

Clark Labs Modified Gstat Code

This document contains information describing the changes made when producing the Clark Labs, Clark University, modified version of Gstat 2.1.0.

All ANSI-C files were imported as .cpp files and compiled in Borland C++ Builder 3.0.

The following files were included in the Clark Labs modified version of Gstat. Please note that not all available files were included.

From Gstat : d:\gstat-2.1.0b\src

data.cpp

direct.cpp

fit.cpp

getest.cpp

getopt.cpp

gls.cpp

glvars.cpp

gstat_main.cpp

lex.cpp

lm.cpp

map2gif.cpp

mapio.cpp

msim.cpp

block.cpp

xvalid.cpp

parse.cpp

parsecmd.cpp /*Note that this file was added from Gstat Version 1.9*/

plot.cpp

polygon.cpp

pqueue.cpp

predict.cpp

random.cpp

read.cpp

reml.cpp

report.cpp

sample.cpp

search.cpp

select.cpp

sem.cpp

sem_main.cpp

sim.cpp

stat.cpp
ui.cpp
userio.cpp
utils.cpp
vario.cpp
vario_fn.cpp
vario_io.cpp
writecmd.cpp

From the Meschach Matrix Library : d:\gstat-2.1.0b\meschach

bdfactor.cpp
chfactor.cpp
copy.cpp
hsehdr.cpp
err.cpp
ivecop.cpp
init.cpp
lufactor.cpp
machine.cpp
matop.cpp
matrixio.cpp
meminfo.cpp
memory.cpp
memstat.cpp
norm.cpp
pxop.cpp
qrfactor.cpp
solve.cpp
sprow.cpp
sparseio.cpp
spchfctr.cpp
sparse.cpp
submat.cpp
update.cpp
vecop.cpp
version.cpp

Individual changes by Clark Labs are indicated in files with the symbol /*KS*/.

Systematic changes for conversion to Windows as well as other significant changes are described here.

All routines that manipulate memory were converted to Windows API calls.

memset à FillMemory

memmove à MoveMemory

memcpy à CopyMemory

malloc à GlobalAlloc

realloc à GlobalReAlloc

free à efree (efree calls GlobalFree)

bzero à ZeroMemory

All if/then statements that inquire whether a pointer variable is NULL were changed in the following way:

If (variable == NULL) à If (!variable)

If (variable != NULL) à If (variable)

These types of statements cause access violations in Windows NT.

All routines that read Idrisi file formats were modified to read Idrisi32 file formats.

The original Gstat file detection process first checks for an Idrisi raster file, and if not found, it then checks for an Idrisi vector file. The checking order was reversed so that a vector is detected first.

The GRIDMAP data structure cellsize element was changed to two new elements :

cellsizeX and cellsizeY. This is because IDRISI permits irregular cell sizes in x and y directions.

The correlogram and crosscorrelogram functions were added. A global variable was added to indicate whether these functions were called. A statement added to the command file, # CORRELOGRAM, is read by the command file parser to set this variable.

All warning and error messages were changed so that they write to a file called gsttmp.err that is written to the working directory of IDRISI. Warning messages have a "W=" added to them and error messages have an "E=" added to them before they are written to the file. The message file is read by IDRISI, and IDRISI displays them.

To make the h-scatterplot option in IDRISI possible, a series of changes were made to Gstat.

A binary file is now produced that is the listing of the x and y coordinates of each datum for each pair and each pairs index which enable them to be connected to their attributes in a separate file.

Because data pair files can be quite large, it was necessary to optimize memory usage and disk accessing to maintain speed when calculating variograms. To do so, class CEFile and class CEFILE2 were added to utils.h and their opening, closing, writing functions to utils.cpp. The elements distx and disty were added to the VARIOGRAM data structure to record changes in x and change in y distances for the variogram cloud. Finally, the cloud and directional variogram calculation routines were modified to accomodate for the timing of reading and writing from memory to the hard disk. The following routines were replaced with three procedures each in sem.cpp:

```

static SAMPLE_VGM *semivariogram(DATA *a, SAMPLE_VGM *ev) à SAMPLE_VGM *
UpFrontSemivariogram(DATA *d, SAMPLE_VGM *ev,CEFile* f2);
SAMPLE_VGM * PageSemivariogram(DATA *d, SAMPLE_VGM *ev, CEFile* f2);
SAMPLE_VGM * c_Semivariogram(DATA *d,DATA *b, SAMPLE_VGM *ev,CEFile* f);
static SAMPLE_VGM *cross_variogram(DATA *a, DATA *b, SAMPLE_VGM *ev) à SAMPLE_VGM *
UpFrontCross_covariogram(DATA *d, DATA *b,SAMPLE_VGM *ev,CEFile* f2);
SAMPLE_VGM * PageCross_covariogram(DATA *a, DATA *b,SAMPLE_VGM *ev,CEFile* f2);
SAMPLE_VGM * c_Cross_covariogram(DATA *d, DATA *b,SAMPLE_VGM *ev,CEFile* f);
static SAMPLE_VGM *covariogram(DATA *a, SAMPLE_VGM *ev) à
SAMPLE_VGM *UpFrontCovariogram(DATA *d, SAMPLE_VGM *ev,CEFile* f2);
SAMPLE_VGM *PageCovariogram(DATA *d, SAMPLE_VGM *ev,CEFile* f2);
SAMPLE_VGM *c_Covariogram(DATA *d,DATA *b, SAMPLE_VGM *ev,CEFile* f);
static SAMPLE_VGM *cross_covariogram(DATA *a, DATA *b, SAMPLE_VGM *ev) à
SAMPLE_VGM * UpFrontCross_variogram(DATA *a, DATA *b, SAMPLE_VGM *ev,CEFile* f2);
SAMPLE_VGM *PageCross_variogram(DATA *a, DATA *b, SAMPLE_VGM *ev,CEFile* f2);
SAMPLE_VGM *c_Cross_variogram(DATA *a, DATA *b, SAMPLE_VGM *ev,CEFile* f);

```

The following routines were added :

```

static void fprintf_sample_vgm(CEFile* f, const SAMPLE_VGM *ev, int n_list1, int n_list2);
static void fprintf_header_vgm(CEFile* f, const DATA *d1, const DATA *d2,const SAMPLE_VGM *ev);
void Pagefprintf_sample_vgm(CEFile* f, const SAMPLE_VGM *ev, int n_list1, int n_list2);
SAMPLE_VGM * SetEVnest(DATA *a, DATA *b, SAMPLE_VGM *ev);
SAMPLE_VGM * AllocVGM(DATA *a, DATA *b, SAMPLE_VGM *ev);
void ResetEVspace( SAMPLE_VGM *ev );
void PageDivide(SAMPLE_VGM *ev);
static int GetIndex(DATA *d, SAMPLE_VGM *ev);
static int GetMaxCross_variogramIndex(DATA *a, DATA *b, SAMPLE_VGM *ev);
static int GetMaxCross_covariogramIndex(DATA *a, DATA *b,SAMPLE_VGM *ev);
static int GetMax_CovariogramIndex(DATA *d, SAMPLE_VGM *ev);

```

Cross validation reports are now written to a file called "xvreport.id\$" rather than automatically to the screen. It is subsequently deleted by IDRISI.

The command file parser that is used in the Clark Labs modified version is based on Gstat version 1.9 parsecmd.cpp rather than Gstat 2.1.0 parse.cpp. Some changes were

made to the older parser – such as – modifications that enable Windows file naming conventions (such as spaces in filenames) to be usable.

The Clark Labs version uses the Gstat version 1.9 "percent_done" utility. Percent_done is modified to include the Windows API command PostMessage. This command uses a Windows handle to communicate with IDRISI. That handle is passed through the command line. A "-t" option was added to the command line options which takes the handle number passed by IDRISI. Gstat 2.1.0 now has its own handler for softwares to use, but IDRISI maintains the modified Gstat 1.9 version of percent_done because the Clark Labs has not yet updated the change.