Geospatial Simulation: A model-independent, open-source geospatial tool for managing point-based environmental model simulations at multiple spatial locations

Dr. Kelly R. Thorp
USDA-ARS, Arid Land Agricultural Research Center
kelly.thorp@ars.usda.gov
October 30, 2013

If you use Geospatial Simulation, please cite the following paper:
Thorp, K. R., Bronson, K. F., 2013. A model-independent, open-source geospatial tool for managing point-based environmental model simulations at multiple spatial locations. *Environmental Modeling & Software*. 50:25-36. doi:10.1016/j.envsoft.2013.09.002

Overview

The purpose of Geospatial Simulation is to provide an interface to run point-based, one-dimensional environmental simulation models using geospatial data contained in a geographic information system (GIS). Geospatial Simulation provides software tools for geoprocessing spatial datasets, for running model simulations for unique polygon features, and for calibrating model parameters to site-specific conditions. Many of the software tools available within Geospatial Simulation permit a user to establish and manage a 'base layer' polygon shapefile, which stores the geospatial data that is passed to and from the input and output files of the simulation model. The software was developed as a plug-in for the open-source Quantum GIS environment. An important feature is the ability of the software to interface with the input and output files of any point-based environmental simulation model. Instructions for use of the six software tools included in Geospatial Simulation and for development of the required files to interface the GIS with a simulation model are given below.

Geoprocessing Tools

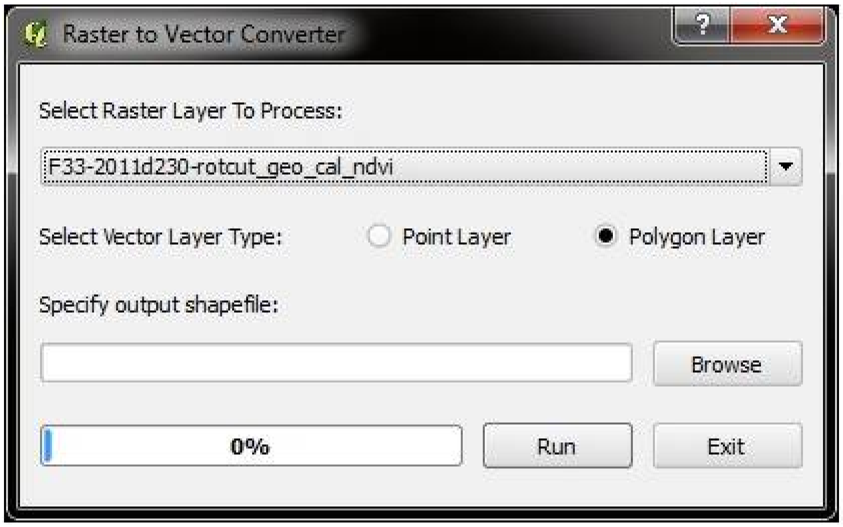
Many of the geoprocessing tools required to process spatial data in Quantum GIS are available through a different software plug-in called fTools (www.ftools.ca). Additional geoprocessing tools within Geospatial Simulation do not aim to duplicate the functionality of fTools but rather to extend it. The fTools plug-in should be implemented to the extent that it is useful to prepare the base layer polygon shapefile required by Geospatial Simulation. Additional tools have been created within Geospatial Simulation for handling raster datasets and for summarizing geospatial data within the base layer polygons.

<u>Tool #1: Raster to Vector Converter</u>

The Raster to Vector Converter (Fig. 1) provides a tool for processing raster data layers, such as kriged soil texture maps or remote sensing images. Essentially, the tool converts the raster layer to a vector layer. The Vector Geoprocessor (discussed below) and other vector data processing tools within Quantum GIS can then be used to process the data. Beginning with Quantum GIS version 1.8.0, a tool to polygonize raster layers to vector layers was provided by default, which was not available when Geospatial Simulation development began. However, it is my understanding that the default tool does not currently handle floating point images. The Raster to Vector Converter supplied with GeoSim does handle floating point; however, the algorithm can be quite slow for large images. To use the tool, complete the following steps in the Quantum GIS environment.

1. Select Plugins->Geospatial Simulation->Raster to Vector Converter
2. Select the raster layer to be processed. The combo box will be populated with all raster data layers available in the current workspace.

3. Select the type of vector layer to be created. By selecting the 'point layer' option, a shapefile of points at center of each raster cell will be created. By selecting the 'polygon layer' option, a shapefile of polygons for each raster cell will be created.
4. Specify a file path and file name for the output shapefile.
5. Click Run. For high resolution raster layers, the process can take many hours and can create quite large shapefiles.


Raster to Vector Converter

Select Raster Layer To Process: F33-2011d230-rotcut_geo_cal_ndvi

Select Vector Layer Type: Point Layer [✓ Polygon Layer]

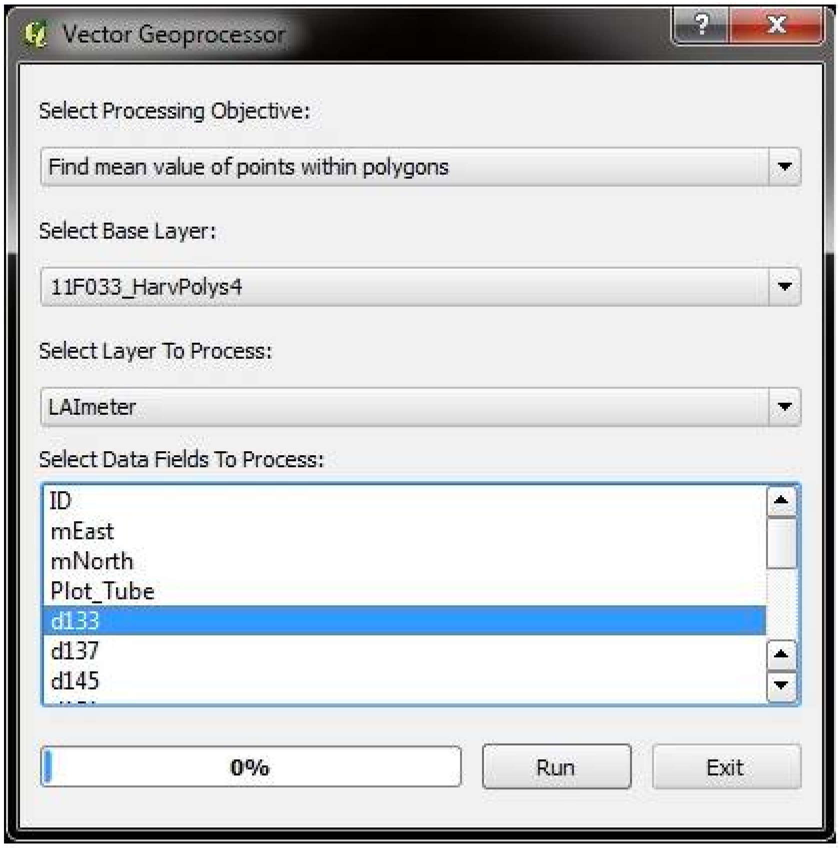
Specify output shapefile:

Figure 1. The graphic user interface for the Raster to Vector Converter tool.

Tool #2: Vector Geoprocessor
The Vector Geoprocessor (Fig. 2) permits a user to process vector data layers to summarize data within the geospatial units established by the base layer polygon shapefile. By selecting 1) the base layer polygon shapefile, 2) another layer containing the data to be processed, 3) the attributes of interest in the layer to be processed, and 4) a processing objective, the tool will process the data contained within the process layer and append the results to the base layer. To use the tool, complete the following steps in the Quantum GIS environment.

1. Select Plugins->Geospatial Simulation->Vector Geoprocessor
2. Select the processing objective from the list of available options, as follows:
    a. Find mean value of points (process layer) within polygons (base layer).
    b. Find median value of points (process layer) within polygons (base layer).
    c. Find maximum value of points (process layer) within polygons (base layer).
    d. Find minimum value of points (process layer) within polygons (base layer).
    e. Find (area-weighted) mean value of polygons (process layer) within polygons (base layer).
    f. Find maximum area polygon (process layer) within polygons (base layer). This option returns the attribute from polygon having the maximum area within the base layer polygons. If two or more polygons have the same attribute value, their areas are summed.
    g. Add attributes of polygons (process layer) to the points (base layer) falling within the polygon. Likely, this is not explicitly needed for Geospatial Simulation but is nonetheless useful for other geoprocessing tasks.

3. Select the base layer polygon shapefile. For processing options 'a' through 'f' above, the combo box will be populated with all polygon shapefiles in the current workspace. For processing option 'g' above, the combo box will be populated with all point shapefiles in the current workspace.
4. Select the layer to be processed. For processing option 'a' through 'd' above, the combo box will be populated with all point shapefiles in the current workspace. For processing options 'e' through 'g' above, the combo box will be populated with all polygon shapefiles in the current workspace.
5. Select the attributes (data fields) from the process layer. For processing options 'a' through 'e' above, the list box will be populated with all numeric data fields in the process layer. For processing options 'f' and 'g' above, the list box will be populated with all text data fields in the process layer.
6. Click Run.


Vector Geoprocessor

Select Processing Objective: Find mean value of points within polygons

Select Base Layer: 11F033_HarvPolys4

Select Layer To Process: LAImeter

Select Data Fields To Process:

ID
mEast
mNorth
Plot_Tube
d133
d137
d145

Figure 2. The graphic user interface for the Vector Geoprocessor tool.

Simulation Control

Geospatial Simulation is designed to control a simulation model that has been previously compiled as a separate executable file. It will automate simulations by passing geospatial data from unique spatial units (the base layer polygons) to the model input files, and it will similarly pass key model outputs back to the GIS database (Fig. 3). An important feature of the software is its ability to control any one-dimensional simulation model that uses input and output files. In other words, the software is designed to be model independent. It accomplishes this using 'template' files (*.gst) to interface with the model input files and 'instruction' files (*.gsi) to interface with the model output files. A control file (*.gsc) instructs the GIS how to utilize the template and instruction files to control the model. Template file, instruction file, and control file concepts are discussed below.

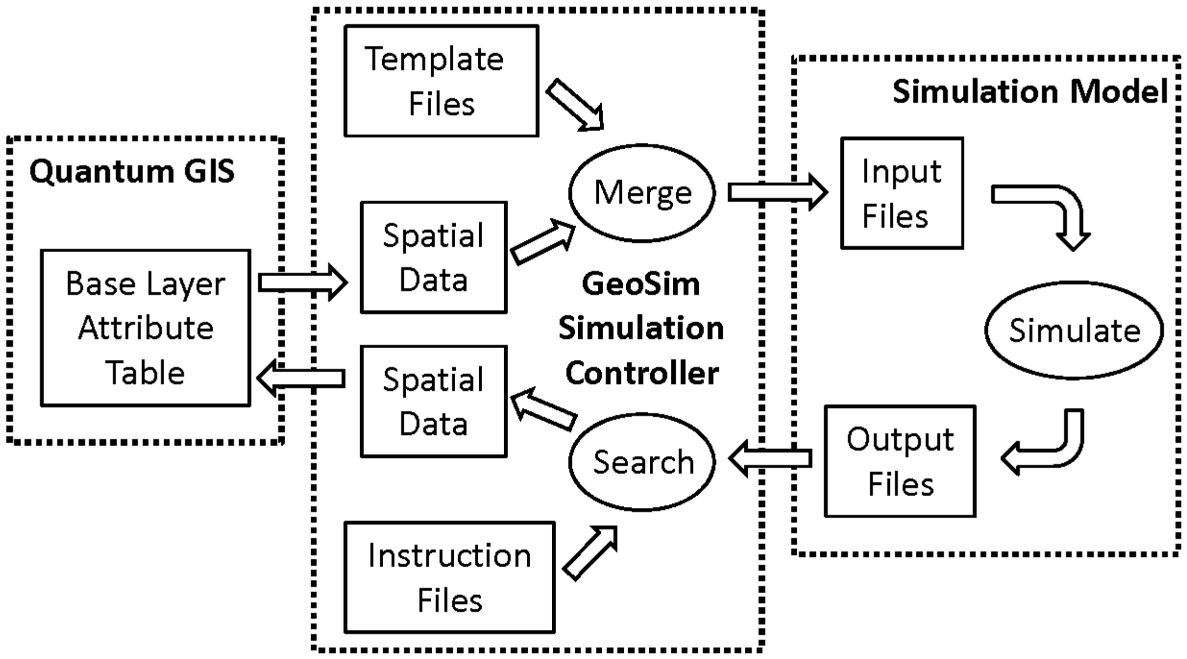
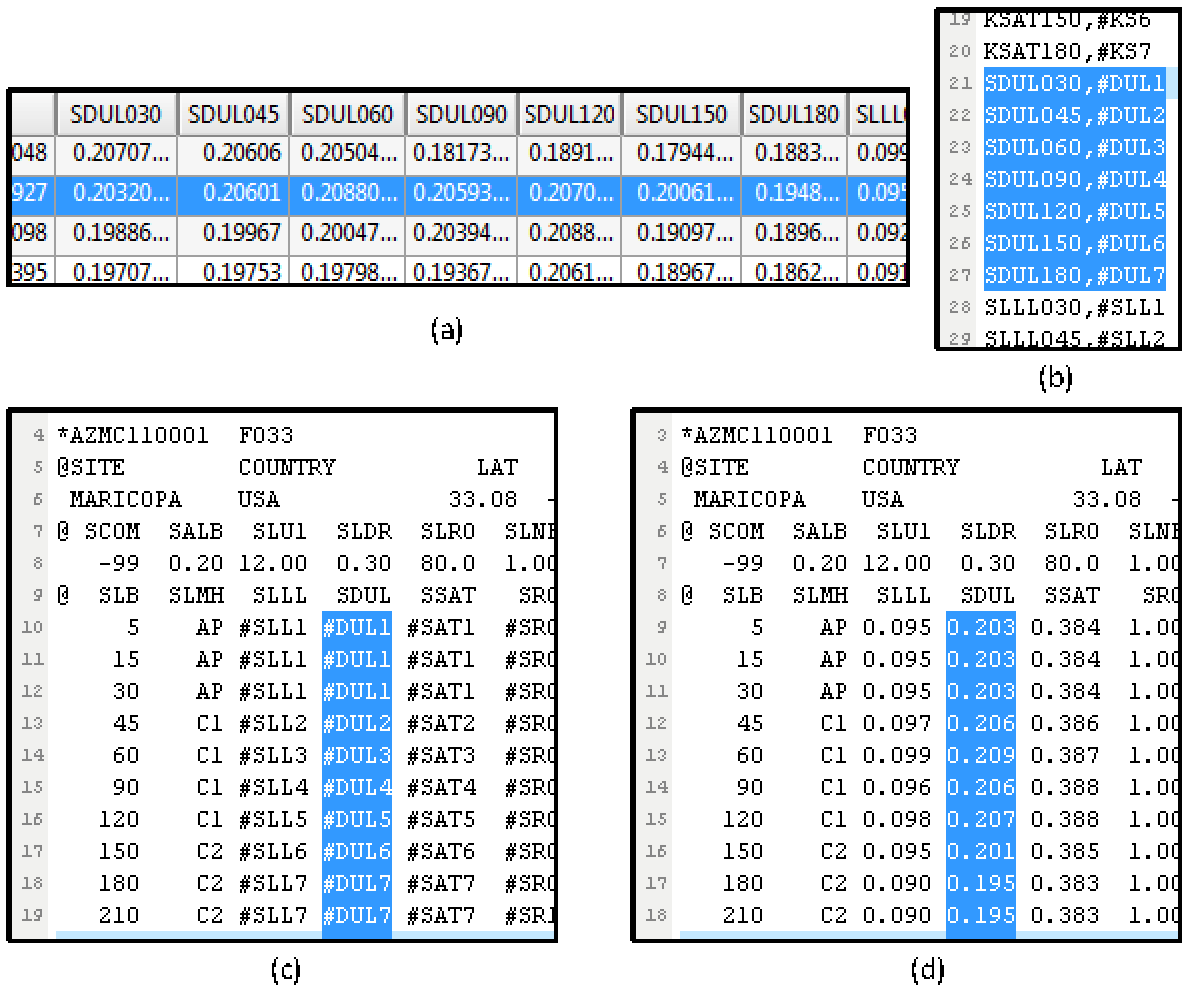


Figure 3. The Simulation Controller tool within Geospatial Simulation (GeoSim) uses template files to interface the geospatial data from the base layer polygon shapefile with the simulation model input files. It uses instruction files to search model output files for the geospatial data to be returned to the base layer.

Template Files
A template file (*.gst) must be created for each model input file that receives spatial data from the GIS. The template file is essentially a replicate of the actual input file that the model will read. Therefore, template files are highly specific to the model to be implemented. Within the template file, a 'unique code' or a unique combination of characters, numerals, symbols, and spaces should be included at the location where Geospatial Simulation will place a geospatial data value. This unique code should not appear anywhere else in any of the model input files other than where geospatial data values should be written. Prior to running a model for a spatial unit, Geospatial Simulation will search the template files for each unique code and overwrite the unique code with a data value for that spatial unit. The control file provides the relationship between each unique code and its respective data field in the GIS database (Fig. 4).



	SDUL030	SDUL045	SDUL060	SDUL090	SDUL120	SDUL150	SDUL180	SLLL
048	0.20707...	0.20606	0.20504...	0.18173...	0.1891...	0.17944...	0.1883...	0.099
927	0.20320...	0.20601	0.20880...	0.20593...	0.2070...	0.20061...	0.1948...	0.09
098	0.19886...	0.19967	0.20047...	0.20394...	0.2088...	0.19097...	0.1896...	0.092
395	0.19707...	0.19753	0.19798...	0.19367...	0.2061...	0.18967...	0.1862...	0.091

(a)

```
KSAT150,#KS6
KSAT180,#KS7
SDUL030,#DUL1
SDUL045,#DUL2
SDUL060,#DUL3
SDUL090,#DUL4
SDUL120,#DUL5
SDUL150,#DUL6
SDUL180,#DUL7
SLLL030,#SLL1
SLLL045,#SLL2
```

(b)

```
*AZMC110001   F033
@SITE         COUNTRY               LAT
 MARICOPA     USA                 33.08  -
@ SCOM  SALB  SLU1  SLDR  SLRO  SLN
   -99  0.20 12.00  0.30  80.0  1.0
@  SLB  SLMH  SLLL  SDUL  SSAT   SR
     5    AP #SLL1 #DUL1 #SAT1  #SR
    15    AP #SLL1 #DUL1 #SAT1  #SR
    30    AP #SLL1 #DUL1 #SAT1  #SR
    45    C1 #SLL2 #DUL2 #SAT2  #SR
    60    C1 #SLL3 #DUL3 #SAT3  #SR
    90    C1 #SLL4 #DUL4 #SAT4  #SR
   120    C1 #SLL5 #DUL5 #SAT5  #SR
   150    C2 #SLL6 #DUL6 #SAT6  #SR
   180    C2 #SLL7 #DUL7 #SAT7  #SR
   210    C2 #SLL7 #DUL7 #SAT7  #SR
```

(c)

```
*AZMC110001   F033
@SITE         COUNTRY               LAT
 MARICOPA     USA                 33.08  -
@ SCOM  SALB  SLU1  SLDR  SLRO  SLN
   -99  0.20 12.00  0.30  80.0  1.0
@  SLB  SLMH  SLLL  SDUL  SSAT   SR
     5    AP 0.095 0.203 0.384  1.0
    15    AP 0.095 0.203 0.384  1.0
    30    AP 0.095 0.203 0.384  1.0
    45    C1 0.097 0.206 0.386  1.0
    60    C1 0.099 0.209 0.387  1.0
    90    C1 0.096 0.206 0.388  1.0
   120    C1 0.098 0.207 0.388  1.0
   150    C2 0.095 0.201 0.385  1.0
   180    C2 0.090 0.195 0.383  1.0
   210    C2 0.090 0.195 0.383  1.0
```

(d)

Figure 4. Excerpts from a) a Quantum GIS attribute table, b) a control file relating attribute names to unique character codes, c) a template file using the unique codes to denote locations for writing data, and d) a model input file with data passed from the GIS database. These excerpts demonstrate the transfer of drained upper limit (DUL) soil parameters at depths of 30, 45, 60, 90, 120, 150, and 180 cm to the model input file for a common crop growth model.

An example of a template file used to create a soil input file for a common crop growth model demonstrates how a template file is constructed (Example 1). The first line of the file should contain the following text: Geospatial Simulation Template (GST) File. Subsequent lines replicate the actual data file expected by the model with unique codes as place holders for data that Geospatial Simulation will obtain from the GIS database. For this particular example, all the unique codes begin with a '#' symbol. However, any unique code can be used as long as the unique string of characters is found only where geospatial data should be written. There are many unique codes in this example (i.e., #SLL1, #SAT4, #SRG9, etc.), each serving as a place holder for various soil hydraulic parameters at different soil depths. Notice that unique codes can be repeated at more than one location in the file. Geospatial Simulation will overwrite the unique code anywhere in the file that it is found. Utilizing the information provided in the template files and the control file (discussed below), Geospatial Simulation will create the model's soil input file for a given spatial unit by using data from the GIS database to overwrite the unique codes (Example 2).

```
Geospatial Simulation Template (GST) File
*SOILS : Maricopa Agricultural Center, Maricopa, Phoenix, Arizona

*AZMC110001  F033
@SITE        COUNTRY          LAT     LONG SCSFAMILY
 MARICOPA    USA            33.08  -111.97
@ SCOM  SALB  SLU1  SLDR  SLRO  SLNF  SLPF  SMHB  SMPX  SMKE
   -99  0.20 12.00  0.30  80.0  1.00  1.00 IB001 IB001 IB001
@  SLB  SLMH  SLLL  SDUL  SSAT   SRGF SSKS  SBDM  SLOC  SLCL  SLSI  SLCF  SLNI
     5    AP #SLL1 #DUL1 #SAT1  #SRG1 #KS1  #BD1  0.58  -99.  -99.  -99.  -99.
    15    AP #SLL1 #DUL1 #SAT1  #SRG2 #KS1  #BD1  0.58  -99.  -99.  -99.  -99.
    30    AP #SLL1 #DUL1 #SAT1  #SRG3 #KS1  #BD1  0.58  -99.  -99.  -99.  -99.
    45    C1 #SLL2 #DUL2 #SAT2  #SRG4 #KS2  #BD2  0.17  -99.  -99.  -99.  -99.
    60    C1 #SLL3 #DUL3 #SAT3  #SRG5 #KS3  #BD3  0.17  -99.  -99.  -99.  -99.
    90    C1 #SLL4 #DUL4 #SAT4  #SRG6 #KS4  #BD4  0.17  -99.  -99.  -99.  -99.
   120    C1 #SLL5 #DUL5 #SAT5  #SRG7 #KS5  #BD5  0.17  -99.  -99.  -99.  -99.
   150    C2 #SLL6 #DUL6 #SAT6  #SRG8 #KS6  #BD6  0.17  -99.  -99.  -99.  -99.
   180    C2 #SLL7 #DUL7 #SAT7  #SRG9 #KS7  #BD7  0.17  -99.  -99.  -99.  -99.
   210    C2 #SLL7 #DUL7 #SAT7  #SR10 #KS7  #BD7  0.17  -99.  -99.  -99.  -99.
```

Example 1. A template file for a soil input file of a crop growth model. For this particular case, all the unique codes begin with a '#' symbol.

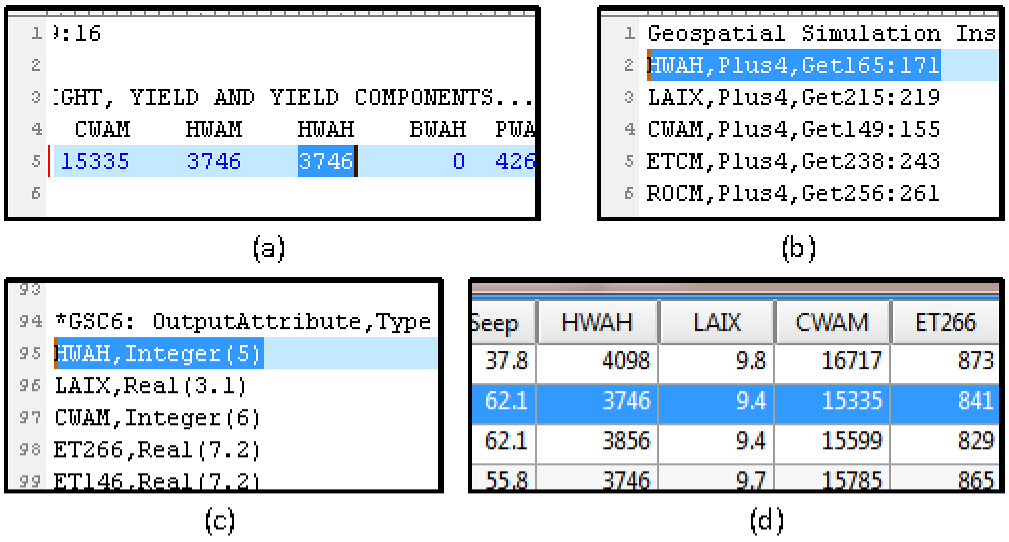
```
*SOILS : Maricopa Agricultural Center, Maricopa, Phoenix, Arizona

*AZMC110001  F033
@SITE        COUNTRY          LAT     LONG SCSFAMILY
 MARICOPA    USA            33.08  -111.97
@ SCOM  SALB  SLU1  SLDR  SLRO  SLNF  SLPF  SMHB  SMPX  SMKE
   -99  0.20 12.00  0.30  80.0  1.00  1.00 IB001 IB001 IB001
@  SLB  SLMH  SLLL  SDUL  SSAT   SRGF SSKS  SBDM  SLOC  SLCL  SLSI  SLCF  SLNI
     5    AP 0.124 0.245 0.405  0.884 0.39  1.58  0.58  -99.  -99.  -99.  -99.
    15    AP 0.124 0.245 0.405  0.824 0.39  1.58  0.58  -99.  -99.  -99.  -99.
    30    AP 0.124 0.245 0.405  0.734 0.39  1.58  0.58  -99.  -99.  -99.  -99.
    45    C1 0.124 0.243 0.403  0.644 0.42  1.58  0.17  -99.  -99.  -99.  -99.
    60    C1 0.124 0.241 0.401  0.554 0.46  1.59  0.17  -99.  -99.  -99.  -99.
    90    C1 0.098 0.204 0.384  0.374 0.94  1.63  0.17  -99.  -99.  -99.  -99.
   120    C1 0.091 0.193 0.383  0.194 1.19  1.64  0.17  -99.  -99.  -99.  -99.
   150    C2 0.087 0.188 0.384  0.014 1.33  1.63  0.17  -99.  -99.  -99.  -99.
   180    C2 0.092 0.197 0.385  0.000 1.09  1.63  0.17  -99.  -99.  -99.  -99.
   210    C2 0.092 0.197 0.385  0.000 1.09  1.63  0.17  -99.  -99.  -99.  -99.
```

Example 2. Soil input file created by Geospatial Simulation using the template file in Example 1.

Instruction Files

An instruction file (*.gsi) must be created for each model output file that contains spatial data to be passed to the GIS database. The instruction file essentially tells the GIS how to read the model output file and extract the data values. Three commands are available to provide the instruction, including Plus, Find, and Get. Plus is used to move forward a given number of lines. Find is used to find a given set of unique characters. Get is used to acquire the data value. The control file provides the relationship between the attribute (field) in the GIS database and the type of data value that is read from the output file (Fig. 5).



```
?:16

IGHT, YIELD AND YIELD COMPONENTS...
  CWAM      HWAM      HWAH      BWAH   PWA
15335      3746      3746         0   426
```

```
Geospatial Simulation Ins
HWAH,Plus4,Get165:171
LAIX,Plus4,Get215:219
CWAM,Plus4,Get149:155
ETCM,Plus4,Get238:243
ROCM,Plus4,Get256:261
```

```
*GSC6: OutputAttribute,Type
HWAH,Integer(5)
LAIX,Real(3.1)
CWAM,Integer(6)
ET266,Real(7.2)
ET146,Real(7.2)
```

Seep	HWAH	LAIX	CWAM	ET266
37.8	4098	9.8	16717	873
62.1	3746	9.4	15335	841
62.1	3856	9.4	15599	829
55.8	3746	9.7	15785	865

Figure 5. Excerpts from a) a crop model output file, b) an instruction file providing instructions to read data for the HWAH attribute (crop yield, kg ha^{-1}) from the model output file, c) a control file relating the attribute name to the data type expected by the GIS, and d) a Quantum GIS attribute table showing the HWAH value passed from the output file. After reading the model output file, the instruction file uses two commands to get the HWAH value: "Plus4" moves down four lines in the file and "Get165:171" retrieves the characters between cursor positions 165 and 171.

An example of an instruction file used to read the evapotranspiration output file for a common crop growth model demonstrates how an instruction file is constructed (Example 3). The first line of the file should contain the following text: Geospatial Simulation Instruction (GSI) File. Subsequent lines provide the commands for locating a data value within the model output file. Data values are obtained in the order given in the instruction file. Commands for one data value should all reside on one line, separated by commas. The first item on each line gives the attribute (field) name in the GIS database where the data value will be stored. The file in Example 3 will read two data values, one for the "ET146" attribute and one for the "ET266" attribute. Subsequent items give "Plus" and/or "Find" commands as needed to reach the appropriate line in the file. Each "Plus" command is followed by an integer number that gives the number of lines to move forward. Each "Find" command is followed by a string of characters that will be located in the file. The goal is to use "Plus" and "Find" commands to locate the appropriate line in the model output file. For consecutive "Plus" and "Find" commands for the same data value, the algorithm searches from the line determined by the previous command. On opening the file, the algorithm is set to read the first line. Thus, if a data value must be read from the first line in the file, no "Plus" or "Find" commands are needed. Once the appropriate line is located, a "Get" command is used to obtain the data value. Each "Get" command is followed by a pair of integers, separated by a colon, which give the cursor positions for the data value to be read. Cursor positions are counted from the left hand side of the file, beginning with zero. Think of it as counting the spaces between characters rather than the characters themselves. The pair of integers for the "Get" command should provide the cursor positions that straddle the data value of interest. After finding one data value, the algorithm begins looking for the next data value from the first line in the file. Thus, if two data values are to be obtained from the same line, the "Plus" or "Find" commands will need to be repeated for the second data value. In Example 3, the algorithm will seek forward 54 lines from the first line in the file and then get the "ET146" data value between cursor positions 101 and 109. It will then start over at the first line of the file and seek forward until it finds "2009 266" as a text string. From that line, it will get the "ET266" data value between cursor positions 101 and 109. These data values will be stored in the "ET146" and "ET266" fields for the current feature within the GIS database.

```
Geospatial Simulation Instruction (GSI) File
ET146,Plus54,Get101:109
ET266,Find2009 266,Get101:109
```

Example 3. An instruction file for reading an evapotranspiration output file of a crop growth model.

Control File

The control file (*.gsc) instructs the GIS how to use the template and instruction files to control a simulation model for unique spatial units. Whereas the template and instruction files are unique depending on the simulation model to be implemented, the format of the control file is specific to the Geospatial Simulation algorithm.

An example of the control file used to operate a common crop growth model is given in the lengthy Example 4. The first line of the file should contain the following text: Geospatial Simulation Control (GSC) File. Seven key character strings are used to denote sections within the file. Table 1 provides these character strings and a summary of the purpose for each of the sections in the file. A blank line should be included between each section in the control file. The first section (*GSC1) provides the path to the model executable file. All template files, instruction files, model input and output files, and the model executable file should exist at this path. The second section (*GSC2) provides the name of the base layer polygon shapefile that contains the geospatial data to be utilized in the simulations. The third section (*GSC3) provides the names of the template files and corresponding model input files. Each set of files is given on a new line and separated by a comma. For simplicity, the name of the template file can be the name of the model input file with the '.gst' extension appended (Example 4). The fourth section (*GSC4) provides the names of the attributes (fields) in the GIS database and their corresponding unique codes that appear in the template files. Geospatial Simulation will search all template files in section '*GSC3' for all the unique codes in section '*GSC4' and overwrite the codes with the data value for the corresponding attribute. Therefore, the unique codes must be unique among all template files included in section '*GSC3'. Many of the unique codes specified in '*GSC4' in Example 4 are used to write the soil parameter data to the model's soil input file, as shown in Example 1 and Example 2. The fifth section (*GSC5) provides the names of the instruction files and corresponding model output files. Each set of files is given on a new line and separated by a comma. For simplicity, the name of the instruction file can be the name of the model output file with the '.gsi' extension appended (Example 4). The sixth section (*GSC6) provides the names of the attributes (fields) in the GIS database and their corresponding data type. If the attribute already exists in the database, its values will be overwritten. Otherwise, a new field will be appended to the existing database. Three data types can be specified, including Integer, Real, and String. Following each data type specification is a numeric value in parentheses. For Integer and String, the numeric value is an integer which specifies the field length. For Real, the numeric value is a decimal number, where the integral number gives the field length and the decimal number gives the decimal precision. The seventh section (*GSC7) provides a command line text string for running the simulation model.

```
Geospatial Simulation Control (GSC) File

*GSC1: ModelDirectory
C:\Users\kthorp\Documents\USDA-ALARC\02-Cotton\2009Cotton\CROPGRO-Cotton\

*GSC2: BaseLayer
09F033_HarvPolys12

*GSC3: TemplateFile,InputFile
AZ.sol.gst,AZ.sol
AZMC0901.COX.gst,AZMC0901.COX

*GSC4: InputAttribute,Code
KSAT030,#KS1
KSAT045,#KS2
KSAT060,#KS3
KSAT090,#KS4
KSAT120,#KS5
KSAT150,#KS6
KSAT180,#KS7
SDUL030,#DUL1
SDUL045,#DUL2
SDUL060,#DUL3
SDUL090,#DUL4
SDUL120,#DUL5
SDUL150,#DUL6
SDUL180,#DUL7
SLLL030,#SLL1
SLLL045,#SLL2
SLLL060,#SLL3
SLLL090,#SLL4
SLLL120,#SLL5
SLLL150,#SLL6
SLLL180,#SLL7
SSAT030,#SAT1
SSAT045,#SAT2
SSAT060,#SAT3
SSAT090,#SAT4
SSAT120,#SAT5
SSAT150,#SAT6
SSAT180,#SAT7
SBDM030,#BD1
SBDM045,#BD2
SBDM060,#BD3
SBDM090,#BD4
SBDM120,#BD5
SBDM150,#BD6
SBDM180,#BD7
SRGF005,#SRG1
SRGF015,#SRG2
SRGF030,#SRG3
SRGF045,#SRG4
SRGF060,#SRG5
SRGF090,#SRG6
SRGF120,#SRG7
SRGF150,#SRG8
SRGF180,#SRG9
SRGF210,#SR10
IrrDOY1,#D1
IrrRate1,#R1
```

```
IrrDOY2,#D2
IrrRate2,#R2
IrrDOY3,#D3
IrrRate3,#R3
IrrDOY4,#D4
IrrRate4,#R4
IrrDOY5,#D5
IrrRate5,#R5
IrrDOY6,#D6
IrrRate6,#R6
IrrDOY7,#D7
IrrRate7,#R7
IrrDOY8,#D8
IrrRate8,#R8
IrrDOY9,#D9
IrrRate9,#R9
EFIR,#EIR
SH2O005,#ICW1
SH2O015,#ICW2
SH2O030,#ICW3
SH2O045,#ICW4
SH2O060,#ICW5
SH2O090,#ICW6
SH2O120,#ICW7
SH2O150,#ICW8
SH2O180,#ICW9
SH2O210,#IC10

*GSC5: InstructionFile,OutputFile
Summary.OUT.gsi,Summary.OUT
ET.OUT.gsi,ET.OUT
SoilWat.OUT.gsi,SoilWat.OUT

*GSC6: OutputAttribute,Type
HWAH,Integer(5)
LAIX,Real(3.1)
CWAM,Integer(6)
ET266,Real(7.2)
ET146,Real(7.2)
SP266,Real(7.2)
SP146,Real(7.2)
ROCM,Integer(5)

*GSC7: CommandLine
dscsm045.exe B DSSBatch.v45
```

Example 4. A control file for running a common crop growth model within Geospatial Simulation.

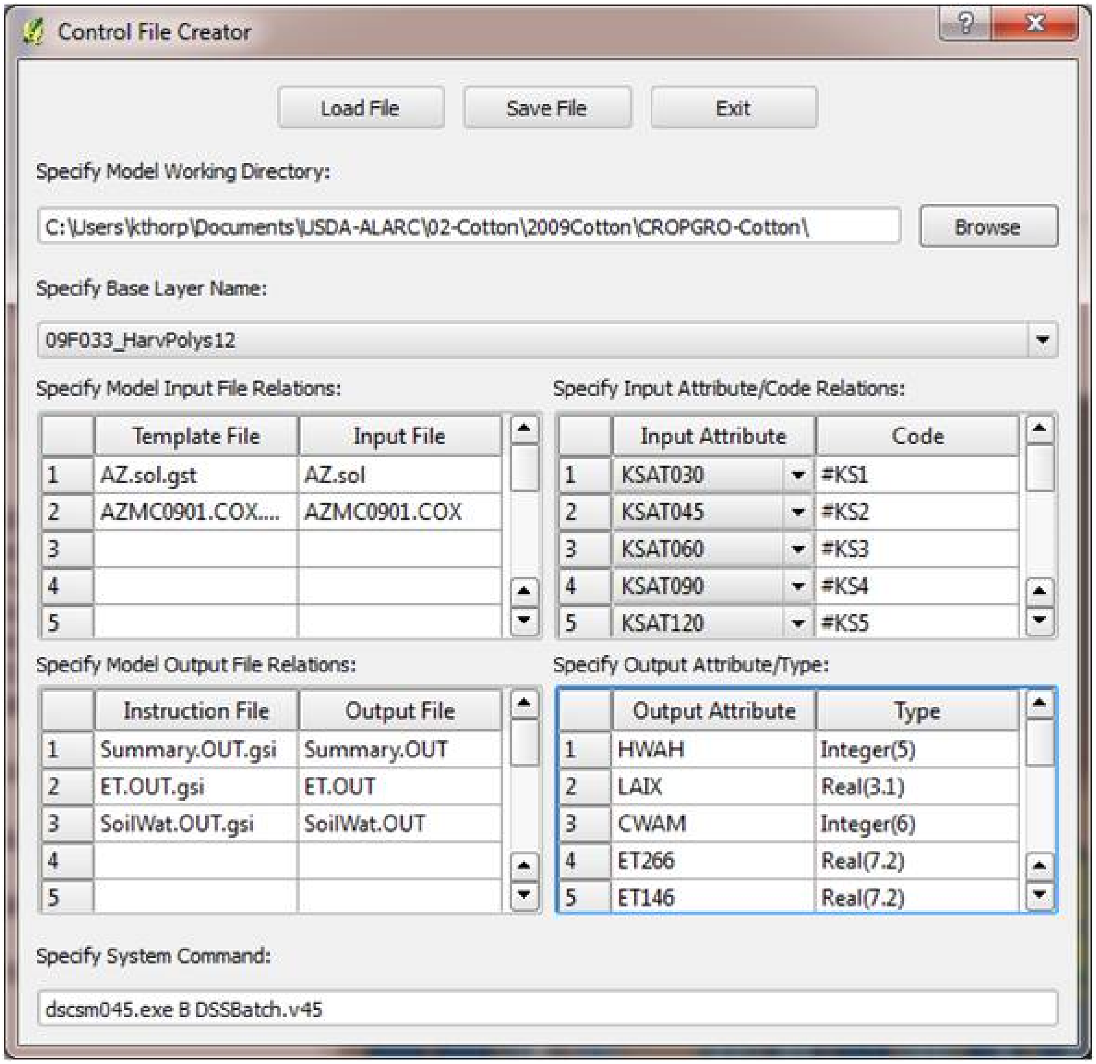
Table 1. Title and purpose of sections in the control file.

Section Title	Section Purpose
*GSC1: ModelDirectory	Provide the path to the model executable file
*GSC2: BaseLayer	Provide the name of the base layer containing the geospatial data
*GSC3: TemplateFile,InputFile	Provide the names of the template files and the corresponding model input file names
*GSC4: InputAttribute,Code	Provide the names of the attributes (fields) in the GIS database and their corresponding unique codes that appear in the template files
*GSC5: InstructionFile,OutputFile	Provide the names of the instruction files and the corresponding model output file names
*GSC6: OutputAttribute,Type	Provide the names of the attributes (fields) in the GIS database and the corresponding data format.
*GSC7: CommandLine	Provide the command line string required to run the simulation model

Tool #3: Control File Creator

The Control File Creator (Fig. 6) is used to generate the control file (*.gsc), which specifies how geospatial data is passed from the GIS database to the model input files and from the model output files to the GIS database. To use the tool, complete the following steps in the Quantum GIS environment.

1. Select Plugins->Geospatial Simulation->Control File Creator
2. To load an existing control file, click 'Load File' and select the file to be loaded.
3. Specify the model working directory by clicking 'Browse' and navigating to the path of the model executable.
4. Specify the base layer name by selecting the base layer containing the geospatial data. The combo box is populated with all polygon shapefiles in the current workspace.
5. Specify the model input file relations by typing the names of the template files and corresponding model input files.
6. Specify the input attribute and unique code relations. Double clicking an item in the 'Input Attribute' column will give a list of attribute (field) names in the selected base layer. Type the unique code for each input attribute.
7. Specify the model output file relations by typing the names of the instruction files and the corresponding model output files.
8. Specify the output attribute and types by typing the name of the output attributes and corresponding data type. If the output attribute already exists in the database, data values will be overwritten. Otherwise, a new attribute will be appended to the existing database. Options for data type are Integer, Real, and String. Each type specification is followed by a numeric value in parenthesis. For Integer and String, the numeric value is an integer giving the field length. For Real, the numeric value is a decimal value, where the integral number gives the field length and the decimal number gives the decimal precision.
9. Specify the system command by typing the command line text used to run the simulation model.
10. Save the control file to the hard disk by clicking 'Save File'.



Specify Model Working Directory:

C:\Users\kthorp\Documents\USDA-ALARC\02-Cotton\2009Cotton\CROPGRO-Cotton\

Specify Base Layer Name:

09F033_HarvPolys12

Specify Model Input File Relations:

	Template File	Input File
1	AZ.sol.gst	AZ.sol
2	AZMC0901.COX....	AZMC0901.COX
3		
4		
5		

Specify Input Attribute/Code Relations:

	Input Attribute	Code
1	KSAT030	#KS1
2	KSAT045	#KS2
3	KSAT060	#KS3
4	KSAT090	#KS4
5	KSAT120	#KS5

Specify Model Output File Relations:

	Instruction File	Output File
1	Summary.OUT.gsi	Summary.OUT
2	ET.OUT.gsi	ET.OUT
3	SoilWat.OUT.gsi	SoilWat.OUT
4		
5		

Specify Output Attribute/Type:

	Output Attribute	Type
1	HWAH	Integer(5)
2	LAIX	Real(3.1)
3	CWAM	Integer(6)
4	ET266	Real(7.2)
5	ET146	Real(7.2)

Specify System Command:

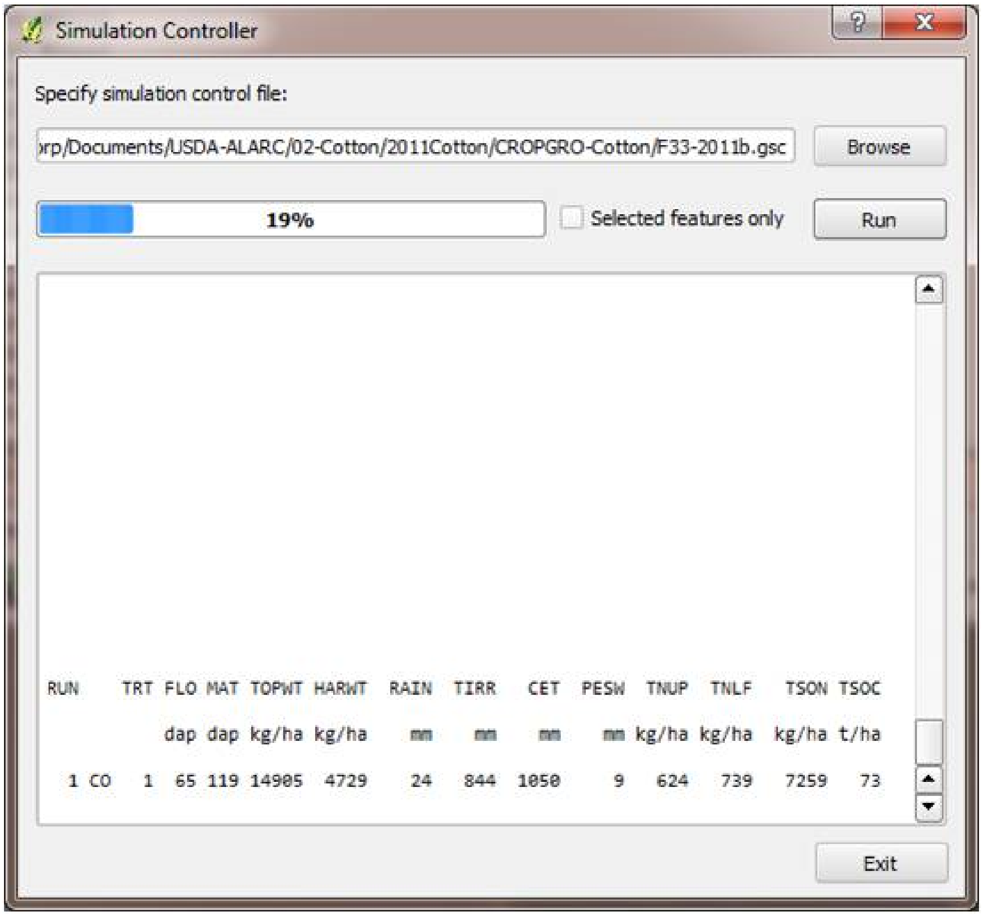
```
dscsm045.exe B DSSBatch.v45
```

Figure 6. The graphic user interface for the Control File Creator tool.

<u>Tool #4: Simulation Controller</u>

After setting up the template files, instruction files, and control file, the Simulation Controller (Fig. 7) can be implemented to conduct the simulations for all geospatial units in the base layer shapefile. Alternatively, the tool will also run simulations only for selected features in the base layer. To use the tool, complete the following steps in the Quantum GIS environment.

1. Select Plugins->Geospatial Simulation->Simulation Controller
2. Click Browse and open the control file.
3. To run only for selected features, check 'Selected features only.' Otherwise, leave unchecked.
4. Click 'Run'
5. Standard output from the model, if any, will be printed in the simulation window.



orp/Documents/USDA-ALARC/02-Cotton/2011Cotton/CROPGRO-Cotton/F33-2011b.gsc

```
RUN     TRT FLO MAT TOPWT HARWT  RAIN  TIRR   CET  PESW  TNUP  TNLF   TSON TSOC
                dap dap kg/ha kg/ha    mm    mm    mm    mm kg/ha kg/ha  kg/ha t/ha
  1 CO    1  65 119 14905  4729    24   844  1050     9   624   739   7259   73
```

Figure 7. The graphic user interface for the Simulation Controller tool.

Simulation Optimization

It is often necessary to calibrate a simulation model uniquely for the conditions at a given location. Geospatial Simulation provides an optimization tool for this purpose. The optimization algorithm utilizes the control file (and thus the instruction files and template files as well) to run the simulation model many times for each geospatial unit. The objective is to adjust model input parameters to minimize error between an observed and simulated quantity. A optimization file must be created to instruct the GIS how to conduct the optimization. The optimization routines represent an additional layer of simulation complexity beyond simply running the simulation model for a geospatial dataset. Thus, it is wise to first set up Geospatial Simulation to run a simulation model (by developing a control file) before attempting to optimize the model. Geospatial Simulation implements a 'simulated annealing' optimization routine.

<u>Optimization File</u>

An optimization file (*.gso) instructs the GIS how to conduct the optimization routine. An example of the optimization file used to optimize the crop coefficient and leaf growth parameters of a common crop growth model is given in Example 5. The first line of the file should contain the following text: Geospatial Simulation Optimization (GSO) File. Five key character strings are used to denote sections within the file. Table 2 provides these character strings and a summary of the purpose for each of the sections in the file. A blank like should be present between each section in the optimization file. The first section (*GSO1) provides the path to the control file, which specifies how to use the template files and instruction files to pass data to model input files and to retrieve data from the model output files. The second section (*GSO2) provides the name of the base layer polygon shapefile that contains the geospatial data to be utilized in the simulations. The third section (*GSO3) provides the parameters to be optimized, the initial values for the parameters, and their upper and lower bounds. Each parameter is given on a new line and associated data is separated by commas. The first entry provides the attribute (field) name of the parameter to be optimized. In most cases, the attribute must also appear in the control file, such that the parameter values can be passed to the model. The attribute name is followed by the initial parameter value, the lower bound value, the upper bound value, and a code value. The code value is used for cases where the optimization occurs collectively over a group of spatial features. In this case, it may be necessary to optimize the same parameter, but uniquely for each spatial feature. The code value is used to specify the parameters in the order in which the model simulations are conducted. Example 6 demonstrates the use of the code value to optimize the 'PercentVol' parameter (which specifies the percent volume of total irrigation water to be distributed among 20 spatial zones) by minimizing the error in spatial yield simulation among the zones. If the model optimization is to be performed independently for each spatial feature or if the parameter does not need to be uniquely optimized for spatial features in a group, a code value of '0' should be used (Example 5). The fourth section (*GSO4) gives the attribute (field) names in the GIS database that contain the measured and simulated values to be compared during the optimization process and a factor value. The factor value is multiplied by the resulting difference between measured and simulated values. It is useful for optimizations where two or more measured versus simulated quantities have errors that vary by an order of magnitude or more, due possibly to different units for the values. The factor value is multiplied by the error term for each quantity and can thus be set such that the errors have reasonably similar orders of magnitude. This will improve the performance of the optimizer. If all the errors have the same units or reasonably similar magnitudes, the factor can be set to one. Measured values to be compared to simulated values during optimization can likely be obtained using the geoprocessing tools described above. Simulated values will need to be passed from a model output file to the GIS database using an instruction file. Examples 5 demonstrates an optimization that simultaneously minimizes error between measured and simulated cotton lint yield and measured and simulated cotton evapotranspiration. The fifth section (*GSO5) provides the eleven input parameters needed for the simulated annealing optimization algorithm. These parameters are further discussed below.

```
Geospatial Simulation Optimization (GSO) File

*GSO1: ControlFile
C:/Users/kthorp/Documents/USDA-ALARC/02-Cotton/2009Cotton/CROPGRO-Cotton/F33-
2009b.gsc

*GSO2: BaseLayer
09F033_HarvPolys27

*GSO3: Parameter,Initial,LB,UB,Code
EORATIO,1.25,1.25,1.65,0
SLAVR,120.0,120.0,220.0,0
SIZLF,240.0,240.0,340.0,0

*GSO4: Measured,Simulated,Factor
LintSeed,HWAH,0.0002
ObsETAdj,SimET,0.0011

*GSO5: AnnealingParameters
10.0
None
```

```
10
0.005
None
20
None
1.0
1.0
1.0
100
```

Example 5. An optimization file for optimization of a common crop growth model within Geospatial Simulation.

```
Geospatial Simulation Optimization (GSO) File

*GSO1: ControlFile
C:/Users/kthorp/Documents/USDA-ALARC/02-Cotton/2009Cotton/CROPGRO-Cotton/F33-
2009.gsc

*GSO2: BaseLayer
09F033_HarvPolys8

*GSO3: Parameter,Initial,LB,UB,Code
PercentVol,0.0286,0.01,0.04,1
PercentVol,0.0286,0.01,0.04,2
PercentVol,0.0286,0.01,0.04,3
PercentVol,0.0286,0.01,0.04,4
PercentVol,0.0286,0.01,0.04,5
PercentVol,0.0286,0.01,0.04,6
PercentVol,0.0286,0.01,0.04,7
PercentVol,0.0286,0.01,0.04,8
PercentVol,0.0286,0.01,0.04,9
PercentVol,0.0286,0.01,0.04,10
PercentVol,0.0286,0.01,0.04,11
PercentVol,0.0286,0.01,0.04,12
PercentVol,0.0286,0.01,0.04,13
PercentVol,0.0286,0.01,0.04,14
PercentVol,0.0286,0.01,0.04,15
PercentVol,0.0286,0.01,0.04,16
PercentVol,0.0286,0.01,0.04,17
PercentVol,0.0286,0.01,0.04,18
PercentVol,0.0286,0.01,0.04,19
PercentVol,0.0286,0.01,0.04,20

*GSO4: Measured,Simulated,Factor
LintSeed,HWAH,1

*GSO5: AnnealingParameters
10.0
None
10
0.0001
None
None
None
1.0
1.0
1.0
100000.0
```

Example 6. An optimization file for conducting a group optimization, where the same parameter is optimized to different values for a series of spatial features.

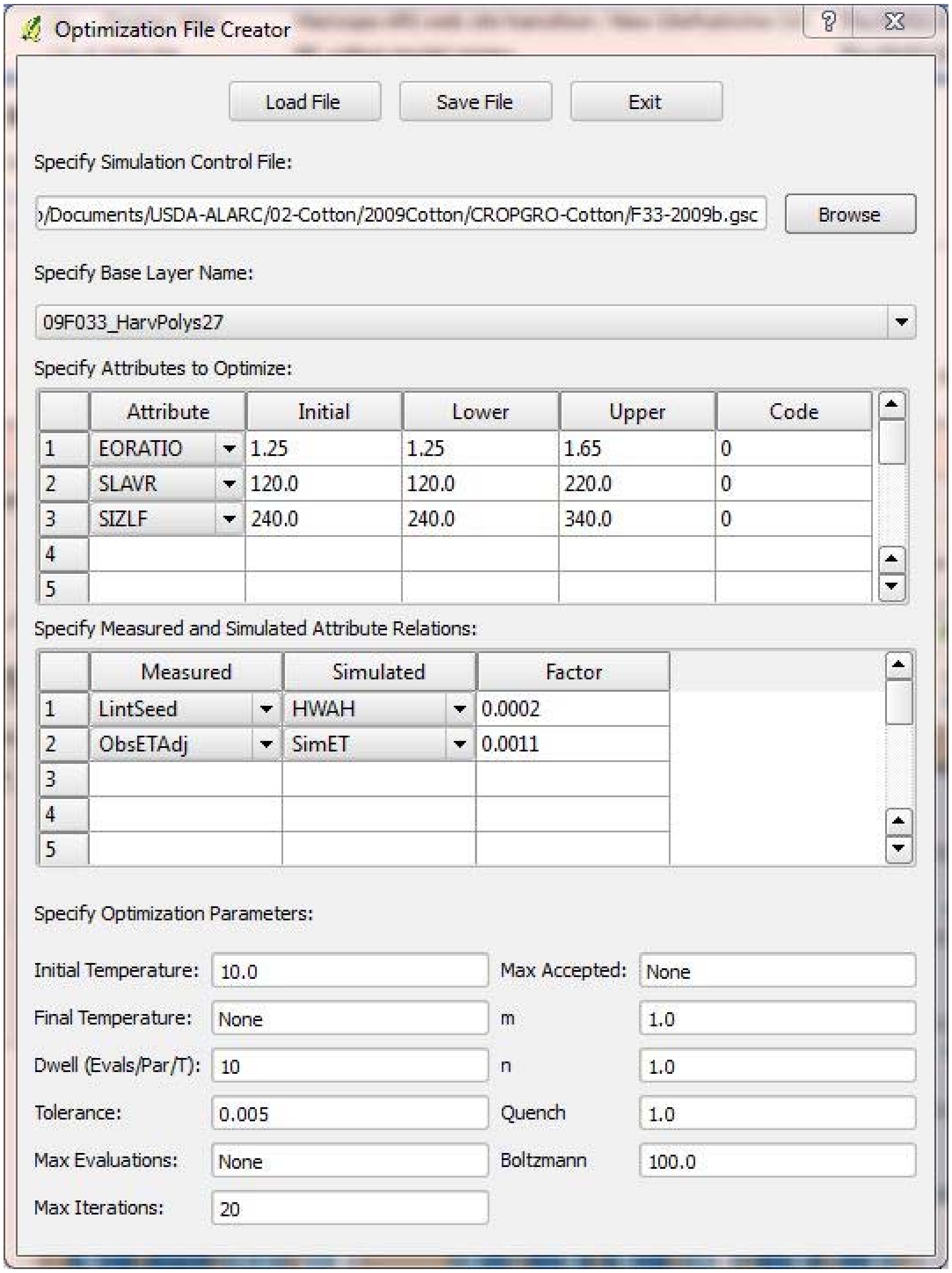
Table 2. Title and purpose of sections in the optimization file.

Section Title	Section Purpose
*GSC1: ControlFile	Provide the path to the control file
*GSC2: BaseLayer	Provide the name of the base layer containing the geospatial data
*GSC3: Parameter,Initial,LB,UB,Code	Provide the names of the attributes (fields) containing values to be adjusted using optimization and give the initial value, lower bound, and upper bound for the parameters
*GSC4: Measured,Simulated,Factor	Provide the names of the attributes (fields) in the GIS database that contain the measured and simulated data to be compared during optimization
*GSC5: AnnealingParameters	Provide the values for the input parameters to the simulated annealing algorithm

<u>Tool #5: Optimization File Creator</u>

The Optimization File Creator (Fig. 8) is used to generate the optimization file (*.gso), which instructs the GIS how to conduct the optimization routine. Prior to conducting optimizations, template files, instruction files, and a control file must be developed. To use the tool, complete the following steps in the Quantum GIS environment.

1. Select Plugins->Geospatial Simulation->Optimization File Creator
2. To load an existing control file, click 'Load File' and select the file to be loaded.
3. Specify the control file by clicking 'Browse' and navigating to the path of the control file.
4. Specify the base layer name by selecting the base layer containing the geospatial data. The combobox is populated with all polygon shapefiles in the current workspace.
5. Specify the input attributes to be optimized, the initial value, the lower bound, the upper bound, and the code value. Double clicking an item in the 'Attribute' column will give a combo box of attribute (field) names in the selected base layer. The code value is used for group optimizations as described above.
6. Specify the attributes that contain the measured and simulated data to be compared during the optimization. Double clicking an item in the table will give a combo box of attribute (field) names in the selected base layer. Specify the factor value to multiply by error term, if needed. Default factor value is 1.
7. Specify the parameter values for simulated annealing optimization.
8. Save the control file to the hard disk by clicking 'Save File'.


Optimization File Creator

Specify Simulation Control File:

)/Documents/USDA-ALARC/02-Cotton/2009Cotton/CROPGRO-Cotton/F33-2009b.gsc

Specify Base Layer Name:

09F033_HarvPolys27

Specify Attributes to Optimize:

	Attribute	Initial	Lower	Upper	Code
1	EORATIO	1.25	1.25	1.65	0
2	SLAVR	120.0	120.0	220.0	0
3	SIZLF	240.0	240.0	340.0	0
4					
5					

Specify Measured and Simulated Attribute Relations:

	Measured	Simulated	Factor
1	LintSeed	HWAH	0.0002
2	ObsETAdj	SimET	0.0011
3			
4			
5			

Specify Optimization Parameters:

Initial Temperature:	10.0	Max Accepted:	None
Final Temperature:	None	m	1.0
Dwell (Evals/Par/T):	10	n	1.0
Tolerance:	0.005	Quench	1.0
Max Evaluations:	None	Boltzmann	100.0
Max Iterations:	20		

Figure 8. The graphic user interface for the Simulation Optimization File Creator.

Simulated Annealing Optimization Algorithm

The simulated annealing algorithm used in Geospatial Simulation was adapted from that developed for the open-source 'scipy' package of the Python programming language. Simulated annealing is a global search algorithm that mimics the annealing process in metallurgy. Eleven parameters govern the performance of the algorithm, including the initial temperature, the final temperature, the dwell

(number of evaluations per parameter at each temperature), the tolerance, the maximum number of evaluations, the maximum number of iterations (cooling steps), and the maximum number of accepted evaluations. The m, n, and quench parameters affect the cooling rate, and the boltzmann parameter affects the acceptance and rejection criteria. During the course of the optimization, the algorithm tracks three parameter sets: the current set, the previous set, and the best (optimum) set. After each function evaluation, the algorithm must decide whether to accept or reject the evaluation. If the current evaluation is better than the previous evaluation, the current parameter set is accepted and the next parameter set will be calculated based on the current parameter set. If the current evaluation is not better than the previous evaluation, the algorithm uses a probabilistic function to determine acceptance or rejection. The algorithm calculates the following P value:
$P = EXP(-E/(B*T))$
where E is the error difference between the current evaluation and the previous evaluation, B is the boltzmann parameter, and T is the current temperature. The value of P will range from 0.0 to 1.0. If P is greater than a random number chosen from a uniform distribution between 0.0 and 1.0, the evaluation is accepted and the next parameter set will be calculated from the current parameter set. Otherwise, the evaluation is rejected and the next parameter set will be calculated from the previous parameter set. If the current evaluation is better than the best evaluation, the current parameter set becomes the optimum parameter set. As the temperature cools, P values will decline, and the chance of accepting an evaluation with higher error than the previous evaluation is reduced.

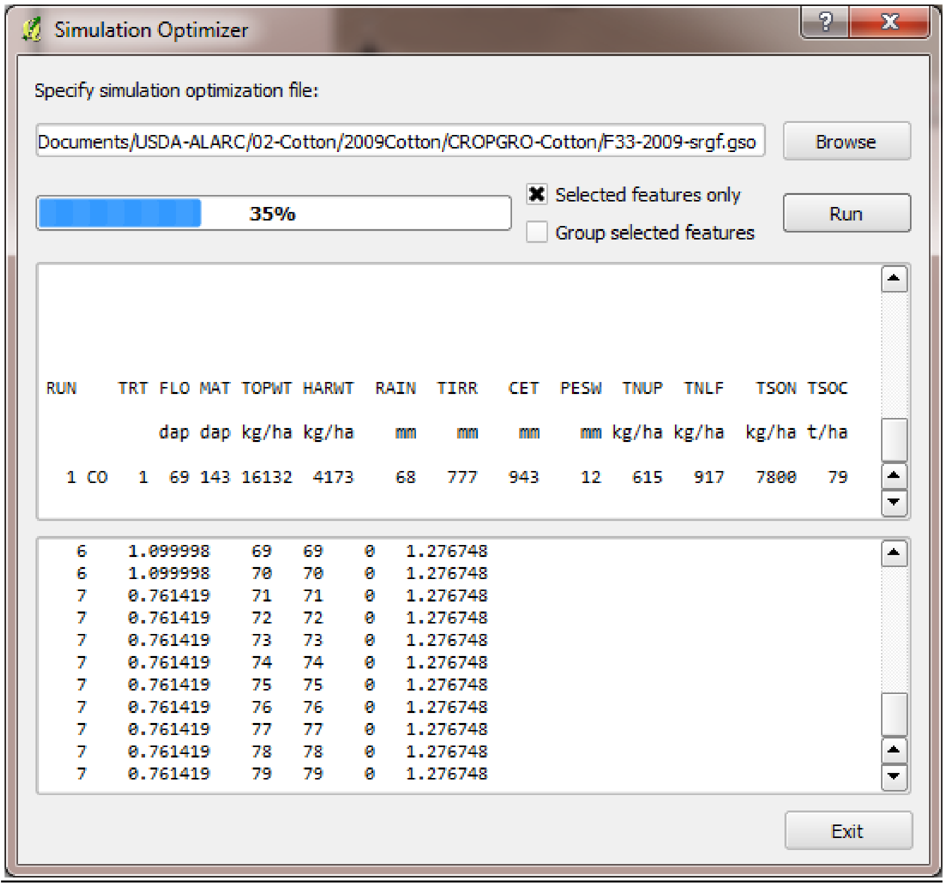
The initial temperature parameter sets the starting temperature for the algorithm. The dwell parameter specifies the number of evaluations per parameter at each temperature. Thus, the number of function evaluations at each cooling step is dwell times the number of parameters to be optimized. The dwell parameter can have a substantial effect on how long the algorithm will run for a given dataset. To calculate the new temperature at each cooling step, the algorithm uses the following equation:
$T = T0*EXP(-C*I^Q)$
where T is the new temperature, T0 is the initial temperature, C is a constant, I is the current iteration or number of cooling steps, and Q is the quench parameter. The constant, C, is calculated as
$C=M*EXP(-N*Q)$
where M and N are input parameters and Q is the quench parameter.

Many of the optimization parameters govern the stop criterion for the algorithm. If the temperature cools below the specified final temperature, the algorithm will stop. If the error of an evaluation is less than the tolerance criteria, the algorithm will stop. If the maximum number of evaluations, iterations, or accepted evaluations is exceeded, the algorithm will also stop. These parameters can be set to 'None' if their stop criterion should be ignored. After the algorithm stops, the best parameter set is passed to the GIS database, and the simulations are run once more using the best parameter set to update other simulation outputs in the database.

Tool #6: Simulation Optimizer

After setting up the optimization file, the Simulation Optimizer (Fig. 9) can be implemented to conduct the optimizations for all geospatial units in the base layer shapefile. Alternatively, the tool will run optimizations only for selected features in the base layer. The simplest approach is to conduct optimizations independently for each spatial unit. In this case, the algorithm loops through the spatial features (or selected features) while conducting optimizations separately for each feature. However, the algorithm can also handle grouped optimizations, where several spatial features are considered together. The optimization file must be properly set up for this type of optimization, as discussed above. To use the tool, complete the following steps in the Quantum GIS environment.

1. Select Plugins->Geospatial Simulation->Simulation Optimizer
2. Click Browse and open the optimization file.
3. To run only for selected features, check 'Selected features only.' Otherwise, leave unchecked.
4. To run the optimization for grouped spatial features, check 'Group selected features.' Otherwise, leave unchecked.
5. Click 'Run'
6. Standard output from the model, if any, will be printed in the simulation (top) window.
7. Output from the optimization algorithm will be printed to the optimization (bottom) window. This information will also be printed to a file with the same name as the optimization file and a '.log' extension appended.


Simulation Optimizer
Specify simulation optimization file:
Documents/USDA-ALARC/02-Cotton/2009Cotton/CROPGRO-Cotton/F33-2009-srgf.gso
Browse
35%
Selected features only
Group selected features
Run
Exit

Selected features only: [✓]
Group selected features: [ ]

```
RUN     TRT  FLO  MAT  TOPWT  HARWT   RAIN   TIRR    CET   PESW   TNUP   TNLF    TSON  TSOC

               dap  dap  kg/ha  kg/ha     mm     mm     mm     mm  kg/ha  kg/ha   kg/ha  t/ha

  1 CO    1   69  143  16132   4173     68    777    943     12    615    917    7800    79
```

```
    6     1.099998    69    69    0    1.276748
    6     1.099998    70    70    0    1.276748
    7     0.761419    71    71    0    1.276748
    7     0.761419    72    72    0    1.276748
    7     0.761419    73    73    0    1.276748
    7     0.761419    74    74    0    1.276748
    7     0.761419    75    75    0    1.276748
    7     0.761419    76    76    0    1.276748
    7     0.761419    77    77    0    1.276748
    7     0.761419    78    78    0    1.276748
    7     0.761419    79    79    0    1.276748
```

Figure 9. The graphic user interface for the Simulation Optimizer.