# 11.0 APPENDIX-B: COMPUTATION

Computational details and the pseudo codes of the time-varying $ARX(p_t)$ model and the MI-SRI composite imputation method will be given by using different statistical packages that have the ability of performing multiple imputation. They are

1. SAS (http://www.sas.com),
2. S-plus (http://www.insightful.com),
3. freeware R-project (http://www.r-project.org),
4. Schafer's NORM (http://www.psu.edu/∼ jls/misoftwa.html)

If we provide the computational pseudo-codes, we are hoping that the applied statisticians will be able to use the techniques without additional search. Programing code in SAS will be given as well as the code written in S-Plus due to the addition of the new missing data library. For the users who do not have access to SAS or S-plus, there are freeware statistical packages such as R-project and Schafer's NORM. R-project is free and works the same way as S-plus does. NORM is a free windows stand-alone software to create multiply imputed data sets from MVN distribution.

The time-varying $ARX(p_t)$ model does not require a special statistical package. Any program which fits the linear regression model and saves the residuals can be used without additional programming. However, MI-SRI composite imputation method requires special computational techniques due to multiple imputation. SAS (SAS Institute, Inc.) has procedures called MI and MIANALYZE. Procedure MI creates multiple imputations from MVN distribution and procedure MIANALYZE combines estimates obtained from multiply imputed data sets by Rubins's rule. S-Plus also has a library called 'missing data' to create multiple imputations from MVN distribution and to combine the results from the multiply imputed data sets. Schafer (http://www.stat.psu.edu/~jls) has four different built-in functions written for S-Plus which have been used as bases for the missing data library in S-plus and the procedures in SAS. One of them is called as NORM which is the only standalone package runs in windows and creates multiple imputations from MVN distribution. R-project, a free statistical package, that works the same way as S-Plus has also had built-in functions. The functions in R-project are comparable with what NORM program does. There are other statistical packages provide multiple imputation techniques to users but the ones that are mentioned here use the data augmentation technique with the specified prior distribution to create multiply imputed data sets. Data augmentation was the technique that we have been implemented to the MI-SRI composite imputation method.

Pseudo codes written in SAS, S-plus, and NORM will be given. Let assume that there are three time points for 319 subjects in the study with one time-dependent and one time-independent covariates. Let $Y_t$ represents the longitudinal continuous outcome, $X_t$ be the time-dependent and $Z$ be the time-independent covariates for t=1,2,3. Number of imputation is set to 3. Let also assume that the range of $X_t$ is (0,40) and Z is a binary variable that takes values 0 and 1.

## 11.1 PSEUDO CODE IN SAS

The pseudo coding in SAS will be based on the version 8.2 with procedures called MI and MIANALYZE. Procedure Multiple Imputation (Proc MI) imputes missing values under the assumption of the MVN distribution. We will use MVN with informative Ridge Prior as an imputation model. Procedure Multiple Imputation Analyzer (Proc MIANALYZE) is used after Proc MI to be able to combine estimates from the results of analyzing multiply imputed data sets. The scaler values are obtained by using the Rubin's rule of combining estimates.

*The following is the code written for SAS version 8.2;
*It performs MI-SRI method with the time-varying $ARX(p_t)$ model;

title 'multiple imputation to missing covariates with informative Ridge prior=0.75';

```
proc mi data=indata nimpute=3 out=mioutdata seed=12345
minimum=(0 0 0 0) max=(40 40 40 1) round= 1 1 1 1;
mcmc chain=multiple prior=ridge=0.75 initial=em;
var x1 x2 x3 z;
run;

title ‘Stochastic Regression Imputation to missing longitudinal outcome variable’;run;
%macro impsri (sridata=&sridata,
resp=&resp, covar=&covar,
outdata=&outdata, outest=&outest,
yresid1=&yresid1, yresid2=&yresid2, phat=&phat,
edf=&edf);
title ‘’;
proc reg data=&sridata noprint;
by _imputation_;
model &resp=&covar;
output out=out1 r=&yresid1 p=&phat;
run;

ods output anova=anovadat;ods listing close;
proc reg data=&sridata;
by _imputation_;
model &resp=&covar;
run;
quit;ods listing;

*impute missing Y by predY+normal(0,MSE) by time;
*missing Y(t) will be imputed predY(t)+random sample of N(0,$MSE_t$);
proc sort data=out1;by _imputation_;run;
data anovadat;set anovadat;
```

```
where source='Error';sqms=sqrt(ms);
keep _imputation_ ms sqms;
run;
data out1;merge out1 anovadat;by _imputation_;run;
proc sort data=out1;by id;run;

data out1;set out1;by id;
if _imputation_=1 and &resp=. then &resp=int(&phat+(sqms)*rannor(987654321));
if _imputation_=2 and &resp=. then &resp=int(&phat+(sqms)*rannor(87654321));
if _imputation_=3 and &resp=. then &resp=int(&phat+(sqms)*rannor(7654321));
run;

proc sort data=out1;by _imputation_;run;
*fits the regression model to imputed response by Stochastic Regression Imputation;
*obtains residuals for the next time point;
*residuals at time t-1 are considered as complete covariate at time t;
proc reg data=out1 outest=&outest covout;
model &resp=&covar;
output out=&outdata r=&yresid2;by _imputation_;
run;
*&outdata is the data set which will be used in the next time point;

*combines estimates of the regression parameters by Rubin's rule;
*where edf=n-number of parameters that needs to be estimated;
proc sort data=&outest;by _imputation_;run;
ods output ParmEst=est1;ods listing close;
proc mianalyze data=&outest multivariate edf=&edf alpha=.05; var intercept &covar;
run;
quit;ods listing;
%mend impsri;run;
```

```
%let covar=x1 z;run;
%impsri (sridata=mioutdata,
resp=y1,covar=&covar,
outdata=sridata1,outest=estbeta1,
yresid1=yres01,yresid2=yres1,phat=phat1,
edf=316);
run;
*check to see whether Multiple Imputation is worked properly;
proc means data=sridata1 n nmiss mean;var y1 x1 z yres1;run;


%let covar=x2 z yres1;run;
%impsri (sridata=sridata1, resp=y2,covar=&covar,
outdata=sridata2,outest=estbeta2,
yresid1=yres02,yresid2=yres2,phat=phat2,
edf=315);
run;
proc means data=sridata2 n nmiss mean;var y2 x2 z yres1;run;


%let covar=x3 z yres1 yres2;run;
%impsri (sridata=sridata3,resp=y3,covar=&covar,
outdata=sridata3,outest=estbeta3,
yresid1=yres03,yresid2=yres3,phat=phat3,
edf=314);
run;
proc means data=sridata3 n nmiss mean;var y3 x3 z yres3 yres2 yres1;run;
```

## 11.2 PSEUDO CODE IN S-PLUS

The following code is written for S-Plus 6.0 professional release 2.

```
#To start S+MissingData, use the following command to include the ”Missing“ library.
library(missing)
# use function miss to explore patterns of missingness first. This function rearranges the row and columns of the data according to the number of patterns of missing values, then summarizes the patterns of missing values
indata.miss < − miss(indata)
summary(indata.miss)
plot(indata.miss,sort.obs=F)
#To save computation in EM and DA routines, create a preGauss object as follows
indata.s < − preGauss(indata)
# To fitting the Gaussian model using EM algorithm
indata.em < − mdGauss(indata.s, prior=”ridge“,na.proc=”da“)
indata.em$paramIter
indata.em$algorithm
```

## 11.3 PSEUDO CODE IN NORM

Schafer (1999) has written general purpose MI software that is called NORM. It uses a MVN model for estimating the parameters (means, variances, and covariances) and for imputing the missing values. A stand alone version of NORM suitable for PCs running Windows (95/98/NT) is offered for free. It is also available as a function in S-Plus for versions 4.5 and lower. NORM has drop-down menus and easy to use without writing extensive code.

1. Select ”New“ from the File menu. It will be prompted for the name of the file that contains your data (named *.dat).
2. NORM reads the data from the specified data file. The data file is displayed in a small

window on the Data file sheet. Make sure that the numeric missing value code is correct.

3. Select variables for the model. We may add a variable to the model or remove it by clicking on its box in the “In model” column.
4. NORM can also report important features of all the variables, including means, standard deviations, rates and patterns of missingness. To produce a summary, go to the summarize sheet by clicking on the “Summarize” tab. Select an appropriate name for the file where the output is to be stored and press the“Run” button. The file will be created and displayed in a small window. This summary contains information on all the variables currently in the model.
5. The data augmentation (DA) algorithm in NORM simulates random values of parameters and missing data from their posterior distribution. It is the method which NORM creates multiple imputations for the missing data. Before running DA, its a good idea to run EM first. Running EM first will provide a nice set of starting values for the parameters. To run DA, go to the “DA sheet” by clicking on the “Data augmentation” tab in your NORM session. Then press the “Run” button. Any run of DA will create two files: an output (*.out) file reporting the results of DA, and a parameter (*.prm) file where the final simulated values of the parameters are stored. When DA is finished running, the output file is automatically displayed but the parameter file is not. Either of these files may be displayed at any time by using the “Display” menu.
6. The number of DA cycles and various other computing options may be set via the “Computing” button. Since we proposed the use of informative Ridge Prior, the option for the Ridge prior should be set via the “Computing” button.
7. “Run” button will create multiply imputed data sets.
8. Once we have created m imputed versions of the data set, we may analyze them in any way that would be appropriate without missing data. NORM does not perform these analysis for us.