UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL
INSTITUTO DE INFORMÁTICA
PROGRAMA DE PÓS-GRADUAÇÃO EM COMPUTAÇÃO

LEONARDO HAX DAMIANI

SHPECK - A Geochemical Speciation Modelling Software

Prof. Dr. Carla Maria Dal Sasso Freitas
Advisor

Prof. Dr. Anthony J. Park
Coadvisor

Porto Alegre, October 2015

SUMMARY

LIST OF ABBREVIATIONS AND ACRONYMS 5

LIST OF FIGURES 7

ABSTRACT 8

1 INTRODUCTION 9
1.1 Objectives of this work 10

2 BASIC CONCEPTS 12
2.1 Computer Science Principles 12
2.1.1 Computer Processing and Modelling 12
2.1.2 Software Architecture and Design 13
2.1.3 Software Development 15
2.2 Hydrogeochemistry Principles 16
2.2.1 Introductions to Thermodynamics 16
2.2.2 Hydrochemical processes 23
2.2.3 Geochemical Modelling 24
2.3 Summary 26

3 COMMERCIAL SOFTWARES REVIEW 27
3.1 Geochemical Speciation Modelling Codes 27
3.1.1 *EQ3/6* 27
3.1.2 *PHREEQC* 30
3.1.3 *MINTEQ* 33
3.1.4 *SOLMINEQ.88* 38
3.2 Commercial Software Review Discussion 40
3.3 Summary 43

4 *SHPECK* - SPECIATION MODEL 44
4.1 Specification 44
4.2 Architecture 44
4.2.1 Technical Specification 45
4.3 Governing equations 45
4.4 Numerical Method 46
4.4.1 Gauss-Newton simple example 47
4.5 Algorithm 48
4.5.1 Complexity of the algorithm 48
4.6 Software Engineering 48

4.6.1 Program organisation 48
4.6.2 Data flow 48
4.7 Graphical User Interface 48
4.8 Database 48
4.9 Summary 49

5 VERIFICATION AND VALIDATION 50

6 CONCLUSION 51

REFERENCES 52

LIST OF ABBREVIATIONS AND ACRONYMS

a	Activity
ASCII	American Standard Code For Information Interchange
CRUD	Create / Read / Update / Delete
CPU	Central Processing Unit
DBH	Debie-Hueckel
E_a	Activation Energy
Eh	Redox Potencial
FLOPS	Floating-Point Operations per Second
GUI	Graphical User Interface
GWB	The Geochemist's Workbench
HCI	Human-Computer Interaction
I	Ionic Strength
IAP	Ion Activity Product
K	Equilibrium Constant
k_{diss}	Dissolution rate constant
k_0	Pre-exponential (Arrhenius) factor
m	Molality
M	Molarity
MVC	Model-View-Controller
OS	Operating System
pH	Power of Hydrogen
R	Universal Gas Constant
SI	Saturation Index
T	Temperature
UI	User Interface
USGS	U.S. Geological Survey

γ	Activity coefficient
β_i	Stability Constant

LIST OF FIGURES

Figure 2.1: Example of Linear regression 19
Figure 2.2: Example of Linear regression 20
Figure 2.3: Example of Polynomial Regression 20

Figure 3.1: User interface of the PHREEQC for Windows (*PfW*) 32
Figure 3.2: Example of the PHREEQC Notepad++ plugin 33
Figure 3.3: PHREEQC software called from the Notepad++ Plugin 33
Figure 3.4: PHREEQC software called from the Notepad++ Plugin 34
Figure 3.5: *MINTEQA2* initial menu options 37
Figure 3.6: *MINTEQA2* main menu 37
Figure 3.7: *MINTEQA2* Level 1 informations 37
Figure 3.8: *PRODEFA2*'s example of adding a specific aquoues specie (Na^+) . . 38

Figure 4.1: Architecture of the *SHPECK* software 45

ABSTRACT

HERE WILL COME THE ABSTRACT

Keywords: Geochemical Modelling, Chemical Equilibrium, Geochemical Speciation, Multiphase System, Software Engineering, Computer Science.

1 INTRODUCTION

Geochemical modelling corresponds to the design of the reactions that occur in a geological structure through the usage of chemical properties (either thermodynamics and kinetics) to describe it. The need to understand the Earth's interior (both at high-temperature - magma - and low-temperature - aqueous solutions near the surface) motivates the effort in this area of study with the development of models and simulations. The applications of geochemical models are essential in several environmental problems, such as calculating the composition of natural waters, measuring flowing groundwater or surface water and the formation and dissolution of rocks and minerals in geologic formations. A geochemical speciation modelling software is responsible for calculating the distribution of dissolved species between free ions and aqueous complexes and also saturation indexes for different minerals.

Any model requires three major components: specific information describing the point of interest; the equations that drive and solve the model; and the model output. A model is an object represented by a set of mathematical expressions previously thought to represent natural processes and output the results of these calculations - something experimentally verifiable. In this sense, a model is a system capable of prediction which uses observational data as input and produces results of past examination. The thermodynamics and kinetics data used to establish the reactions and mimic the nature are directly responsible for the accuracy and precision of the geochemical model.

In this work, we develop a software that through the stoichiometry formulation calculates the chemical equilibrium of a geochemical system using the approach of imposing mass-balance conditions according to the species of the system. This process is known as chemical speciation, and the software was named as *SHPECK*. It accepts any general combination of elements, species and reactions, allowing the user to create different environments, simulations and, therefore, fully control any aspect and configuration of the model. Also in this work, we show a thorough analysis of the available existing solutions, and we made clear the uniqueness of our computational approach to the geochemical modelling problem.

Using a high-level and object-oriented programming language, we could implement an efficient solution that models geochemical speciation. *SHPECK* provides an interactive and intuitive user interface - unique among geochemical speciation software - as well as the support of a built-from-the-ground database structure that handles the management of the whole information used by *SHPECK*. These two contributions are presented as the result of an extensive study about the available software normally in use to perform geochemical speciation simulations.Their flow of information (input and output) are old, complexes and prone to error. It is also important to mention that these software fetch the information from flat file databases. Both of these characteristics are responsible for frequent errors, problems and wrong interpretations.

The principles of chemical equilibrium calculation rely on the law of conservation of mass (also known as the principle of mass conservation), stated by Antoine Lavoisier, and chemical speciation, which was presented by Garrels and Christ (GARRELS, 1965). The law of conservation of mass establishes that the total mass of an isolated system will remain constant and is independent of any chemical and physical changes taking place within the system. Therefore, the challenge of chemical equilibrium calculations is finding the number of moles that satisfies a system of equilibrium constraints at the moment where forward and reverse reactions rates are the same. These constraints are organized in a form of linear conservation equations, which may be expressed in the form of either linear algebraic atom and charge balance equations or chemical equations (**?**). For the sake of simplicity, in this work we will only deal with chemical equilibrium and not with chemical kinetics calculations since the first one requires only the solution of algebraic equation. It is planned to integrate kinetics reactions in the future.

The system of equations will drive and represent all the interactions between the components of the simulation. Newton-Raphson's method uses the previous guess for the equilibrium calculation in a subsequent step, recursively, until finding a suitable solution that satisfies the system and the convergence criteria. One must note that the initial guess is generated automatically and used as a seed for the iterations. This method requires the usage of a Jacobian matrix and a residual vector during the algebraic calculations. Geochemical modelling speciation has an important application in processes that occur in turbidite reservoirs. The process of the water coming from a salt dome contains a high concentration of salts as sodium (Na^+), chlorine (Cl^-) and potassium (K^+). Compactation, cementation, dissolution or recrystalization can be observed inside turbidites when this process happens. These processes might change drastically, for example, the porosity of the rock and, therefore, the storage capacity of oil and gas.

1.1 Objectives of this work

Soils and aquifers are heterogeneous, subsurface systems composed of a large number of components - dissolved salts, minerals, metals, gases, natural organics, microorganisms, animals and plants. The subsurface is one of the most complex systems studied by scientists and engineers today. Because of this, geochemical modelling has gained importance and is being accepted as a useful tool to interpret subsurface geochemical processes. Geochemical speciation is based on thermodynamics concepts and the assumption of chemical equilibrium in geochemical reactions. The idea of our own geochemical speciation software has emerged as an application where it would be possible to apply all the physical, chemical aqueous, geochemistry and linear algebra concepts, and develop a useful tool with an intuitive and interactive user interface. The most usual approach found in the area of geochemical modelling is a geochemical expert that develops a solution to solve his/her particular problem and generates a specific code/algorithm - a solution that most of the times is not very reliable and has no scalability. In this work, the approach was for the computer science expert to made the necessary efforts to understand and learn all the complex aspects of a geochemical speciation model and develop a software based on a solid knowledge in computer architecture, algorithms and software engineering.

The main purpose of this work is to develop a geochemical speciation modelling software following a structured computational approach.

The rest of this work is structured as follow. In chapter 2, we present an overview of the basic concepts needed and technical concepts involved in this work. Chapter 3 shows a thoroughly analysis and review of the commercial software available. Chapter 4 presents the *SHPECK* implementation with a detailed description of the whole system: design options; mathematical treatment and details; implementation and user interface (UI) details; algorithm validation and complexity; architecture and organization of the software as well as the database; data-flow; and iteration control. In chapter 5, it is presented a study case with an interesting and relevant scenario; the results that validate *SHPECK* and a broad comparison between solutions previously addressed in this work. Chapter 6 brings the conclusion of this work. Finally, this work contains an Appendix A, which is a presentation and an analysis of a linear algebra library used for the development of *SHPECK* called *Armadillo C++*. – THIS NEEDS TO BE VERIFIED LATER

2 BASIC CONCEPTS

At this point, it is important to understand all the different multidisciplinary aspects that are present in the development of a geochemical speciation modelling software. By definition, applying Computer Science to solve problems and create solutions requires to redefine obstacles outside normal boundaries and generate a new understanding of complex situations by thinking across two or more academic disciplines.

To develop this work, we had to delineate common goals for the different profiles that would take part on it along the way. All of them with a clear view of their roles and with a noiseless communication in any direction. Furthermore, it is vital and benefits crucially the whole work to be able to take advantage of all the different point of views from the diverse professionals profiles participating in this work. All of the mentioned above are fundamental factors to a successful multidisciplinary work.

Therefore, we present a meticulous and detailed review of all the basic concepts necessary to follow the development of this work, both from the computer science side and also from the hydrogeochemistry and geochemical modelling angle.

Along this chapter, we will first address topics of computer science relevant to this work: computer processing and modelling, software architecture and design, and software development. After, we explain the fundamentals hydro-geochemistry principles: an introduction to thermodynamics; and Hydrochemical processes; And to finish we focus on the geochemical modelling with a special section for it. If the reader feels comfortable with these topics, we recommend that you proceed to chapter 3.

2.1 Computer Science Principles

2.1.1 Computer Processing and Modelling

A processor is a small chip that resides in computers and electronic devices. Its job is to receive input, do something with it and provide the appropriate output. Modern processors, whose location is inside the *central processing unit* or *CPU*, can handle trillions of calculations per second and even work together to solve complex instructions. Within that *CPU* is an electronic clock responsible for creating series of synchronized electrical pulses. These pulses are the key to integrating all the computer's components and perform calculations with the data pulled from the memory. In 2013 the supercomputer *NUDT Tianhe-2* performed $33.86 Pflops$. $Pflops$ stands for Peta Floating-Point Operations Per Second and is the regular unit to measure computer performance. Nowadays, human's power of abstraction and modelling is what set the boundaries of the application and systems that we build. Countless factors influencing and driving the process are not a problem anymore for the computing power of the machines that are available. Not many years ago the bottleneck was on the computing power.

The advances in computer processing made possible scientific modelling, which generates part or feature of the real world to understand, define, quantify, visualize or simulate (HUMPHREYS, 2004). Modelling such systems require a previous knowledge of all the characteristics, the behavior of this domain and what is the goal of this modelling allied with a big *"piece"* of abstraction. Popular models are, for example, conceptual models, operational models, mathematical models, graphical models. The advantages of a model are: help us to communicate; allow us to clarify and test understanding; create credibility and accountability; organize the thoughts; simplify and solve problems;

The series of "orders" clearly expressed by the modeller is what defines the model. These stack of "orders" are known as algorithms - a series of instructions for how to do something. Instructions that tell the computer how to make decisions and when to do calculations - which are different according to the type of model.

Among the many computer science areas, there is one that studies the complexity of algorithms. As algorithms are programs that perform a series of instructions, complexity analysis allows us to measure how fast a program is when it performs computations. The analysis enables us to explain how an algorithm behaves in the worst case scenario, for instance. This information will come in hand when we analyze the complexity of *SHPECK*'s algorithm on chapter 4.

2.1.2 Software Architecture and Design

Software Architecture is the process of finding a structured solution that achieves all of the technical and operational requirements also addressing attributes as performance, security, value for the user and management. The architecture and design of software are the *art* of considering all of the several factors and tracing the best path available. Without compromising the impact on quality, interface, database structure, performance, maintainability and success of the software.

The software architecture is responsible not only for the algorithms and the data structure but also by the organization, communication, synchronization, functionality and design of the desired elements, scaling and performance. There is no well-defined recipe for a good software architecture - it takes time, practice and efforts to start taking the right paths and weighing the options according to the needs. Recognizing paradigms and building relationships among systems can be a handful tool to perform successfully as a software architect. The software architect is responsible for structuring the software with a consolidated and dense foundation - anything other than this implies a risk for the application. Studying the scenarios and requirements before designing the application is a must. Poor architecture results in deployment problems, instability, lack of support and the complete failure of the software (sometimes even the entire business). A software architect aims to:

- Catalog all the requirements of the application, as well as the use cases and scenarios;

- Analyze and reduce the risk either for the application and for the business (if there is one involved);

- Be able to adapt the design decisions around the reality - that will most likely change over time;

- Develop a structure where the tradeoffs of all attributes are clear and the impact of any change will be controlled;

During the development of *SHPECK*, we adopt the Object-Oriented, also know by its abbreviation *OO* or *OOP*, architectural style. *OO* is a paradigm based on the division of responsibilities into reusable and self-sufficient objects, each one of them containing the data and the behavior desired to its functionalities and responsibilities. For the purpose of illustration we listed a few other important architectural styles: client/server; domain driven design; service-oriented architecture (SOA);

Before defining the architecture of *SHPECK*, we have analyzed points as attributes, application type, technologies and deployment options. Only after this section was possible to identify which of the design architectures would fit best to our needs - small refinements were done along the way (which is normal).

2.1.2.1 Software Architecture Principles

The origin of software architecture princippes is the need to minimize costs, address properly maintenance requirements and promote the *Seven Basic Principles of Software Engineering* as in (BOEHM, 1983), which are:

1. Manage using a phased life-cycle plan;

2. Perform continuous validation;

3. Maintain disciplined product control;

4. Use modern programming practices;

5. Maintain clear accountability for results;

6. Use better and fewer people;

7. Maintain a commitment to improve the process;

Along these principles, it is mandatory that the software architect or designer to see the large picture of the software that is under his management. The large picture is responsible to make sure that no feature is overlaping (nor duplicating) with another - this will lead to a low coupling and highly cohesive software.

2.1.2.2 Design Principles

Designing software is composing a structure with different layers responsible for various tasks or properties. This layering must be consistent with any operation and must respect the hierarchy as well as the orientation of this structure. These layers must be connected but never overlapping themselves: duplicating properties/functionalities/responsibilities is a mistake and prone to error. Overlapping layers is the signal of potential inconsistencies and elevated software maintenance costs. Design patterns are also one important term to keep in mind. Establishing a coding style, naming standards and conventions provide a consistent model that will make the software's life longer and more adaptable.

2.1.3 Software Development

Software development is a process that requires extremely careful planning and execution to meet the proposed goals. The proposed goal is software, but sometimes people forget that to achieve this goal is necessary many hours of computer programming, documenting, testing, bug fixing and decisions making. Software development may also include research, new development, prototyping, modification, reuse, re-engineering and maintenance. The following sections discuss the most interesting and relevant points.

2.1.3.1 Life Cycle of a Software Development Projet

This topic is relevant to clarify that lots of work are done before writing any line of code. We can mention tasks as requirements definition; functional specification; architecture and design decisions; implementing and testing; software deploy; documentation; and maintenance; There may be additional functions according to the reality of each software development, but the idea of life cycle must be a clear notion in the reader's mind.

2.1.3.2 Software Engineering

The contrast in time doesn't destroy the importance of software engineering among different periods as can be verified in the following quotes:

- *"The establishment and use of sound engineering principles in order to obtain economically software that is reliable and works efficiently on real machines."*, from (BAUER, 1968).

- *"Software engineering is the application of a systematic, disciplined, quantifiable approach to the development, operation, and maintenance of software, and the study of these approaches; that is, the application of engineering to software."* by the *IEEE Computer Society's Software Engineering Body of Knowledge* from 2004.

The understanding that overlaps in both quotes is that to achieve a proper *software*, engineering principles (for example management issues, documentation, infrastructure, directing teams, scheduling and budgeting) are necessary and will be fundamental to reach that goal.

2.1.3.3 Database

The database is responsible for organizing, storing and retrieve the data so it can be used efficiently and smoothly. A collection of schemes composes it in a way that it supports processes requiring information that are utilized by the application's internal operations. They are organized according to their approach: relational database; tabular database; distributed database; OO database; and flat file database;

Among several types of database, the software engineering previously done should identify which of them suits better the needs of the software. Details of *SHPECK*'s database are presented in chapter 4.

*2.1.3.4 User Interface (*UI*) and Human-Computer Interaction (*HCI*)*

Psychology, ergonomics, engineering, graphic design and others fields of study influence the *UI* and *HCI* areas from computer science. Both areas take into account and are products of how humans interact with computers.

Good *UI* are not user-expensive nor task-expensive, they behave naturally as the extension of the user's needs and desires. The software will easily bring more value to its users if the *HCI* happens in a mutually beneficial way - by reaching the software's goal and not being an embarrassment or annoyance for the user. Losses of productivity, efficiency, money and usability are expected consequences from a software that has skipped the preparation parts to the development of *UI* and *HCI*.

UI and *HCI* were extensively studied and analysed along the development of *SHPECK*.

2.2 Hydrogeochemistry Principles

2.2.1 Introductions to Thermodynamics

In thermodynamics, equilibrium is a state of dynamic balance where the ratio of the product and the reactant concentrations is constant. There are three general approaches to calculating the composition of a solution at equilibrium (PETRUCCI, 2007).

1. Manipulation of equilibrium constants (K): The final concentrations are achieved by mathematical handling of the equilibrium constants; the idea is to express all the parts in terms of the measured equilibrium constant and initial conditions. Thermodynamics databases contain the value for the equilibrium constants obtained through experiments. Demonstration of this can be found in (KEHEW, 2000). The disadvantages of this method is when using this method for a huge number of reactions it may never converge.

2. Gibbs Energy of the system: At equilibrium, the Gibbs Energy (G) is at a minimum. When the object of the study is a close system - no particles entering nor leaving - the total number of atoms of each element will remain constant, therefore, achieving the minimum free energy. Due to the complexity in demonstrating how this method works, it will be supressed here. An interesting algorithm for equilibrium calculation that uses Gibbs energy is described in (ALLAN, 2015). One of the disadvantages of this method lies in the effect of species which appear only in tiny quantities at equilibrium.

3. Manipulation of mass-balance: The total concentration of species that compose the system is the base for this method, (SMITH, 1980) explains this stoichiometric formulation approach. This method takes into account the stoichiometric approach among the species, which generates a system of non-linear mass-action equations. Mass-balance manipulation is the method chosen for this work, and the details are explained further in this work.

Stoichiometric approaches have two general advantages over non-stoichiometric: in the case of real systems and for multiphase problems - in which singularities can occur in the linear equations (SMITH, 1980). It is important to remind the reader that any of the methods described above are equivalent and can be verified in (ZEGGEREN, 1970)

It is important to mention that any analysis resulting from a water sample must be carefully taken. Any geochemical investigation is useless if the integrity of the water of the solid phase is compromised. Results of interpretation and modelling might be incorrect if the sampling was not done properly. A principal objective is to obtain a water sample with the same chemical composition as those of water in its original environment, for example, an aquifer or a surface water. (DEUTSCH, 1997)

2.2.1.1 Thermodynamic Equilibrium Reactions

There are mainly two ways to describe thermodynamic equilibrium reactions: Equilibrium and Kinetic. Both of them formulates a closed system and describe the position of the maximum thermodynamic equilibrium. Equilibrium is the moment where there is no more chemical energy to alter the distribution of mass between reactants and products in the system. The way to model a reaction depends on its rate: an equilibrium reaction is relatively fast on the mass transport process while the kinetic reaction is slow. Therefore, when applying an equilibrium model to a reaction is assumed that the whole mass transfer happens at the same time when the reactant and product are putted together, and this will configurate an equilibrium situation. If the reaction rate is slow, it requires a kinetic description of the reaction. On this work, it will be addressed only equilibrium reactions. (NORDSTROM, 1986)

Assuming the independent equilibrium reactions:

$$0 \rightleftharpoons \sum_{i=1}^{N} v_{ji}\alpha_i \qquad (j = 1, ..., M) \tag{2.1}$$

where v_{ji} is the stoichiometric coefficient of the *i-th* species in the *j-th* reaction; and M represents the number of reactions and N the number of species, with $M < N$. The sign convention is to assign the stoichiometric coefficient negative for reactants and positive for products. Assuming that all the reactions in the system are in equilibrium, the chemical system must also satisfy the mass-action equations:

$$K_j = \prod_{i=1}^{N} a_i^{v_{ij}} \qquad (j = 1, ..., M) \tag{2.2}$$

where K_j denotes the equilibrium constant of the *j-th* reaction; a denotes the activity of the *i-th* chemical species. The equilibrium constant depends on the temperature of the system; therefore, the equilibrium constant needs to be calculated according to the temperature of the system.

It has been known that the driving force of a chemical reaction is related to the concentration of the constituents that are reaction and the concentrations of the products of the reaction. The law of mass action states that any reaction will proceed to the right (dissolution) or to the right (precipitation) until the mass-action equilibrium is achieved, important to keep in mind that it may take years or even thousands of years for that equilibrium to be achieved and after a disturbance in the system, such as an addition of reactants, removal of products, changes in the temperature or pressure, the system will continue to proceed toward this new equilibrium (if the disturbances are frequent compared to the reaction rate, equilibrium will never be achieved) (FREEZE, 1979). Each of the dissolved species will have one representation of the nonideal behavior of components in the solution, which is called *activity* and is presented in details later on this chapter.

Kinetic descriptions is applicable to any reaction but it is needed necessary to describe reactions that are slow in relation to mass transport. The following reaction has a k_1 and k_2 rates for the forward and reverse reactions, respectively

$$aA + bB \underset{k_1}{\overset{k_2}{=}} dD + eE \tag{2.3}$$

Each ion has a reaction rate related to the stoichiometry and is expressed as

$$-\frac{r_A}{a} \quad = \quad -\frac{r_B}{b} = \frac{r_D}{d} = \frac{r_E}{e} \tag{2.4}$$

where a, b, d and e are stoichiometric coefficients of each one of the ions in the reaction. r_A, r_B, r_D and r_E are reaction rates, and they describe the time rate of change of concentration as function of rate constants and concentration. Each one of them express the rate of change at the chosen ion as the difference between the rate at which the component is being used in the forward reaction and generated in the reverse reaction and is described as follow

$$r_A \quad = \quad -k_1(A)^{n1}(B)^{n2} + k_2(D)^{m1}(E)^{m2} \tag{2.5}$$

where $n1, n2, m1$ and $m2$ are empirical stoichiometric coefficients. When there are reactions in parallel or series the rate laws are even more complex. The dissolution rate constant ($\mathrm{k_d}$iss) of a chemical reaction depends on temperatue. The relation between constant and temperature is given by the *Arrhenius equation*, described as

$$k_d iss = A * exp(\frac{-E_a}{R * T}) \tag{2.6}$$

where $\mathrm{k_0}$ is the pre-exponential (Arrhenius) factor, E_a is the activation energy, R is the universal gas constant, and T is the temperature in Kelvin. During the development of *SHPECK*, we will not deal with kinetic reactions.

2.2.1.2 Thermodynamic Equilibrium Constant

The *equilibrium constant* (*K*), also known as *stability constant*, is the value of the reaction quotient when the reaction has reached equilibrium, as stated in equation 4.1. *K* depends only on the temperature and on the ionic strength of the solution. According to known reactions' equilibrium constant value, it is possible to determine the value for at any temperature by a polynomial fitting technique or polynomial regression.

Equilibrium constants are determined by measurements of the relevant concentrations of the species under differing experimental conditions. Concentrations of species can be measured in multiple ways, and the use of these values in modelling requires adjustment to the conditions in the system being modelled. These adjustments, as well as the differences in conditions and different methods for determination, can lead to uncertainty in chemical speciation constants.

Several thermodynamics database are available and really popular nowadays. They include reaction constants, reaction descriptions, solutes, species, enthalpy values, activity coefficient parameters, etc. During the development of *SHPECK* we selected the *Geochemist's Work Bench's* (*GWB*) database - it contains also the values of a 8^th degree polynomial which allows the user to calculate the equilibrium constant to any temperature. Also another source of data used along this work is (PALANDRI, 2004).

In geochemical modelling, the usage of polynomial regression is specifically to calculate the equilibrium constant of the compound at the desired temperature. Polynomial regression is one of several methods of curve fitting, which is a process of constructing a curve that hast the best fit to a series of data points. The polynomial regression is a statistic method that is a form of linear regression in which the relationship between the independent variable *x* and the dependent variable *y* is modelled as an *nth* degree polynomial. In our case, the polynomial regression is necessary in order to acchieve the equilibrium constant for compounds found in the solution system. Polynomial regression is considered to be a special case of multiple linear regression. A polynomial is a function that takes the form

$$f(x) = c_0 + c_1 * x + c_2 * x^2 + ... + c_n * x^n \tag{2.7}$$

where n is the degree of the polynomial and c is a set of coefficients. Polynomial regression models are usually solved using the method of least squares. Likewise performing polynomial regression with a degree 0 on a set of data returns a single constant value. It is the same as the mean average of that data. This makes sense because the average is an approximation of all the data points, as shown in figure 2.1. The average line mostly follows the path of the data points. Thus the mean average is a form of curve fitting and likely the most basic.

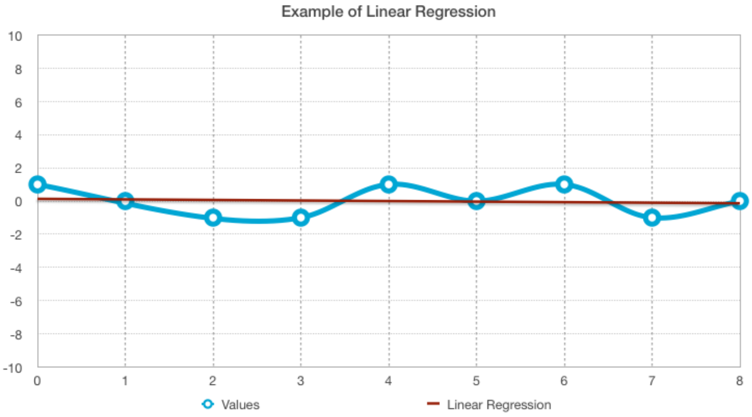


Figure 2.1: Example of Linear regression

Linear regression is polynomial regression of degree 1, and generally takes the form

$$f(x) = c_0 + c_1 * x \tag{2.8}$$

where c_0 is the y-intercept and c_1 being the slope. Figure 2.2 shows clearly that the linear regression line running along the data points approximate the data. Mean average and linear regression are the most commom forms of polynomial regression, but not the only.

The next step of polynomial would be the quadratic regression, now the regression becomes non-linear and the data is not restricted to straight lines. With figure 2.3 is possible to visualize a data with a quadratic regression trend line. Basically, the idea is simple: find a line that best fits the data which is find the coefficients to a polynomial that best fits the data.

Polynomial regression is an overdetermined system of equations that uses least squares as a method of approximating an answer. To understand this, some linear algebra is required.

2.2.1.3 Activity of a solute

Activity (a_i) is *"thermodynamic concentration"* (or informally known as *"effective concentration"*). It is calculated as a product of activity coefficient and concentration (where *i* means the solute involved):

$$a_i = \gamma_i * m_i \tag{2.9}$$

Activity coefficient (γ_i) is a function of ionic strenght (I), which is a measure of the concentration of ions in the solution.

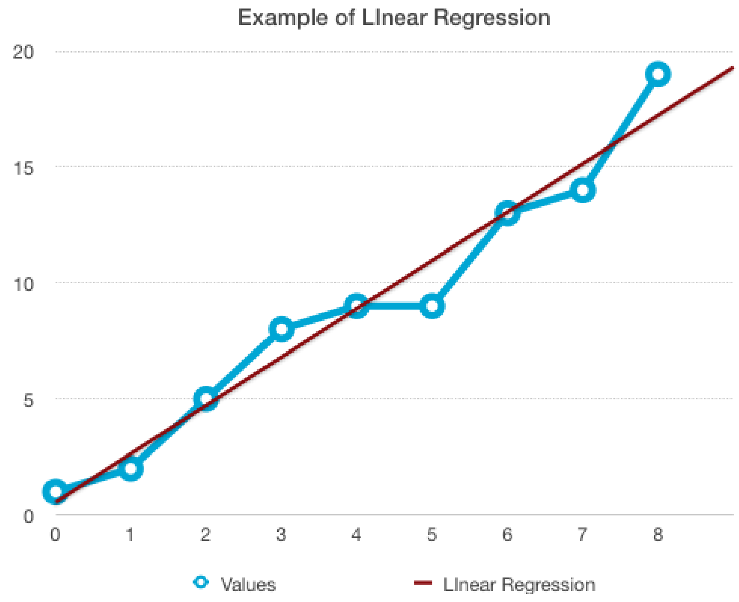


Figure 2.2: Example of Linear regression

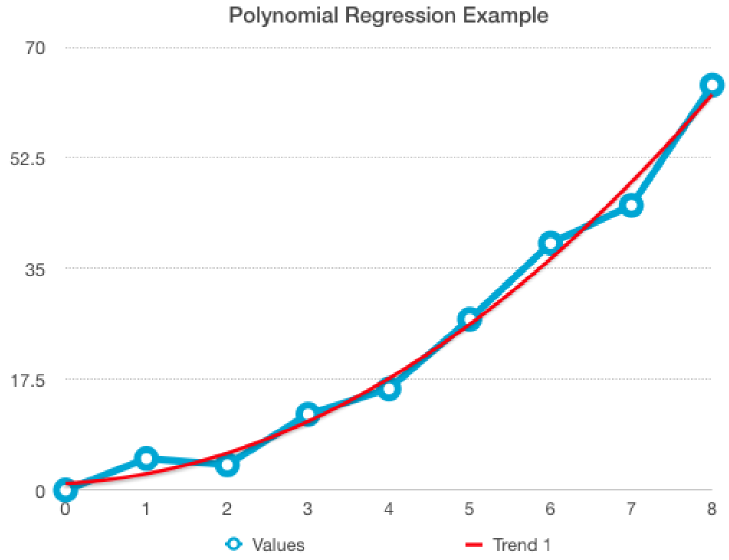


Figure 2.3: Example of Polynomial Regression

2.2.1.4 Ionic strength

Mathematically the ionic strength of the solution is calculated according to

$$I = 0.5 \sum M_i z_i^2 \tag{2.10}$$

where M is the molar concentration of the specie i having a charge z.When I increases, activity coefficients decrease. In very diluted solutions activity coefficient is equals to *1.0* and activity is equal to concentration. The decreasing trend is related to the "cage" of opposite charge particles around ions. There is reversal of the trend in extremely concentrated solutions (brines) because beyond ionic strenght of about $1mol/L$ there is an increase of activity coefficients with increasing ionic strength. This is related to decreasing amount of free water because most of water is already bound around dissolved species.

For a matter of explanation, we will calculate the ionic strength of a $CaCl_2$ solution (composed by $0.5mol$ of Ca^{+2} and $1mol$ Cl^{-1}):

$$I = \frac{1}{2}(z_{Ca}^2[Ca^{+2}]) + \frac{1}{2}(z_{Cl}^2[Cl^{-1}]) \tag{2.11}$$

$$I = \frac{1}{2}(2_{Ca}^2[Ca^{+2}] + (-1)_{Cl}^2[Cl^{-1}]) \tag{2.12}$$

$$I = \frac{1}{2}(4 * 0.5 + 1 * 1) \tag{2.13}$$

$$I = 1.5mol/L \tag{2.14}$$

2.2.1.5 Activity Coefficient

There are different methods to calculate γ for ions:

- Debie-Hueckel: They assumed that ions behave like spheres with charges located at their center points. The ions interact with each other by coulombic forces and the result of their analysis is as follows

  $$log\gamma_i = -Az_i^2\sqrt{I} \tag{2.15}$$

  where *A* is a constant that is a function of temperature, z_i is the ion charge and *I* is the ionic strength of the solution.

- Davies equations: Is a variation of Debie-Hueckel that can be used when the ionic strength is relatively high. The equation is as follow

  $$log\gamma_i = -Az_i^2\Big(\frac{\sqrt{I}}{1+\sqrt{I}} - 0.3I) \tag{2.16}$$

- B-dot: This model is presented as an activity model based on an equation similar to Davies and parameterized for solutions up to 3 molal ionic strength.

  $$log\gamma_i = -\frac{Az_i^2\sqrt{I}}{1+a_iB\sqrt{I}} + \dot{B}I) \tag{2.17}$$

  where *å* is the ion size for each specie and *A, B and* $\dot{B}$ are coefficients that vary with the temperature.

Important to mention that there are other methods available to calculate activity coefficients which are not going to be addressed here. Is important to keep in mind that pure solids have an activity equals to one. Each one of the methods has its advantages and limitations. Debye-Hueckel equations are simple to apply and extensible to include new species in the solution due to the fact that it requires a low number of arguments and specific arguments. Besides, Debye-Hueckel can be applied to the most important temperatures in the field of aqueous geochemist. Important to keep in mind that it works poorly when regarding moderate or high ionic strength. Regarding dissolution and precipitation there is clearly a reaction happening during these processes, which means that some reactions are not in equilibrium.

2.2.1.6 Saturation Index

The saturation index (*SI*) indicates the degree of saturation with respect to a given mineral, in other words, it defines if a reaction will be in equilibrium or not. *SI* is expressed as

$$SI \;=\; log(IAP/K) \tag{2.18}$$

when a mineral is in equilibrium with a solution the *SI* is zero, a negative *SI* indicates undersaturation and a positive *SI* supersaturation. Ion Activity Product (*IAP*) is calculated according to

$$IAP \;=\; \frac{[C]^c[D]^d}{[A]^a[B]^b} \tag{2.19}$$

where [A], [B], [C] and [D] are activitys of each ion. The interpretation of *IAP* is the following:

- IAP > K : The reaction is progressing from right to left, producing more products. In a ground water solution, the water is supersaturated.
- IAP = K : The reaction is in equilibrium, there is no flow neither to the right nor to the left. In a ground water solution, the water and the mineral are in equilibrium.
- IAP < K : The reaction is progressim from left to right, producing more reactants. In a ground water solution, the water is undersaturated.

With the *SI* approach is possible to predict the reactive mineralogy of the subsurface from the groundwater data without collecting samples of the solid phase and analyzing the mineralogy. If the *SI* for a mineral calculates to be less than zero, the aqueous solutions is undersaturated with respect to that mineral - which corresponds to the fact that the mineral will not precipitate and may dissolve in order to reach equilibrium concentrations. If the *SI* is greater than zero, then the mineral is not reactive and the mineral may precipitate from the aqueous solution (oversaturated). To conclude, when the *SI* is close to zero (it is ok to consider a small range of values to be in equilibrium) , it means that the water is saturated with respect to that mineral (ALLEY, 1993). From the mentioned before, it is possible to state the following:

- SI < 0 : Mineral is undersaturated;
- SI = 0 : Mineral is in equilibrium with the solution;
- SI > 0 : Mineral is oversaturated;

2.2.1.7 Hydrogeochemistry common units

Molarity (*M*), defined as mass in moles in 1 liter of solution and molality (*m*), defined as mass in moles in 1 kilogram of solution. In dilute solution molarity is approximately equal to molality. Concentration in miliequivalents per liter is concentration in milimoles per liter multiplied by charge of an ion.

2.2.2 Hydrochemical processes

2.2.2.1 Acid-Base Reactions

The importance of acid-base reactions is cleary when it is understood its influence on the pH. The pH is a master variable in charge of controlling chemical systems and is described as

$$pH = -log([H^+]) \tag{2.20}$$

where $[H^+]$ is the activity of the hydrogen ion. The interpretation of the values is as follows:

- pH < 7 : acid solution;
- pH = 7 : neutral solution;
- pH > 7 : basic solution;

The acid substance has tendecy to lose protons while a base substance has tendency to gain protons and the interation between acids and bases is called acid-base reactions and is described as

$$Acid_1 + Base_2 \quad = \quad Acid_2 + Base_1 \tag{2.21}$$

The reaction must be understood as that in the forward reaction, the proton lost by $Acid_1$ is gained by $Base_2$ and in the reverse reaction the proton lost by $Acid_2$ is gained by $Base_1$. The strength of an acid or base refers to the proportion of its protons are lost or gained.

2.2.2.2 Complexation and Speciation

A complexation is when an ion that forms by combining simpler cations, anions and sometimes molecules, this process facilitates the transport of potentially toxic substances and form what is called a complex. Due to the importance of this process in contamination problems it has acquired a huge importance in practical and commercial fields. A simple example of complexation is the following

$$Mn^{2+} + Cl^- \quad = \quad MnCl^+ \tag{2.22}$$

Calculation of distribution of metals mong complexes (*speciation*) involves the solution of a series of mass-law transport equations. The mass law equation of the reaction 2.22 is described bellow

$$K_{MnCl^+} \quad = \quad \frac{[MnCl^+]}{[Mn^{2+}][Cl^-]} \tag{2.23}$$

Each one of the complex has a variable associated called stability constant (β_i) and it contains the basic information necessary to determine how the total concentration of a metal in a solution is distributed as a metal ion and the other various complexes possible.

2.2.2.3 *Oxidation-Reduction Reactions*

Groundwater environemnt's reactions involve transfer of electrons between its components (gaseous, dissolved or solid constituents). As result, there are changes in the oxidation states of the reactants and products. It is important to stress that the oxidation number is a hypothetical charge that an atom would have if the ion or molecule were to dissociate. This state can be different according to the solution. During this work, *redox reactions* (as oxidation-reduction reactions are also known) are not going to be addressed. In order to get deeper understanding on this topic, we refer to (PETRUCCI, 2007)

2.2.2.4 *Adsorption and ion exchange*

Adsorption systems treat water by adding a substance, such as activated carbon or aluminia, to the water supply. Adsorbents attract contaminants by chemical and physical processes that cause them to *stick* to their surfaces for later disposal. This mechanism is often used to remove contaminants like *arsenic* or *fluoride* (mostly organic contaminants) from water reservoir. Ion exchange work simillarly but it is focused in inorganic contaminants in a particle-free water. Ion exchange is most often used to remove hardness or nitrate (mostly inorganic soluble molecules). During this work, adsorption and ion exchange are not going to be addressed. In order to get deeper understanding on this topic, we refer to (FREEZE, 1979)

2.2.3 Geochemical Modelling

The geochemical modelling is the design of the geochemical reactions responsible for the migration of dissolved species. Geochemical models can be divided into two groups:

- Geochemical Equilibrium Models: Based on the assumption of thermodynamic equilibrium reached in a relatively short time (no time factor is included in calculation). It takes in consideration only equilibrium reactions.

- Geochemical Kinetic Models: It takes into account also kinetic reactions and includes the time factor. As kinetic data is measured experimentally, there is still a lack of kinetic data available for many geochemical processes.

As mentioned before, this work will focus on the first one - *geochemical equilibrium models*.

Inside geochemical equilibrium models we can mention three divisions: speciation models; inverse models (also called mass balance models); forward models (also called reaction path models); and reactive transport (coupled) models. Regarding the relation to spatial coordinates, geochemical equilibrium models are considered *batch models* - which are basically closed vessels or reactors.

2.2.3.1 *Geochemical Speciation Modelling*

Speciation represents modelling based in the equilibrium of the system. A geochemical speciation modelling program calculates the distribution of dissolved species between free ions and aqueous complexes and also saturation indexes for different minerals. Sodium, for example, can be present in water as free ion Na^+, and also in the form of complexes with anions:

$$\mathrm{Na^+}_{total} = \mathrm{NaCl}_{aq} + \mathrm{NaOH} + \mathrm{Na^+} \tag{2.24}$$

where Na^+_{total} is total sodium concentration from chemical analysis. Na^+_{total} is a component (e.g., chemical formula unit used to describe a system) and Na^+, $NaCl_{aq}$ and NaOH are species (chemical entities which really exist in the system). Information about the distribution of dissolved species is important, for example, for risk assessment of contamination by metals because toxicity of metails depends on their speciation in solution. Carbonate complexes of metails, for example, are less toxic than their free ions. Saturadio index (SI) is used to determine the direction of geochemical processes. When *SI > 0* the mineral precipitates from the water and when *SI < 0*, the mineral dissolves in contact with the water, if it is present in solid phase. Field data necessary for input of speciation program are temperature, pH and results of laboratory chemical analysis (results from a sampling of the solution of interest).

Common problems solved using speciation programs are:

- There is a sample with high concentration of dissolved sodium and we need to know the distribution of sodium between Na^+ and different complexes (for example, $aCl$$NaCl_{aq}$ or NaOH) because different forms of sodium have different characteristics;

- There are ground water samples that had been in contact with granitic masses and we want to verify the possibility of precipitation of minerals like *Albite* (a plagioclase feldspar mineral whose formula is $NaAlSi_3O_8$).

Note that the details of several available programs are going to be presented and discussed in chapter 3).

The development of our software *SHPECK*, which is a geochemical speciation modelling software will be detailed, presented and thoroughly discussed in chapter 4. Also important to mention that this work is the first work that will completely guide anyone to generate a geochemical speciation modelling software from the ground.

2.2.3.2 Other Types Of Geochemical Modelling

- Inverse geochemical modelling This type of models, also known as mass balance models, are used when chemistry of groundwater and solid phase composition are already known, and reactions that have already happened should be determined. It is used when we have access to 2 hydraulically connected points and composition of solid phase between these points; with these data in hand, it is possible to calculate and produce the reactions that will explain the changes of the water's chemistry. This approach leads to some uncertainties: Stoichiometry of minerals in solid phase is not often well known; solution may be non-unique; and programs can produce several possible models for the same input; An interesting work about inverse geochemical modelling can be verified in (SHARIF, 2007).

- Forward geochemical modelling This type of models, also called reaction path models, are used for prediction of water chemistry evolution along a flowline. Initial water chemistry is known and the aim of the program is to predict water chemistry at some point along flow path. This kind of modelling introduce problems regarding kinetic and adsorption data, which are ofter missing and frequently limited.

2.3 Summary

- Importance of multidisciplinary problems: The amazing power of computer science to adapt itself to other areas of knowledge and do something brilliant is incredible. The best way to push the limits is by redefining obstacles outside normal boundaries and reach solutions based on new understanding of complex situations.

- Computer science principles: Understanding what is a software and how it connects with the *"machine world"* to the *"real world"* is something not trivial. We reinforce the idea that the extension of a software goes way beyond the lines of code and the interface showing up on the screen.We often find the comparison that building software is somehow like building a house - this certainly is a helpful example to understand everything that is behind a software. Both house and software require: estimating costs, thinking about the requirements, plans, rules, standards, best practices, specifications timelines, reviews, milestones, testing, alterations, handover and warranty.

- Hydrogeochemistry principles: Thermodynamics is one of the substantial basis for the history of physics, chemistry and the science in general as we know it. It is crucial to understanding the role that it has in the natural aspects of the world that we live. The comprehension of thermodynamics goes through the analysis, awareness and interrelations of the several factors that compose this complex *maze*. From the wide range of factors, we can mention the equilibrium and kinetic reactions, the activity and the activity coefficient of the solute, the ionic strength of the solution, the saturation index, etc.

3 COMMERCIAL SOFTWARES REVIEW

The importance of geochemical models has increased lately due to its variety of applications, but the first models date back to the 70's as in (WESTALL, 1976) and (WOLERY, 1979). Since then, these models are used to solve problems as speciation; determination of minerals' saturation indexes; adjustment of equilibrium for minerals; mixing of different waters; calculation of stoichiometric reactions; mixing of solids, fluids and gaseous phases; calculation of equilibrium/kinetic controlled reactions; reactive transport; mass-law calculations;

The quality of the chemical analysis depend on the methods used, thermodynamic data and theoretical concepts applied. Therefore, it is crucial to check the results and is clear that there will be some differences in the results according to the software used. Among the enormous variety of software available, some of them are developed for batch-type simulations only while others have transport capabilities. Several of them do not incorporate graphical interfaces and are written in FORTRAN, newer distributions are mainly written in C/C++ and due to proprietary reasons code is not distributed with the software. Important to mention that, even those who have an integrated graphical user interface (GUI), it is ofter very tedious and time-consuming to generate input files for groundwater simulation software.

The goal of this chapter is to note other codes that perform modelling of aqueous geochemical systems. It is not possible nor the purpose of this work to present all the existing codes but to critically review and compare some aspects of the codes. These systems are being widely used among the community and have enough documentation published in the open literature that makes a comparison effort sustainable and enjoyable.

During this work, we will focus exclusively on programs that do speciation models specifically. Which are the following: *EQ3/6*; *PHREEQC*; *MINTEQ*; and *SOLMINEQ*;

We will present each relevant detail about the geochemical point of view and then critically analyze and discuss from the computer science point of view. It's worth mentioning that in any aspect this review is meant to be impolite. Quite the opposite, we value their efforts and are thankful for the knowledge we could acquire by using and analyzing them. We ask the reader's comprehension to understand that sometimes is not possible to analyse the same aspects of different software due to the lack of information available.

3.1 Geochemical Speciation Modelling Codes

3.1.1 *EQ3/6*

EQ3/6 consists of two programs: EQ 3 is a pure speciation code whose results EQ 6 subsequentially process. It is a software package for geochemical modeling of aqueous systems written in FORTRAN77. EQ3/6 includes a speciation-solubility solver (useful for analyzing groundwater chemistry data, calculating solubility limits and determining whether certain reactions are in states of partial equilibrium or disequilibrium). It also offers a reaction path calculation that models water/rock interaction or fluid mixture. EQ3/6 supports several thermodynamic data files (these data files contain support for Davies, B-dot, Debye-Hueckel equations, as well as support data for standard state and activity coefficient-related). It is developed to run under UNIX, and the full package distribution is not free (it requires a license). (WOLERY, 1979) (WOLERY, 1990) (WOLERY, 1992).

3.1.1.1 Input/Output Options

The *datafilekey* and *inputfile* given to the program as arguments must be consistent with the options and methods. For example, if they have different methods for calculating the activity coefficient there will be problems, and the results will be meaningless. Another point to be taken into consideration and that follows the same idea, the *inputfile* must be using chemical data (for example, elements, species and compounds) that is known by the *datafilekey*.

Inside each file, there are a series of *blocks* that are combined to do the geochemical speciation. They are presented bellow:

- *Datafilekey*: Title; Miscellaneous parameters (temperature limit, activity coefficient parameters, pressure); Chemical elements block; Aqueous species block; Pure minerals block; Pure non-aqueous liquids block; gas species blocks; solid solutions blocks; references blocks;

- *Inputfile*: Title; Special basis switches; temperature; pressure option; density; total dissolved salts (TDS) option; electrical balancing option; redox option; basis species constraints; ion exchanger creation flag; ion exchanger compositions; solid solutions compositions; alter/suppression options; iopt options; iopg options; iopr options; iodb options; numerical parameters; ordinary basis switches; saturation flag tolerance; aqueous phase scale factor;

EQ3/6 package produces different output depending on the software that is used. We will exemplify the output files in general by using the *EQ3NR* and *EQ6* output formats.

- *EQ3NR*: Two output files are generated. A *pickup* file and the normal output file. The *pickup* file can be used as input to *EQ6* software. The normal output file consists of six *blocks*: header section; input file echo; recap of input data; iterative calculations; principal results; and finally the end of EQ3NR run;

- *EQ6*: This program generates three output files. An *tab* output file, a *pickup* output file and the normal output file. The *tab* file contains information that can be used to plot output results. The *pickup* file is the input to *EQ6*. The normal output file consists of six *blocks*: header; input echo; input recap; iterative calculations; principal results; and finally the end of *EQ6* run;

A small excerpt of an *EQ6* outpuf file is shown in code 3.1.

Code 3.1: Excerpt of *EQ6* output file

```
...
 Entity Date Base Dimension Current Problem
 Chemical Elements 81 81 6
 Basis Species 201 259 7
 Phases 1135 1159 29
 Species 3031 3523 0
 Aqueous Species 1769 1769 22
 Pure Minerals 1120 1120 26
 Pure Liquids 1 3 1
 Gas Species 93 93 2
 Solid Soutions 12 12 0
...
```

3.1.1.2 User Interactions

In *EQ3/6* the command prompt is responsible for all the interactions with the user. There are several codes inside *EQ3/6*; the appropriate command will trigger each one of them. From the existing software, there are *EQ3NR*, *EQ6* and *EQPT* just to name a few.The user must enter the command from the keyboard and must use this "command prompt". By pressing "CTRL+C" at any time the software execution stops, literally "breaking" the process. For example, EQ3 is run by commands of the form as in code 3.2.

Code 3.2: Running EQ3 in *EQ3/6* package

```
>runeq3 datafilekey inputfile(s)
```

Where *datafilekey* and *inputfile* are arguments used by the program; The former is a three-character identification associated to which database should be used while the latter is specifically the name of the input file (can be more than one). Depending on which program from the package the user is using, it generates from two to several output files (always in the *ASCII* format). As mentioned, the input file will be entered in the program as an argument. Any regular text editor is sufficient to create or modify an input file (although we recommend that the user does not create an input file from scratch). There are several pre-existing input templates available and if none of them matches the need of the user, *EQ3/6* recommends that the user generate a new one by copying existing blocks from these templated provided.

The menu format inside the input files tries to recreate known user interactions method as shown in code 3.3.

Code 3.3: Menu Option inside *EQ3/6* input files that mimics a "radio button"

```
iopr(4) - Print a Table of Aqueous Species Concentrations, Activities, etc.:
 [ ] (-3) Omit species with molalities < 1.e-8
 [ ] (-2) Omit species with molalities < 1.e-12
 [ ] (-1) Omit species with molalities < 1.e-20
 [x] ( 0) Omit species with molalities < 1.e-100
 [ ] ( 1) Include all species
```

3.1.1.3 File Formats

All the files discussed above are in the *ASCII* text files format and, therefore, any regular text editor can be used. When the text file is a database, it is converted to binary format and this binary will be used directly by the software. This conversion from text files (*ASCII*) to a binary file is interesting. An *ASCII* file is a binary file that consists of *ASCII* characters. *ASCII* characters are 7-bit encodings stored in a byte. Thus, each byte of an *ASCII* file has its most significant bit set to 0 - this represents a waste of a bit for every byte. By scaling this to a large amount of *ASCII* information and it is possible to understand the reason for this conversion. Another reasonable advantage for the binary files is that the search in binary files is typically faster than in *ASCII* files.

3.1.1.4 Software Environment

The *EQ3/6* package runs on Windows (95 and upper versions) and was developed in FORTRAN77. The support for UNIX computers has been discontinued. It has been developed and run at Lawrence Livermore National Laboratory on an Alliant FX/80 and Sun SPARCstations.

3.1.1.5 Installation Procedures

The installation of *EQ3/6* is explained in details in (WOLERY, 1992). Due to the purpose of this work, details of the installation will be suppressed here. At the same time, it is interesting to mention that the whole installation process requires some level of experience with command prompt and *DOS*.

3.1.2 *PHREEQC*

PHREEQC stands for *PH RE*dox *EQ*uilibrium (in *C* language) and is a widely used public-domain geochemical modelling software available from the USGS. It is available to download for free with versions for PC and Mac. Designed to perform a wide variety of low-temperature aqueous geochemical calculations based on an ion-association aqueous model and has capabilities to:

- Speciation and Saturation Index calculations;
- Batch reaction and one-dimensional (1D) transport calculations involving reversible reactions (including aqueous, minerals, gas, solid-solution, surface-complexation, and ion-exchange equilibrium) and irreversible reactions (including specified mole transfer of reactants, kinetically controlled reactions, mixing of solutions and temperature changes);
- Inverse modelling, which finds sets of mineral and gas mole transfers that takes into account differences in composition between waters.

The software was written in C and has free distributions for Windows, Linux and MAC (PARKHURST, 1995).

3.1.2.1 Input/Output Options

The input data for *PHREEQC* is arranged by *keyword data blocks*. Each block organized with a keyword on the first line followed by lines containing data related to the keyword. Keywords and its context are read from the database at the beginning of the run to define all the necessary parameters. After this database reading procedure, it will continue reading the input file until the it reaches the *END* keyword. This process goes on until the end of the file. As the input file is being read, the program will start putting the pieces together to perform the necessary calculations. An example of a *keyword data block* is shown in code 3.4. Among the possible keywords available in *PHREEQC* are EQUILIBRIUM PHASES, EXCHANGE, GAS PHASE, INVERSE MODELING, PHASES, REACTION, PRINT, SAVE, SOLUTION SPECIES, etc. Each one of these keywords contains specific arguments and parameters to be used. Certain keywords require some specific others keywords - somehow like a puzzle - and if something is missing from the input file, the results are going to be inconclusive and wrong.

Code 3.4: *PHREEQC* keyword data block example

```
EQUILIBRIUM_PHASES
Chalcedony  0.0     0.0
CO2(g)      -3.5    1.0
Gibbsite(c) 0.0     KAlSiO8  1.0
Calcite     1.0     Gypsum   1.0
pH_Fix      -5.0    HCl      10.0
```

The output file will contain the results of the simulation defined in the input file also divided into *keyword blocks*. Among those, we can mention solution composition, description of the solution, redox couples, distribution of species (as can be seen in code 3.5), saturation indices.

Code 3.5: *PHREEQC*'s excerpt from the output file

```
...
----------------------------Distribution of species----------------------------

                                               Log       Log       Log    mole V
   Species          Molality    Activity  Molality  Activity     Gamma   cm3/mol

   OH-            2.705e-006  1.647e-006    -5.568    -5.783    -0.215     -2.63
   H+             7.983e-009  6.026e-009    -8.098    -8.220    -0.122      0.00
   H2O            5.551e+001  9.806e-001     1.744    -0.009     0.000     18.07
C(4)         2.257e-003
   HCO3-          1.238e-003  8.359e-004    -2.907    -3.078    -0.170     27.87
   NaHCO3         6.168e-004  7.205e-004    -3.210    -3.142     0.067     19.41
   MgHCO3+        2.136e-004  1.343e-004    -3.670    -3.872    -0.201      5.82
   MgCO3          7.301e-005  8.527e-005    -4.137    -4.069     0.067    -17.09
   CaHCO3+        3.717e-005  2.572e-005    -4.430    -4.590    -0.160      9.96
   CO3-2          3.128e-005  6.506e-006    -4.505    -5.187    -0.682     -0.34
   CaCO3          2.256e-005  2.636e-005    -4.647    -4.579     0.067    -14.60
   NaCO3-         1.477e-005  9.972e-006    -4.831    -5.001    -0.170      1.77
...
```

3.1.2.2 User Interactions

PHREEQC's distribution differs drastically according to the environment (*Windows* or *UNIX*). By this reason, the analysis of user interactions will be done separately.

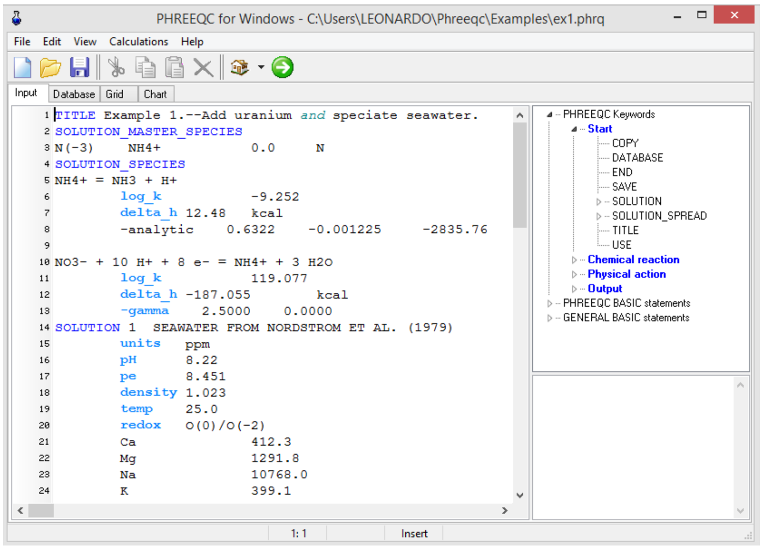
- *PHREEQC* for Windows: Due to the need of a geochemical modelling software and the lack of interface for *PHREEQC* on the first versions of the software, many efforts were done to create an interface to *PHREEQC*. The program *PhreeqcI* is a graphical user interface to *PHREEQC* that provides data entry screens for the keyword data blocks with a description of each input data item. Organizing the input file by using some *project tree* which facilitates viewing, selecting, editing and running the *PHREEQC* simulations. Critical to mention that *PhreeqclI* does not implement all the keyword blocks. After this, another effort was done by the launch of *PHREEQC* version 2 with a graphical interface - which was kept later in version 3. This graphical interface was baptized *PfW* (Phreeqc for Windows) but it has no updates since 2011, it can be seen in figure 3.1. The last effort in this sense was the development of an adaptation of the popular general-purpose text editor *Notepad++*. This modification comes with the following capabilities: syntax highlighting; autocompletion of keywords and identifiers; tips; colored numbers; parenthesis matching; commenting and uncommenting multiple lines at once; column editor; few shortcuts; and file recognition; This can be seen in picture 3.2. When running PHREEQC from the *Notepad++* adaptation, the command prompt is called and the simulation is executed, as in picture 3.3. There are some options and shortcuts available from the *Notepad++*'s interface, which is shown in detail in figure 3.4.

- *PHREEQC* for UNIX: Under UNIX's distribution, *PHREEQC* runs from the command prompt. It can be used by using the command as in code 3.6

Code 3.6: Command to run UNIX's *PHREEQC*

```
phreeqc input output database screen_output
```

Where *"input"* file will contain the description of the simulation, the *"output"* file will store the results of the simulation, *"database"* will specify which database should be used and *"screen_output"* stores the information that will be shown on screen. If the user doesn't specify the names of *"output"*, *"database"* or *"screen_output"* the geochemical modelling software will choose default values.



```
TITLE Example 1.--Add uranium and speciate seawater.
SOLUTION_MASTER_SPECIES
N(-3)     NH4+            0.0      N
SOLUTION_SPECIES
NH4+ = NH3 + H+
        log_k               -9.252
        delta_h 12.48   kcal
        -analytic     0.6322     -0.001225     -2835.76

NO3- + 10 H+ + 8 e- = NH4+ + 3 H2O
        log_k               119.077
        delta_h -187.055         kcal
        -gamma     2.5000     0.0000
SOLUTION 1  SEAWATER FROM NORDSTROM ET AL. (1979)
        units      ppm
        pH         8.22
        pe         8.451
        density 1.023
        temp       25.0
        redox      O(0)/O(-2)
        Ca                  412.3
        Mg                  1291.8
        Na                  10768.0
        K                   399.1
```

Figure 3.1: User interface of the PHREEQC for Windows (*PfW*)

3.1.2.3 *File formats*

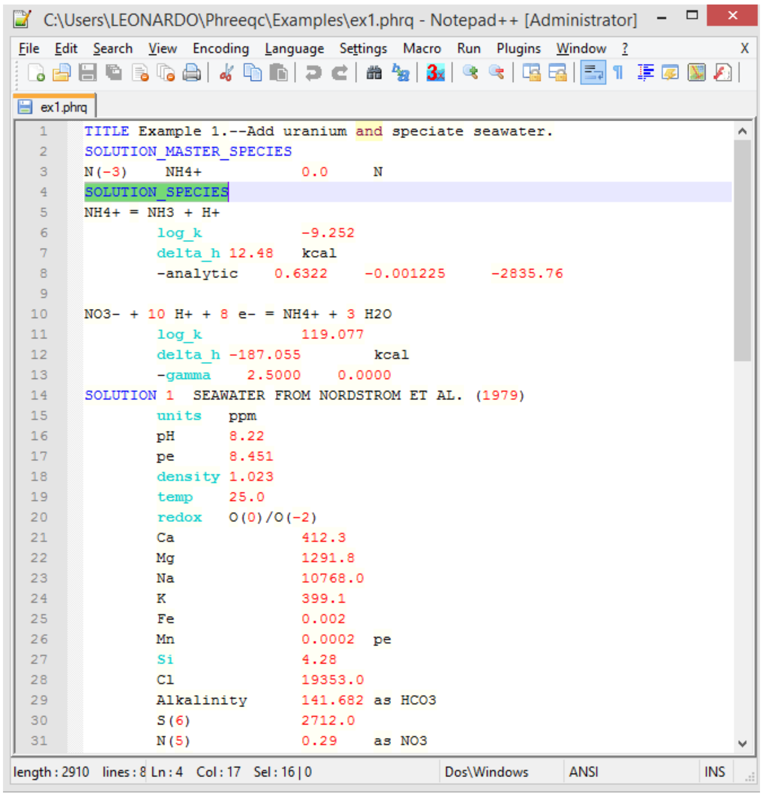
All the files discussed above are in the *ASCII* text files format and, therefore, any regular text editor can be used. It is recommended that the edition of the *PHREEQC* files is done by using its NotPhreeqcce or the notepad++ adapted version (APELLO, 2011).

3.1.2.4 *Software Environment*

PHREEQC has support for Windows (32 and 64-bit), MacOS (OS 10.6+) and Linux. *PHREEQC* is currently on version 3 and with frequent updates, bug fixes and maintenance.

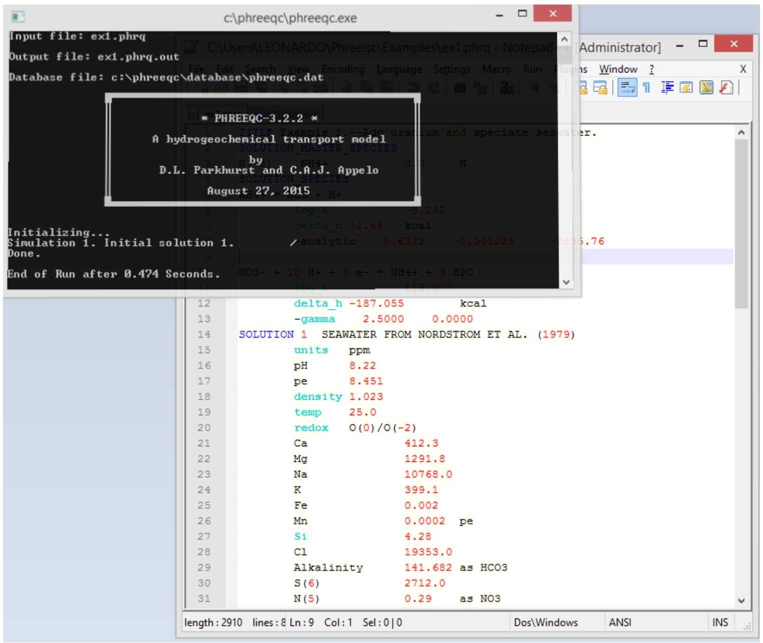
3.1.2.5 *Installation Procedures*

The *PHREEQC* version for Windows has a self-extracting file that can is available for download from the USGS website and easily installed. The *UNIX* distribution comes with additional scripts and a makefile and the user should follow some steps to compile and install the program.


C:\Users\LEONARDO\Phreeqc\Examples\ex1.phrq - Notepad++ [Administrator]
File Edit Search View Encoding Language Settings Macro Run Plugins Window ?
ex1.phrq
length : 2910 lines : 8 Ln : 4 Col : 17 Sel : 16 | 0 Dos\Windows ANSI INS

```
TITLE Example 1.--Add uranium and speciate seawater.
SOLUTION_MASTER_SPECIES
N(-3)     NH4+            0.0      N
SOLUTION_SPECIES
NH4+ = NH3 + H+
        log_k           -9.252
        delta_h 12.48   kcal
        -analytic     0.6322    -0.001225     -2835.76

NO3- + 10 H+ + 8 e- = NH4+ + 3 H2O
        log_k           119.077
        delta_h -187.055        kcal
        -gamma     2.5000    0.0000
SOLUTION 1  SEAWATER FROM NORDSTROM ET AL. (1979)
        units   ppm
        pH      8.22
        pe      8.451
        density 1.023
        temp    25.0
        redox   O(0)/O(-2)
        Ca              412.3
        Mg              1291.8
        Na              10768.0
        K               399.1
        Fe              0.002
        Mn              0.0002  pe
        Si              4.28
        Cl              19353.0
        Alkalinity      141.682 as HCO3
        S(6)            2712.0
        N(5)            0.29    as NO3
```

Figure 3.2: Example of the PHREEQC Notepad++ plugin


c:\phreeqc\phreeqc.exe
length : 2910 lines : 8 Ln : 9 Col : 1 Sel : 0 | 0 Dos\Windows ANSI INS

```
Input file: ex1.phrq
Output file: ex1.phrq.out
Database file: c:\phreeqc\database\phreeqc.dat

                * PHREEQC-3.2.2 *
          A hydrogeochemical transport model
                      by
           D.L. Parkhurst and C.A.J. Appelo
                August 27, 2015

Initializing...
Simulation 1. Initial solution 1.
Done.

End of Run after 0.474 Seconds.
```

Figure 3.3: PHREEQC software called from the Notepad++ Plugin

3.1.3 *MINTEQ*

Much simpler options on reaction treatment, developed by the United States' Environment Protection Agency (EPA). *MINTEQ* is a geochemical program to model aqueous solutions and the interactions of aqueous solutions with hypothesized assemblages of solid phases. It has a particular inclination to equilibrium speciation to calculate equilibrium

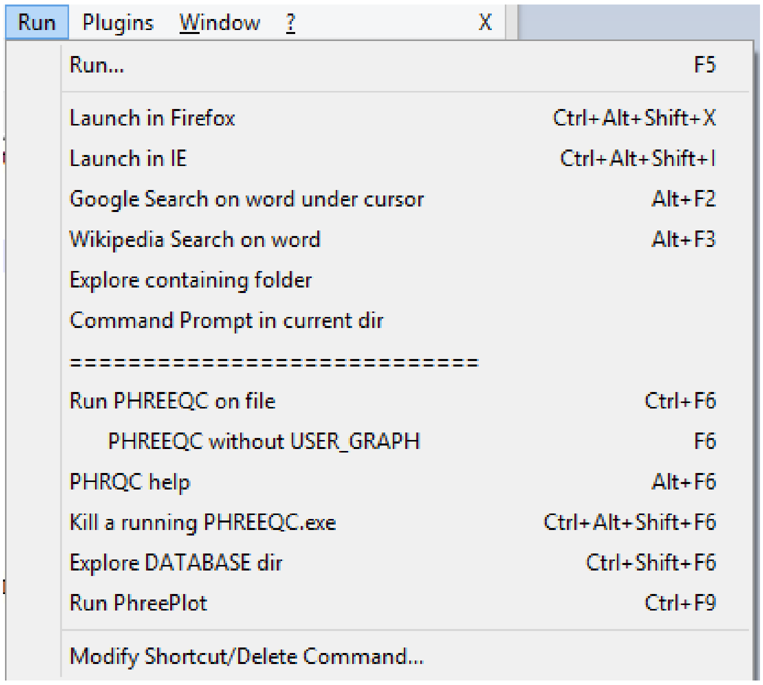

Run
Plugins
Window
?
X
Run... F5
Launch in Firefox Ctrl+Alt+Shift+X
Launch in IE Ctrl+Alt+Shift+I
Google Search on word under cursor Alt+F2
Wikipedia Search on word Alt+F3
Explore containing folder
Command Prompt in current dir
============================
Run PHREEQC on file Ctrl+F6
PHREEQC without USER_GRAPH F6
PHRQC help Alt+F6
Kill a running PHREEQC.exe Ctrl+Alt+Shift+F6
Explore DATABASE dir Ctrl+Shift+F6
Run PhreePlot Ctrl+F9
Modify Shortcut/Delete Command...

Figure 3.4: PHREEQC software called from the Notepad++ Plugin

composition of dilute aqueous solutions. The model is useful for calculating the equilibrium mass distribution among dissolved species, adsorbed species and multiple solid phases although it has a much simpler treatment of the reactions. It was originally developed in FORTRAN77 by Battelle Pacific Northwest Laboratory (*PNL*) and continued by the Environmental Protection Agency (*EPA*) to perform the calculations necessary regarding waste, sediments and ground water. *MINTEQ* does not consider the kinetic reactions and works at fixed temperature (25 degrees Celcius). An extensive database adequate to a broad range of problems is part of the software, and there is no need for the user to change nor add anything (BROWN, 1987) (ALLISON, 1991). The latest update on *MINTEQ* has been in 1990, since then, some upgrades and improvements especially on the usability and calculations. This new version was baptised *MINTEQA2* and uses a well-developed thermodynamic database from the *USGS*. This review will always refer to this continued version of *MINTEQ*.

3.1.3.1 Input/Output Options

The input files for *MINTEQA2* can be generated manually, but there is another software called *PRODEFA2* that guides the user to accomplish this task. *PRODEFA2* is an interactive program used to create input files that will be addressed in details in section 3.1.3.2. Four parts compose the input file:

- Input file: The input file that contains the data input by the user. Typically, this file contains dissolved (i.e. Ca concentrations, pH, temperature) and solid phase (i.e. minerals, sorption sites) information for a water sample.
- Database file: This file contains the thermodynamic constants that govern the processes of interest (i.e. complexation constants, mineral solubilities, activity constants) which will be used to conduct calculations.
- Algorithm or executable file: These files contain the workings of the code, which

solve the specified problem (usually using an iterative numerical approach) within the constraints imposed by the Database files and the information in the Input file.

- Output file: This file contains the results of the calculations performed by the Algorithm Files.

Amont the input file's options, there are 4 levels of complexity that when putted together generate a *MINTEQA2* simulation.

1. Displays the current settings of system parameters such as temperature as well as program flag settings such as the number of iterations allowed;

2. Specify the chemistry of the system;

3. This level works as a "line editor" in displaying by category or TYPE those species that have been explicitly entered through level 2;

4. Deals with utility functions (output file details, for example);

If database, algorithm and output files are not specified, some default options will be used. Code 3.7 brings a *MINTEQA2*'s input file for a simulation that computes pH and comes inside the package as an example.

Code 3.7: *MINTEQA2*'s input file

```
TEST1A - Compute total H+ of a solution of known pH.
This together with TEST1B illustrate a two run set to compute pH.
25.00 MOLAL  0.000  0.00000E+00
0 0 1 0 0 0 0 0 1 1 0 0 0
0   0   0
    330  0.000E+00   -2.50                    /H+1
    732  1.580E-03   -2.80                    /SO4-2
    410  7.700E-05   -4.11                    /K+1
    140  0.000E+00  -16.00                    /CO3-2

  3   2
3301403    21.6600    -0.5300                 /CO2 (g)
    330     2.5000     0.0000                 /H+1
```

The *MINTEQA2*'s output file is divided into six parts:

1. Reproduction and interpretation of the input file;

2. Detailed listing of species read from the database files;

3. Iteration information and detailed information for each specie;

4. Percentage distribution of components among dissolved and adsorbed species;

5. Provisional or equilibrated mass distribution, provisional or equilibrium ionic strength, equilibrium pH and pE, electrostatic surface potencial and charge for electrostatic adsorption models;

6. Saturation indices for all database solids with respect to the solution;

Code 3.8 brings an excerpt of *MINTEQA2*'s part 3 output.

Code 3.8: *MINTEQA2*'s excerpt from the output file

```
...
________________________________________________________________________________
_____________________________ PART 3 of OUTPUT FILE ____________________________
  MINTEQA2  v4.02   DATE OF CALCULATIONS:  5-JUN-2000  TIME: 14: 6:27



PARAMETERS OF THE COMPONENT MOST OUT OF BALANCE:

     ITER      NAME       TOTAL mol/L   DIFF FXN   LOG ACTVTY    RESIDUAL
      0   SO4-2          1.580E-03   6.594E-07    -2.91757    5.014E-07
      1   SO4-2          1.580E-03   4.193E-04    -2.91775    4.192E-04
      2   SO4-2          1.580E-03   8.125E-06    -3.01997    7.967E-06
      3   SO4-2          1.580E-03   1.693E-07    -3.02220    1.135E-08

 ID No       Name    Total Conc(M)    Conc (M)  log Activity   Diff fxn
  410  K+1           7.700E-05     7.649E-05    -4.14759    5.093E-11
  732  SO4-2         1.580E-03     1.266E-03    -3.02224    3.557E-09
    2  H2O           0.000E+00    -1.049E-05    -0.00004    0.000E+00
  330  H+1           0.000E+00     3.398E-03    -2.50000    0.000E+00
  140  CO3-2         0.000E+00     2.916E-17   -16.66004    0.000E+00

-----------------------------------------------------------------------------
...
```

3.1.3.2 User Interactions

MINTEQA2 and *PRODEFA2* interactions are completely independent programs and *PRODEFA2* is used before *MINTEQA2* in order to generate the input file that will be consumed by the latter. Everything is done through the command prompt and on this work will detail *PRODEFA2*'s interaction. It provides a "walk-through" to generate an input file for *MINTEQA2*.

After opening the software and providing a valid name, it will ask which part of the input file the user wants to create or edit as shown in figure 3.5. We will follow the suggested order and go through 4 levels, as previously discussed in this section. Figure 3.6 shows the main menu, it is the organization of all the levels and works as a central hub of information. Figure 3.7 displays the necessary information about level 1, in order to change any of the entries on this screen, the user must enter the number to the left of the entry and respond to the questions presented. All the four levels do this kind of interactions. Through these interactions, the user has access to all the information in the database and can choose specifically about what is the model.

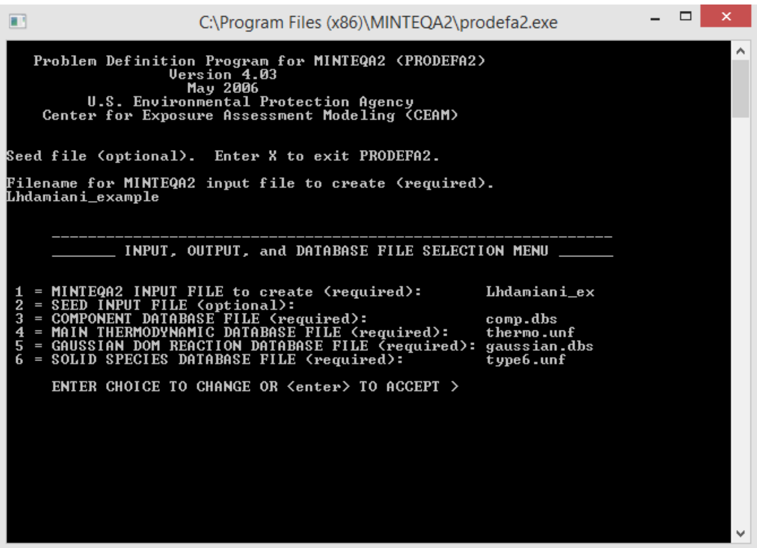
During this work, we chose to add a specific aqueous specie to show the *PRODEFA2* interactions and present it step-by-step bellow:

1. Choose level 2 on main menu;

2. Choose option 1 (Specify AQUEOUS COMPONENTS: TOTAL CONCENTRATIONS or FIXED ACTIVITIES) inside the menu from level 2;

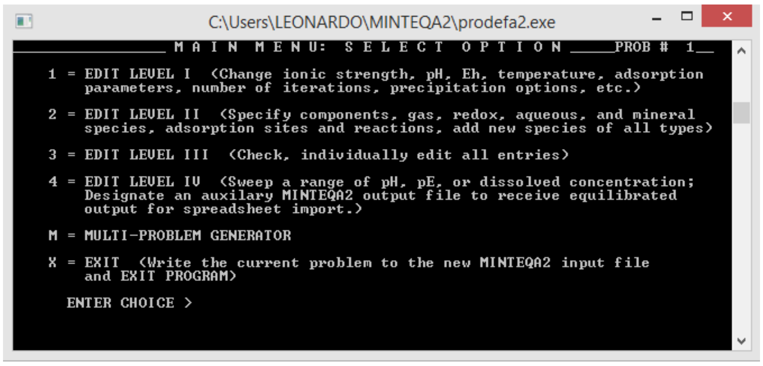
3. Choose option 1 (TOTAL DISSOLVED CONCENTRATION), it identifies how we want to entre this new component.

4. At this step, we are requested to enter the first letter for the component. Alternatives to this approach is to enter "-1" if you know the components id number or quit. We will add Na^+ to the system. So, we type the letter "N" and hit enter button.



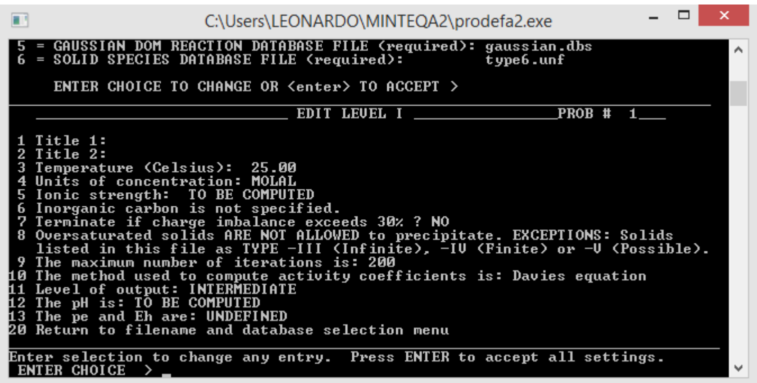
```
   Problem Definition Program for MINTEQA2 (PRODEFA2)
                  Version 4.03
                    May 2006
         U.S. Environmental Protection Agency
    Center for Exposure Assessment Modeling (CEAM)

Seed file (optional).  Enter X to exit PRODEFA2.

Filename for MINTEQA2 input file to create (required).
Lhdamiani_example

     ______________________________________________________________
     ________ INPUT, OUTPUT, and DATABASE FILE SELECTION MENU ______

 1 = MINTEQA2 INPUT FILE to create (required):       Lhdamiani_ex
 2 = SEED INPUT FILE (optional):
 3 = COMPONENT DATABASE FILE (required):             comp.dbs
 4 = MAIN THERMODYNAMIC DATABASE FILE (required):    thermo.unf
 5 = GAUSSIAN DOM REACTION DATABASE FILE (required): gaussian.dbs
 6 = SOLID SPECIES DATABASE FILE (required):         type6.unf

     ENTER CHOICE TO CHANGE OR <enter> TO ACCEPT >
```

Figure 3.5: *MINTEQA2* initial menu options



```
_________________ M A I N  M E N U:  S E L E C T  O P T I O N ____PROB #  1__

   1 = EDIT LEVEL I   (Change ionic strength, pH, Eh, temperature, adsorption
       parameters, number of iterations, precipitation options, etc.)

   2 = EDIT LEVEL II  (Specify components, gas, redox, aqueous, and mineral
       species, adsorption sites and reactions, add new species of all types)

   3 = EDIT LEVEL III  (Check, individually edit all entries)

   4 = EDIT LEVEL IV  (Sweep a range of pH, pE, or dissolved concentration;
       Designate an auxilary MINTEQA2 output file to receive equilibrated
       output for spreadsheet import.)

   M = MULTI-PROBLEM GENERATOR

   X = EXIT  (Write the current problem to the new MINTEQA2 input file
       and EXIT PROGRAM)

      ENTER CHOICE >
```

Figure 3.6: *MINTEQA2* main menu



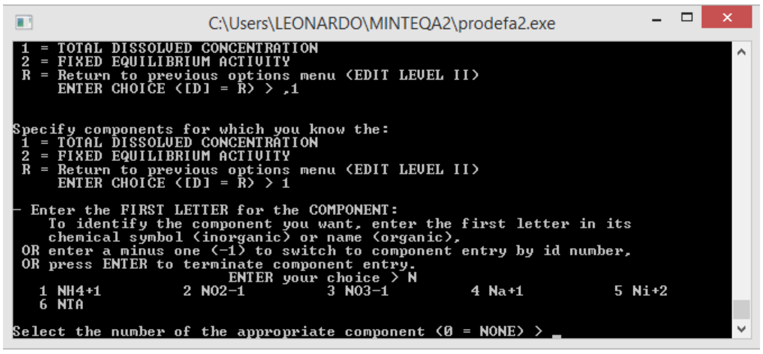
```
 5 = GAUSSIAN DOM REACTION DATABASE FILE (required): gaussian.dbs
 6 = SOLID SPECIES DATABASE FILE (required):         type6.unf

     ENTER CHOICE TO CHANGE OR <enter> TO ACCEPT >
_____________________________________________________________________________
   ____________________________ EDIT LEVEL I ________________PROB #  1___

 1 Title 1:
 2 Title 2:
 3 Temperature (Celsius):  25.00
 4 Units of concentration: MOLAL
 5 Ionic strength:  TO BE COMPUTED
 6 Inorganic carbon is not specified.
 7 Terminate if charge imbalance exceeds 30% ? NO
 8 Oversaturated solids ARE NOT ALLOWED to precipitate. EXCEPTIONS: Solids
   listed in this file as TYPE -III (Infinite), -IV (Finite) or -V (Possible).
 9 The maximum number of iterations is: 200
10 The method used to compute activity coefficients is: Davies equation
11 Level of output: INTERMEDIATE
12 The pH is: TO BE COMPUTED
13 The pe and Eh are: UNDEFINED
20 Return to filename and database selection menu
_____________________________________________________________________________
Enter selection to change any entry.  Press ENTER to accept all settings.
 ENTER CHOICE  > _
```

Figure 3.7: *MINTEQA2* Level 1 informations

5. At this step, all the existing options available are shown listed with an identifier and we are requested to identify which one of these we want to add. This can be seen in 3.8.

6. After pressing Na^{+}'s id (at this case was the number 4). We can finally enter the TOTAL DISSOLVED CONCENTRATION (MOLAL) of COMPONENT. Defined earlier on step 3 After adding the molal concentration for this specie the system

goes back to step 4 in case we want to continue adding other components.



```
 1 = TOTAL DISSOLVED CONCENTRATION
 2 = FIXED EQUILIBRIUM ACTIVITY
 R = Return to previous options menu <EDIT LEVEL II>
     ENTER CHOICE <[D] = R> > ,1


Specify components for which you know the:
 1 = TOTAL DISSOLVED CONCENTRATION
 2 = FIXED EQUILIBRIUM ACTIVITY
 R = Return to previous options menu <EDIT LEVEL II>
     ENTER CHOICE <[D] = R> > 1

- Enter the FIRST LETTER for the COMPONENT:
    To identify the component you want, enter the first letter in its
    chemical symbol <inorganic> or name <organic>,
 OR enter a minus one <-1> to switch to component entry by id number,
 OR press ENTER to terminate component entry.
                       ENTER your choice > N
   1 NH4+1          2 NO2-1          3 NO3-1          4 Na+1           5 Ni+2
   6 NTA

Select the number of the appropriate component <0 = NONE> > _
```

Figure 3.8: *PRODEFA2*'s example of adding a specific aquoues specie (Na^+)

There is also a software called *Visual Minteq* that tries to "humanize" *MINTEQ* and was maintained by the KTH Royal Institute of Technology, located in Stockholm, Sweden. *Visual Minteq* latest release date is December 2013 at the version 3.0 and is available only for *Windows* operating systems since it was developed in Visual Basic.

3.1.3.3 File formats

All the files follow a regular text format (*ASCII*) and their purpose is defined on the extension of the file.

- Input files have the extension *"INP"*;
- Test and help files have the extension *"HLP"*;
- Output files have the extension *"LST"*;
- Database files have the extension *"DBS"* or *"UNF"*;
- Input file have the extension *"INP"*;

3.1.3.4 Software Environment

It is currently at the version 4.03 (Windows only), interesting to mention that this release date back to May 2006. The latest *UNIX* distribution is version 3.12 (which was also called BETA for UNIX) and date back to August 1996.

3.1.3.5 Installation Procedures

MINTEQA2 is easily installed by a self-extractor installer that can be downloaded from *EPA*'s website (MINTEQ, 2006) and (MINTEQ, 1996). Included in the distribution package are also some important documentation, *PRODEFA2* software and several input/output template files. The UNIX version distribution comes with the source files and must be compiled to run.

3.1.4 *SOLMINEQ.88*

SOLMINEQ.88 is an FORTRAN 77 written geochemical modeling program based on SOLMNEQ (KHARAKA, 1973) with improved algorithms that result in a faster program execution and tighter convergence. The software has a database with the focus on organics aqueous species. It calculates the distribution of mass among aqueous species and complexes and calculates saturation indexes of minerals at different temperatures and pressures. It includes options as boiling, mixing of solutions and partitioning of gases between water, oil and a vapour phase. *SOLMINEQ.88* also contemplates mass transfer with the effects of dissolution and precipitation of minerals and options to calculate activity coefficient model (KHARAKA, 1988). The original version has no UI, but there were further studies with the intention of creating a user-friendly program that can be used to generate, edit and analyze input and output files - *SOLINPUT*. *SOLMINEQ.88* model has not been developed any further nor improved since its first release.

3.1.4.1 Input/Output Options

The input of *SOLMINEQ.88* consists of two sets of data: fixed and variable. The first contains the chemical composition of an aqueous fluid and options for processing these data; and the last Consists of input data required for using them.

The input file is composed by six parts, *SOLINPUT* guides the user throught all of them:

1. Basic Parameters: enter the chemical and physical data for that sample;
2. Flags: controls how the software interprets, process and display the data;
3. pH: controls the details of how the pH calculation is done;
4. Mass transfer: defines which mass transfer capabilities are used;
5. User Log K: make temporary changes and extensions to the database;
6. Additional ions and minerals: temporarily adds user defined ions and minerals to a particular simulation;

The output file contains the results of the computations by *SOLMINEQ.88* and it consists of six parts: An input data echo that shows the values and options selected for each sample; A table listing the calculated tolerance factor for successive iterations on the anions; A list of input to *SOLMINEQ.88* including sample description, pH, Eh, temperature and os on; A table showing the distribution of species in solution; Ratios of a number of cations and anions of importance in geochemical processes; and th e last one contains a table indicating the states of reactions for minerals considered;

Code 3.9: *SOLMINEQ.88*'s excerpt of the output file

```
 : Test Sample #1 for SOLMINEQ.88 - Modified Seawater at 25 C
TEMP HI TEMP DENS PRESS
0.2500E+02 O.OOOOE+00 0.1023E+01 O.OOOOE+00
PH EHM EHMC EMFZSC
0.8200E+01 0.5000E+00 0.9000E+01 0.9000E+01
CONCENTRATION UNITS : PPM
Na K Li Ca
0.1077E+05 0.3991E+03 0.1810E+00 0.4123E+03
Si02 Cl S04 H2S
0.4280E+01 0.1935E+05 0.2712E+04 O.OOOOE+00
F P04 N03 NH3
0.1390E+01 0.6000E-01 0.2900E+00 0.3000E-01
Pb Zn Cu Mn
0.5000E-04 0.4900E-02 0.7000E-03 0.2000E-03
As U V
0.4000E-02 O.OOOOE+00 O.OOOOE+00
Acetate Oxalate Succinate CH4
0.1000E+00 0.1000E+00 0.1000E+00 0.1000E+00
```

3.1.4.2 *User Interactions*

The software that accompany *SOLMINEQ.88* and handles the generation of input files and all the interactions is named *SOLINPU*. All the interactions use the command prompt - the menus with several options are generated and displayed. The user selects the option by entering the indication number and pressing enter (the indication number stays on the left of the option). Figure 3.10 shows this example of interaction.

Code 3.10: *SOLMINEQ.88*'s example of user interaction

```
        pH OPTIONS
1) Gas Addition Option
2) Gas-Water-Oil Distribution Option
3) Carbonate Mineral Saturation Option
4) C02 Option
5) Tolerance factor for Mineral and C02 Options
6) Return to Options Menu
Enter Choice (1-6)   __
```

3.1.4.3 *File formats*

The input files are regular *ASCII* text files and any regular text editor is good for creating or editing.The database files from *SOLMINEQ.88* have the extension "TBL" and the output files

3.1.4.4 *Software Environment*

As mentioned, *SOLMINEQ.88* had only one release and has been discontinued since then. It is available only for the *Windows* operating system.

3.1.4.5 *Installation Procedures*

SOLMINEQ.88 distribution requires knowledge in compiling and linking FORTRAN77 programs. It also comes with the software *SOLINPUT* that makes the user interactions, as explained before, easier.

Interesting to point that along the installation manual, they recommend that curious compiling options are activated, for example *"do not check for array bounds"*, and *"do not check subprogram interfaces"*.

3.2 Commercial Software Review Discussion

On this section we compare, analyze and discuss some important aspects and issues of each one of the software presented earlier in this work. Regarding geochemical features, table 3.1 brings certain aspects and features of the software.

Regarding the computer science point of view, we analyse and evaluate the software always thinking how good is the software engineering behind it. Taking into account all the aspects described and extensively discussed in chapter 2 is essential.

The points that we discuss and take into consideration are the following:

- The costs: Costs are probably the most thing that people look first when choosing software; however, it should not be the deciding factor. Different solutions use different pricing models and according to the purpose of the solution it will be decided.

- Setup and versioning: The installation of software is the act of making it ready for execution. According to the software, a particular installation process is done -

Table 3.1: Geochemical comparison between speciation softwares

	Aqueous Complexation	Precipitation and Dissolution Mass Balancing	Reaction path	Kinetics	Multi-Activity Coefficient methods
EQ3/6	✓	✓	✓	✓	✓
PHREEQC	✓	✓	✓	✓	
MINTEQA2	✓	✓			
SOLMINEQ.88	✓	✓	✓	✓	✓

which involves copying/generating files from the installation files to the local computer to be accessed by the operating system (*OS*). The *OS* also influences how this process is done; each software has a different distribution package to each *OS* - or not. Not hard to find software with distribution only to specifics *OS*, meaning that if the user uses other *OS* it is impossible to use that software. Also not hard to find same software with disparate versions according to the *OS* - resulting in divergent features available on the same software defined by the *OS* that is being used.

- Customization and Integration: The software is a standard solution, and its supplier is not interested in making changes? This is the typical scenario where the user has great chances of finding problems ahead. Therefore, an interesting exercise is to think of how this software communicate with others? Options like “import” and “export” are vital when working with a large amount of data. With the advances in software, you might want to use a different software to analyse the output and to be able to reach deeper insights about the information available. These insights might be unique and lead to incredible breakthroughs.

- Security and Control: Security is one of the main issues one face when considering solutions. Nobody wants to share private data and details with others. Ensuring that the software can guarantee no data loss or data leakage is important. The solutions that provide direct control to the database, details and process is most likely to have a private required by some users.

- Infrastructure: When choosing a software, the infrastrucutre that it requires need to be carefully analyzed to verify if it matches the disponibility. Does it requires internet access? How many space on disk and memory it uses? Extra costs may occur if this is not thought earlier.

- Core functionality: This is one of the most important points to be analyzed. How good the software focus on the needs of the user and how good is the value that the software brings to its users by performing the core activity successfully.

Table 3.2: Computer Science comparison to Geochemical Speciation Software

	Costs	Setup and versioning	Customization and Integration	Security and Control	Infrastructure	Core functionality	Graphical User Interface	Support and Maintenance	Database	Overall Average
EQ3/6	2	2	1	1	3	5	1	1	1	1.88
PHREEQC	4	4	2	2	3	4	2	3	1	2.77
MINTEQA2	4	2	1	1	2	2	1	1	1	1.66
SOLMINEQ.88	3	1	1	1	1	5	1	1	1	1.66

Graphical User Interface (*GUI*) and visualization: Handles the interactions between the users and the electronic devices. When the software has a complex domain, such as geochemical modeling, the *GUI* is even more important. It is responsible for allowing the user to think all the possible options and to take all the advantages of the software. The perfect *GUI* takes into consideration all the human behaviour, senses and how we interact with our world (from electronic devices to human relationships).

- Support and Maintenance: If the software, for some unknown reason, goes down, is the user able to reach someone to question about the issue or discuss what happened? Will the user be able to find a users' community to debate and share knowledge? If the software has a support team working to fix bugs, improve the performance, add new features and sharpen some of the old features, it means that the user will have a better infrastructure to work.

- Database: All the data manipulated inside a software is stored and organized in a database. There are multiple ways of doing this; many important things must be taken into account to decide which database fits best to the software. Since the 80's the relational database model represented by the *SQL* language has been the most popular. A conceptual database model is strongly recommended to produce a schema that consider all the structure and information needed by this software. Along the database schema, the security of this database must be addressed properly - either for consistency and privacy reasons. A good database design avoids redundant data (unnecessarily duplicated data). Poorly designed database generates inconsistent data (inaccurate data), which will lead to wrong decisions and, therefore, can result in failure of the software.

To achieve an applicable comparison we give grades from 1 to 5 to each aspect (where 1 is the lowest and worst and 5 the highest and best possible grade). This "grading system" is done with the intention to obtain a normalization towards differents aspects an interesting output for the comparison and in table 3.2 we can in details.

3.3 Summary

- EQ3/6: It represents a landmark in Geochemical Modelling. Unfortunately, it is inaccessible due to the elevated cost of licensing. The vast amount of information available online about *EQ3/6* makes it an excellent knowledge encourager and pushes the understanding of geochemical modelling further. It used all the computing tools and options that were available until the release date. Since then, computing has clearly evolved, making it an obsolete and hard to use the software. It has a large database with many pieces of information on it, but it is not exactly clear to the user. It is hard to understand what exactly is the software doing; verifying if that is what the users wants is even more difficult. This might be the reason that geochemical modellers do not understand how it is put together.

- PHREEQC: It is the best option for users not experienced with software - it has a *GUI* and comes with a self-extractor installer. Important to mention here that *PHREEQC*'s *GUI* is far from what a regular user might want - it is not clear in many aspects, and its usability is far from regular. In the geochemical modelling area, not all the users have intimacy with tasks as compiling and linking computer programs. *PHREEQC* allows anyone with an interest to have a chance to perform a geochemical modelling simulation even though many people have different ways of defining the problem. Database in *PHREEQC* seems to be a problem - it uses a flat file database.

- MINTEQA2: From the geochemical point of view, *MINTEQA2* is the simpler from the four software analyzed in this work. The user interaction can be painful for anyone that is not used with command prompt, besides that, the complexity of the input file makes any way a hard way. Creating the input file without the subsidiary software *PRODEFA2* is a task close to impossible and learning how to use this subsidiary software is a very costly task. Taking into account that its last release date back to 2006, it is hard to motivate and try to give *MINTEQA2* any consideration nowadays.

- SOLMINEQ.88: An interesting software that was also a pioneer and layed the ground for many others to improve and progress the knowledge in the geochemical modelling field. *SOLMINEQ* used the computing tools that were accessible when it was developed - nowadays there is no tendency that someone will start using it and learning its complex input and output. There are less costly options easily attainable, and that produce analogous results.

4 *SHPECK* - SPECIATION MODEL

SHPECK is a equation-based simulator, there is a governing theory that can guide the construction of mathematical models based on a set of equations. The “equation-based” term refers to simulations based on the kinds of global equations we associate with physical theories - many of these theories present in the foundation concepts of *SHPECK* originally from the *Phase Rule*. Phase rule allows us to predict the number of stable phases that may exist in equilibrium for a particular system and is describe fully in (GARRELS, 1965) and is the base principle used in chemical speciation.

As a computer simulation softare, *SHPECK* represents the entire process. This process includes choosing a model; finding a way of implementing that model in a form that can be run on a computer; calculating the output of the algorithm; and visualizing and studying the resultant data.

This chapter will guide through all the aspects of the development of *SHPECK* - a geochemical speciation modelling software.

4.1 Specification

SHPECK is a geochemical speciation modelling software responsible for calculating the distribution of dissolved species between free ions and aqueous complexes and also saturation indexes for different minerals.

4.2 Architecture

The architecture of a software is resposible to assure that all the internal and external stakeholders’ concerns are preserved, addressed and satisfied. In order to not develop something that will not need restructuration and code refactoring later, it is necessary to see the big picture of the whole *SHPECK*’s project. This will drive and be present during the whole development, starting from the technical decisions (which programming language, libraries, database model and so on) until the final user (which value the software really deliver).

Figure 4.1 brings the software architecture of *SHPECK*, which is modelled following the popular concept of *Model-View-Controller* (MVC) (GAMMA, 1994). *MVC* is an architectural pattern that divides the software into three interconnected parts:

- Model: Is an object representing data or even activity. For example, the algorithm and math behind calculating the activity coefficient by Debye-Huckels’ formula.
- View: Is some form of visualization of the state of the model. For example, which are the solutes that the user wants to add into the simulation?

- Controller: Offers facilities to change the state of the model. For example, define which algorithm for calculating the activity coefficient should be used accoriding to the user's choice or the value of the ionic strength (if the user did not specify which one to use).

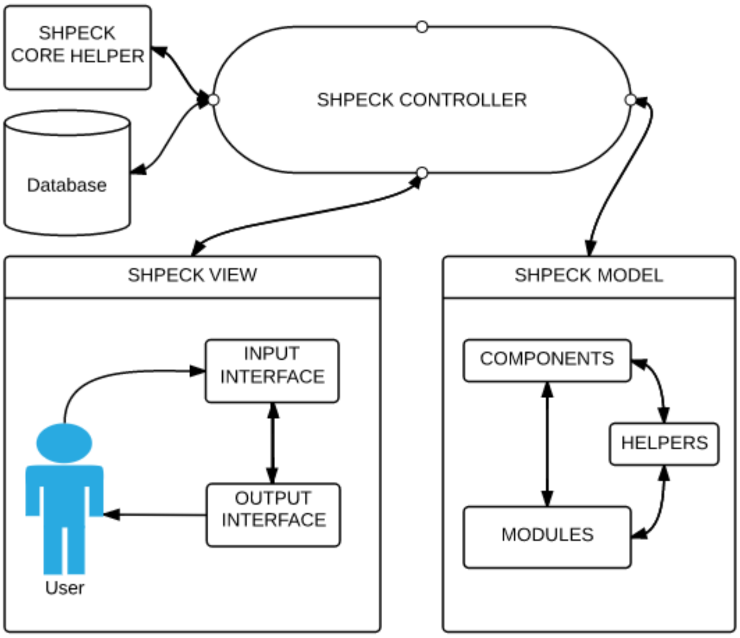


Figure 4.1: Architecture of the *SHPECK* software

4.2.1 Technical Specification

SHPECK is cross-platform software (*Windows* and *UNIX*) developed with an object oriented approach using C++ (STROUSTRUP, 1997) and Qt (Qt, 2014). We use also an external library called *Armadillo C++* that implements some of the linear algebra needed in the algorithm (**?**).

4.3 Governing equations

SHPECK uses the thermodynamic equilibrium reaction as equations for the calculation of a multiphase systems in equilibrium, the details of these reactions are discussed in details in chapter 2. The system will be composed by a set of mass-actions equations (as in equation 2.2) defined by the number of species that coexist in the system. These equations model the geochemical speciation closed system and take into account all the chemical properties.

Aside with the mass action equations we must have another constraints in order to solve the euilibrium state of the system. In *SHPECK*, we use the the concentration of the specie (other possible constraints are activity, number of moles, total volume, etc) to deal with the equilibrium state of the system. This configuration generates a system where there are U unknowns, M mass action equations and E equilibrium constraints.

$$U = E + M \tag{4.1}$$

The unknowns values represents the number of moles of each specie in the system.

4.4 Numerical Method

In order to solve the system composed by the equilibrium state of the mass action equations and the equilibrium constraints, *SHPECK* uses a numerical methodology that computes simultaneously and find the value of the unknowns.

The numerical method applied a modification of the *Newton's method* (also known as Newton-Raphson method), which is a method for finding successively better approximations to the roots of a set of equations. It works with a derivative approach to the equations, which optimizes the time consumed to find the roots and makes the representation of the system easier.

SHPECK uses the *Gauss-Newton*'s method to solve a nonlinear system of equations which results in finding the roots of continuously differentiable equations.

The *Gauss-Newton*'s method is applied in order to acchieve the best approximation possible to the solution that is being seeked since it is an iterative algorithm where each step consists in minimizing the first-order approximation of the solution. *Gauss-Newton*'s method is a modification of *Newton*'s method (also known as *Newton-Raphson*'s method) for finding a minimum of a function. Its difference from regular *Newton*'s method is that second derivatives are not required. *Gauss-Newton*'s method is used to solve a system of coupled nonlinear equations. The first-order approximation of the function starts with an initial guess for the minimum values, the method proceeds by the iterations as shown in equations 4.2 and 4.3.

$$F(x+1) = F(x) - J^{-1} * R \tag{4.2}$$

or

$$F(x+1) = F(x) - \frac{F(x)}{F'(x)} \tag{4.3}$$

Where F is function's result for the applied x, J^{-1} is the inverse of the Jacobian matrix and R is the residual matrix (all the answers of all equations for the applied x). For details see (ISAACSON, 1966).

The R residual vector is defined as a vector containing the values for each equation using the interations's guess.

$$R = \begin{pmatrix} F(x_1) \\ \vdots \\ F(x_m) \end{pmatrix} \tag{4.4}$$

Where m is the number of unknowns (or mass action equations plus equilibrium constraints).

The algorithm consists of iteratively calculating new approximations for the unknown values, through the matrix equation:

$$[J]^{-1}_{iteration} * \alpha[U]_{iteration+1} = [R]_{iteration} \tag{4.5}$$

Where J is the Jacobian Matrix; $\alpha[U]_{iteration+1}$ is the unknown composition at the next iteration; iteration is the iteration number and R is the residual matrix. With this, is possible to state the $[U]_{iteration+1}$ value with:

$$[U]_{iteration+1} = [U]_{iteration} + \alpha[U]_{iteration+1} \tag{4.6}$$

The initial guess of the solutions is an approximation that the user provides to the *Gauss-Newton*'s method in order to have a way to start (normally this guess is used for $F(0)$). If the guess is close to the real root value than the number of iterations necessary to obtain the solution will be small, if the guess is something completely nonsense and far from the real solution, more iterations are going to be necessary to find the correct solution.

The equations that were organized before are now placed in a *Jacobian Matrix*. The *Jacobian Matrix* is the matrix of all first-order partial derivatives of the equation and it is defined as

$$J_{mn} = \frac{\partial y^m}{\partial x^n} \tag{4.7}$$

where the y^i's are a new coordinate system defined in terms of the original coordinate system, the x^i's. In differential equation theory, the Jacobian matrix plays a key role in defining the stability of solutions.

Specifically, *SHPECK*'s equations can be described as: $F_1(x_1, ..., x_n), ..., F_m(x_1, ..., x_n)$. The partial derivatives of all these equations with respect to the variables $x_1, ..., x_n$ can be organized in a m-by-n matrix, the Jacobian matrix, as bellow:

$$J = \begin{pmatrix} \frac{\partial F_1}{\partial x_1} & \cdots & \frac{\partial F_1}{\partial x_n} \\ \vdots & \ddots & \vdots \\ \frac{\partial F_m}{\partial x_1} & \cdots & \frac{\partial F_m}{\partial x_n} \end{pmatrix} \tag{4.8}$$

In our case, $m = n$ and the *jacobian matrix* is a square matrix and is generated once. With all the equations selected and organized, the derivatives of each reaction towards each variable are calculated and the *jacobian matrix* is modeled and stored.

Important to realize that the main complication with using a the *Gauss-Newton*'s method to solve a system of nonlinear equations is having to define all the functions included in the *Jacobian Matrix*. As the number of equations and unknowns increases (n), so does the number of elements in the *Jacobian* (n^2).

4.4.1 Gauss-Newton simple example

Consider the function $f(x) = x^2 - 612$, the derivative of this function is $f'(x) = 2x$, and the initial guess is 10. According to *Gauss-Newton Method*: $x_n = x_{n-1} - \frac{f(x)}{f'(x)}$

$x_1 = 10 - \frac{10^2-612}{2*10} = 35.6$

$x_2 = 35.6 - \frac{35.6^2-612}{2*35.6} = 35.6$

$x_3 = ... = 24.790635...$
$x_4 = ... = 24.738688...$
$x_5 = ... = 35.738633...$
...

The accurate of the solution is given according to the number of decimal places accuracy wanted. Increasing the number of iterations will automatically increase the precision of the solution.

4.5 Algorithm

PARA MUDAR!!!!! The algorithm takes as its input a specification of the system's state (the value of all of its variables) at some time t. It then calculates the system's state at time t+1. From the values characterizing that second state, it then calculates the system's state at time t+2, and so on. When run on a computer, the algorithm thus produces a numerical picture of the evolution of the system's state, as it is conceptualized in the model. PARA MUDAR!!!!!

4.5.1 Complexity of the algorithm

4.6 Software Engineering

4.6.1 Program organisation

4.6.2 Data flow

4.7 Graphical User Interface

4.8 Database

criterios de comparacao do pq usar um banco de dados estruturado; comparacao de tempo com o arquivo texto; simular pesquisa com os 2 formatos; consulatas elaboradas; concatenacao de elementos com queries; espaco, tempo e expressividade; uso de BD em area nao difundida.

4.9 Summary

5 VERIFICATION AND VALIDATION

As computer simulation methods have gained importance in more and more disciplines, the issue of their trustworthiness for generating new knowledge has grown, especially when simulations are expected to be counted as epistemic peers with experiments and traditional analytic theoretical methods.

- Explanation of the data that will be used to compare the results:
- Results comparison
- Summary

Results from initial studies indicated that the uncertainty for thermodynamic values is much greater around the very dilute range and the more concentrated range, where data for the thermodynamic constants are comparatively sparse.

6 CONCLUSION

Geochemical speciation is critical for understanding the form of chemicals of interest in natural systems. It is crucial in many different aspects of our daily life nowadays: assessing bioavailability, risk to humans and ecosystems. Geochemical speciation models are generaly determined through analytical methods that measure free ions or total concentrations, used in conjunction with thermodynamically based models. As presented on this work, these models rely on the local equilibrium assumption, and on experimentally determined reaction constants. It is clear, though, that if the chemical system does not achieve equilibrium state or if the equilibrium constants are not certain, the geochemical speciation model's prediction show significant uncertainty. This emphasizes the clear need for a proper computation approach while treating with such influential information like a geochemical speciation model. The information flow is tightly connected and any mistake will propagate errors and carry that wrong information until the very end of the model.

Calculation of speciation is conducted by substitution of the equilibrium constants into the mass balance expressions for the total concentration of a particular component. This results in a series of nonlinear equations which are solved iteratively using numerical techniques, such as the Newton-Raphson iteration used in the MINTEQA2 model. Iterations continue until the total calculated component concentrations, derived from the equilibrium expressions, calculated activity coefficients, and component mass balances, and the measured total concentrations converge to a prescribed limit.

- Review of the results emphasizing our approach and advantages
- Geological explanations for the results
- Final discussion about *SHPECK*
- CONTRIBUTION (EXPLICIT AND WELL DESCRIBED)

REFERENCES

R. M. GARRELS AND C. M. CHRIST **Solutions, Minerals, and Equilibria**: Harpers' Geoscience Series. Harper and Row, New York, 1965.

KEHEW, A. **Applied Chemical Hidrogeology**: Prentice Hall, 2000. 368p.

FREEZE, R. A. and CHERRY, J. A. **Groundwater**: Prentice Hall 1979

DOMENICO, P. A. and SCHWARTZ, F. W. **Physical and Chemical Hydrogeology - Second Edition**: John Wiley and Sons Inc.

DALE, NELL B. **Programming and problem solving with C++**: Jones Bartlett Publishers, 2004.

WOLERY, T. J. **Calculation of chemical equilibrium between aqueous solution and minerals: *the EQ3/6 software package***: Lawrence Livermore National Laboratory, Livermore CA, U.S.A.

WOLERY, T. J., JACKSON, K. J., BOURCIER, W. L., BRUTON, C. J., VIANI, B. E., KNAUSS, K. G. and DELANY, J. M. **Current status of the EQ3/6 software package for geochemical modeling in Chemical Modeling of Aqueous System**: ACS Symposium series, No 416, p 104-116. American Chemical Society, Washington, DC.

WOLERY, T. J. **EQ3/6, a software package for geochemical modeling of aqueous systems: package overview and installation guide (Version 7.0)**: Lawrence Livermore National Laboratory, Livermore CA, U.S.A.

KHARAKA, Y. K. and BARNES, I. **SOLMNEQ: Solution-Mineral Equilibrium Computations**: NTIS Tech Rept. PB214-899, Springfield, VA, 82p

DEBRAAL, J. D. and KHARAKA, Y. K **SOLINPU: A computer Code to Create and Modify Input Files for the Geochemical Program SOLMINEQ.88**

KHARAKA, Y. K., GUNTER, W. D., AGGARWAL, P. K., PERKINS, E. H., AND DEBRAAL, J. D., **SOLMINEQ.88; a computer program for geochemical modeling of water-rock interactions: Water-Resources Investigations Report.**

NORDSTROM, D., MUNOZ, K. **Geochemical Thermodynamics**: Prentice Hall, 200. 477p

PARKHURST, D. L. **User's guide to PHREEQC - A Computer program for speciation, reaction-path, advective-transport, and inverse geochemical calculations**: U.S. Geological Survey Water-Resources Investigations Report 95-4227, 143p

BROWN, D. S. and ALLISON, J. D. **MINTEQA1, an equilibrium metal speciation model: user's manual**: Environmental Research Laboratory, Office of research and development, U.S. Environmental Protection Agency.

ALLISON, J. D., Brown, D. S. and Novo-Gradac, K. J. **MINTEQA2/PRODEFA2, a geochemical assessment model for environmental systems: version 3.0**: Environmental Research Laboratory, Office of research and development, U.S. Environmental Protection Agency.

Qt **Application Framework**: Available at http://www.qt-project.org : Accessed in July 2014.

Arma **Armadillo C++ linear algebra library**: Available at $http : //www.arma.sourceforge.net$: Accessed in July 2014.

LEE, L. and GOLDHABER, M **The Geochemist's Workbench Computer Program**:

BETHKE, C. M. **Geochemical Reaction Modeling, concepts and applications**: Oxford University Press, 397p

BETHKE, C. M. **Geochemical and Biogeochemical Reaction Modeling**: Cambridge University Press, 547p

XU, T., SONNENTHAL, E.L., SPYCHER, N., PRUESS, K. **TOUGHREACT User's guide: A Simulation program for non-isothermal multiphase reactive geochemical transport in variably saturated geologic media**: Lawrence Berkeley National Laboratory, Berkeley, California, U.S.

HARBAUGH, A. W., BANTA, E. R., MCDONALD, M. G. **MODFLOW-2000, the U.S. Geological Survey modular ground-water model - User guide to modularization concepts and the Ground-Water Flow Process**: U.S. Geological Survey

NOFZIGER, D. L., WU, J. **CHEMFLO : Interactive software for simulating water and chemical movement in unsaturated soils**

PETRUCCI, RALPH H. **General Chemistry: Principles & Modern Applications**: New Jersey : Prentice-Hall

ZEGGEREN, F. VAN., STOREY, S. H. **The Computation of Chemical Equilibria**: Cambridge University Press

SMITH, W. R. **The Computation of Chemical Equilibria in Complex Systems**: Americam Chemical Society

LEAL, M. M. ALLAN, ET. AL. **A chemical kinetics algorithm for geochemical modelling**: Applied Geochemistry

DEUTSCH, J. WILLIAN, **Groundwater Geochemistry: fundamentals and applications to contamination**: CRC Press

PALANDRI, L. JAMES, KHARAKA, K. YOUSIF, **A compilation of rate parameters of water-mineral interaction kinetics for application to geochemical modeling**: U.S. Geological Survey

SHARIF, M. U. ET. AL. **Inverse geochemical modeling of groundwater evolution with emphasis on arsenic in the Mississippi River Valey alluvial aquifer, Arkansas (USA)**: Journal of Hidrology

ALLEY, WILLIAN M. **Regional Ground-Water Quality**

BOEHM, BARRY W., **Seven Basic Principles of Software Engineering**

HUMPHREYS, PAUL. **Extending Ourselves: Computational Science, Empiricism, and Scientific Method**

BAUER, FRITZ **Software Engineering: A Report on a Conference Sponsored by NATO**

WESTALL, J.C., J.L. ZACHARY AND F.F.M. MOREL **MINEQL, a computer program for the calculation of chemical equilibrium composition of aqueous systems**

APELLO, C.A.J. **Notepad++ with PHREEQC Plugin - www.hydrochemistry.eu/ph3**

Center for Exposure Assessment Modeling (CEAM) $http : //www2.epa.gov/exposure - assessment - models/minteqa2$: accessed in 01/10/2015

Center for Exposure Assessment Modeling (CEAM) $http : //eng.odu.edu/cee/resources/model/minteqa_unix.shtml$: accessed in 01/10/2015

GAMMA, ERICHT. ET AL **Design Patterns. Elements of Reusable Object-Oriented Software.**

STROUSTRUP, BJARNE **The C++ Programming Language (Third ed.)**

ISAACSON, E., KELLER, H. B. **Analysis of Numerical Methods**: Wiley, New York