
Customization, extension and reuse of outdated hydrogeological software

A. Serrano-Juan¹ R. Criollo^{1,2,5} E. Vázquez-Suñé^{1*} M. Alcaraz^{3,4} C. Ayora¹ V. Velasco¹ L. Scheiber¹

¹Institute of Environmental Assessment and Water Research (IDAEA-CSIC)

Jordi Girona 18-26, 08034 Barcelona (Spain)

²GHS, Department of Geotechnical Engineering and Geosciences, Universitat Politècnica de Catalunya (UPC)

Jordi Girona 1-3, Building D 2, 08034 Barcelona (Spain)

³Instituto de Hidrología de Llanuras “Dr. Eduardo Usunoff”

República de Italia 78, B7300 Azul, Provincia de Buenos Aires, Argentina

⁴Consejo Nacional de Investigaciones Científicas (CONICET)

Av. Rivadavia, 1917, C1066AAJ Ciudad Autónoma de Buenos Aires, Argentina

⁵Barcelona Cicle de l'Aigua SA (BCASA)

Carrer de l'Acer 16, 08038 Barcelona, Spain

*Corresponding author

| A B S T R A C T |

Each scientist is specialized in his or her field of research and in the tools that he or she uses during the research in a specified site. Thus, he or she is the most suitable person for improving the tools by overcoming their limitations to realize faster and higher quality analysis. However, most scientists are not software developers. Hence, it is necessary to provide them with an easy approach that enables non-software developers to improve and customize their tools. This paper presents an approach for easily improving and customizing any hydrogeological software. It is the result of experiences with updating several interdisciplinary case studies. The main insights of this approach have been demonstrated using four examples: MIX (FORTRAN-based), BrineMIX (C++-based), EasyQuim and EasyBal (both spreadsheet-based). The improved software has been proven to be a better tool for enhanced analysis by substantially reducing the computation time and the tedious processing of the input and output data files.

KEYWORDS | Software improvement. Software customization. MIX. EasyQuim. EasyBal. BrineMIX.

INTRODUCTION

Over the past few decades, the rapid evolution of computer processing power has enabled the scientific community to solve various problems in the vast variety of geoscience fields, such as mineralogy, petrology, geochemistry, geology, geophysics, hydrology, and hydrogeology, among others. As a result, most scientists are aware of the importance

of computer-aided analysis since geoscience algorithms manage many variables, resulting in laborious calculations that are impossible to conduct without a computer tool (Wang and Huang, 2012).

For decades, scientists have searched for repeatable and predictable processes that would improve the productivity and the quality of the computer architecture and programming

languages to facilitate geoscientific calculations. The first programming languages, such as FORTRAN, COBOL, and BASIC, appeared in the mid-1960s and were widely used until the 1990s. Most were devised for the creation of individual programmes for handling specified tasks and short sets of data (at that time, the data were limited and sometimes difficult to collect). The compilers generated the well-known “*exe*” files, which typically required additional “*txt*” files such as input, output or conditional data during the execution (Wang *et al.*, 2012), thereby resulting in a set of many files that contained the information for one analysis. Many geoscientists have developed tools based on these programming languages (*e.g.* Bea, 2009).

The development of new technologies in both computer architecture and programming languages continues apace, thereby modifying the landscape. Current programming languages such as Python, Matlab, Visual Basic and Visual C are known as visual languages and are more user-friendly than their predecessors. Most integrate all the required information (*e.g.* input, output, and sources) into a single file and enable the user to directly conduct the whole analysis. Furthermore, the higher computing power has been accompanied by increasing data availability. In the last few decades, digital data collection, aggregation and integration have increased exponentially (*e.g.* streaming in from a growing number of satellites and sensors and the Internet). Geoscientists are overrun by data while having access to ever-increasing computing power.

In addition, Graphical User Interfaces (GUIs) became commonly used to facilitate rapid, rigorous and interactive analysis (Jones *et al.*, 2014). Many GUIs have been developed in geoscience (*e.g.* Phong *et al.*, 2012) to make software more user-friendly (*e.g.* screen selection of the input and output arrangement for instant comprehension of the results). Since new software programmes are dynamic, visual and interactive, some old fashioned programming language-based software programmes, such as FORTRAN-based programmes, are becoming outdated due to their complex analysis processes (preparing input text files, analysing the output text files and displaying limited graphical options). However, despite the limitations of these geoscientific software programmes, some remain the best option for resolving specified problems.

The academic (*e.g.* Ibrahim, 2009) and scientific communities (*e.g.* Asuncion, 2013) have also widely accepted the combination of spreadsheets with Visual Basic for Applications (VBA) for the development of software applications. This acceptance has mainly occurred because i) spreadsheet interfaces are user-friendly and facilitate numerical and statistical computations; ii) data can be easily queried, analysed and visualized; iii) a macro programming interface provides satisfactory end-

user guidance that facilitates the user in writing correct and more reliable programmes (Cunha *et al.*, 2014); iv) this approach saves time due to its low barrier since most researchers are already adept at manipulating spreadsheets and v) there are available tools that have been specially designed for the correction of potential errors (Jannach *et al.*, 2014) and inconsistent data storage (Cunha *et al.*, 2014). Consequently, a substantial variety of new tools are available that facilitate geoscientific calculations (*e.g.* Aliane, 2010; Jones *et al.*, 2014; Wang *et al.*, 2013). In hydrogeology, many spreadsheets have been developed for the facilitation of calculations in the analysis and interpretation of pumping tests, hydrogeochemical data, and analytical and numerical solutions for groundwater flow and pollution problems, among others (*e.g.* Elmore, 2007; Molano, 2013).

For instance, MIX (Carrera *et al.*, 2004) is a FORTRAN-based software that computes mixing ratios with uncertain end-members. It is the only available tool that estimates mixing ratios while considering the uncertainty in the end-member concentrations. However, the use of MIX is highly time-consuming since it is difficult to prepare all input text files (MIX is highly sensitive to typing errors, among other errors) and it is difficult to analyse the output files (which contain more than 10,000 text lines). Thus, it is necessary to improve MIX to automate input and output data treatment to reduce errors and to accelerate the analysis. More information about the code could be found in Carrera *et al.* (2004); Vázquez-Suñé *et al.* (2010); as well as, its previous application to real case studies (Canovas *et al.*, 2012; Jurado *et al.*, 2016; Scheiber *et al.*, 2018; Tubau *et al.*, 2014). Additional examples are EasyQuim and EasyBal. EasyQuim is a widely used tool (see section 3.4) for representing hydrochemical data and performing calculations such as ionic relationships, unit conversion and balance errors. However, EasyQuim was initially designed to plot up to 24 samples, while current projects typically collect many more samples. A similar difficulty is encountered with EasyBal, which is a software that evaluates the water balance per unit of soil. In this case, the programme is limited by a rigid data period range and requires a tedious input data process.

The scientific community is highly specialized. The combination of the field of research, the site of research and the tools that are utilized render the scientist the most specialized person in his or her field of research and in the application of the tools that he or she uses in a specified site. Thus, he or she is the most suitable person for improving his or her tools by overcoming their limitations to realize faster and higher quality analysis. However, most scientists are not software developers. Hence, it is necessary to provide them with an easy approach that enables non-software developers to improve and customize their tools.

This paper presents an approach for easily improving and customizing any hydrogeological software. It is the result of experiences with updating several interdisciplinary case studies. Since the programming language differs among case studies, it has been possible to determine whether this approach can be generalized. The main insights of this approach have been demonstrated using four examples: MIX (Carrera *et al.*, 2004) (FORTRAN-based), BrineMIX (C++-based), EasyQuim and EasyBal (both spreadsheet-based). However, only MIX will be discussed in detail to enable the reader easily to follow a step-by-step application of the presented approach. This paper also attempts to answer the following research questions: Q1 is it possible to easily update any hydrogeological software via this approach? Q2 do the improved versions lead to fewer errors during the analysis compared to the original approaches? Q3 are end users more efficient when using an improved version than when using the original version?

METHODOLOGY

General systems development

In both Object-Oriented Analysis (OOA) and the Systems Development Life Cycle (SDLC), programme creation can be regarded as the following flow process:

$$ID = GUI \rightleftharpoons DT/RUN$$

where ID is the identification of the problem (SDLC (1), OOA (1)), GUI denotes the graphical user interface (SDLC (2), OOA (2)), DT represents the required data treatment and RUN describes the solution computation (SDLC (3-5), OOA (3)). The maintenance phase of the SDLC has not been included.

The first step is problem identification (ID), which facilitates understanding of the problem and the answering of questions regarding, *e.g.* the available information and the desired outcome. Only when the programmer truly understands the nature of the problem is it possible to identify the necessary and available information, display it, arrange it, request it and determine which options should be offered. After identification, a GUI should be designed. Through this interface, the programme requests the input data, visualizes the output data and offers the possibility of setting up any option that the programme offers. Finally, all the requested data in the GUI could require Data Treatment (DT) to realize the suitable format for computation (RUN). Afterwards, the output data should be again displayed in the GUI, thereby maintaining a continuous interaction between the GUI and the computation of the programme.

Based on this scheme, an updating approach has been established as a decision flow chart (following the Unified

Modeling Language, UML standards), as software programs differ and require various types of updating (Figure 1).

Updating Approach

This paper presents an approach for easily improving and customizing software. This chapter follows the decision flow chart in Figure 1, and it describes each step and discusses the flow options. All the presented codes have been developed to run in an MS Excel environment.

To fully investigate the problem, four main issues should be addressed Problem identification (ID): 1) input data and output data, 2) computation, 3) improvements and 4) communication. 1) INPUT DATA AND OUTPUT DATA. What is computed? It is necessary to clearly identify all the data that are involved in the process, which include not only the required data but also the available data. 2) COMPUTATION. How is the result computed? Which software programs are involved in the computation? Is it possible to recompile the available code (access to the source code)? At this step, the developer should understand how the programme works, the complexity of the algorithm, the accessibility (open code access or not) and the possibility of combining various software programmes to define various software configurations, among other aspects. 3) IMPROVEMENTS. Are any changes needed? Which improvements are possible? The strategy is not just to reuse and adapt outdated software but to add new features and functionalities that will improve the performance of the analysis (*e.g.* allowing data storage, enhancing graphic outputs or connecting the results to other software platforms such as Geographic Information System, GIS). 4) COMMUNICATION. What do I know? What does the final user know? Finally, it is necessary to understand who will use the software and to consider factors such as background knowledge (both in computers and in science) and language. The ways in which information is solicited and displayed are significant. The ID process takes longer than the previous steps as the future GUI, the input and output DT and how the solution will be computed are defined.

A suitable (GUI^N) should request and display sufficient data while being aesthetically pleasing, comprehensible, simple and responsive. 5) INPUT DATA AND OUTPUT DATA. From where is the input information obtained? In hydrogeology, the information is commonly obtained from maps (GIS), tables (matrices), independent numbers (cells or input boxes) or is selected from an available dataset (*e.g.* buttons or lists). The process is similar for the output, where results of the analysis are commonly displayed as maps (GIS) or tables (matrix). 6) COMPUTATION and IMPROVEMENTS. How is the analysis conducted? VBA offers a large set of options such as button clicks or events

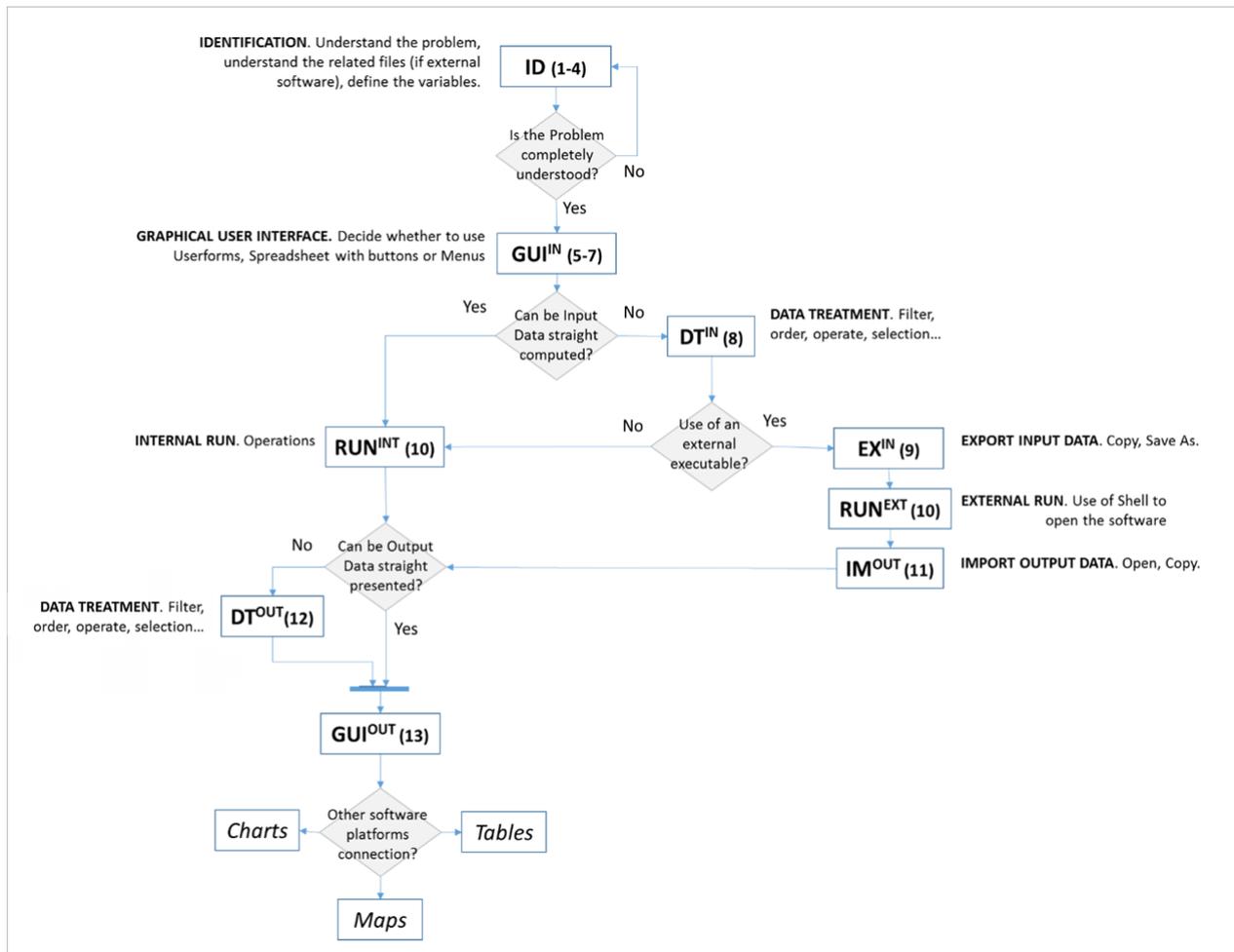


FIGURE 1. Decision Flow Diagram to improve and customize existing software.

(*e.g.* when adding information or modifying the content of a cell). 7) COMMUNICATION. Again, who is the final user? Many options are available for displaying information or for ordering and selecting it. The MS Excel environment can substantially improve the power of the analysis by considering whether the results should be static (*e.g.* simple tables or maps) or dynamic (*e.g.* pivot tables and charts). Finally, it is necessary to adapt all the new programme capabilities according to the knowledge of the final user.

Verplank (1985) and Marcus (1995) defined general principles of GUI design and its effectiveness in visual communication. In addition, many reliable resources are available on the Internet, such as (Jisc Digital Media website, 2019).

The input data are typically the available data, which are not necessarily required data. As these available data are not always provided in the correct order, 8) INPUT DATA TREATMENT (DT^{IN}) is essential. Depending on each programme, filters, calculations, unit conversions and

data rearrangement will sometimes be necessary for the preparation of the required input for analysis, whereas in other programmes. The input will be already in the desired format. Non-Excel-based programmes will need to 9) EXPORT THE INPUT DATA (EX^{IN}) in various formats and call external executables to perform the analysis, whereas Excel-based programmes (*e.g.* solvers, macros) will 10) RUN (EXTERNAL RUN (RUN^{EXT}) and INTERNAL RUN (RUN^{INT})) as a matter of course. In contrast, depending on the computational core format, 11) OUTPUT DATA will be IMPORTED (IM^{OUT}) into the GUI or prepared to be computed by another external programme. As not all the output must be presented in GUI, the Output Data (OD) can also be partially disregarded, rearranged in new tables and plotted. This 12) OUTPUT DATA TREATMENT (DT^{OUT}) is typically necessary for satisfying the 13) GUI output (GUI^{OUT}) requirements. Occasionally, it will be interesting to export the results to other software or platforms to obtain additional results and to conduct in-depth analyses (*e.g.* connecting to GIS platforms adds a time-space dimension). Common considerations during the DT process are the

decision to use code to evaluate formulas and create objects or to use pre-set formulas and charts in the spreadsheets. Typically, data storage (input/output) will be necessary before the data are recalled by the GUI or exported in various formats.

Figure 1 presents a decision flow diagram, and possible code actions are specified at each step. The Appendix presents another figure (Figure I, see the appendix) and a list of code sentences that could help future programme developers conduct each step of the process.

Additionally, during the reuse process, the time that is needed for the development of each step was analysed. According to the analysis, the conceptual model design (identification of the problem, design of the GUI and identification of the necessary DT) requires longer time than coding. Along this line (Buccella *et al.*, 2013) presents similar time distributions during their reuse development case study in GIS, which is also similar to the Rational Unified Process (RUP) (Kruchten, 2003) hump chart and the unified process (Jacobson *et al.*, 1999). Even if the user experience can significantly impact the total time that is necessary for improving any software, the time task distributions typically remain the same.

APPLICATION EXAMPLES

Several application examples have been created and tested to develop the presented approach. The combination of spreadsheets and VBA has been used to implement the software improvement and customization.

The MIX software will be discussed at length to enable the reader to follow a step-by-step application process of the presented approach. This example will emphasize the improvements over the previous versions, *e.g.* automatic and instant graphical output interaction, automatic formula refill to avoid heavy documents, connection to non-Excel-based software such as FORTRAN or GIS, automatic graphical output generation, and automatic data selection and rearrangement.

Three additional examples will be briefly described to improve the understanding regarding how spreadsheet-based and C++-based software can be improved and customized via the same approach.

MIX 2.0

MIX (Carrera *et al.*, 2004) was created for the assessment of a methodology for computing mixing ratios with uncertain end-members. Problem identification (ID). 1) INPUT DATA AND OUTPUT DATA. The input

file contains information about different waters (which can be divided into “end-members” and “samples”. Additional information such as restrictions (impossible mixing ratios) or known mixing ratios) can also be set. 2) COMPUTATION. Since the software was developed in FORTRAN, it requires one input file and generates two output ones, both of which are very long (the output files can exceed 10,000 lines). MIX considers three different degrees of freedom in the generation of the input matrix: the number of chemical species, the number of wells and the number of end-members. This matrix plus the user decision to include initial solutions and restrictions results in a complex input generation process. Moreover, the input file is highly sensitive to typing errors. The source code is not available for recompiling changes. 3) IMPROVEMENTS. The first requirement is automatic input file generation for handling typing errors. The new MIX should also offer the possibility of using the main input matrix as a database, thereby offering the possibility of selecting the chemical species, wells or end-members that the user wants to consider in the analysis. A selection of the analysis results should be automatically displayed in the spreadsheet. 4) COMMUNICATION. The final user should minimally feel or not feel that he or she is working with multiple platforms (in this case, MS Excel and FORTRAN). Moreover, output files can be lengthy and monotonous to read with unnecessary information for the analysis. For a standard analysis, only a selection of the data from these files should be displayed in tables and in various types of charts.

To design the GUI, both spreadsheets and UserForm were chosen. This enables the user to predefine the magnitude of the problem in the UserForm and to represent the input data in the spreadsheet as a matrix. 5) INPUT DATA AND OUTPUT DATA. In this case, the input data tables are established in separate spreadsheets (concentrations, standard deviations, initial solutions and restrictions) and are activated when the user navigates through the buttons of the UserForm. The input information can be directly set in the matrix or can be imported from a GIS (using macros that enable spatial selection and filling of the matrix). This GUI also offers the possibility of interacting with Windows by opening folders and available files.

6-7) COMPUTATION, IMPROVEMENTS and COMMUNICATION. After introducing the data, selecting the data that will be suitable for analysis and setting up the desired options, 8) INPUT DATA TREATMENT transforms all these data into a single text file and changes the formats, data types and units. This is the real input file that is 9) EXPORTED AND CALLED by the “FORTRAN-based programme MIX” from the Excel environment 10) (RUN^{EXT}). The “FORTRAN executable” is automatically called by Excel, thereby giving the user the impression that

he or she is not working with two software programmes (*e.g.* using the Shell statement). See the appendix for further information regarding the code. To 11) IM^{OUT} import the results, 12) DT^{OUT} is required despite the difficulty of managing the data. The storage of the input numbers of chemical species, wells and end-members in variables and the use of the functions to find key words enable us to select the information that merits consideration. After the DT, two types of plots are generated: pie charts that show the proportions of the end-members in each sample (well) and scatter plots of measurements versus calculated values for each chemical specie. Additional results such as contributions to the objective function and the eigenvalues are also presented in the form of tables. If the user desires to revise the two complete output files, these files are imported as two spreadsheets even if this also enables us to access the MIX windows folder where all files are stored.

One of the advantages of this case study is that the number of automatically generated plots and tables changes according to the data set that is input by the user. Another advantage is that the new version is connected to GIS-based software QUIMET (Velasco *et al.*, 2014) and AKVAGIS (Criollo *et al.*, 2019). This enables us to 13) export data GUI^{OUT} as a spatial representation in GIS and to 3) import the selected input temporal and spatial GIS environment data to fill the input data tables for analysis in the new MIX. Additionally, the programme enables the storage of large amounts of data (for use as a database) and the selection of a portion of the data for analysis. Last, a UserForm automatically appears when the programme starts, which presents the title, the logo and the designers of the programme. The UserForm can be set to disappear when the user clicks a button, or it can automatically vanish after a few seconds. All the presented UserForms can be minimized to avoid inconveniencing the user when he or she is checking the data. Figure 2 compares the input and the output software environments between the old and the new versions.

In summary, the new MIX version satisfies the need for a GUI by proposing a GUI that is based on the MS Excel environment. This GUI prepares input templates based on the user's requirements for the analysis in external software. A subset of the generated output is plotted and rearranged in the GUI, thereby enabling the user to check the entire output data files. Additional advantages are its potential use as a database (by providing the opportunity to select combinations of chemical species, wells and end-members for analysis) and its connection to a GIS environment.

Other examples

EasyQuim was designed in 1999 for the graphical representation of hydrochemical data. It conducts calculations such as unit conversion and balance error and ionic relationship

identification. It also plots Piper, Schöeller-Berkaloff, salinity and stiff diagrams of 24 samples and enables the user to select which to present. Everything is set in spreadsheets with functions, except one small macro, which activates the “No representation of samples” option, which can only be activated once. The new version should provide three main advantages: First, the maximum number of samples is increased (up to 200). Second, a “Sample Selector” is added. Third, a space-time analysis is possible. The “Sample Selector” provides a powerful tool for using the updated EasyQuim as a database and for plotting various sample combinations, whereas the connection to several GIS-based softwares as QUIMET (Velasco *et al.*, 2014); FREEWAT (Rossetto *et al.*, 2018); AKVAGIS (Criollo *et al.*, 2019), will enable analyses in the spatial and temporal dimensions.

EasyQuim is an example of the energization of a spreadsheet that was originally created for plotting in hydrochemical data analysis. The new version adds functionalities such as conversion of the main data spreadsheet into a database and creation of a data selector, thereby enabling the final user to decide which analyses merit comparison. Additionally, new programme connections such as the connection with GIS were established, thereby enabling further temporal and spatial data analyses.

EasyBal was designed in 1999 for the evaluation of the water balance per unit of soil area as a function of the precipitation, the Potential EvapoTranspiration (PET), the temperature and the irrigation. The outputs are the deficit and the recharge of the aquifer. Older versions required up to six steps to introduce the input data into six Excel sheets. All data analysis periods had to be between January 1970 and December 1997, and calculations and adaptations were required if the user required a different period. Each month had to contain exactly 30 days instead of the real number of days. It is necessary to eliminate the current data period restriction by allowing conditional sums, which will enable us to realize the automatic calculation of monthly and yearly totals. All functions should be reorganized to enable the autofill of each formula in a single line. These improvements enable the user to conduct the analysis simultaneously and to obtain all the results so that they can be clearly structured and organized in a single Excel sheet. Additional features are also included in the new EasyBal version: The user can select English or Spanish as the programme language. The PET can be introduced as input data or can be automatically calculated (using the Hargreaves and Thornthwaite methods) and graphically compared with the input data, thereby enabling the user to select the best option in a menu or graph.

EasyBal provides an example of an improvement to a current calculation spreadsheet. In this case, the process involved reorganizing all data functions to realize automatic

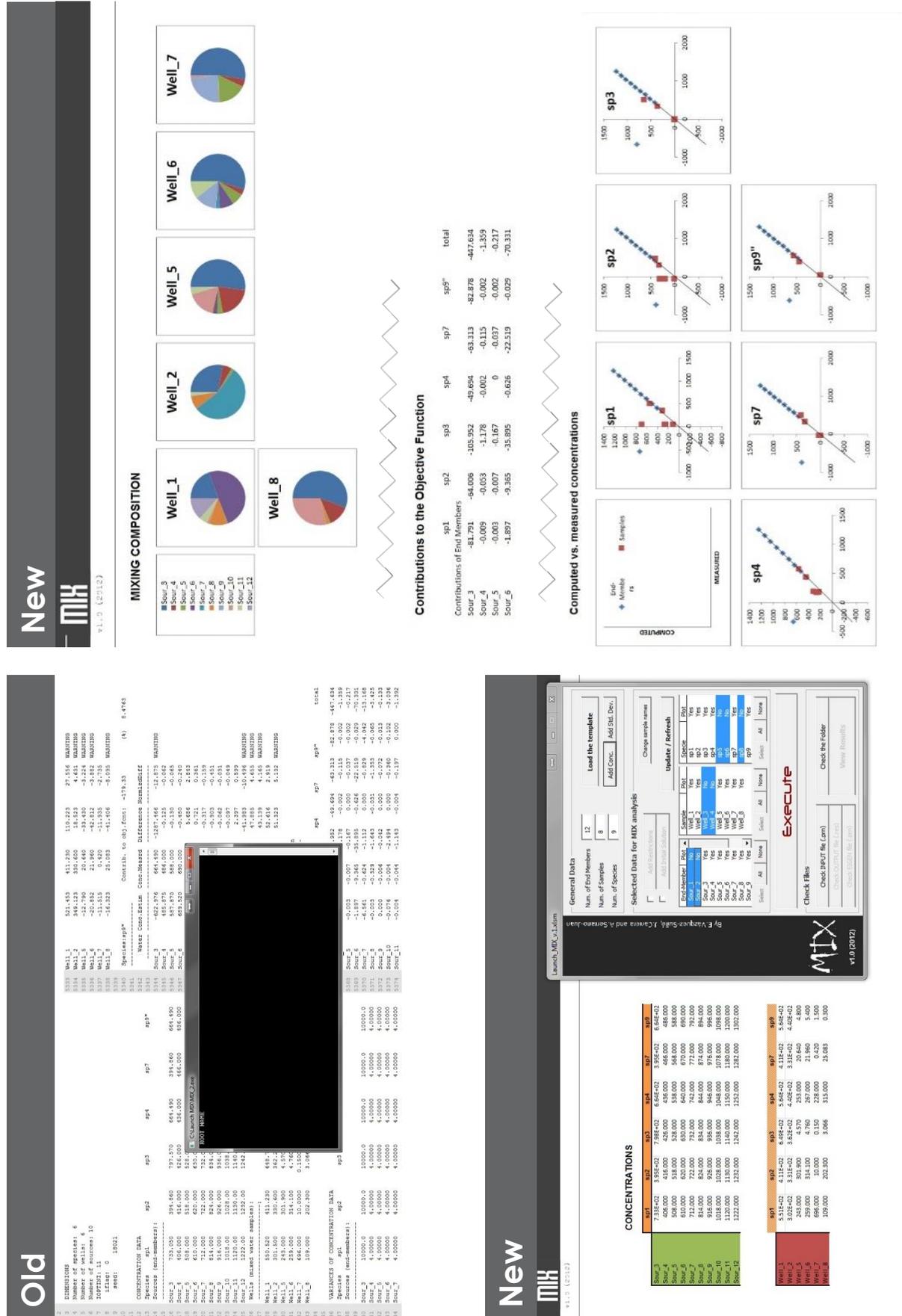


FIGURE 2. Comparison between the old MIX txt input file (top right) and old MIX txt output file (with more than 6,000 lines), and the new MIX v2014 MS Excel-based input dynamic table (down left) and output Graphic User Interface (GUI) (right).

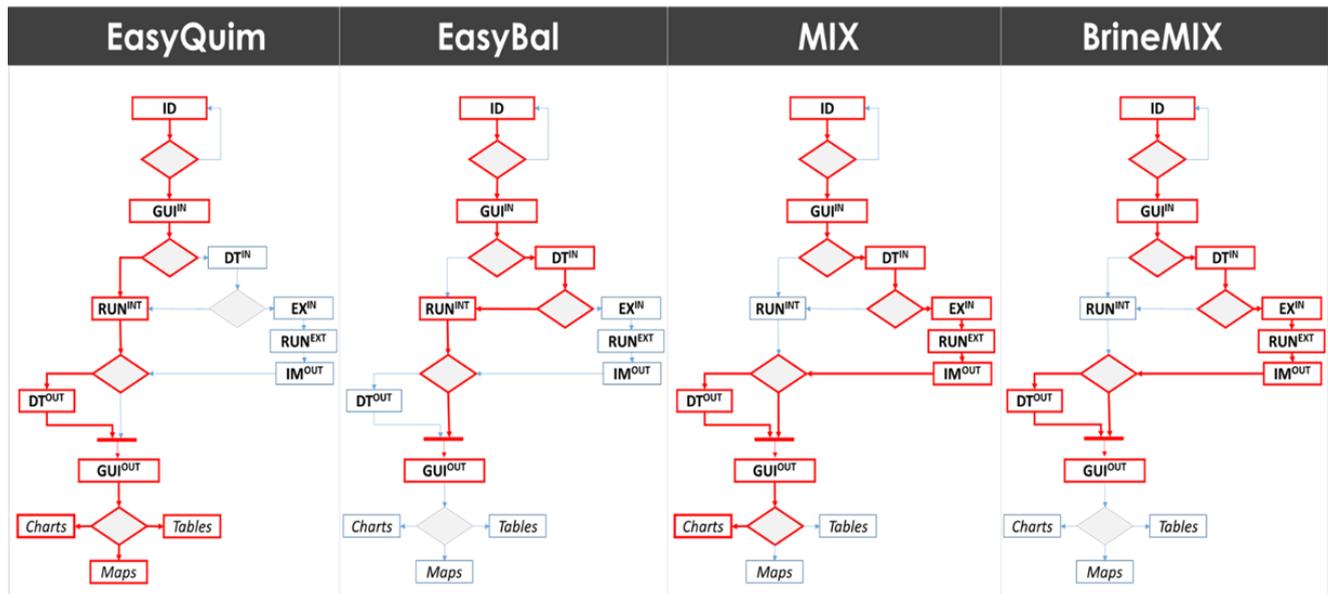


FIGURE 3. Flow Diagram Paths followed by each of the presented case studies: EasyQuim, EasyBal, MIX and BrineMIX.

formula refill and adding the language selector and the PET graph selector. By changing formulas, it was possible to accept any input data period and to automatically calculate monthly and yearly totals.

In contrast to the earlier examples, BrineMIX is a new programme, not an update. In this case, BrineMIX seeks to create a GUI that automatically generates the input and reads the output of PHREEQC (Parkhurst and Appelo, 2013) for a specified water mixing analysis. In the input, only the chemical water samples, the mixing percentage and the mineral selection are set, whereas the output specifies the chemical composition of the final water and its chemical precipitates. The objective of this new programme is to simplify a specified PHREEQC analysis for a user who does not typically work with it.

BrineMIX provides an example of the externalization of part of a larger software. PHREEQC can conduct many analyses, but not all are necessary for non-advanced chemical users. BrineMIX was created to simplify specified analyses by using an Excel environment to facilitate these users in conducting them.

Figure 3 shows the flow diagram paths that are followed in each of the presented case studies: EasyQuim, EasyBal, MIX and BrineMIX.

Software validation

The improvement of software is typically regarded as an empirical discipline. However, authors (*e.g.* Suri and

Garg, 2008) have used quantitative and qualitative metrics to measure the benefits of improving software. These metrics are typically related to quality (such as the error density, fault density, ratio of major errors to total faults, rework effort, module deltas, and developer perception), to productivity (lines of code per effort) and to the time-to-market (development cycle time). Many empirical studies have been reported in the literature in both industry and academia in which the relationship between software improvement and metrics is assessed (*e.g.* Devanbu *et al.*, 1996).

Quantitative metrics are used to obtain the same or better results in less operational time compared to the original version. In our software, most of the codes cannot be recompiled. Even if the time that is required for strictly computing the solution remains the same, the total time that is required for computing the whole analysis has been dramatically reduced. This is possible via the automation of preparing the input files, setting up the problem, reading the output files and preparing the output for a correct interpretation. All four examples have been tested to evaluate the total necessary time for conducting a complete analysis: while EasyBal and MIX save at least three quarters of the time, EasyQuim and BrineMIX save half of it.

In contrast, qualitative metrics measure the quality of the response that the user obtains from the software. The addition of new functionalities, the display of the results in a suitable format or the addition of exporting improves the performance of the analysis and the experience of the user. Automatic input/output data treatment not only saves time but also can substantially reduce the errors during

the process (*e.g.* FORTRAN-based programmes are highly sensitive to any typing error). It is also possible to obtain qualitative feedback through an increase of the system reliability, namely, by automating the error-prone human processes or by displaying warnings when values are out of range. The main improvements in our examples rely on dynamic data comparison, a wide range of data values, the addition of a GUI and the automation of the input and output data treatments.

Finally, software validation can also be measured by its acceptance and use in academia and by professionals. This approach has been used in educational, research and technical projects. EasyQuim and EasyBal, the previous versions of which were widely used in the hydrogeological international community (especially in Latin American countries), are taught in various international master courses by institutions such as the Universitat Politècnica de Catalunya (www.upc.edu) or by nonprofits such as the Fundació Centro Internacional de Hidrología Subterránea (FCIHS, www.fcih.org). All four improved software programmes have been applied in various technical and research projects, and Criollo *et al.* (2016, 2019); Scheiber *et al.* (2015, 2016, 2018); Serrano *et al.* (2016, 2018); Tubau *et al.* (2017); Velasco *et al.* (2014) have applied this approach for reusing these programmes.

System requirements and program availability

The four software examples can be obtained by making a request to the author or by downloading them from the URL: <https://www.idaea.csic.es/research-group/groundwater-and-hydrogeochemical/>

CONCLUSIONS

This paper presents a new approach for improving and customizing any hydrogeological software and provides insights into the process for its application to four cases. According the objectives and the stated questions we summarize the main outcomes:

i) It is possible to easily update hydrogeological software via this approach. Through these case studies, the reader can understand how software (*e.g.* in C++, FORTRAN, or VBA) can be improved via the same approach. Moreover, this approach enables the creation of new GUIs for the automatic generation of input and reading of output files from other analysis. Finally, the MIX case study has been discussed in detail to enable the reader to easily follow a step-by-step process for the application of the presented approach.

ii) The improved versions lead to fewer errors during the analysis compared to the original approaches. It

is demonstrated that the new versions are more user-friendly and avoid errors such as typing mistakes. An MS Excel environment enables us to perform the same action in a variety of ways. This is helpful since it enables the programme developer to design anything he or she considers suitable, thereby resulting in highly personalized programmes. Moreover, VBA offers the possibility of using messages in pop-up windows or colour changes to caution the user; *e.g.* indicating to him or her which values are out of range or that the required values are numbers instead of letters.

iii) End users are more efficient when using an improved version than when using the original version. In addition, the new versions easily generate input files and show, rearrange and plot the most important parts of the output. Through VBA, it was possible to assess complex input matrix generation and difficult output selections and to generate several chart types. We also demonstrated how VBA interacts with Windows by executing other programme and by opening Windows folders. In all cases, the GUI is highly important as it not only makes each programme easier to manage but also improves its organization.

Additionally, this methodology was evaluated during the improvement processes of several case studies, and a qualitative trend of the time distribution was observed throughout the process. This supports that the conceptual model design requires longer time than the other steps.

This approach has been used in education and research, and it is being applied in several technical projects.

Our approach realizes the objectives by providing the necessary steps for the facile development of any hydrogeological software to enable the advancement of the current understanding in hydrogeology by any scientist. The simplified methodology in a decision flow chart facilitates the programme developer in the assessment of any type of programme. However, although this approach has been developed for the reuse of hydrogeological software by hydrogeologists, it can also be applied to other fields, thereby creating synergies among scientists and expert programme developers.

ACKNOWLEDGMENTS

R. Criollo thanks Barcelona Cicle de l'Aigua (BCASA) and the Catalan Industrial Doctorates Plan of the Secretariat for Universities and Research, Ministry of Economy and Knowledge of the Generalitat de Catalunya for their financial support. The staff of the Department of Geotechnical Engineering and Geosciences of the Universitat Politècnica de Catalunya (UPC) and the Institute of Environmental Assessment and Water Research (IDAEA-

CSIC) are gratefully acknowledged for their useful comments and suggestions. We also thank Scott Tacker and Dadiyorto Wendy for their contributions which improved the manuscript.

REFERENCES

- Aliane, N., 2010. Data acquisition and real-time control using spreadsheets: interfacing Excel with external hardware. *ISA Transactions*, July 2010, 49(3), 264-9. DOI: 10.1016/j.isatra.2010.03.009
- Asuncion, H.U., 2013. Automated data provenance capture in spreadsheets, with case studies. *Future Generation Computer Systems*, October 2013, 29(8), 2169-2181. DOI: 10.1016/j.future.2013.04.009
- Bea, S.A., Carrera, J., Ayora, C., Batlle, F., Saaltink, M.W., 2009. CHEPROO: A Fortran 90 object-oriented module to solve chemical processes in Earth Science models. *Computers & Geosciences*, 35 (6), 1098-1112. DOI: 10.1016/j.cageo.2008.08.010
- Buccella, A., Cechich, A., Arias, M., Pol'la, M., Doldan, M.D.S., Morsan, E., 2013. Towards systematic software reuse of GIS: Insights from a case study. *Computers & Geosciences*, 54, 9-20. DOI: 10.1016/j.cageo.2012.11.014
- Buregio, V.A.D.A., Almeida, E.S.D., Lucredio, D., Meira, S.R.D.L., 2007. 31st IEEE Annual International Computer Software and Applications (COMPSAC) Conference, Beijing (China), Short paper.
- Cánovas, C.R., Olias, M., Vázquez-Suñé, E., Ayora, C., Nieto, J.M., 2012. Influence of releases from a fresh water reservoir on the hydrochemistry of the Tinto River (SW Spain). *Science of the Total Environment*, 416, 418-428. DOI: 10.1016/j.scitotenv.2011.11.079
- Carrera, J., Vázquez-Suñé, E., Castillo, O., Sánchez-Vila, X., 2004. A methodology to compute mixing ratios with uncertain end-members. *Water Resources Research*, 40(12), n/a. DOI: 10.1029/2003WR002263
- Criollo, R., Velasco, V., Vázquez-Suñé, E., Serrano-Juan, A., Alcaraz, M., García-Gil, A., 2016. An integrated GIS-based tool for aquifer test analysis. *Environmental Earth Sciences*, 75, 1-11. DOI: 10.1007/s12665-016-5292-3
- Criollo, R., Velasco, V., Nardi, A., de Vries, L.M., Riera, C., Scheiber, L., Jurado, A., Brouyère, S., Pujades, E., Rossetto, R., Vázquez-Suñé, E., 2019. AkvaGIS: An open source tool for water quantity and quality management. *Computers & Geosciences*, 127, 123-132. DOI: 10.1016/j.cageo.2018.10.012
- Cunha, J., Mendes, J., Saraiva, J., Visser, J., 2014. Model-based programming environments for spreadsheets. *Science of Computer Programming*, 1, 1-22. DOI: 10.1016/j.scico.2014.02.002
- Deursen, A.V., Klint, P., Visser, J., 2000. Domain-specific languages: An annotated bibliography. *ACM SIGPLAN Notices*, 35(6), 26-36.
- Devanbu, P.T., Karstu, S., Melo, W.L., Thomas, W., 1996. Analytical and empirical evaluation of software reuse metrics. Berlin (Germany), 18th International Conference on Software Engineering (ICSE), 189-199.
- Elmore, A., 2007. Applying one-dimensional mass transport model using groundwater concentration data. *Journal of Environmental Engineering*, 133(4), 372-379.
- Ibrahim, D., 2009. Using the excel spreadsheet in teaching science subjects. *Procedia-Social and Behavioral Sciences*, 1(1), 309-312. DOI: 10.1016/j.sbspro.2009.01.058
- Jannach, D., Schmitz, T., Hofer, B., Wotawa, F., 2014. *Journal of Systems and Software*. DOI: 10.1016/j.jss.2014.03.058
- Jisc Digital Media, 2019. Accessed: November 10, 2019. Link: <https://www.webarchive.org.uk/wayback/archive/20160101151319/http://www.jiscdigitalmedia.ac.uk/guides>
- Jones, W.R., Spence, M.J., Bowman, A.W., Evers, L., Molinari, D.A., 2014. A software tool for the spatiotemporal analysis and reporting of groundwater monitoring data. *Environmental Modelling & Software*, 55, 242-249. DOI: 10.1016/j.envsoft.2014.01.020
- Jurado, A., Vázquez-Suñé, E., Carrera, J., Tübau, I., Pujades, E., 2015. Quantifying chemical reactions by using mixing analysis. *Science of the Total Environment*, 502, 448-456. DOI: 10.1016/j.scitotenv.2014.09.036
- Kruchten, P.A., 2003. *The Rational Unified Process: An Introduction*. Boston (MA United States). Addison-Wesley Longman Publishing Co., Inc. 75 Arlington Street, Suite 300, 320pp. ISBN: 978-0-321-19770-2.
- Krueger, C., 1992. Software reuse. *ACM Computing Surveys*, 24(2), 131-183.
- Lucredio, D., Almeida, E.S.D., Prado, A.Ed., 2004. A Survey on Software Components Search and Retrieval. In: 30th IEEE EUROMICRO Conference, Component-Based Software Engineering Track. Rennes (France), IEEE Computer Society Press, Los Alamitos, 152- 159.
- Lucrécio, D., dos Santos Brito, K., Alvaro, A., Garcia, V.C., de Almeida, E.S., de Mattos Fortes, R.P., Meira, S.L., 2008. Software reuse: The Brazilian industry scenario. *Journal of Systems and Software*, 81(6), 996-1013. DOI: 10.1016/j.jss.2007.08.036
- Marcus, A., 1995. *Principles of Effective Visual Communication for Graphical User Interface Design*. In: Readings in Human-Computer Interaction. Baecker, Grudin, Buxton and Greenberg, Morgan Kaufman, Palo Alto, 2nd edition, 425-441. ISBN: 1-55860-246-1.
- Molano, C., 2013. *Groundwater Spreadsheets: Efficient and Practical Resource for Solving Simple and Complex Flow, Pollution, and Environmental Problems*. William A. McElhiney Distinguished Lecture Series in Water Well Technology, 2013 NGWA Groundwater Expo and Annual Meeting, NGWA.
- Parkhurst, D.L., Appelo, C.A.J., 2013. Description of input and examples for PHREEQC version 3—A computer program for speciation, batch-reaction, one-dimensional transport, and inverse geochemical calculations. U.S. Geological Survey Techniques and Methods, book 6, chap. A43, 497pp. Available at: <http://pubs.usgs.gov/tm/06/a43/>

- Phong, V.V., Kumar, P., Drewry, D.T., Quijano, J.C., 2012. A graphical user interface for numerical modeling of acclimation responses of vegetation to climate change. *Computers & Geosciences*, 49, 91-101. DOI: 10.1016/j.cageo.2012.07.007
- Robin, B., 1995. Supporting geoscience with Graphical-User-Interface internet tools for the Macintosh. *Computers & Geosciences*, 21(6), 737-751.
- Rossetto, R., De Filippis, G., Borsi, I., Foglia, L., Cannata, M., Criollo, R., Vázquez-Suñé, E., 2018. Integrating free and open source tools and distributed modelling codes in GIS environment for data-based groundwater management. *Environmental Modelling and Software*, 107, 210-230. DOI: 10.1016/j.envsoft.2018.06.007
- Scheiber, L., Ayora, C., Vázquez-Suñé, E., Cendón, D., Soler, A., Custodio, E., Baquero, J.C., 2015. Recent and old groundwater in the Niebla-Posadas regional aquifer (southern Spain): Implications for its management. *Journal of Hydrology*, 523, 624-635. DOI: 10.1016/j.jhydrol.2015.01.076
- Scheiber, L., Ayora, C., Vázquez-Suñé, E., Cendón, D., Soler, A., Baquero, J.C., 2016. Origin of high ammonium, arsenic and boron concentrations in the proximity of a mine: Natural vs. anthropogenic processes. *Science of the Total Environment*, 541, 655-666. DOI: 10.1016/j.scitotenv.2015.09.098
- Scheiber, L., Ayora, C., Vázquez-Suñé, E., 2018. Quantification of proportions of different water sources in a mining operation. *Science of the Total Environment*, 619-620, 587-620. DOI: 10.1016/j.scitotenv.2017.11.172
- Serrano-Juan, A., Vázquez-Suñé, E., Monserrat, O., Crossetto, M., Hoffmann, C., Ledesma, A., Criollo, R., Pujades, E., Velasco, V., Garcia-Gil, A., Garcia-Alcaraz, M., 2016. Gb-SAR interferometry displacement measurements during dewatering in construction works. Case of La Sagrera railway station in Barcelona, Spain. *Engineering Geology*, 205, 104-115. DOI: 10.1016/j.enggeo.2016.02.014
- Serrano-Juan, A., Pujades, E., Vázquez-Suñé, E., Velasco, V., Criollo, R., Jurado, A., 2018. Integration of groundwater bypass facilities in the bottom slab design for large underground structures. *Tunnelling and underground space technology*, 71, 231-243. DOI: 10.1016/j.tust.2017.07.020
- Suri, P.K., Neeraj Garg, 2008. Simulator for evaluating Reliability of Reusable Components in a Domain Interconnection Network. *International Journal of Computer Science and Network Security*, 8(3), March 2008, 251-259.
- Torlapati, J., Prabhakar Clement, T., 2013. Benchmarking a Visual-Basic based multi-component one-dimensional reactive transport modeling tool. *Computers & Geosciences*, 50, 72-83. DOI: 10.1016/j.cageo.2012.08.009
- Tubau, I., Vázquez-Suñé, E., Jurado, A., Carrera, J., 2014. Using EMMA and MIX analysis to assess mixing ratios and to identify hydrochemical reactions in groundwater. *Science of the Total Environment*, 470-471, 1120-1131. DOI: 10.1016/j.scitotenv.2013.10.121
- Tubau, I., Vázquez-Suñé, E., Carrera, J., Valhondo, C., Criollo, R., 2017. Quantification of groundwater recharge in urban environments. *Science of the Total Environment*, 592, 391-402. DOI: 10.1016/j.scitotenv.2017.03.118
- Vázquez-Suñé, E., Carrera, J., Tubau, I., Sánchez-Vila, X., Soler, A., 2010. An approach to identify urban groundwater recharge. *Hydrology and Earth System Sciences (HESS)*, 14, 2085-2097. DOI: 10.5194/hess-14-2085-2010
- Velasco, V., Tubau, I., Vázquez-Suñé, E., Gogu, R., Gaitanaru, D., Alcaraz, M., Serrano-Juan, A., Fernández-García, D., Garrido, T., Fraile, J., Sánchez-Vila, X., 2014. GIS-based hydrogeochemical analysis tools (QUIMET). *Computers & Geosciences*, 70, 164-180. DOI: 10.1016/j.cageo.2014.04.013
- Verplank, W.L., 1985. Graphics in human-computer communication: Principles of graphical user-interface design. *Proceedings of the IFIP-IMIA Second Stockholm Conference on Communication in Health Care*. Stockholm (Sweden), H.E. Peterson and W. Schneider, 13-130.
- Wang, J.-P., Huang, D., 2012. RosenPoint: A Microsoft Excel-based program for the Rosenblueth point estimate method and an application in slope stability analysis. *Computers & Geosciences*, 48, 239-243. DOI: 10.1016/j.cageo.2012.01.009
- Wang, J.-P., Huang, D., Yang, Z., 2012. Deterministic seismic hazard map for Taiwan developed using an in-house Excel-based program. *Computers & Geosciences*, 48, 111-116. DOI: 10.1016/j.cageo.2012.05.014
- Wang, J.-P., Huang, D., Cheng, C.-T., Shao, K.-S., Wu, Y.-C., Chang, C.-W., 2013. Seismic hazard analyses for Taipei city including deaggregation, design spectra, and time history with excel applications. *Computers & Geosciences*, 52, 146-154. DOI: 10.1016/j.cageo.2012.09.021

Manuscript received August 2019;
revision accepted March 2020;
published Online June 2020.

APPENDIX I

This appendix provides a compilation of the most basic code sentences that allows any program developer to create and design any similar software comparable to that presented above. Every title contains different code examples for performing the title action. Figure A1 locates each action in the decision flow diagram steps.

‘Select a Cell: A1
 Range(“A1”).Select
 Cells(1, 1).Select
 Range(“B2”).Offset(-1, -1).Select

‘Select a range A1:C1
 Range(“A1:C1”).Select
 Range(“A1”, “C1”).Select
 Range(Cells(1, 1), Cells(1, 3)).Select
 Range(“A1”).Resize(1, 3).Select

‘Concatenate words and variables
 Variable = “C”
 Range(“A1:” & Variable & “1”).select

‘Select final value in a row or column
 FinalRow = Cells(Rows.Count, 1).End(xlUp).Select
 FinalColumn = Cells(1, Columns.Count).End(xlToLeft).Select

‘Avoid repetition
 Cells(1, 1).Select
 Selection.Value = “Test”

‘Use straightforward
 Cells(1, 1).Value = “Test”

‘Error management
 On Error Resume Next
 On Error GoTo 0
 On Error GoTo Label1
 Label1:

‘Hide many Rows and Columns at once
 Range(“A2:AA2”).SpecialCells(xlCellTypeFormulas, xlTextValues)
 .EntireColumn.Hidden = True
 Range(“B1:B50”).SpecialCells(xlCellTypeFormulas, xlTextValues)
 .EntireRow.Hidden = True

‘Open
 Workbooks.Open Filename:=Application.
 GetOpenFilename(, “Title”)
 Workbooks.Open (“C:\ OutputFile.out”)

‘Save as
 New_file_name=Application.GetSaveAsFilename(“C:\
 “)
 ActiveWorkbook.SaveAs Filename:=fname,
 FileFormat:=xlNormal

‘Another way of saving is printing line by line
 Open thisfile For Output As #1
 For j = 1 To FinalRow
 Print #1, Cells(j, 1).Value
 Next j
 Close

‘Execute external software
 Variable = Shell(“C:\Launch MIX\MIX_2.exe “;
 vbMaximizedFocus)
 Call Shell (“C:\ Program.exe C:\InputFile.txt”; 1)

‘Find
 Cells.Find(What:=”Word To Find”, After:=ActiveCell,
 LookIn:=xlFormulas, _
 LookAt:=xlPart, SearchOrder:=xlByRows,
 SearchDirection:=xlNext

‘Working with charts
 ‘Add charts
 Charts.Add
 ‘Add data to the chart
 ActiveChart.SetSourceData = Range(“A1:A5”)
 ‘Define the chart type
 ActiveChart.ChartType = xlColumnClustered
 ‘Very important, store its name somewhere to future
 references to the chart
 ActiveChart.Location Where:=xlLocationAsObject,
 Name:=”Sheet1”

‘Fasten the code
 Application.ScreenUpdating = False
 Application.Calculation = xlCalculationManual
 Application.EnableEvents = False
 ActiveSheet.DisplayPageBreaks = False
 ‘...Code...
 Application.ScreenUpdating = True
 Application.Calculation = xlCalculationAutomatic

Application.EnableEvents = True
 ActiveSheet.DisplayPageBreaks = True

'Declare Public Variables to use its values in different macros
 Public Variable As Variant

'Use StrSafe when declaring Functions to adapt them from 32 to 64-bits
 #If VBA7 Then
 Declare PtrSafe Function ...
 #Else
 Declare Function ...
 #End If

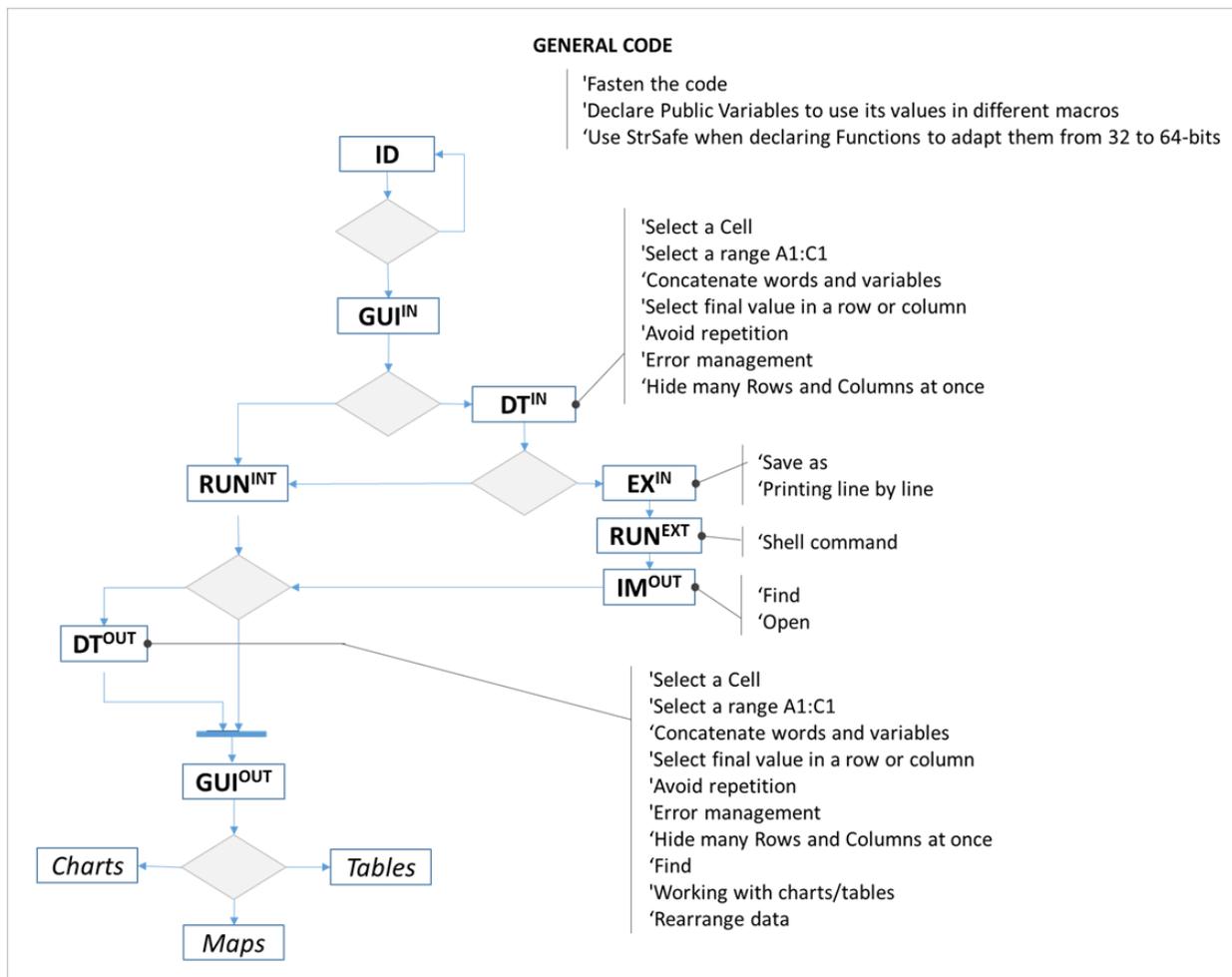


FIGURE 1. Decision Flow Diagram with some code comments related to the appendix.