call sound(3000,10);
RUN;
PROC PRINT;RUN;
**************;
data free;merge freeb freea;by f;
run;
proc iml;
use qpout(WHERE=( UPCASE(_TYPE_)='PARMS' ));
read all var{&acts} into parms;
use qpout(WHERE=(UPCASE(_TYPE_)='GRAD' & _iter_~=0));
read all var{&acts} into grads;
use free;
read all var{&acts} into free;
read all var{name} into name;*print name;

do x=1 to nrow(free);
do y=1 to ncol(free);
if free(|x,y|) then  free(|x,y|)=1;
if ^free(|x,y|) then  free(|x,y|)=0;
end;
end;
*print free;
Freegrads=free#grads;
cnames={&acts};
*print freegrads(|c=cnames|);* parms;
_rhs_=freegrads*t(parms);
```

```
print _rhs_(|r=name f=comma20.2|);
create freeOut var {_rhs_ name};
append var {_rhs_ name};
quit;run;
*proc print data=FreeOut;run;
%if &BaseOrScen=Base %then filename Frees dde "excel|INCOMES!r3c2:r6c3";;
%if &BaseOrScen=Scenario %then filename Frees dde "excel|INCOMES!r3c4:r6c5";;
data _null_;set FreeOut;
file Frees;Put name _rhs_ ;
run;
DATA _NULL_;
M=TIME()-&_H;
PUT "The combined SAS execution time is " M TIME10.0;
RUN;

%mend ExcelSolve;

%macro loop;*Hotlinks with Excel spreadsheet;
option noquotelenmax nonotes;dm output 'autopop off' ;
filename cmds dde 'excel|system';
data _null_;
   file cmds;
put '[app.activate]';
put '[app.maximize]';
put '[window.maximize]';
run;
%global flaga _H;
%do tt=1 %to 5; *This determines the number of solutions with each SAS run;
%let stop=;
filename start1 dde 'excel|A!r17c1';
data start;*This gives value of entry in cell A!r17c1;
 infile start1; input a $ @;call symputx('flaga',a);stop;run;
%put flaga=&flaga;
filename start dde 'excel|A!r17c1' hotlink;
data start;*This indicates when to start the optimization;
 infile start;input b $ @;
 if upcase(b) eq "STOP" then do;call symputx('Stop','STOP');call
sound(2500,2);stop;end;
 if b ne "&flaga" then do;call sound(2500,2);stop;end;run;
%LET _H=%SYSFUNC(TIME());

%if &stop~=STOP%then%ExcelSolve;%else%let tt=100;
data _null_;call sound(300,150);run;
%end;
```

```
data _null_;file start1;put "STOPPED";run;
data _null;  file Esystem;
  put '[ALERT("SAS program stopped. Press F8 in SAS program window to resume
Proc NLP.", 2)]';
  run;
option notes;
%mend loop;%loop;
```