AMSIMOD

User's guide

Guy R. Larocque

Charles-David Lachance

Mikael Jonathan Luce

Patrick Munro

Pier-Olivier Pichette Côté

Mildred Deldago

Pascale-Nicole Lapointe

Jonathan Hamel

André Cloutier

Natural Resources Canada, Canadian Forest Service

Laurentian Forestry Centre

1055 du P.E.P.S., Québec (QC), G1V 4C7, Canada

Table of content

List of Figures

List of tables

1- Introduction

AMSIMOD (Application for the Management of SImulation MODels) is a modelling software platform that facilitates the development and management of different types of simulation models. In particular, many forest simulation models have been developed, but few of them come with utilities that facilitate the management of simulation runs and the visualization of simulation results. Several software platforms have been developed to manage models. They can be classified in three categories: (1) software frameworks, (2) modelling tools and (3) simulators. However, these applications generally do not include utilities to integrate a variety of models or display simulation results on different scales or formats.

The end-users of AMSIMOD can define projects that contain models, input and output files and, if necessary, intervention files to simulate the effects of disturbance scenarios. The objective of this user guide is to describe the different functionalities of AMSIMOD.

2- Installation of AMSIMOD

AMSIMOD can be installed anywhere on a hard disk. For instance, it can be installed on "C:\Program Files (x86)" using the installer file provided. However, it is preferable to install it on the first level of a hard drive or on the partition C:\, particularly for end-users who do not have an administrator account on their computer.

3- Structure of AMSIMOD

The main window of AMSIMOD with first-level menus is illustrated on Figure 1. The active workspace and project (''project loaded'') are indicated. A workspace is the name and location of a folder (directory) that contains projects, models, input and output files or any other type of information or data that may be used in simulations. A project contains the information needed, including models, input and output files, to run simulations (see below).

3.1- File menu

The first two commands in the file menu can be used to create a new workspace or change the active workspace, respectively (Figure 2). A workspace can be located anywhere on a hard disk. For instance, a workspace called "Sim_Forest" may be located on the C drive within a folder "Projects". Thus, the workspace is "C:\Projects\Sim_Forest" and becomes the active workspace. The command "Change Workspace" allows end-users to change the active workspace. As it is possible to define several workspaces within hard disks, this submenu facilitates workspace browsing.

Note : It is preferable not to insert spaces in workspace names. However, the character underline ("_") is acceptable as separator of workspace names, such as, "Sim_Forests".

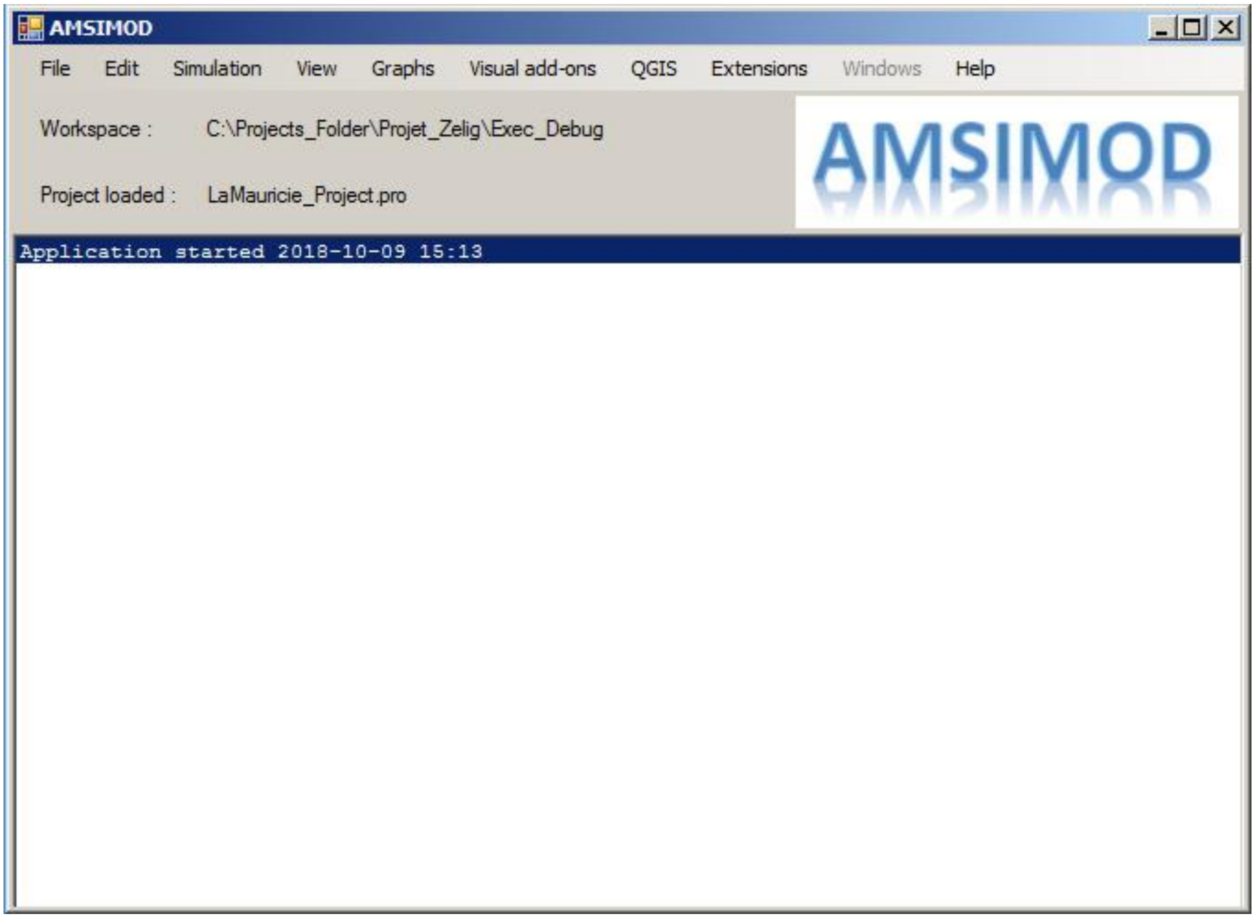


Figure 1: Illustration of the main window and menu bar of AMSIMOD.

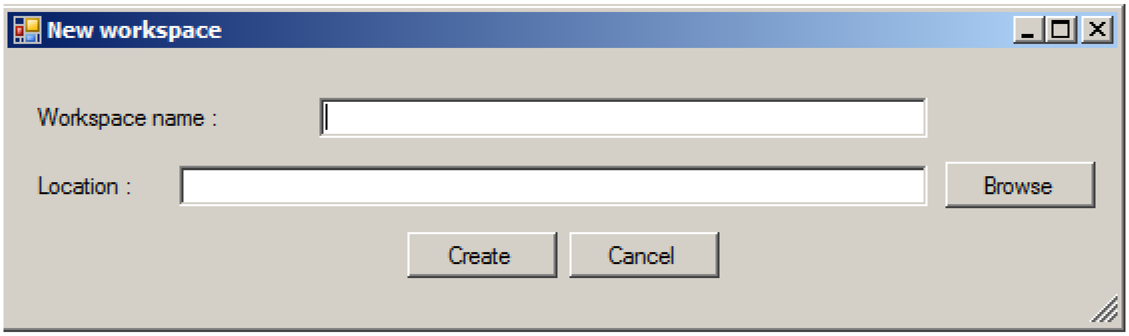


Figure 2: Form that opens when the command « New workspace» is selected in the file menu to enter the name of a workspace and select its location on a hard disk.

The third command, “New Project” is used to define a new project in a workspace. When selected, a form opens to request a project name (Figure 3). When the “Ok” button is selected, the extension “.pro” is added to the name of a project and a folder (directory) containing the name of the project followed by “_results” is created within the current workspace. All the output files created by models following their execution will be saved in this folder. For instance, if the name of a project is Forest_Sim”, the folder “Forest_Sim_results” will be created and the output files of simulation runs will be saved in this folder.

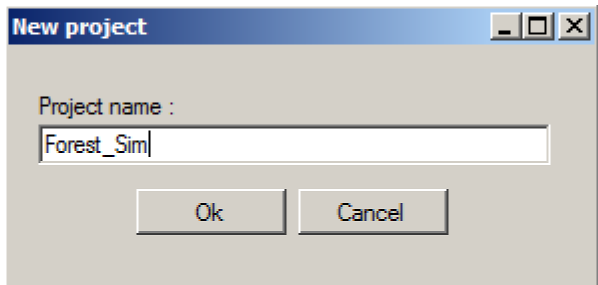


Figure 3: Input window to enter the name of a new project.

If the name of a project contains distinct words that should be separated, the “_” (underline) character must be used instead of a blank character (space) to separate them (e.g., Project_Forest). Once the name of a project is entered, a form opens to add one or several models in a project (Figure 4). By clicking on “Add a model”, the form “Simulation Setup” opens to enter the simulation name, the executable file (ending with .exe) of a model (e.g., Growth_Forest.exe) and initialisation (input), intervention and output files (Figure 5). The executable, initialisation and intervention files must located within the active workspace. Only one executable file of a model is allowed within a simulation, but a project may contain several simulations using different models. A simulation may contain as many input, intervention and output files as necessary to run a simulation. For every file of a simulation, it is possible to include comments by clicking on the button “Comments”. This utility is very useful when a project contains several models and input, intervention and output files. An example of the information provided to run a model within a project is included in Appendix 1. For AMSIMOD, the code of a model can be developped in any programming language, as long as a stand-alone executable file (.exe or .bat) is included in the definition of a project. Input, intervention and output files provided in a simulation must be sequentially associated with the instructions in the code of a model that requests input, intervention and output files. Appendix 2 contains an example that reads input and output file names in the C/C++ language.

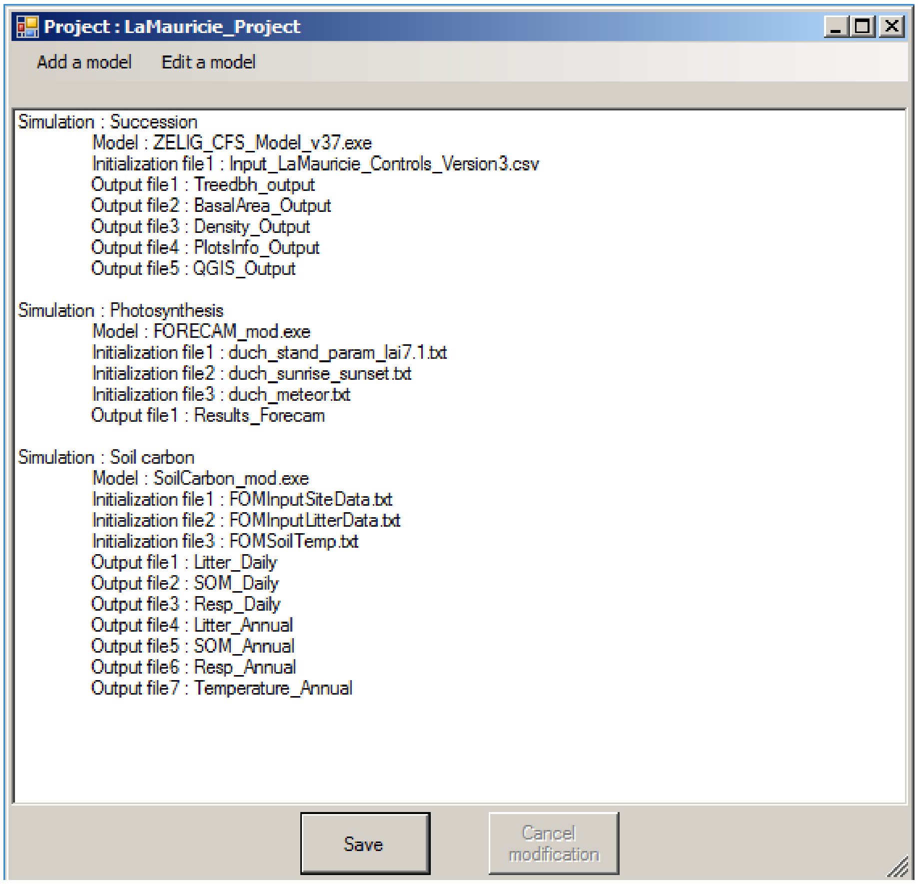


Figure 4: Input window to add or edit a model in a project.

The fourth command, ‘’Open a project’’, is used to open an existing project in the active workspace. When a project is selected, the project form opens and all the simulations of a project are listed. It is possible to add or edit simulations of a project.

The fifth command, “Open graphic file”, is used to import simulation results that were saved in a graphic format.

The sixth command “Language” is used to choose the language of AMSIMOD, either English or French. When selected, all the menus, commands and the texts in the main window and forms will appear in the language selected by the user. This command must be executed when AMSIMOD is installed for the first time to ensure that the file $Language_AMSIMOD.txt is present.

The last command, ‘’Exit’’, ends the execution of AMSIMOD.

3.2- Edit menu

The ‘’Edit’’ menu is used to modify an existing project. The project window opens and lists all the simulations that are included in a project.

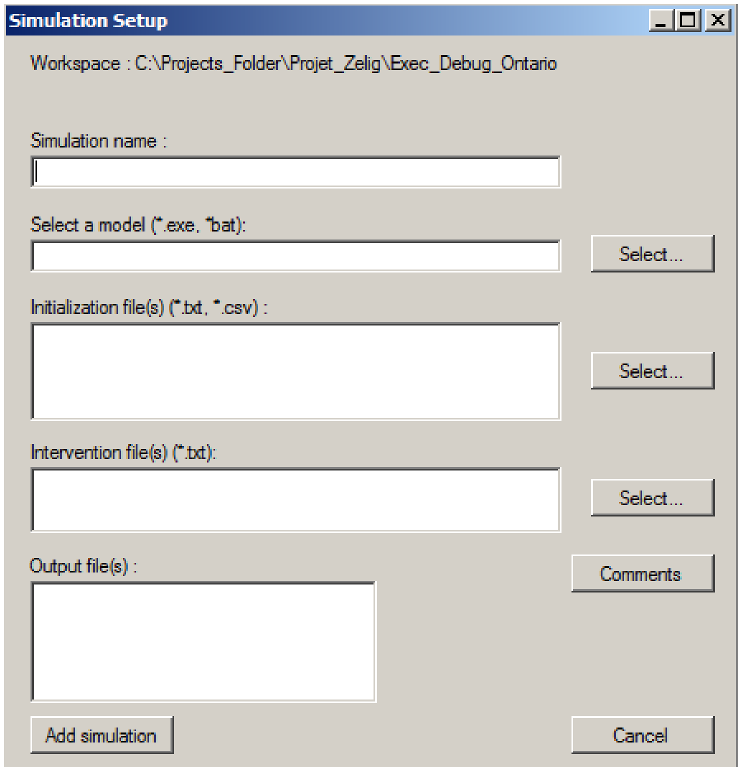


Figure 5: Input window to enter the name of a simulation, the executable file of a model or application and the initialization, intervention and output files.

3.3- Simulation menu

When ‘’run’’ is selected, all the models of a project are executed sequentially. It is possible to stop the execution of the models within a project by selecting ‘’Stop simulation’’.

3.4- View menu

The command ‘’Project comments’’ displays the comments that were provided in the system when a new project was defined or an existing project modified. This functionality may be very useful as a checklist, particularly when a project contains several models and input, intervention and output files. The command ‘’Available projects’’ lists all the projects that exist in the workspace identified just below the menu bar.

3.5- Graphs menu

Modellers or end users of models generally prefer to display simulation results on graphs. AMSIMOD includes utilities in the “Graphs” menu to draw line graphs or histograms. Once a graph type is selected, a data file type “.txt” must be entered. Input files to draw graphs contain simulation results that are normally found in the folders of simulation results (ending with ‘’_results’’). The first line in simulation result files must contain the names of the variables. The variable name for time (e.g., year, day or hour) must be placed in the first column. A maximum of 20 characters may be used for variable names. Letters, digits or the underline (“_”) character are allowed in variable names. The ampersand (&) and dollar ($) signs are used at the end of a variable name to identify stratification or category variables (see below).

3.5.1- Line graph Submenu

The submenu ‘’line graph’’ allows the visualization of simulation results on simple or stratified graphs. For both simple and stratified graphs, it is possible to display the results of a particular simulation by selecting “Single results” or the average of several simulation results by selecting “Average results”.

3.5.1.1- Simple graphs for visualizing simulation results

The “Simple graph” second-level submenu may be used to display single source simulation results by selecting the command “Single results”, such as the simulation results in Table 1. Once a file is selected, a form lists the variables that are available for display on a simple graph (Figure 6). A maximum of five variables can be selected for a simple graph, but there is no limit on the number of graphs that can be created.

Table 1: Example of simulation results that can be displayed in a ‘’Simple graph’’.

Year	Resp_Litter_Roots	Resp_FH	Resp_Min	Resp_FH_Min	
1	1356	908	1119	1766	2885
2	1368	947	1058	1614	2672
3	1380	965	982	1463	2446
4	1391	973	914	1372	2287
5	1401	977	859	1319	2179
Etc.					

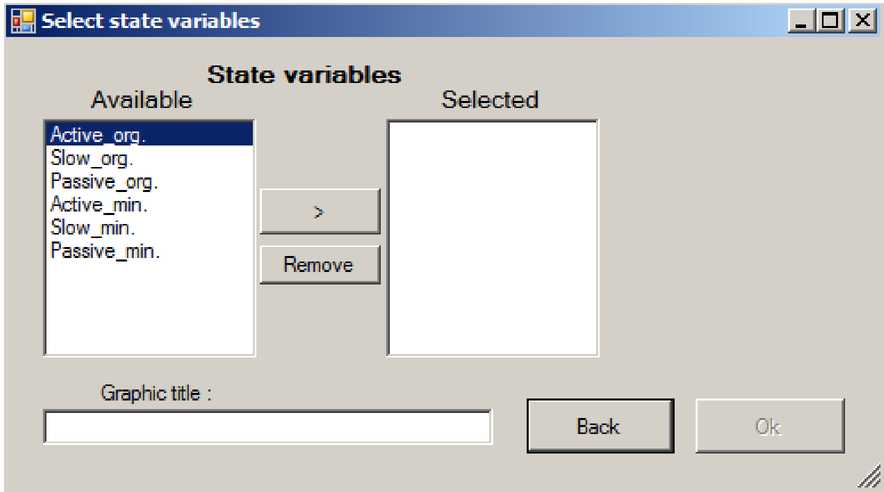


Figure 6: Example of a form that includes the list of the state variables in an output file that can be displayed on a simple graph.

The results of several simulation results may be combined from different files to display average values by selecting the command "Average results" in the "Simple graph" submenu. The files must be selected from the list of files within a result directory (Figure 7). An example of a graph that displays average results is shown on Figure 8. Error bars, which consist of standard deviations, can be added by selecting "Manage error bars" in the "Action" menu of the graph (Figure 9). The number of error bars to display can be selected in the Combo box on the right.

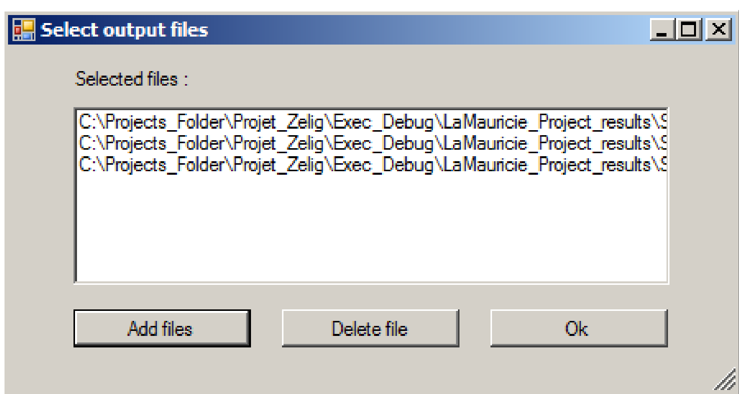


Figure 7: Selection of different output files to display average simulation results.

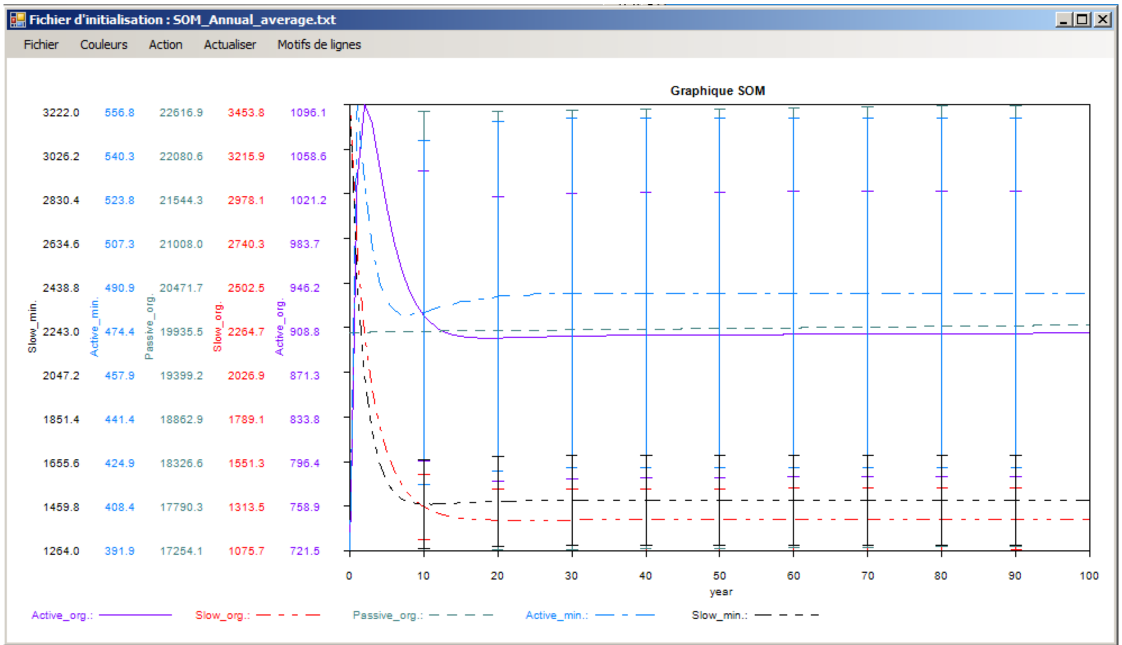


Figure 8: Example of a simple graph showing average results by combining several output files.

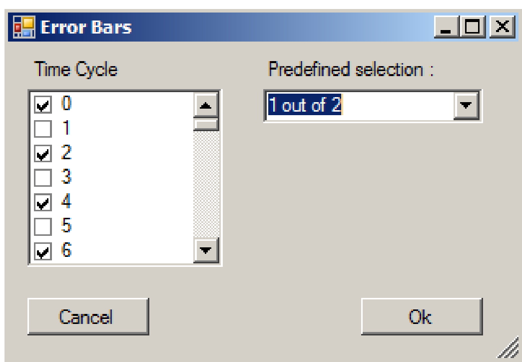


Figure 9: Selection of the number of error bars to display in a graph that contains average simulation results.

3.5.1.2- Stratified graphs for visualizing simulation results

A stratified graph facilitates the visualization of simulation results from different sources of results contained in a single file, such as the results obtained from a forest model that can simulate the development of several sample plots over time. Table 2 is an example of an output file with different sources of results. ‘’Plot_ID$’’ and ‘’ Plot_num$’’ are stratification variables that are used to differentiate the sources of simulation results. Stratification variables end with the character ‘’$’’. In the example of Table 2, ‘’Parc_ID$’’ and ‘’Parc_num$’’ indicate that the simulation results were output for different sample plots identified by a forest type (e.g., Edouard_CO ou Edouard_VI_O) and a sample plot number. A maximum of three stratification variables can be used to draw a stratified graph. ‘’Species&’’ in Table 2 is used to identify the instances of a state variable that will be displayed on a graph. For example, ‘’Species&’’ in Table 2 indicates that basal area will be displayed on a graph for different species and simulation periods. When “Single results” is selected in the “Stratified graph” submenu, a result file must first be selected. Following the selection of a result file, a form opens to select a maximum of two stratification variables (listed under the heading “Available”) by clicking on “>”, one category variable (selected from a Combo box) and 1 output variable (Figure 10). Once this information is provided, a second form opens to select specific simulation results associated with the attributes of a category variable to display (Figure 11). For the example of Figure 11, the end user can select from the Combo boxes the simulation results of a particular sample plot identified by its forest type (Plot_ID) and plot number (Plot_num). A maximum of five attributes of a category variable (Species in the example of Figure 11) can be displayed on a graph, but several stratified graphs can be created during an AMSIMOD session.

Table 2: Example of a simulation result file with different sources of results. Plot_ID$ and Plot_num$ are stratification variables and Species& is a category variable

Year	Plot_ID$	Plot_num$	Species&	Basal_area_(m**2/ha)
0	Edouard_CO	65	Yellow_birch	2.15
0	Edouard_CO	65	White_birch	1.80
0	Edouard_CO	65	Red_spruce	22.12
0	Edouard_CO	65	Red_maple	0.31
0	Edouard_CO	65	Balsam_fir	10.76
0	Edouard_CO	65	Northern_whitecedar	0.72
25	Edouard_CO	65	Yellow_birch	0.11
25	Edouard_CO	65	White_birch	0.24
25	Edouard_CO	65	Red_spruce	28.80
25	Edouard_CO	65	Balsam_fir	3.11
25	Edouard_CO	65	Northern_whitecedar	0.17
0	Edouard_VI_O	67	Yellow_birch	6.78
0	Edouard_VI_O	67	White_birch	4.74
0	Edouard_VI_O	67	Red_spruce	2.99
0	Edouard_VI_O	67	Red_maple	0.53
0	Edouard_VI_O	67	Sugar_maple	0.65
0	Edouard_VI_O	67	Balsam_fir	1.41
0	Edouard_VI_O	67	Northern_whitecedar	0.49
20	Edouard_VI_O	67	Yellow_birch	5.41
20	Edouard_VI_O	67	White_birch	4.71
20	Edouard_VI_O	67	Red_spruce	4.37
20	Edouard_VI_O	67	Mountain_maple	0.90
20	Edouard_VI_O	67	Striped_maple	0.02
20	Edouard_VI_O	67	Sugar_maple	0.24
20	Edouard_VI_O	67	Balsam_fir	3.53
20	Edouard_VI_O	67	Northern_whitecedar	0.97

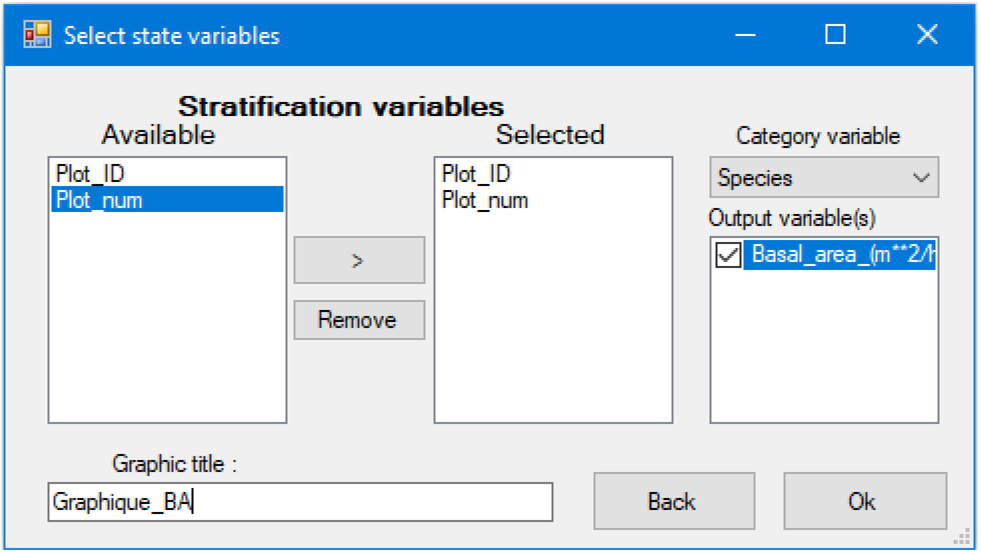


Figure 10: Selection of two stratification variables, 1 category variable and 1 output variable for a stratified graph.

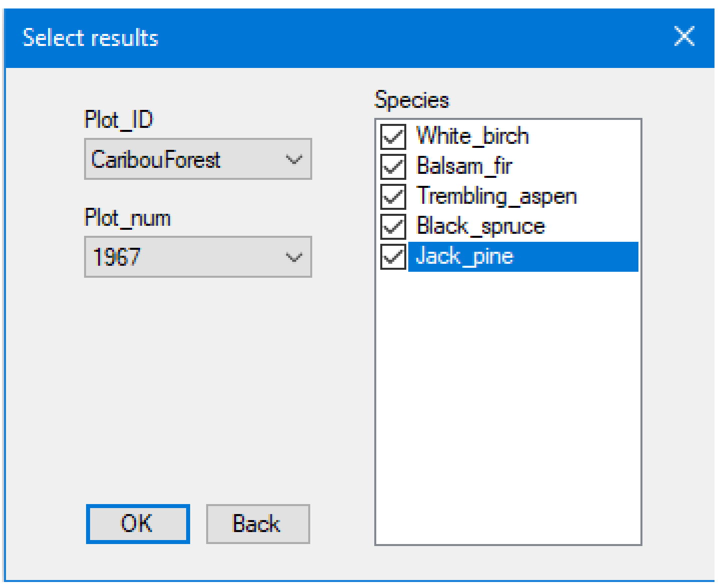


Figure 11: Selection of specific simulation results associated with the attributes of a category variable to display on a stratified graph.

Stratified graphs can also display average results from different sources of variation within a result file or from different result files. For a graph showing average results from different sources of variation within a result file, the third-level menu “Average results – Within a result file” is selected. First, a form opens to select stratification, category and output variables (Figure 10). Then, a second form opens to select cases within the two stratification variables (Figure 12).

In this example, the sample plot numbers (plot_num) that are used to display the output variable are selected. The graph that is created also contains a utility to show error bars (sub-menu error bars in the Action menu). The averages of several simulation results from different result files are displayed by selecting “Average results” in the “Stratified graph” second-level submenu. The result files are selected from the list of files within a result folder (see Figure 7). Once the selection of result files is completed, a form opens to select the stratification and category variables (Figure 13). In the example of Figure 13, the two stratification variables are “Plot_ID” and “Plot_num”. It is possible to select three forest types (Edouard_CO, Edouard_VI_O or Edouard_O_CO) and sample plot numbers associated with each forest type. As the category variable is “Species” in the example of Figure 13, the list of species that can be displayed are then listed to produce a graph showing average simulation results along with error bars (standard deviations) (Figure 14).

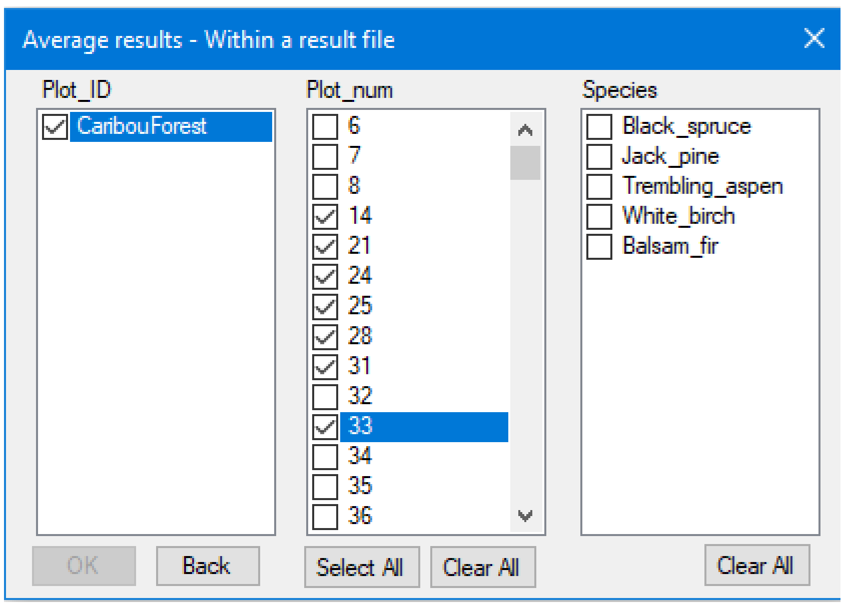


Figure 12: Selection of different sources of variation, such as plot numbers, to display results from the sub-menu “Average results – Within a result file”.

3.5.2- Histograms

The “Histogram” submenu is used to draw histograms to visualize simulation results of frequency distributions. The number of trees by diameter at breast height (dbh) classes is a good example of histograms commonly used to visualize stand density values by dbh classes. An example of the format of a result file to draw a histogram is found in Appendix 3. The first line

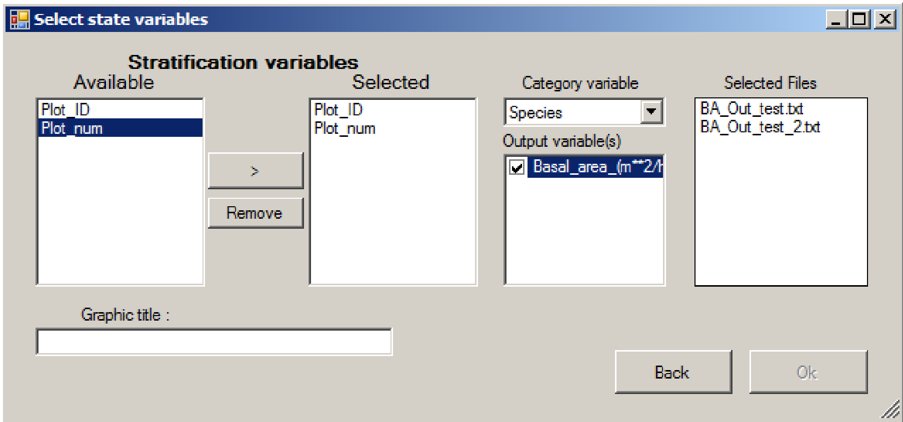


Figure 13: Form that opens after the selection of output files from different simulation results is completed.

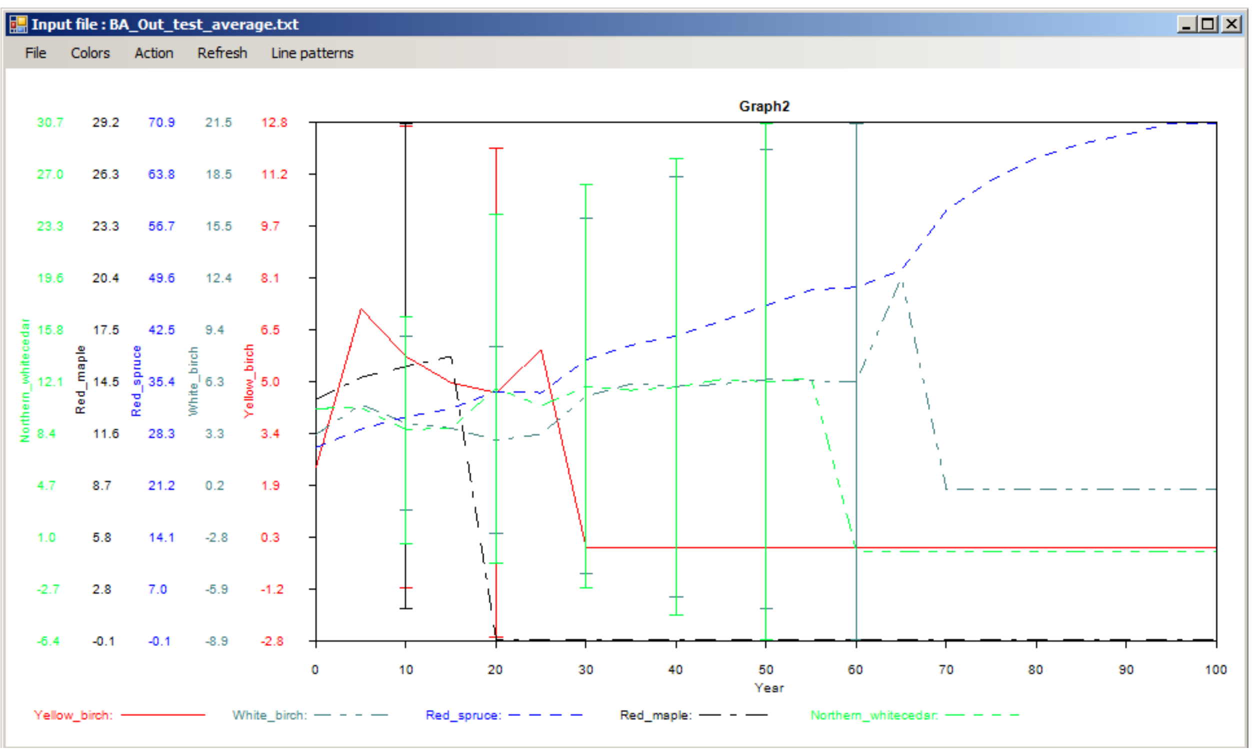


Figure 14: Example of a stratified graph showing average results with error bars.

contains the labels for the time cycle, which must be in the first column, stratification variables (Plot_ID$ and Plot_#$ in the example of Appendix 3), a category variable (Species& in the example of Appendix 3), the label for the X-axis, minimum and maximum values for each class of the variable in the X-axis and the label for the Y-axis. The label for the X-axis ends with the

"@" (at) symbol. The minimum and maximum values for each class are separated using the hyphen mark. For AMSIMOD, the minimum class value is interpreted as data in this class that are greater or equal (≥) to this value, while the maximum value is interpreted as smaller (<) than this value. The last class is identified by a single value followed by the hyphen mark and is interpreted as greater or equal. There is no limit on the number of frequency classes, but it is preferable not to have more than 10 classes.

There are two options to draw histograms. The command "Distribution histogram" is used to draw histograms for a single data source, such as a sample plot. Once the result file is selected, the stratification variable(s) must be selected along with a category variable (Figure 15). Then, based on the stratification variable(s), Plot_ID and Plot_# in the example of Figure 15, the values for a particular data source, such as a sample plot, may be selected to draw histograms (Figure 16). For the example in Appendix 3, the result is a window that displays density histograms for each species and time cycle in the input file, with menus to modify the size and colors of the histograms.

The command "Average histogram" is used to draw histograms that contain average values from different simulation result files. Figure 17 is an illustration of the form used to identify the input files. Following the selection of the input files, a form opens to select stratification and category variables. This step is followed by a form to select the values of a particular data source, such as a sample plot, and the time cycles to display. This functionality is used to display average values of different simulation scenarios performed on the same data source, such as a sample plot. Error bars based on standard deviations can be displayed.

3.6- ''Visual Add-ons'' menu

The menu ''Visual Add-ons'' may contain tools to visualize simulation results using virtual applications. The Stand Visualization System (SVS), developed by the USDA Forest Service, is currently the only virtual application that was implemented. (Note: SVS is currently non-functional).

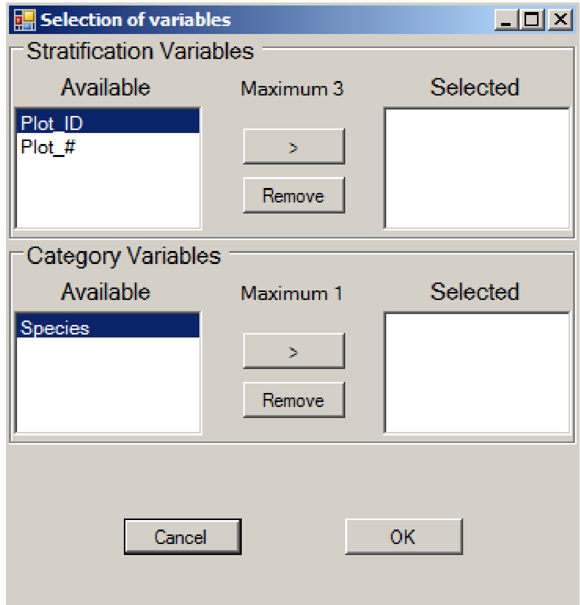


Figure 15: Input window to select the stratification and category variables to produce a distribution histogram.

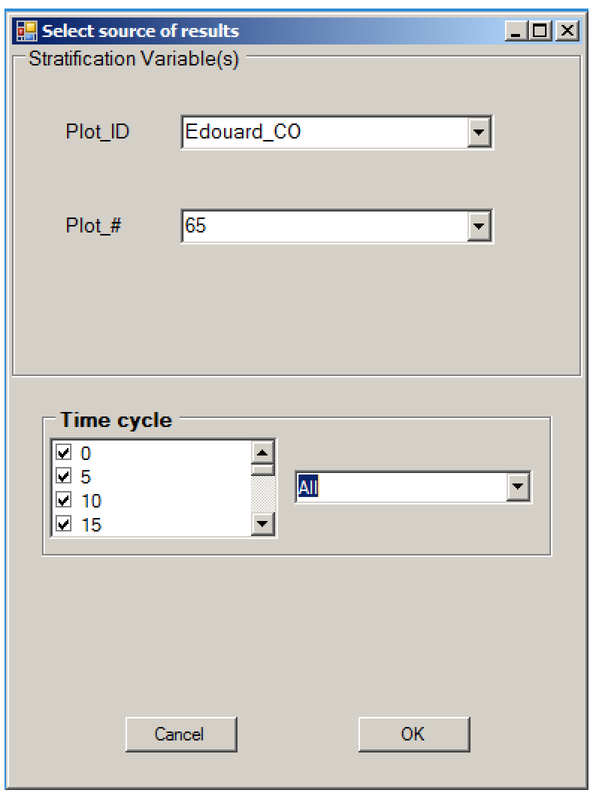


Figure 16: Following the identification of stratification and category variables, this form allows the selection of a specific source of results, such as a sample plot, at different time cycles.

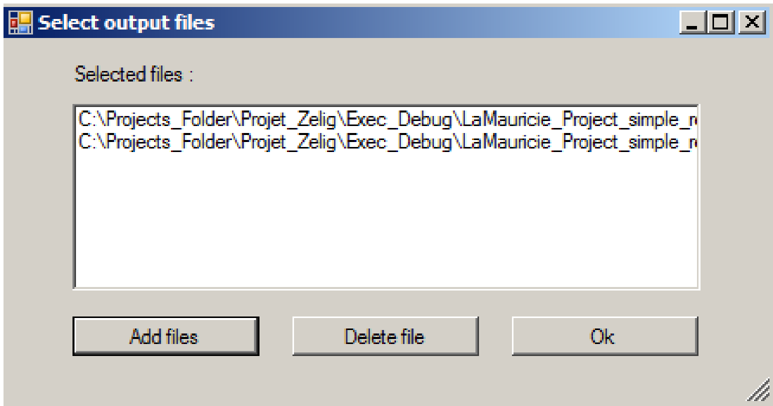


Figure 17: Selection of input files to display average values on histograms.

3.7- QGIS menu

3.7.1- Installing and running Quantum GIS

QGIS can be launched and executed independently of AMSOMOD. However, the QGIS menu was implemented to facilitate the management of QGIS, which is used to visualize simulation results on digital maps at both site and landscape levels. The command “Setup” is used to identify the location of the folder that contains the executable file of Quantum GIS (Usually on C:\Program Files\ QGIS*[1]\bin) and the location (folder) and QGIS file of the digital map that will be used to display results in the current AMSIMOD session (Figure 18). The command “Start” launches the execution of QuantumGIS using the digital map provided as input. The two parameters in the form illustrated in Figure 18 need to be provided only once in a workspace, as AMSIMOD records this information. Thus, it is possible to proceed directly to the command “Start” during a simulation session, unless a new digital map is loaded.

It is recommended to install the version 2.18.20 of Quantum GIS to ensure compatibility with the plugins associated with AMSIMOD. This version of Quantum GIS is available on the AMSIMOD web site: QGIS-OSGeo4W-2.18.20-1-Setup-x86.exe for the 32 bit version or QGIS-OSGeo4W-2.18.20-1-Setup-x86_64.exe for the 64 bit version. When Quantum GIS is launched, it is recommended to select “Automatically enable on the fly reprojection if layers have different CRS” in the Options | CRS tab of the Settings menu. This option ensures that layers in a project will use the same coordinate reference system.

However, the launching of Quantum GIS from AMSIMOD can take considerable time. This long delay can be avoided by keeping Quantum GIS active after the initial launching. New simulation

[1]The asterisk indicates that the name of the folder beginning with QGIS can be followed by an identifier of the version of QGIS, such as Valmeira or Essen.

results can be displayed simply by loading a new map or re-loading the existing map in the "New" and "open" sub-menus in the project menu.

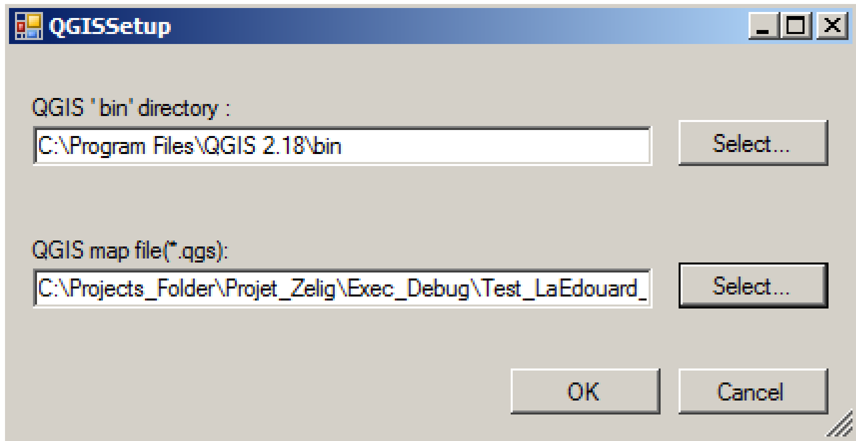


Figure 18: Form that opens when the command "Setup" is selected in the QGIS menu to enter the location of the executable file of Quantum GIS and the location and name of a digital map.

3.8- Extensions menu

The menu "Extensions" is used to implement special applications or tools in AMSIMOD, such as numerical analytical applications to analyse simulation results or create intervention files. These applications or tools normally consist of executable files (with the extension ".exe") that can be developed in any programming language. It is possible to add or delete extensions (Figure 19). The submenu "Run" is used to execute one of the applications in the "Extensions" menu.

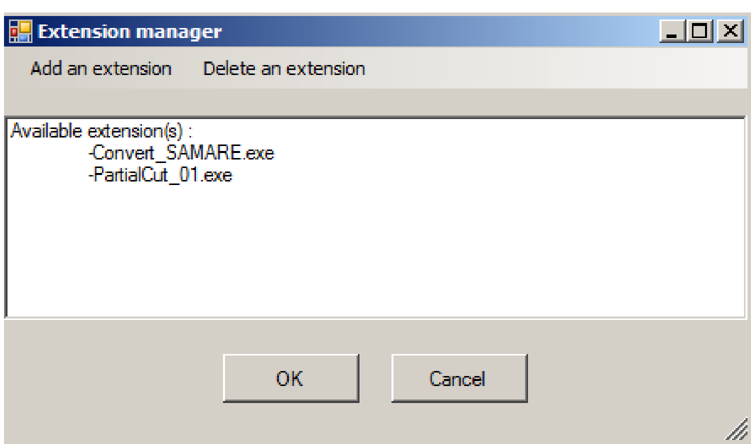
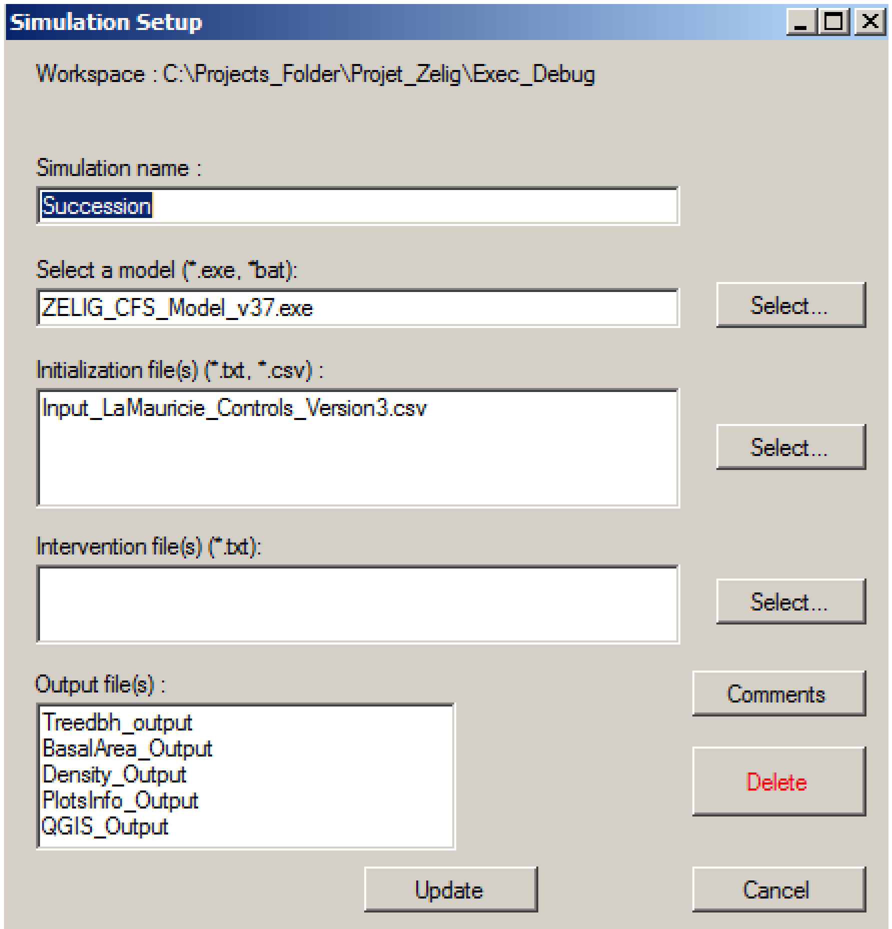


Figure 19: Examples of available applications in the menu ''Extensions''.

3.9- Windows menu

This menu contains the list of all the windows created during an AMSIMOD session. This list is updated automatically when windows are created or deleted. A window that is selected in this menu becomes active.

Appendix 1: Example of the information provided to run a model within a project. There is one input file and 5 output files.



Appendix 2: Example of C/C++ code to read input and output file names that are included in a project simulation

In the example of Appendix 1, there is one initialization (input) file that must be opened to read the initialization data for a model and five output files that must be created. For the input file, the following code can be used in a model:

```
/* Read the name of the input file and open it to read the initialisation data of the
simulation. */
       gets(InpfileNameDriver);
       if ((InpFileDriver = fopen(InpfileNameDriver, modeinp)) == NULL)
        {
               /* Print error message, as the file did not open correctly. */
               /* If there is an error, gets out of the function
                      and terminate the program. */

               printf("\n Problem to open the file %s \n", InpfileNameDriver);
               printf("\n <Press return>");
               gets(InpfileName);
               goto endpgr;
        }
```

AMSIMOD automatically reads the character variable “InpfileNameDriver” using the function gets(), which is the name of an input file provided in an AMSIMOD project. An example of an input file is “Input_LaMauricie_Controls_Version3.csv”. The next line contains the instructions to open the file “Input_LaMauricie_Controls_Version3.csv” and assign it to InpFileDriver. This line of code also verifies if the file exists within the workspace. In case of a problem, an error message is displayed and the execution of the model is stopped. The content of the input file is subsequently assigned to specific variables in the model using C/C++ functions such as “fscanf()”. The same instructions to read input file names and open the files must be repeated for each input file provided to AMSIMOD in the definition of simulations in a project.

In the example of Appendix 1, there are five names of output files provided to AMSIMOD. The names of the output files are read and opened using the C/C++ functions “gets()” and “fopen()”. For instance:

```
 gets(OutputExceldbh_Dr);
 OutExceldbh = fopen(OutputExceldbh_Dr, 'w');
```

The function ‘’gets()`` reads the name “Treedbh_output” (adds “.txt” if not there) and the function ‘’fopen()’’ opens the file “Treedbh_output.txt“. If this file does not exist, it is created. However, if the file already exists, data in the file is deleted. Then, the function ‘’fprintf()’’ is used to write simulation results in the file. For instance, “`fprintf(OutExceldbh,"\n %4d", kyr);`” writes the variable kyr in the ouput file “Treedbh_output.txt”.

The following sequential calls must be included in the code of the model to create or open the output files listed in Appendix 1:

For BasalArea_Output:

```
gets(OutputBasalArea_Dr);
OutBasalArea = fopen(OutputBasalArea_Dr, 'w');
```

For Density_Output:

```
gets(OutputDensity_Dr);
OutDensity = fopen(OutputDensity_Dr, 'w');
```

For PlotsInfo_Output:

```
gets(OutputPlotInfos_Dr);
OutPlotInfos = fopen(OutputPlotInfos_Dr, 'w');
```

For QGIS_Output data:

```
gets(OutputQGISfiles_Dr);
OutQGISfiles = fopen(OutputQGISfiles_Dr, 'w');
```

Appendix 3: Example of the format of a data file to draw histograms. A typical file may contain several sample plot data.

An	Parc_ID$	Plot_#$	Species&	Dbh_class@	0.5-1	1-9	9-20	20-30	30-	Num_trees/ha
0	AmericanC	1	White_birch	0	25	37	87	0		
0	AmericanC	1	Balsam_fir	0	12	0	12	0		
0	AmericanC	1	Trembling_aspen	0	25	161	420	0		
0	AmericanC	1	Black_spruce	0	87	111	185	12		
0	AmericanC	1	Total	0	148	309	705	12		
1	AmericanC	1	White_birch	0	25	37	87	0		
1	AmericanC	1	Balsam_fir	0	12	0	12	0		
1	AmericanC	1	Trembling_aspen	0	25	161	420	0		
1	AmericanC	1	Black_spruce	0	87	111	185	12		
1	AmericanC	1	Total	0	148	309	705	12		
2	AmericanC	1	White_birch	0	25	37	87	0		
2	AmericanC	1	Balsam_fir	0	12	0	0	0		
2	AmericanC	1	Trembling_aspen	0	12	136	346	0		
2	AmericanC	1	Black_spruce	0	74	111	185	12		
2	AmericanC	1	Total	0	124	284	618	12		
3	AmericanC	1	White_birch	0	25	37	87	0		
3	AmericanC	1	Balsam_fir	0	12	0	0	0		
3	AmericanC	1	Trembling_aspen	0	0	111	321	0		
3	AmericanC	1	Black_spruce	0	62	111	185	12		
3	AmericanC	1	Total	0	99	260	593	12		