Developing Modeling Software for Engineering Majors with Emphasis on Water Resources.

Samuel Camacaro[1], David Rivas[2], Jairo Diaz-Ramirez[3]

Abstract -Currently, modeling tools with graphical user interfaces are beneficial for teaching engineering. In teaching water-related courses, the complexity of rainfall-runoff theory can be softened by using computer models. In rainfall-runoff analyses, calculating soil infiltration rates is key to computing how much water enters the soil column and how much water flows over the ground surface as runoff in a finite lapse of time. The goal of this project was to develop a stand-alone graphical user interface tool where users can select different soil infiltration equations using the MATLAB programming language. The infiltration analysis tool is called WATIC and consists of over 20 formulas divided into three groups: empirical, conceptual, and other models. The WATIC software will support engineering problem-based learning activities in water-related majors.

Key words: MATLAB, software development, infiltration models, empirical models, conceptual models.

Introduction

The learning environment for engineering majors in agricultural, civil, and environmental requires computer models for better understanding of underlying principles and processes of the continuous movement of water on, above, and below the surface of the earth. The use of computer programs allows the students to participate in hands-on engineering activities. The U.S. Environmental Protection Agency (USEPA [1]) defined a model as "*a representation of the behavior of an object or process, often in mathematical or statistical terms*." In using computer programs, the graphical user interface (GUI) plays a vital role. A GUI [2] is a type of user interface that uses windows, icons, and menus that can be operated by a mouse rather than text commands (e.g., MS-DOS). In today's professional market, the use of computer software with GUIs can speed up the users work and allow them to take advantage of running multiple programs of modern operating systems [2].

In teaching water-related courses, the complexity of rainfall-runoff theory can be softened by using computer models. A particular application case is the analysis of rainfall-runoff processes that includes calculating soil infiltration rates. These infiltration rates are key to computing how much water enters the soil column and how much water flows over the ground surface as runoff in a finite lapse of time. Infiltration is required for determination of the so-called effective rainfall (also referred to as the rainfall excess) and consequently the direct runoff. It is generally acknowledged that approximately 70 percent of annual precipitation infiltrates the soil in the continental United States, and approximately 30 percent becomes streamflow [3]. Infiltration is a useful parameter in water balance calculations data and hydrology assessment of interest areas. In some cases, infiltration rates are related to

[1] Undergraduate student at Mississippi State University –Department of Computer Science and Engineering, PO BOX 3075, Mississippi State, MS 39762, sec224@msstate.edu

[2] Undergraduate student at National University of Colombia –Department of Civil and Agricultural Engineering, 215 Louisville Street Unit 9, Starkville, MS 39759, darivast@unal.edu.co

[3]Assistant research professor at Mississippi State University –Department of Civil and Environmental Engineering, 501 Hardy Road, office 224, Mississippi State, MS 39762, jd216@cee.msstate.edu

soil erosion, as well as with soil-water components including soil-moist storage, percolation and evapotranspiration. Different authors [5], [6], [7] consider that this infiltration phenomenon is one of the most complex elements of the hydrologic water cycle.

Several approaches to compute soil infiltration rates, such as empirical-based and physics-based models, have been proposed by different authors. The interpretation of the infiltration processes leads to many derivations. Empirical models were developed over the basis of experimental data; therefore, these models do not employ physics of flow through porous media. Conceptual models employ physics of flow through porous media in measurable physical properties of the soil-water system. The other infiltration models refer to empirical or conceptual models, which do not explicitly depend on rainfall intensity [4]. Infiltration data analysis is beneficial in multiple fields of study, such as hydrology modeling, environmental engineering, irrigation and landscaping design.

The goal of this project was to develop a stand-alone graphical interface tool where users can select different soil infiltration equations. Water Infiltration Calculator (WATIC) was coded using the MATLAB programming language. The formulas that are available in WATIC are[4]: Horton, Kostiakov, Kostiakov-Lewis, Philips, Holtan, Overton, SCS, HEC, Snyder, Hydrograph, Philip Version, Fok - Hansen Version, Li-Sterns-Simons, Hall Model, Mein-Larson, Morel-Seytoux and Khanji, Paralange Model, Gardner – Widtsoe, Kirkham –Feng, Green-Ampt, Philip Two-Parameters, Collins-George, Smith Model and the Stanford Watershed Model used in HSPF model. These are commonly used equations in the fields of civil engineering, agricultural sciences and engineering, environmental sciences and engineering and in geosciences fields. The end goal is to be able to easily distribute this software, so it can be used by water resources professionals. WATIC was programmed in the MATLAB programming language because of its efficient Graphical Users Interface (GUI) and its portability.

Methods

To develop the WATIC software, we relied heavily on the MATLAB integrated functionalities as well as scientific literature related to soil infiltration processes and approaches. The development of WATIC, started using C++ and Java programming. After some analysis we decided to use MATLAB computing language for several defined reasons. MATLAB version 10b can be deployed in a .fig and .m format that does not require the MATLAB license (for non commercial distribution, just the MCR compiler attached to the .exe files is necessary). Also MATLAB version 10b has a high portability that allows for cross-platform interaction, meaning that the software can be run in different operating systems, such as Mac OS, Linux and Microsoft. The cross-platform functionality and the license waiver let our software reach many users, hence giving it a wider audience. Another notable factor in coding WATIC in MATLAB is its multiple language portability, including language conversion of code written in the MATLAB environments that can be transferred easily into C++. Hence if it is necessary for implementation purposes to change the programming language, this could easily be achieved due to MATLAB built in features.

The graphical users interface (GUI) in WATIC was a key design element. The GUI interface was developed using the MATLAB GUIDE function. The user would input the data into the program through the use of different input mechanisms such as radio buttons, panel boxes, text boxes and checkboxes. The input of data would be sequential and transparent for the user, meaning that depending on the first choice a new set of choices would be generated. By using this approach it would allow the user to focus on solving their particular problem without being overwhelmed by the numerous infiltration equations available in the WATIC software. The way that MATLAB GUIDE works is that programmers literally draw out the GUI. Then, MATLAB automatically generates the code behind it. This is similar to the GUI system used in Netbeans for Java applications. Once the GUI has been designed, any modification of the MATLAB code will affect the GUI. Also any modification of the GUI will affect the code. Therefore, bi-directional programming in the code was established. This bi-directional coding was an important feature in the software development. Some changes are more easily done in the code than with the GUIDE on the GUI- also inversely many design features like the boxes sizes are more easily done on the GUIDE, instead of the code directly.

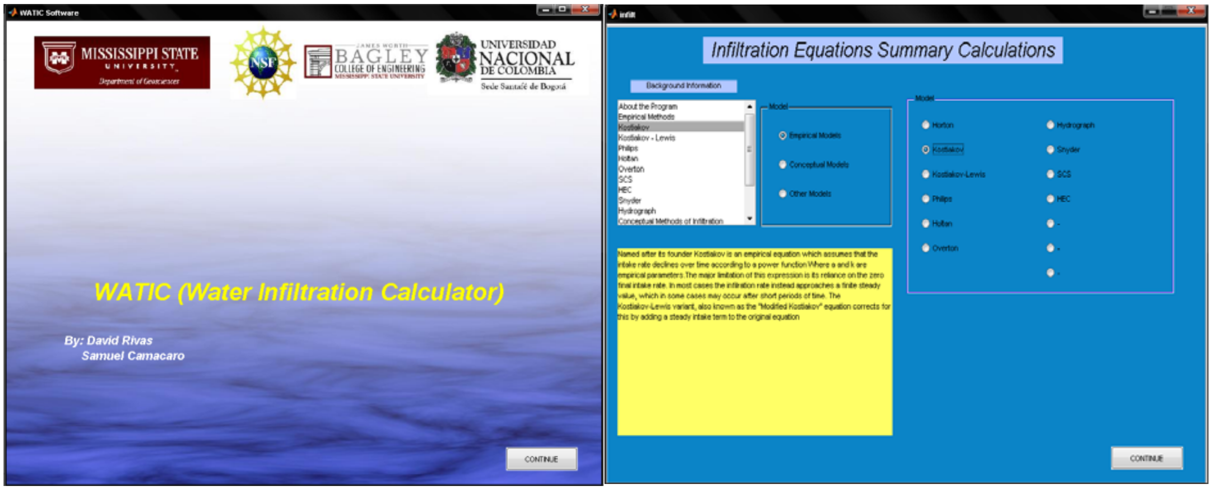
A key focus of the software development was to program the software. Different approaches were taken into consideration such as automata-based programming, declarative programming, and concurrent programming, among others. We were focused on four main programming paradigms: the event driven programming, the component based programming, the function level programming and the modular programming. After analyzing the different programming paradigms, we decided to develop the software using a modular programming approach. The main reason to use modular programming was to code individual equations, and then add them to the main code. In WATIC, infiltration equations do not interact between them; therefore, programmers took into consideration other approaches like using a code repository such as SVN, so that the code interaction would go smoother. This approach means that the code updates would be made visible to all of us and previous versions of the code would be available to programmers always. Such an approach of using repositories is used in industrial environments. The use of modular programming allowed skipping the use of repositories, as it has been known that small repositories in an academic environment tend to be a hassle because of the interworking of the push and pull functionalities. Since, programmers individually maintain their WATIC code, once their code is fully functional they can add it to the main component

Applications like the WATIC tool, serve to introduce more ideas for engineering applications. In that way, MATLAB functions are usually used to deploy more programming skills for any engineering applications. Most of programmers use MATLAB programming language to link data, as well as for computation and user interface for different situations. An example of this is to link data sources on EXCEL such as input data for calculations, and MATLAB for calculus and graphical interface. In addition, MATLAB can be used to import data for different sources like java, www, *.txt (files), com and *.net.

The different compilers involved in calculations and deploying tools are usually known by programmers and can be used for any situation. The ability of programmers to understand this capability allows them to successfully develop tools or applications. In this case, the WATIC tool is the first step in configuring a complete tool for water resources engineering. Modeling in hydrology is more complex with the use of infiltration equations, however, these equations are fundamental for engineers to model the water cycle.

Results

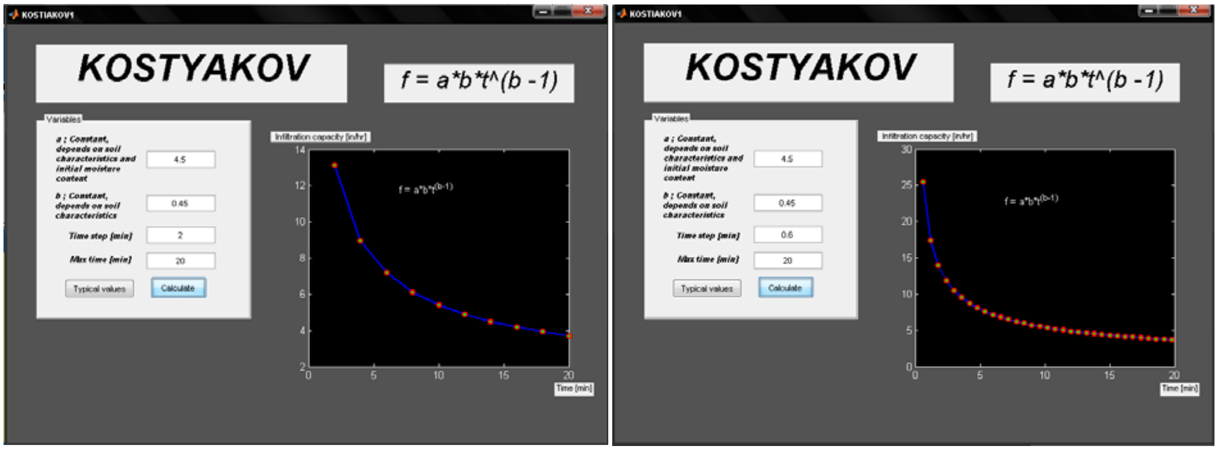
This section is an overview of the WATIC tool and user interface platform. Figure 1 depicts the presentation screen of the tool and the selection menu of the different water infiltration models. The user can chose among three different kind of models, empirical, conceptual and others models.



Named after its founder Kostiakov is an empirical equation which assumes that the intake rate declines over time according to a power function Where a and k are empirical parameters The major limitation of this expression is its reliance on the zero final intake rate. In most cases the infiltration rate instead approaches a finite steady value, which in some cases may occur after short periods of time. The Kostiakov-Lewis variant, also known as the "Modified Kostiakov" equation corrects for this by adding a steady intake term to the original equation

Figure 1.Presentation of WATIC Graphical User Interface for infiltration models

Figure 2 shows the input interface for the empirical model of Kostyakov [4]. The left picture shows the infiltration of Silt Loam soil with *a* = 4.5, *b* = 0.45 (*a* and *b* are constants that depends on soil characteristics), time step every 2 minutes and maximum time display calculations of 20 minutes. The picture on the right shows the same type of soil with same value constants, but a shorter time step every 0.6 minutes. This example shows how the user can manipulate input parameters values of coded equations in WATIC.



KOSTIAKOV1

KOSTYAKOV

f = a*b*t^(b -1)

Variables

a ; Constant, depends on soil characteristics and initial moisture content: 4.5

b ; Constant, depends on soil characteristics: 0.45

Time step [min]: 2

Max time [min]: 20

KOSTIAKOV1

KOSTYAKOV

f = a*b*t^(b -1)

Variables

a ; Constant, depends on soil characteristics and initial moisture content: 4.5

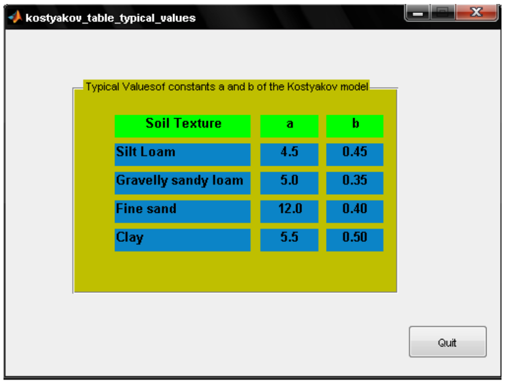
b ; Constant, depends on soil characteristics: 0.45

Time step [min]: 0.6

Max time [min]: 20

Figure 2.Example of Kostyakov model for a Silty Loam soil water infiltration with different time resolution.

In Figure 2, the user interface allows users to introduce parameters for the computational process. If the user doesn’t know the parameter range of coded equations, there is a help button that shows the typical constants values for different soils (see Figure 3 for typical values for the Kostyakov equation). These values were extracted from a literature review of each model that included hydrology books, technical papers, manuals, etc.



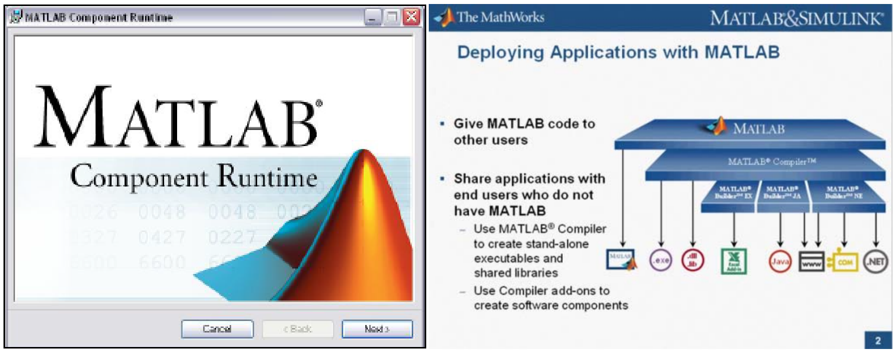
kostyakov_table_typical_values

Typical Valuesof constants a and b of the Kostyakov model

Soil Texture	a	b
Silt Loam	4.5	0.45
Gravelly sandy loam	5.0	0.35
Fine sand	12.0	0.40
Clay	5.5	0.50

Figure 3. Typical values table of Kostyakov empirical model of Infiltration [9].

The WATIC tool is a GUI application of MATLAB that requires MATLAB software for execution. For users that don’t have this software, we found a solution using the MCR compiler of MATLAB to execute the WATIC package in computers that do not have MATLAB, which allows execution of the tool with no legal problems for license permission. Figure 4 shows the MCR (MATLAB Component Runtime).



Deploying Applications with MATLAB

- Give MATLAB code to other users
- Share applications with end users who do not have MATLAB
  - Use MATLAB® Compiler to create stand-alone executables and shared libraries
  - Use Compiler add-ons to create software components

Figure 4. MATLAB Component Runtime (MCR)[8].

The executable files need to be installed following the user guide for better execution.

Discussion

Programming the .m files of infiltration equations in WATIC tool required a comprehensive literature review and mathematical manipulation. Some equations like Holtan's model presented difficult mathematical interpretation in terms of time. This model does not depend directly of time- it depends on soil to store water on a lapse of time [1].

The purpose of any infiltration model is to show the infiltration rates with time for different time resolutions. In order to help the users, a compendium of typical input values of each infiltration equation was developed during this investigation.

In general, the WATIC tool can calculate soil infiltration rates using empirical to physics-based models in an easy-to-use graphical user interface. The application and analysis is up to the user. One of the features of the WATIC tool is the option that the user can compare field test data with any model. This can help in finding the best model for a particular study area.

Conclusions and Future Work

The WATIC tool is an engineering software application that displays in one interface many approaches for computing water infiltration rates through the soil. The interface was developed to easily manipulate data, including help with typical values of each model along with tabular and graphical outputs of soil infiltration curves.

The GUI application of MATLAB combined with the MCR, resulted to be a useful combination for academic purposes and for modeling assessment in hydrology, irrigation, landscaping, environmental engineering, etc. The capability of open .m files allows any person to modify the programming environment and add other components.

Future development includes:

- Link models for the same type of soil
- Test models with observed data
- Compute statistical analysis between simulated and observed data
- Code an optimization tool that finds the best model for the observed data
- Link WATIC with geographical information systems
- Improve the import and export data interface
- Improve plotting options

Acknowledgments

The authors would like to thank Dr. Brenda Kirkland from the Mississippi State University's Geosciences Department for providing scholarships to Samuel Camacaro and David Rivas through a National Science Foundation grant titled "Diversity Enhancement through Research Experiences (DEGReE)." Mr. Rivas was also funded by the Mississippi State University's Bagley College of Engineering and Office of Graduate School

References

[1] USEPA, U.S. Environmental Protection Agency, *Guidance on the Development, Evaluation, and Application of Environmental Models*, available from: http://www.epa.gov/crem/library/cred_guidance_0309.pdf, 02-04-2012.

[2] The Linux Information Project, *GUI Definition*, available from: http://www.linfo.org/gui.html, 02-04-2012

[3] Chow, V. T., *Handbook of Applied Hydrology*, McGraw-Hill Book, New York, 1964, pg. 98,103.

[4] Singh, V. P., *Hydrologic Systems: Watershed Modeling*, Prentice hall, New Jersey, 1989, pg. 39, 47, 70

[5] Horton, R. E., 1933. “The Role of Infiltration in the Hydrologic Cycle,” *Transactions of the American Geophysical Union*, 1933, 14:446-60.

[6] Horton, R. E., “Analysis of Runoff- Plat Experiment with Varying Infiltration Capacity,” *Transactions of the American Geophysical Union*, 1939, 20:693-711.

[7] Cook, H. L., “The Infiltration Approach to the Calculation of Surface Runoff,” *Transactions of the American Geophysical Union*, 1946, 27(5):726-47.

[8] Mathworks, *Build Standalone Executables and Software Components from MATLAB Code*, available from: http://www.mathworks.com/products/compiler/, 11-28-2011.

[9] Haan, C. T., Barfield, B. J., and Hayes, J. C., *Design Hydrology and Sedimentology for Small Catchments*, Academic Press, San Diego, 1994, pg. 58.