CHALMERS

LJUNGSKILE 2.4

Expansion and modernization of LJUNGSKILE 2.3

Master's Thesis in the Master Degree Programme, Nuclear Engineering

MAGNUS MALMBERG

Department of Chemistry
Division of Nuclear Chemistry
CHALMERS UNIVERSITY OF TECHNOLOGY
Gothenburg, Sweden, 2015
Report No. TIFX03

LJUNGSKILE 2.4 - Expansion and modernization of LJUNGSKILE 2.3
Development of software intended for uncertainty assessment of chemical speciation calculations
MAGNUS A. MALMBERG

Technical report no TIFX03
Department of Nuclear Chemistry
Chalmers University of Technology
SE-412 96 Göteborg
Sweden
Telephone + 46 (0) 31-772 1000

[tryckeriets namn]
Göteborg, Sweden 2015

LJUNGSKILE 2.4 - Expansion and modernization of LJUNGSKILE 2.3
Development of software intended for uncertainty assessment of chemical speciation calculations
MAGNUS A. MALMBERG
Department of Nuclear Chemistry
Chalmers University of Technology

SUMMARY

In this paper the development of a new version of LJUNGSKILE, a program used to calculating chemical speciation. Chemical speciation is a technique used to determine the concentration of different complexes formed in an aqueous system. Among other areas, chemical speciation can be used to analyze how radioactive material would be spread in the ground water in the event of a large scale nuclear accident.

Most tools which are available for calculating chemical speciation does not take statistical uncertainties into account. And as such does not give any information regarding the magnitude of deviation between concentrations in a real sample and the concentrations which are determined in the chemical speciation calculation. The LJUNGSKILE program however is intended for qualitative analysis of the uncertainties in chemical speciation calculation.

More simulation and plotting capabilities has been added to the LJUNGSKILE program adding the option of initiating 2-dimensional simulation setups and display these as 3D-dimensional plots to illustrate how species may vary as a function of both e.g. acidity and ionic strength.

The new features are demonstrated through 3 simulations which illustrate both the possibilities of 2-dimensional simulation setups and how varying the options of a simulation setup can be expected to impact on the end result.

TABLE OF CONTENTS

1. INTRODUCTION

This Master's Thesis describes the development of LJUNGSKILE 2.4, which is a program used for chemical speciation. In this report, both the applications for chemical speciation and the development process of the LJUNGSKILE2.4 source code will be covered. The report is written for readers with knowledge in chemistry but without previous programming experience. As such, any previous experience in computer programming or chemical speciation will be beneficial but not required.

1.1. BACKGROUND

The properties and transport of metals in nature, whether it is natural deposits or deposits created by men, is largely determined by aquatic systems. As there are plenty of metals that are valuable resources and quite a few that are hazardous, it is important to be able to describe how these metals behave. The description of which species are present in a given solution is called speciation, or chemical speciation.

Speciation is important when aquatic systems, as describing a solution only by determining the concentration on all elements present in the system is rarely sufficient to predict the behavior of the solution. The behavior of dissolved metal is highly dependent on the coordination of metal ions and as such, is very complex.

The application for speciation calculations range between e.g. safety analyses and design processes for mineral extraction. Speciation is for example used to determine the appropriate safety measures needed for long-term storage of nuclear waste disposal. The repository must be designed to prevent the effect of leakage. If a leak in the spent nuclear fuel containment should occur the repository must ensure that the fuel will not dissolve in groundwater and eventually reach sea level until it has decayed sufficiently to be considered harmless. To successfully perform a speciation on spent fuel dissolved in groundwater both the water chemistry and the fuel content must be considered.

The basics of speciation calculations are standard thermodynamic equilibrium calculations, but with each potential species that can be formed the complexity of the calculations is increased. The number of equilibrium reactions that has to be considered in an aqueous is normally too large for the system to be solved manually. Instead, computer programs have been developed to perform the speciation. At the core of these programs is the chemical and physical data of the ions in the solution. These data have to be determined experimentally and thus involve inherent uncertainties that will affect the outcome of the speciation analysis.

1.2. LJUNGSKILE

The program LJUNGSKILE 2.3, hereby referred to as LJUNGSKILE, is a designed to perform a chemical speciation analysis, as a function of e.g. pH or reduction potential (henceforth abbreviated pE) .

LJUNGSKILE is specialized in presenting its results with standard deviations taken into account, compared to similar tools which typically do not consider statistical uncertainties. While it is always important to be aware of the accuracy of calculations performed, the uncertainty of tabulated equilibrium constants may for example mean that the predominating species in a sample differ from those calculated.

The program is used for this purpose all over the world, among others the Japanese radiation protection agency. The program is made to work as a graphical user interface for PHREEQC (abbreviation for pH-REdox-EQuilibrium, written in C) which is the program that performs the actual speciation calculations based on input code. Being a text based program, PHREEQC requires more time to master than a similar program with a graphical interphase. The data achieved is then assorted and can be presented in plots. Many users have limited computer experience and need a program that is easy to use and does what it’s told.

1.3. PURPOSE

The main aim of this master thesis project is to expand the calculating and plotting capabilities of LJUNGSKILE to incorporate predominance diagrams. The simulations will be made in two-dimensions instead of one and the data from these simulations will be presented graphically in a comprehensible manner in order to make the program more useful overall.

The introduction of two dimensional simulations is made in order to be able to present results in 3- dimensional plots and predominance diagrams. Both of these presentation types are useful to visualize possible speciation chemistry over an area of varying acidity and electric potential. In the case of leaking nuclear waste deposits the predominance diagram might provide data for potential leakage scenarios and optimal conditions for final repository sites.

The incorporation of predominance diagrams brings a demand for additional plotting capabilities. While three-dimensional plotting is not required for predominance diagrams, the introduction of two-axis variables for speciation will also be used for three dimensional plotting. To ensure that the user experience remains familiar and coherent, the display program will be substituted by an internal component of the LJUNGSKILE program, which will essentially retain the same graphical layout as the previous plotting suite.

2. THEORY

2.1.METALS IN AQUEOUS SOLUTION

Surrounded by water, metal ions form complexes with water molecules or ions present in the solution. Metals in aqueous solution typically form complexes with six ligands, albeit some metals have higher and some have lower coordination numbers. What ligands will coordinate around a given metal ion is determined by the chemistry in the vicinity of the metal ion, which in turn is determined through a vast series of thermodynamic equilibrium reactions. As the metal solution chemistry is governed by the equilibrium reactions, the metal complexes formed will vary with the water chemistry of the solution. (Martell, A E. 1952)

2.1.1. CHEMICAL SPECIATION

Chemical speciation describes how an element is distributed among chemical species within a given system. It is a crucial instrument to describe the behavior of e.g. metals forming various complexes in aqueous solution. However, to provide a speciation analysis, analytical chemistry alone will generally not suffice. In many cases the concentration of metals are very low and cannot be measured directly. (Hanrahan, G. 2010)

Instead of trying to measure trace amounts of metals, chemical speciation may be performed analytically. The basic mechanism governing the behavior of metals in aqueous solution can be described with a simple thermodynamic theory. From the reaction

$$\mathrm{A} + \mathrm{B} \iff \mathrm{C} + \mathrm{D}$$

With the equilibrium constant K, assuming ideal behavior, the concentration of D is described by

$$\log\,[\mathrm{D}] = \log \mathrm{K} - \log\,[\mathrm{C}] + \log\,[\mathrm{A}] + \log[\mathrm{B}]$$

However, a reaction formula only describes a single possible reaction, while the concentration of the species described in the reaction may be involved in more reactions dependent yet another set of species. The codependency of species in a solution rapidly creates a complex network of equations that needs to be solved simultaneously.

Computer modeling of chemical speciation is the main way to mitigate the barriers associated with analytical speciation. There are quite a few programs designed to model the chemical speciation available, for example PHREEQC, which is used for the speciation calculations in the LJUNGSKILE program. (Hanrahan, G, 2010)

2.2. STATISTICAL CONSIDERATIONS

No matter how advanced the speciation modeling is, the accuracy of the model will always be limited by the accuracy of the equilibrium data provided. The uncertainties of the equilibrium constants stems from the fact that these are achieved through experimental studies. Minimizing the uncertainties of the equilibrium constants will of course minimize errors in the model, but it is impossible to completely avoid errors. The reliability of a speciation calculation will thus be strongly dependent on the interaction between ions in a given solution and the magnitude of the uncertainties of used constants.

Acknowledging the existence statistical errors and measuring their impact on a speciation study will provide information about the reliability of the study, which is of great concern in e.g. the design of final repository for nuclear waste.

2.3.MONTE CARLO SIMULATIONS

In Monte Carlo Simulations a large amount of calculations are used to statistically determine a mean value and standard deviation of a function is calculated from an array of random sampling distortion cases of the function's variables.

[When trying to determine a variable Y that is a function of several variables a_1, a_2, …, a_k, the result is dependent on the values used for the variables of which Y is a function of. If, for any reason the value of the variable a_1, a_2, …, a_k cannot be determined exactly (e.g. due to uncertainties in measurement of the variables) then the variable Y cannot be determined exactly either. In such cases, it is important to keep track of the uncertainties, as a large uncertainty in the parameter a_i might have a minor impact on the calculated value of Y, while a small uncertainty in another parameter a_j might have a major impact on the value of Y.

In some cases it is possible to assess these uncertainties analytically, while in other cases, the sheer number of variables or equations makes analytical solutions impractical, or even impossible. In such cases a commonly used method is Monte Carlo simulations.

In a Monte Carlo simulation the variable Y is calculated by sampling values from the probability density functions of the variables a_1, a_2, …,a_k. The calculation is performed a large number of times, how many times it is performed is determined by the desired accuracy. More calculations naturally require more computational time, which means a higher cost to perform the calculation. The number of calculations needed to produce satisfactory statistical data for the variance of the variable Y will depend on the function being evaluated and the accuracy of the variables used in said function.

Higher accuracy is always something that is always desirable, albeit not always time- or cost-effective. The trade-off between accuracy and computer time can be partially bypassed by clever sampling methods.

To perform the Latin hyperbolic sampling method a probability density function for each variable that is sampled has to be determined. The probability density function is a function which returns the probability of a variable to be equal or less to any given value. The Latin hyperbolic sampling method then divides the probability density function of each variable into smaller intervals with equal probability. One sample is drawn from each interval using the probability density function of that interval. The samples of each interval are then randomly combined ith samples from all other variables, drawn in the same way. The result is a matrix with k columns and n rows, where k is the number of variables used and n is the number of samples for each variable. The advantage of this method is that a comparably small number of samplings will yield a high degree of coverage for all variables used, ensuring both that a broad interval of samples is used for each variable and that the overall samplings are not used in a narrow interval.

The following example will illustrate the main differences between normal random sampling and latin hyperbolic sampling. A number of uniform random numbers R between 0 and 1 are generated for each variable. For normal sampling, the sampled value of the variable is picked using the inverse probability density function. In latin hyperbolic sampling, values are instead picked in equally probable intervals, using the inverse probability density function, this corresponds to:

$$RLHS, j = \frac{R}{n} + \frac{j-1}{n}$$

Where $R_{LHS,j}$ is the number generated in a given interval j, R is a uniform random number between 0 and 1 n is the number of samples taken and j is the current interval.

To account for the uncertainties present in equilibrium data, both normal sampling and Latin hyperbolic sampling are used in the LJUNGSKILE program. In each sampling, the equilibrium constants are varied within an interval given by the sampling method, based on the confidence interval supplied with each constant. The two sample methods used when analyzing the accuracy of speciation calculation are random sampling and Latin hypercube sampling. (Hernández.solís, A. 2007)

2.3.1. *RANDOM SAMPLING*

A straightforward way to account for the uncertainties of equilibrium constants is to run Monte Carlo Simulations (MCS) without any specialized sampling method. With random sampling all used equilibrium constants are simply varied within a normal distribution. The output collected from these simulations is then used as statistical data to determine mean values and standard deviation of each species at the specified aqueous conditions.

2.3.2. *LATIN HYPERCUBE SAMPLING*

Where random sampling utilizes brute force to determine the uncertainties through sheer number of simulations, the Latin Hypercube sampling (LHC) divides the distribution into several equally probable intervals. The data is then picked from each of these intervals to ensure that the sampling is representative for the solution. The divided intervals and hence the representative ensure that reliable statistical data is provided with a minimum of calculations.

2.3.3. *CHOOSING SAMPLING METHOD*

LHC provides rapid results and it can often prove to be a valuable tool for quickly assessing uncertainties associated with chemical speciation. The efficiency of the LHC depends on the fact that the LHC assumes no dependency between the variables used, which is rarely the case in chemical speciation. It should however be emphasized that the errors induced by the assumption that no dependency between variables exists in many cases are negligible compared to the accuracy of the equilibrium data provided

Random sampling however excels in the areas where LHC lacks; basically the only assumption made is that the statistics of a series of simulation with normal distributed variations is sufficient to describe the uncertainty of the whole system.

As both methods have its inherent strengths and weaknesses, both options are available as options in the LJUNGSKILE program.

3. DEVELOPMENT METHOD

3.1. DEVELOPMENT TOOLS

Overall, very basic tools were chosen to be utilized in the development process. Care was also taken to find older alternatives to the programs needed, to ensure that the program will be executable even on older computers.

For writing source code in C++ and compilation of the main program, Borland builder 6 was used. Builder 6 is developed for Windows XP service pack 1, which was released in 2001. This means that the program should be compatible with most computers which are presently in use.

For code review and editing e. g. data files and plotting instructions notepad++ was used.

Finally, for plotting the data, GNUPLOT 4.0 was used. GNUPLOT 4.0 was released in 2004.

3.2. PRELIMINARY STUDY

To provide input to the planning of the master's thesis and the project planning report, a preliminary study was carried out. The source code for LJUNGSKILE was studied and prepared for the project. Besides from providing vital input for the construction of the project's time plan, the aim of the preliminary study was to allow variation of and additional variable, going from one to two variables (e.g. pH and pE in predominance diagrams).

The preliminary study was initiated by creating a new graphical user interface for MULTIRUN and adapting the source code to be compatible with LJUNGSKILE. The MULTIRUN is a subprogram that is responsible for the setup of the interval and step size of the variable that is used for the calculations. The focus of the graphical design was to use the same functions and structure as MULTIRUN had before the update but to make the design slightly more compact to limit the window size as it had to incorporate support for two variables instead of one. The rewriting of the source code was focused on avoiding deviations from the original code when possible to limit the time needed to perform the preliminary study.

Unexpected problems relating to lack of independency between different subprograms led to the decision to evaluate the preliminary study before a fully functional alpha version of the program could be presented.

3.2.1. INSIGHTS OF THE PRELIMINARY STUDY

The general conclusions of the preliminary study was that creating the graphical user interface would not contribute to a significant part of the time needed for the project, even though some modernization of the design of the applications will have to be performed.

In the project's time schedule, the functional source code programming is expected to require less time than expected before the preliminary study, due to the structure of the LJUNGSKILE-PHREEQC interface being more general than the documentation of PHREEQC suggests. The fact that the present source code does not utilize the full potential of PHREEQC might result in unnecessary long computational time, however this was not found to be a major issue, and updating the interface between LJUNGSKILE and PHREEQC was not found to be necessary.

As for the code structure, the previous version of LJUNGSKILE needed to be reworked thoroughly. As the author of the original source code is not involved in the project, the structure of the updated parts of the code will differ from those parts that do not need to be updated if a complete revision would not be performed. This could potentially cause accumulating readability issues resulting in making debugging and future updates of the program extremely time consuming.

3.3. Implementation

The project is mainly aligned to programming, which results in difficulties in continuous evaluation of the project progression. To evaluate the progress, each subprogram involved is programmed in four steps, analysis, coding, (in some cases adaptation) and compilation. The order in which the subprograms are updated is determined by the level of relation to previous steps, starting with MULTIRUN which is included in the pre-study.

During the analysis stage the code of the sub-program is translated into pseudo code to provide a good overview of the subprogram functionality. The idea of the pseudo code was to describe the program in layman's terms with some basic logical functions still left more or less unaltered, without any actual functionality. Based on the pseudo code, the features to be altered, added or removed in the coding stage are identified.

The coding stage involves the actual work with coding as well as the graphical design of each subprogram.

The adaptation stage is only performed in the plotting sub-program. It involves backtracking into previously written code to make sure the program features perform as desired.

The last stage of the sub-programming is the compilation stage, were errors in each sub-program are identified and corrected, in some cases leading to additional need of coding and program adaption.

Once the functional coding stages are finished, the code readability is enhanced. Deactivated features will be removed and explanatory comments added for all functions that are not sufficiently comprehensible. The visual program shell is also slightly updated to give LJUNSKILE a more finished look.

4. PROGRAM DESCRIPTION

4.1. INTRODUCTION TO THE PROGRAM

LJUNGSKILE consists of three main parts. The main parts are Chemical calculations, statistical method and visual presentation.

While the actual calculations are performed in PHREEQC, the LJUNGSKILE provides a graphical user interface to perform multiple simulations automatically and without having to learn the PHREEQC code language. The statistical method varies the equilibrium constants within a normal distribution using either MCS or LHC, depending on the user's preference. The equilibrium constants and deviations are provided by the user, with the option of using predefined equilibrium libraries. Merged with the chemical calculations, these parts are responsible for the calculations performed in the LJUNGSKILE program. Finally, the visual presentation performs all the post-processing of data and a range of options for visualization through diagrams presented in GNUPLOT.

4.2. PROGRAM STRUCTURE

The LJUNGSKILE program consists of several dialog windows. These windows include separate source codes, and each window is responsible for a specified set of functions within the LJUNGSKILE program. This specified set of functions will hereby be referred to as subprograms. Though the subprograms do not necessarily have to be included into a window, the windows and the functions that they are associated with will be regarded as a single unity for the sake of clarity.

The table on the next page is a quick reference for the program structure, the subprograms and their general functions. Along with LJUNGSKILE, the PHREEQC is used for calculations and *GNUPLOT* for plotting.

Table 1. Subprogram structure of LJUNGSKILE

Subprogram	function
LJUNGSKILE	This is the main program, which lists the source codes to be included in LJUNGSKILE.
Init	Main window, responsible for loading projects and initiating simulation calculation
Project description	Options for project parameters
Method description	Options for sample method (Latin hypercube or Monte Carlo)
Water description	Options for water chemistry parameters
Simulation start	Responsible for interaction between LJUNG-SKILE and PREEQC. Dialog window is not displayed
Graph	Responsible for compiling PHREEQC output data for the plotting programs. Dialog window is not displayed
Version	Displays current version of the LJUNGSKILE program
Multiple runs	Options for running multiple simulations, including options for number of variables to be used
LJUNGSKILE internal display program (LiDP)	Visual shell for initiation of graphs using GNUPLOT for 3D and surface plotting.
Variables	Sub menu in LiDP with options for regular 2D plotting in the LiDP

4.3. Implemented changes

The LJUNGSKILE program have been reworked to support two-dimensional variation of selected variables and to display the results using *GNUPLOT* instead of the previously used *LJUNGSKILE display program* (henceforth *LDP*). To ensure program stability and avoid unintentional changes in the code that leads to bugs and calculation errors in the final program, care has been taken to avoid altering iteration instructions and communication with *PHREEQC* where possible.

Overall, the appearance of the program remains the same, including the graphical user interface used for *GNUPLOT*, which mimics the design of the previous *LDP*. New dialog windows are designed to resemble older windows and the raw data is stored in the same way as before.

4.3.1. LJUNGSKILE

Changes include defining additional variables to store and distribute instructions for simulating with two variables and also defining functions and variables used in the internal display program (LiDP).

4.3.2. Init

The instructions for initiating *LDP* have been rerouted to initiate LiDP; also, the extra variables needed for instruction of dual variables for the *multiple runs* subprogram have been added in the *init* program to allow access in other parts of the program, such as simulation and graphing routines.

4.3.3. Project, Method and Water description

No changes have been implemented in the *Project description*, *Method description* or *Water description* subprogram. The function files were clear enough to be easily interpreted and no additional comments in the code where necessary.

4.3.4. Simulation Start

The previous code was written to vary a single variable in a preset interval; these instructions have been expanded to incorporate the option of using two variables and this is the area where the upgrade of the code impacts the simulation routines of PHREEQC.

4.3.5. Graph

The data files produced by PHREEQC have been programmed to be compiled to matrices and initial plot instructions have been implemented to instruct GNUPLOT's start-up routines.

4.3.6. LJUNGSKILE Display Program (LDP and LiDP)

The whole plotting tool has been rewritten, the main reason being that the previous tool was not supplied with source code. The LiDP performs the calculations needed to supply matrices with appropriate margins of error as *GNUPLOT* does not support mathematical operations when handling data. The LiDP also provides a visual shell for performing instruction manipulation of the *GNUPLOT* program.

Care has been taken to write the LiDP in such a way that advanced users can choose to rewrite or replace the plotting instructions for *GNUPLOT* according to their needs. The measurements taken to allow this include an editable template file with basic instructions for *GNUPLOT*, which forms the basis of all *GNUPLOT* instructions and self-explanatory nomenclature on data files created.

4.3.7. Multiple Runs

As described in the preliminary study included in the development method description the multi run instructions have been expanded to include a second variable. This also included minor updates in various subprograms (e.g. Init and Simulation Start). To support the inclusion of second variable, additional safety barriers were added. The barriers were made to prevent conflicting instructions associated with choosing the same variable for both axes of the predominance diagram.

5. SIMULATION

The introduction of 2-dimensional matrices in LJUNGSKILE was successful, and while 3-dimensional plotting is now supported, pure predominance diagrams could not be displayed using GNUPLOT. In order to demonstrate the new features of LJUNGSKILE a monte carlo simulation of an iron complex was run. As compability could potentially be an issue when switching to a newer version of a code package, the data files used for this simulation was from an older version of the LJUNGSKILE program.

While a few minor tweaks are required to execute the simulation using the data file, in this case only updating LJUNGSKILE version info in the simulation instruction files (see appendix I) the new program is considered to be compatible with older data files.

As the LJUNGSKILE program will terminate if an error occurs in the input variables to the simulation in PREEQC, a stable interval for the simulation was determined using very low resolution and only ten samplings per node. This results in a fairly rapid simulation that gives an idea of how the distribution between species will be at every node, but not enough data for a good statistical analysis.

When the low resolution simulation has been successfully executed, the resolution of samplings is increased to yield a more detailed distribution matrix with higher accuracy.

The low resolution simulation takes approximately 20 minutes on a modern standard computer (the one used in these tests where produced in 2012 with windows 7 as opperating system) while the high resolution calculation can run for more than a day on the same system, hence the need for low resolution simulations to ensure successful execution of the high resolution simulation.

As the 3-dimensional are harder to read than 2-dimensional plots, the plotting program incorporates the possibility to get the cordinates for a desired points in the matrix from an overhead view, and input the cordinates in order to display 1- and 2-dimensional plots of the desired cordinates. It is of course still possible to get the exact values of a given node by reading the data files generated in the simulation, but as the purpouse of the LJUNGSKILE code is to present qualitative and not quantitative uncertainty analysis, this fubction has not been introduced into the program suite.

The first simulation was performed between pH 0 and 6 and pE -2 and 4, both at a step size of 2. Each node in the matrix was simulated 50 times using the Latin hypercube method. The purpose of this simulation was to analyze if the range for pH and pE which was to be used in the following simulations would be executable, as the PHREEQC is prone to failure at simulation conditions outside a stable interval.

The second simulation was performed at the same range (pH 0 to 6 and pE -2 to 4) but the step size was decreased to 0.2, yielding a hundred times finer mesh. However, the number of simulations per node was decreased to 20. This simulation was performed partly to ensure that all points in the mesh would be stable for the last simulation, and partly to illustrate the impact of varying the number of runs for each point on uncertainties and accuracy.

The third simulation where performed on the same range and step-size as the second simulation (pH 0 to 6 and pE -2 to 4, step-size 0.2) but with 50 simulations per node in the matrices. The simulation time was considered to be reasonable bearing in mind the risk of the program terminating due to errors or convergence problems in the PREEQC program. Increasing the nodes significantly also reduces the performance of GNUPLOT.

Table 2. The setup for simulation 1-3, including pH, pe, simulation resolution and number of simulations per node.

	pH	pe	Resolution	simulations per node
Simulation 1	0 to 6	-2 to 4	2	50
Simulation 2	0 to 6	-2 to 4	0.2	20
Simulation 3	0 to 6	-2 to 4	0.2	50

6. RESULTS

All three simulation setups described in the simulation section where run successfully. The first simulation, with a total of 16 nodes (see figure 6-1 and 6-2 below), indicates that there are four larger areas where which species are predominating are fairly certain. While the standard deviations are very large due to the small number of samplings, the overall trends are the same as in the second and third simulation.

Except for small deviations, the second and third simulation yield the same mean values for each species in each node.

All three simulations where found to be fairly similar and with a few exceptions the predominating species where the same. Naturally, the lower resolution of the first simulation results in discrepancies as the plotting program interpolates between the nodes, but comparing the nodes of the first simulation with the same nodes in simulation three it can be seen that they yield roughly the same results.

6.1. COARSE MESH

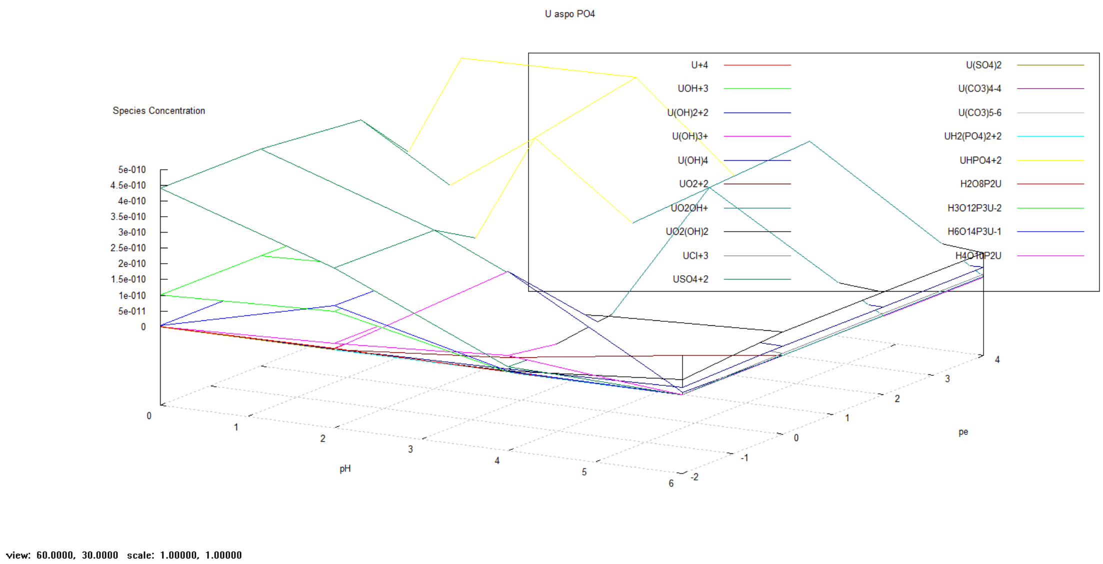

Figure 6-1. 3-dimensional plot of the coarse mesh simulation.

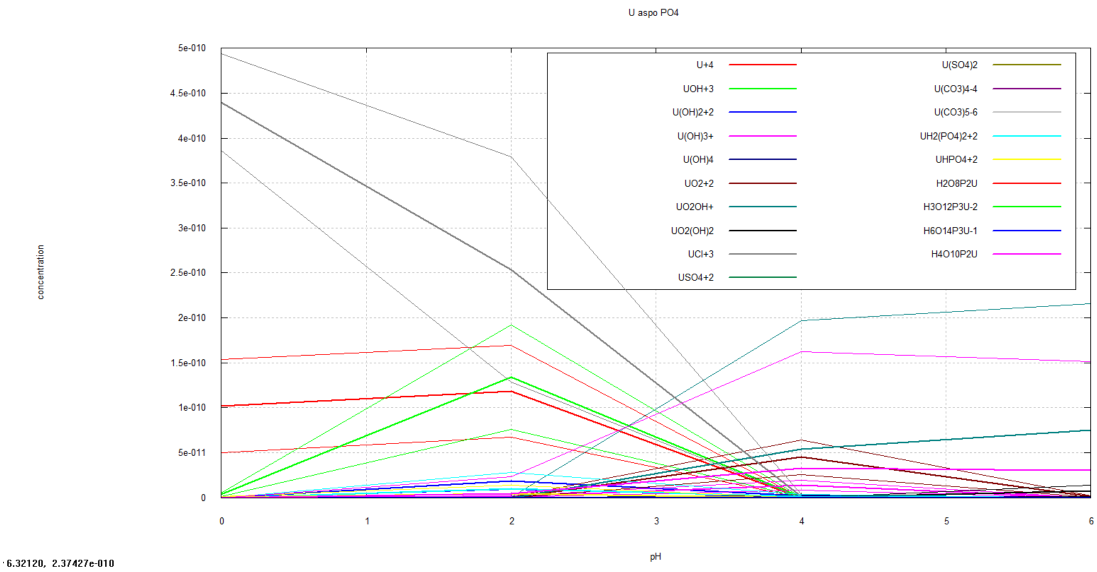

Figure 6-2. 2-dimensional plot of the coarse mesh at pe=0. Thin lines represent the 95%-certainty interval for the species of corresponding colours.

6.2. Fine mesh, low run count

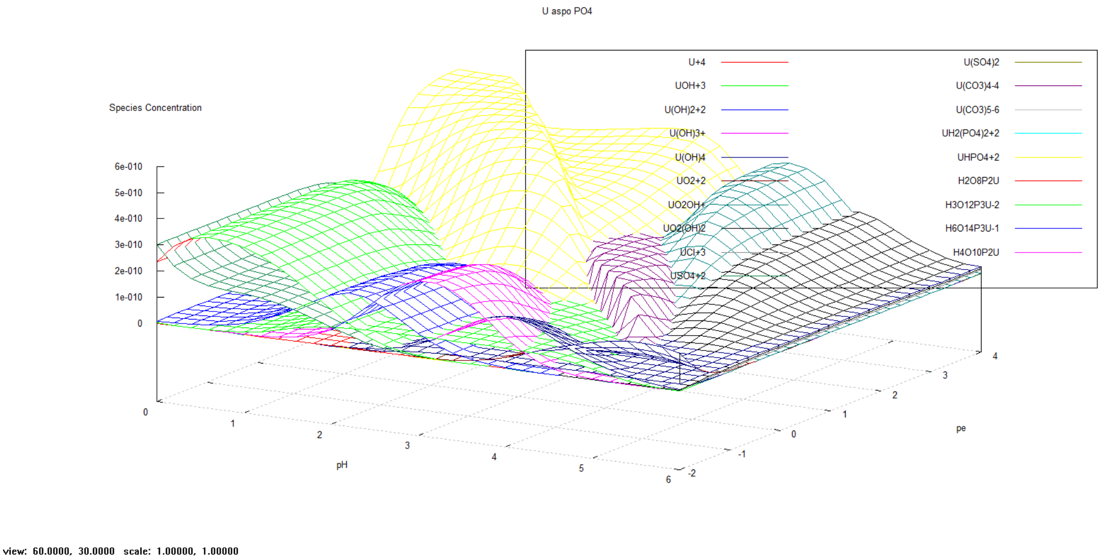

Figure 6-3. 3-dimensional plot of the fine mesh simulation with a low run count.

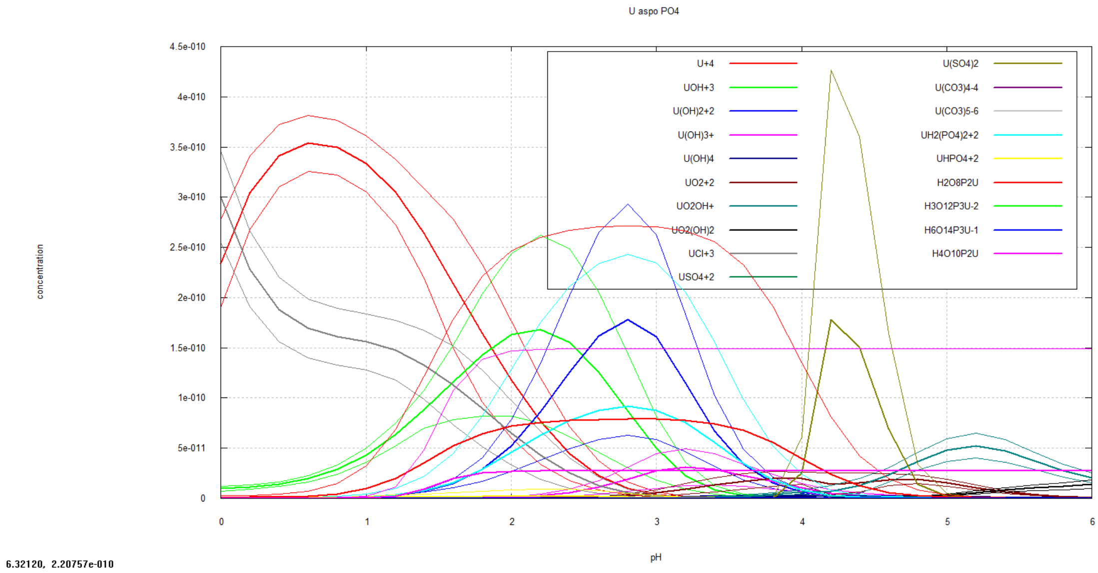

Figure 6-4. Figure 6-5. 2-dimensional plot of the fine mesh simulation with a low run count at pe=0. Thin lines represent the 95%-certainty interval for the species of corresponding colours.

6.3. Fine mesh, high run count

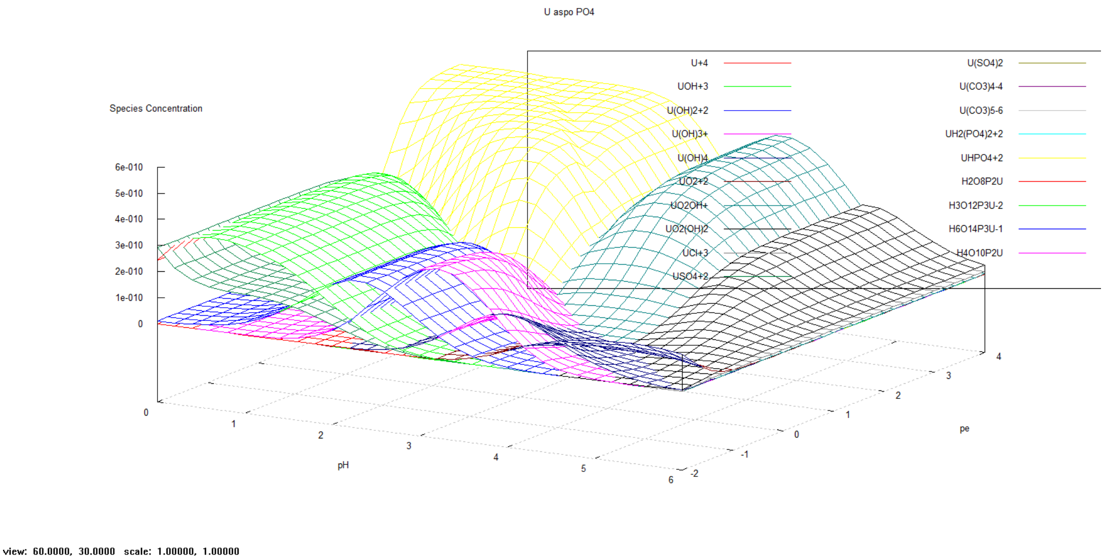

Figure 6-5. 3-dimensional plot of the fine mesh simulation with a high run count.

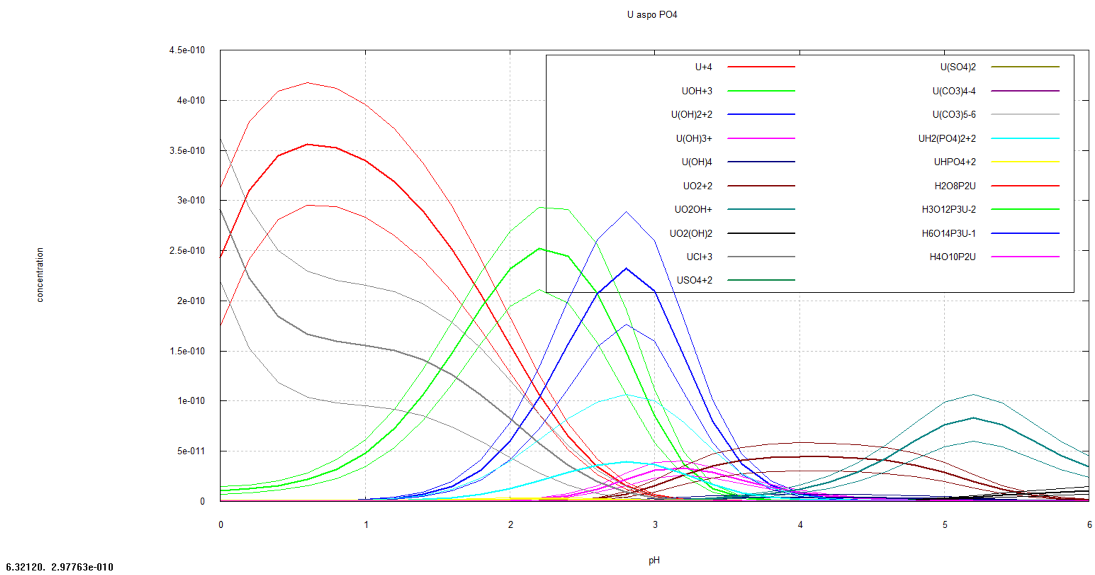

Figure 6-6. 2-dimensional plot of the fine mesh simulation with a high run count at pe=0. Thin lines represent the 95%-certainty interval for the species of corresponding colours.

7. Conclusions

The implementation of 2-dimensional simulations and 3-dimensional presentation of the simulated data was successful. The limitations of GNUPLOT meant that the predominance diagrams merely is a 2-dimensional projection of the 3-dimensional plotting area and not an actual level map. However, as the predominating specie will be the only specie which is visible, one could argue that the basic functions and applications of these projections means that they are in fact predominance diagrams, albeit not presented as a level map.

The addition of predominance diagrams means that it is easier to identify areas of large uncertainty and probable optimum regions for formation of species. This can be used to assess the uncertainties in the calculated behavior of final repositories for nuclear waste. While the search for optimum conditions for formation of a certain specie would benefit from adding additional dimensions to the simulations, there are limited ways of presenting data in an accurate and comprehensible manner when 3-dimensional projections does not suffice. While limitations in the original code restricts the resolution of the simulation, the lack of hardware acceleration in GNUPLOT means that higher resolution would not be practical to plot anyway. This is, as stated in chapter 2, a deliberate decision to ensure compatibility with older computers.

While the success in creating predominance diagrams was limited, the program is now a more powerful tool for assessing how variations in water chemistry impact the abundances of the examined species in a solution and demonstrates the how false conclusions might be drawn from modelling water chemistry if statistical errors are not properly accounted for.

The second one of the simulations presented requires some additional comments. It had the same basic shape as the other simulations, but some species that where expected to be almost non-existent in the other simulations where at some points predominating. As the graphs where all smooth, this behavior cannot be fully explained by that the lower number of simulations in a given point would yield more inaccurate results. However, LJUNGSKILE provides PHREEQC with initial guesses for each species in a node based on the values of the previous nodes in the same row. This would mean that a small error could propagate through the matrix. The first explanation is supported by an increased uncertainty that can be observed in the areas where the second simulation deviates from the other simulation, while the second explanation is supported by the fact that the deviations seems to increase as the pH value increases.

Conclusively, the number of simulations per node has a large impact on the final results.

8. References

Hanrahan, G. (2010) Modelling of Pollutants in Complex Environmental Systems Volume II. Glendale, AZ: OLM Publications.

Hernández.solís, A. (2007) Uncertainty and sensitivity analysis applied to LWR neutronic and thermal-hydraulic calculations. Göteborg: Chalmers University of technology. (Thesis for the degree of doctor of philosophy department of chemical and biological engineering, division of nuclear chemistry).

Martell, A E. (1952) the behavior of metal complexes in aqueous solutions. Journal of chemical education, 29 (6), p 270

APPENDIX I. SIMULATION FILES

Table 3. Project file used for the last simulation described in chapter 5 and 6.

Current .PRJ file [U aspo PO4.prj]
LJUNGSKILE ProjectFile version 2.0 Begin Project Solid Phase Solid Phase Amount 0 CO2(g) N CO2(g)partial pressure 0 No Species 19 Species ('name' 'mean' 'SD or max value' 'distribution' U+4 -14.9 0.1 Master UOH+3 -0.6 0.2 Normal U(OH)2+2 -2.5 0.2 Normal U(OH)3+ -5.5 0.2 Normal U(OH)4 -10 0.2 Normal UO2+2 -9.1 0.2 Normal UO2OH+ -14.9 0.2 Normal UO2(OH)2 -21 0.2 Normal UCl+3 1.72 0.5 Normal USO4+2 6.58 1 Normal U(SO4)2 10.5 1 Normal U(CO3)4-4 32.9 1 Normal U(CO3)5-6 34 1 Normal UH2(PO4)2+2 47.105999 1 Normal UHPO4+2 24.4 1 Normal H2O8P2U 46.689999 1 Normal H3O12P3U-2 67.739998 1 Normal H6O14P3U-1 56.68 1 Normal H4O10P2U 35.43 1 Normal End Project Begin Water Description Äspö groundwater pH 5 pe -4.37 Temperature 15 No elements 19 Elements ('name' 'concentration') Ca 0.0472 Mg 0.00173 Na 0.0913 K 0.000207 Fe 4.37E-6 Mn 5.28E-6 Al 1E-6

```
Cl 0.181
C 0.000164
S 0.00583
F 7.9E-5
Br 0.000501
P 1.61E-7
Si 0.000146
U 5.45E-10
Sr 0.000399
Li 0.000144
N 3.52E-6
Th 1E-7
End Water
Begin Sampling Method
Seed
-1
Method ( 'nr' 'name' )
1 Latin
Method parameters
50
1000
End Sampling Method
Begin Multiple runs
Multirun
Y
Species
pH
Start
0
Stop
6
Interval length
0.2
Logaritmic scale
N
End Multiple runs
End Project
```

Table 4. Data file used for all three simulation described in chapter 5 and 6.

```
# This is a PHREEQC database file converted from
# PHREEQE format with PHCBC, PHreeqe to preeqeC (data)Base Converter
#
SOLUTION_MASTER_SPECIES
#
#element species                 alk       element_gfw
#
H        H+       0.000000       H         1.007970
H(1)     H+       -1.    0.0
E        e-       0.000000       0.0       0.000000
O        H2O      0.000000       O         15.999400
O(-2)    H2O       0.0    0.0
Ca       Ca+2     0.000000       Ca        40.080002
Mg       Mg+2     0.000000       Mg        24.305000
Na       Na+      0.000000       Na        22.989799
K        K+       0.000000       K         39.098301
Fe       Fe+2     0.000000       Fe        55.847000
```

```
Mn      Mn+2    0.000000        Mn      54.938000
Al      Al+3    0.000000        Al      26.981501
Ba      Ba+2    0.000000        Ba      137.330002
Sr      Sr+2    0.000000        Sr      87.620003
Si      H4SiO4  0.000000        Si      60.084301
Cl      Cl-     0.000000        Cl      35.452999
C       CO3-2   2.000000        C       44.009800
S       SO4-2   0.000000        S       96.059998
N       NO3-    0.000000        N       62.004902
B       H3BO3   0.000000        B       10.810000
P       PO4-3   2.000000        P       94.971397
F       F-      0.000000        F       18.998400
Li      Li+     0.000000        Li      6.941000
Br      Br-     0.000000        Br      79.903999
U       U+4     0.000000        U       238.000000
Np      Np+4    0.000000        Np      239.000000
Cs      Cs+     0.000000        Cs      132.910004
Cx      Cx+2    0.000000        Cx      58.930000
Pu      Pu+4    0.000000        Pu      239.052200
Th      Th+4    0.000000        Th      232.038100
Im      Im      0.000000        Im      40.000000
Rx      Rx      0.000000        Rx      30.000000
Ip      Ip      0.000000        Ip      40.000000
#
SOLUTION_SPECIES
#
#1  H+
 H+ = H+
        logk    0.000000
        delta_h 0.000000        kcal

#2  e-
 e- = e-
        logk    0.000000
        delta_h 0.000000        kcal

#3  H2O
 H2O = H2O
        logk    0.000000
        delta_h 0.000000        kcal

#4  Ca+2
 Ca+2 = Ca+2
        logk    0.000000
        delta_h 0.000000        kcal
        -gamma  5.000000        0.165000

#5  Mg+2
 Mg+2 = Mg+2
        logk    0.000000
        delta_h 0.000000        kcal
        -gamma  5.500000        0.200000

#6  Na+
 Na+ = Na+
        logk    0.000000
        delta_h 0.000000        kcal
        -gamma  4.000000        0.075000

#7  K+
```

```
K+ = K+
        logk     0.000000
        delta_h  0.000000        kcal
        -gamma 3.500000          0.015000

#8  Fe+2
Fe+2 = Fe+2
        logk     0.000000
        delta_h  0.000000        kcal

#9  Mn+2
Mn+2 = Mn+2
        logk     0.000000
        delta_h  0.000000        kcal

#10  Al+3
Al+3 = Al+3
        logk     0.000000
        delta_h  0.000000        kcal

#11  Ba+2
Ba+2 = Ba+2
        logk     0.000000
        delta_h  0.000000        kcal

#12  Sr+2
Sr+2 = Sr+2
        logk     0.000000
        delta_h  0.000000        kcal

#13  H4SiO4
H4SiO4 = H4SiO4
        logk     0.000000
        delta_h  0.000000        kcal

#14  Cl-
Cl- = Cl-
        logk     0.000000
        delta_h  0.000000        kcal
        -gamma 3.500000          0.015000

#15  CO3-2
CO3-2 = CO3-2
        logk     0.000000
        delta_h  0.000000        kcal
        -gamma 5.400000          0.000000

#16  SO4-2
SO4-2 = SO4-2
        logk     0.000000
        delta_h  0.000000        kcal
        -gamma 5.000000          -0.040000

#17  NO3-
NO3- = NO3-
        logk     0.000000
        delta_h  0.000000        kcal

#18  H3BO3
H3BO3 = H3BO3
```

```
            logk      0.000000
            delta_h   0.000000          kcal

#19  PO4-3
 PO4-3 = PO4-3
            logk      0.000000
            delta_h   0.000000          kcal

#20  F-
 F- = F-
            logk      0.000000
            delta_h   0.000000          kcal

#21  Li+
 Li+ = Li+
            logk      0.000000
            delta_h   0.000000          kcal

#22  Br-
 Br- = Br-
            logk      0.000000
            delta_h   0.000000          kcal

#23  U+4
 U+4 = U+4
            logk      0.000000
            delta_h   0.000000          kcal

#24  Np+4
 Np+4 = Np+4
            logk      0.000000
            delta_h   0.000000          kcal

#25  Cs+
 Cs+ = Cs+
            logk      0.000000
            delta_h   0.000000          kcal

#26  Cx+2
 Cx+2 = Cx+2
            logk      0.000000
            delta_h   0.000000          kcal

#27  Pu+4
 Pu+4 = Pu+4
            logk      0.000000
            delta_h   0.000000          kcal

#28  Th+4
 Th+4 = Th+4
            logk      0.000000
            delta_h   0.000000          kcal

#29  Rx
 Rx = Rx
            logk      0.000000
            delta_h   0.000000          kcal

#29  Im
 Im = Im
```

```
        logk     0.000000
        delta_h  0.000000       kcal

#30  Ip
 Ip = Ip
        logk     0.000000
        delta_h  0.000000       kcal

CO3-2 = CO3-2
    log_k       0.000
    -gamma   5.4000   0.0000

#31  OH-
 H2O = OH- +  H+
        logk     -13.998000
        delta_h  13.345000      kcal

#32  O2_aq
2 H2O = O2 + 4 H+ + 4 e-
        logk     -86.080002
        delta_h  134.789993     kcal

#33  H2_aq
2 e- + 2 H+ = H2
        logk     -3.150000
        delta_h  -1.759000      kcal

#34  HCO3-
 CO3-2 + H+ = HCO3-
        logk     10.330000
        delta_h  -3.604000      kcal
        -gamma 5.400000         0.000000
        -a_e     -6.498000      0.023790        2902.389893

#35  H2CO3
 CO3-2 + 2 H+ = O2C +  H2O
        logk     16.681000
        delta_h  -5.847000      kcal
        -a_e     -21.341499     0.056576        6307.100098

#36  CH4
 CO3-2 + 8 e- + 10 H+ = CH4 + 3 H2O
        logk     41.070999
        delta_h  -61.039001     kcal

#37  UOH+3
 U+4 + H2O = UOH+3 +  H+
        logk     -0.600000
        delta_h  11.810000      kcal

#38  U(OH)2+2
 U+4 + 2 H2O = U(OH)2+2 + 2 H+
        logk     -2.500000
        delta_h  17.780001      kcal

#39  U(OH)3+
 U+4 + 3 H2O = U(OH)3+ + 3 H+
        logk     -5.500000
        delta_h  22.639999      kcal
```

```
#40  HSO4-
 SO4-2 + H+ = HSO4-
        logk      1.987000
        delta_h   4.910000        kcal
        -a_e      -5.350500       0.018341        557.246094

#41  S-2
 SO4-2 + 8 e- + 8 H+ = S-2 + 4 H2O
        logk      20.735001
        delta_h   -28.040001      kcal

#42  HS-
 SO4-2 + 8 e- + 9 H+ = HS- + 4 H2O
        logk      33.652000
        delta_h   -40.139999      kcal

#43  H2S
 SO4-2 + 8 e- + 10 H+ = H2S + 4 H2O
        logk      40.644001
        delta_h   -65.440002      kcal

#44  U(OH)4
 U+4 + 4 H2O = U(OH)4 + 4 H+
        logk      -10.000000
        delta_h   24.770000       kcal

# do not exist
#45  U(OH)5-
# U+4 + 5 H2O = U(OH)5- + 5 H+
#       logk      -19.000000
#       delta_h   27.580000       kcal

#46  U2OH2+6
2 U+4 + 2 H2O = H2O2U2+6 + 2 H+
        logk      -1.000000
        delta_h   0.000000        kcal

#47  UCO35-6
 U+4 + 5 CO3-2 = O15C5U-6
        logk      36.500000
        delta_h   0.000000        kcal

#48  NO2-
 NO3- + 2 e- + 2 H+ = NO2- +  H2O
        logk      28.570000
        delta_h   -43.759998      kcal

#49  N2
2 NO3- + 10 e- + 12 H+ = N2 + 6 H2O
        logk      207.080002
        delta_h   -312.130005     kcal

#50  NH3
 NO3- + 8 e- + 9 H+ = NH3 + 3 H2O
        logk      109.830002
        delta_h   -174.580002     kcal

#51  NH4+
 NO3- + 8 e- + 10 H+ = NH4+ + 3 H2O
        logk      119.077003
```

```
          delta_h  -187.054993      kcal

#52  NH4SO4-
 NO3- + SO4-2 + 8 e- + 10 H+ = NH4SO4- + 3 H2O
          logk      120.190002
          delta_h  -187.054993      kcal

#53  UOH3CO3-
 U+4 + CO3-2 + 3 H2O = U(OH)3CO3- + 3 H+
          logk      0.000000
          delta_h  0.000000         kcal

#54  UO2OH
 U+4 + 3 H2O = UO2OH + 5 H+ +  e-
          logk      -17.799999
          delta_h  31.600000        kcal

#55  UO2CO3-
 U+4 + CO3-2 + 2 H2O = UO2CO3- + 4 H+ +  e-
          logk      -2.800000
          delta_h  28.840000        kcal

#56  UO2CO32-3
 U+4 + 2 CO3-2 + 2 H2O = UO2(CO3)2-3 + 4 H+ +  e-
          logk      2.200000
          delta_h  31.600000        kcal

#57  H2BO3-
 H3BO3 = H2BO3- +  H+
          logk      -9.240000
          delta_h  3.224000         kcal

#58  BFOH3-
 F- + H3BO3 = BF(OH)3-
          logk      -0.400000
          delta_h  1.850000         kcal

#59  BF2OH2-
2 F- + H3BO3 + H+ = BF2(OH)2- +  H2O
          logk      7.628000
          delta_h  1.635000         kcal

#60  BF3OH-
3 F- + H3BO3 + 2 H+ = BF3OH- + 2 H2O
          logk      13.666000
          delta_h  -1.580000        kcal

#61  BF4-
4 F- + H3BO3 + 3 H+ = BF4- + 3 H2O
          logk      20.274000
          delta_h  -1.795000        kcal

#62  UO2CO33-5
 U+4 + 3 CO3-2 + 2 H2O = UO2(CO3)3-5 + 4 H+ +  e-
          logk      7.500000
          delta_h  31.600000        kcal

#63  UO2+
 U+4 + 2 H2O = UO2+ + 4 H+ +  e-
          logk      -7.800000
```

```
            delta_h  31.600000          kcal

#64  UO2+2
 U+4 + 2 H2O = UO2+2 + 4 H+ + 2 e-
            logk       -9.100000
            delta_h  34.480000          kcal

#65  HPO4-2
 PO4-3 + H+ = HPO4-2
            logk       12.346000
            delta_h  -3.530000          kcal

#66  H2PO4-
 PO4-3 + 2 H+ = H2PO4-
            logk       19.552999
            delta_h  -4.520000          kcal

#67  UO2OH+
 U+4 + 3 H2O = UO2OH+ + 5 H+ + 2 e-
            logk       -14.900000
            delta_h  -23.520000         kcal

#68  UO2OH2
 U+4 + 4 H2O = UO2(OH)2 + 6 H+ + 2 e-
            logk       -21.000000
            delta_h  -17.049999         kcal

#69  HF
 F- + H+ = HF
            logk       3.169000
            delta_h  3.460000           kcal

#70  HF2-
2 F- + H+ = HF2-
            logk       3.749000
            delta_h  4.550000           kcal

#71  UO2OH3-
 U+4 + 5 H2O = UO2(OH)3- + 7 H+ + 2 e-
            logk       -30.100000
            delta_h  -34.480000         kcal

#72  UO22OH2+2
2 U+4 + 6 H2O = (UO2)2(OH)2+2 + 10 H+ + 4 e-
            logk       -23.799999
            delta_h  -24.389999         kcal

#73  UO23OH5+
3 U+4 + 11 H2O = H5O11U3+1 + 17 H+ + 6 e-
            logk       -42.900002
            delta_h  -9.860000          kcal

#74  UO2HPO4
 U+4 + PO4-3 + 2 H2O = UO2HPO4 + 3 H+ + 2 e-
            logk       11.640000
            delta_h  0.000000           kcal

#75  CaOH+
 Ca+2 + H2O = CaOH+ +  H+
            logk       -12.598000
```

```
        delta_h  14.535000        kcal

#76  CaCO3
 CO3-2 + Ca+2 = CaCO3
        logk     3.153000
        delta_h  4.023000         kcal
        -a_e     -27.393000       0.056170        4114.000000

#77  CaHCO3+
 CO3-2 + Ca+2 + H+ = CaHCO3+
        logk     11.345000
        delta_h  1.806000         kcal
        -a_e     -9.448000        0.037090        2902.389893

#78  CaSO4
 SO4-2 + Ca+2 = CaSO4
        logk     2.309000
        delta_h  1.470000         kcal

#79  CaPO4-
 PO4-3 + Ca+2 = CaPO4-
        logk     6.459000
        delta_h  3.100000         kcal

#80  CaHPO4
 PO4-3 + Ca+2 + H+ = CaHPO4
        logk     15.085000
        delta_h  -0.230000        kcal

#81  CaH2PO4+
 PO4-3 + Ca+2 + 2 H+ = CaH2PO4+
        logk     20.961000
        delta_h  -1.120000        kcal

#82  CaF+
 F- + Ca+2 = CaF+
        logk     0.940000
        delta_h  3.798000         kcal

#83  UO2H2PO4-2
 U+4 + 2 PO4-3 + 2 H2O = H2O10P2U-2 + 2 H+ + 2 e-
        logk     33.759998
        delta_h  0.000000         kcal

#84  UO2H2PO4+
 U+4 + PO4-3 + 2 H2O = UO2H2PO4+ + 2 H+ + 2 e-
        logk     13.440000
        delta_h  0.000000         kcal

#85  MgOH+
 Mg+2 + H2O = MgOH+ +  H+
        logk     -11.794000
        delta_h  15.419000        kcal

#86  MgCO3
 CO3-2 + Mg+2 = MgCO3
        logk     2.980000
        delta_h  2.713000         kcal
        -a_e     0.991000         0.006670
```

```
#87  MgHCO3+
 CO3-2 + Mg+2 + H+ = MgHCO3+
         logk      11.396000
         delta_h   -2.527000       kcal
         -a_e      -4.179000       0.012730       2902.389893    0.000023

#88  MgSO4
 SO4-2 + Mg+2 = MgSO4
         logk      2.250000
         delta_h   1.400000        kcal

#89  MgPO4-
 PO4-3 + Mg+2 = MgPO4-
         logk      6.589000
         delta_h   3.100000        kcal

#90  MgHPO4
 PO4-3 + Mg+2 + H+ = MgHPO4
         logk      15.216000
         delta_h   -0.230000       kcal

#91  MgH2PO4+
 PO4-3 + Mg+2 + 2 H+ = MgH2PO4+
         logk      21.066000
         delta_h   -1.120000       kcal

#92  MgF+
 F- + Mg+2 = MgF+
         logk      1.820000
         delta_h   4.674000        kcal

#93  UO2CO3
 U+4 + CO3-2 + 2 H2O = UO2CO3 + 4 H+ + 2 e-
         logk      1.000000
         delta_h   -37.240002      kcal

#94  UO2CO32-2
 U+4 + 2 CO3-2 + 2 H2O = O8C2U-2 + 4 H+ + 2 e-
         logk      7.600000
         delta_h   -30.879999      kcal

#95  NaCO3-
 CO3-2 + Na+ = NaCO3-
         logk      1.268000
         delta_h   8.911000        kcal

#96  NaHCO3
 CO3-2 + Na+ + H+ = NaHCO3
         logk      10.080000
         delta_h   -3.604000       kcal

#97  NaSO4-
 SO4-2 + Na+ = NaSO4-
         logk      0.700000
         delta_h   1.120000        kcal

#98  NaHPO4-
 PO4-3 + Na+ + H+ = NaHPO4-
         logk      12.636000
         delta_h   -3.530000       kcal
```

```
#99  UO2CO33-4
 U+4 + 3 CO3-2 + 2 H2O = O11C3U-4 + 4 H+ + 2 e-
        logk      14.700000
        delta_h   -44.180000      kcal

#100  KSO4-
 SO4-2 + K+ = KSO4-
        logk      0.850000
        delta_h   2.250000        kcal

#101  KHPO4-
 PO4-3 + K+ + H+ = KHPO4-
        logk      12.636000
        delta_h   -3.530000       kcal

#102  UO23C6-6
3 U+4 + 6 CO3-2 + 6 H2O = O24C6U3-6 + 12 H+ + 6 e-
        logk      32.799999
        delta_h   -103.440002     kcal

#103  UO22OH3C-
2 U+4 + CO3-2 + 7 H2O = H3O10CU2-1 + 11 H+ + 4 e-
        logk      -19.200001
        delta_h   -68.959999      kcal

#104  UO23OH3C+
3 U+4 + CO3-2 + 9 H2O = H3O12CU3+1 + 15 H+ + 6 e-
        logk      -26.299999
        delta_h   -103.440002     kcal

#105  FeOH+
 Fe+2 + H2O = FeOH+ +  H+
        logk      -9.500000
        delta_h   13.200000       kcal

#106  FeOH2
 Fe+2 + 2 H2O = H2O2Fe + 2 H+
        logk      -20.570000
        delta_h   28.565001       kcal

#107  FeOH3-
 Fe+2 + 3 H2O = H3O3Fe-1 + 3 H+
        logk      -31.000000
        delta_h   30.299999       kcal

#108  FeSO4
 SO4-2 + Fe+2 = FeSO4
        logk      2.250000
        delta_h   3.230000        kcal

#109  Fe(HS)2
2 SO4-2 + Fe+2 + 16 e- + 18 H+ = Fe(HS)2 + 8 H2O
        logk      76.250000
        delta_h   -120.279999     kcal

#110  Fe(HS)3-
3 SO4-2 + Fe+2 + 24 e- + 27 H+ = Fe(HS)3- + 12 H2O
        logk      111.936996
        delta_h   -180.419998     kcal
```

```
#111  FeHPO4
 PO4-3 + Fe+2 + H+ = FeHPO4
            logk       15.946000
            delta_h  -3.530000           kcal

#112  FeH2PO4+
 PO4-3 + Fe+2 + 2 H+ = FeH2PO4+
            logk       22.253000
            delta_h  -4.520000           kcal

#113  UO2H2PO42
 U+4 + 2 PO4-3 + 2 H2O = H4O10P2U + 2 e-
            logk       35.430000
            delta_h  0.000000            kcal

#114  UO2H2PO43-
 U+4 + 3 PO4-3 + 2 H2O + 2 H+ = H6O14P3U-1 + 2 e-
            logk       56.680000
            delta_h  0.000000            kcal

#115  Fe+3
 Fe+2 = Fe+3 +  e-
            logk       -13.032000
            delta_h  10.000000           kcal

#116  FeOH+2
 Fe+2 + H2O = FeOH+2 +  H+ +  e-
            logk       -15.220000
            delta_h  20.400000           kcal

#117  FeOH2+
 Fe+2 + 2 H2O = H2O2Fe+1 + 2 H+ +  e-
            logk       -18.700001
            delta_h  10.000000           kcal

#118  FeOH3
 Fe+2 + 3 H2O = H3O3Fe + 3 H+ +  e-
            logk       -26.629999
            delta_h  10.000000           kcal

#119  FeOH4-
 Fe+2 + 4 H2O = H4O4Fe-1 + 4 H+ +  e-
            logk       -34.630001
            delta_h  10.000000           kcal

#120  Fe2OH2+4
2 Fe+2 + 2 H2O = H2O2Fe2+4 + 2 H+ + 2 e-
            logk       -29.010000
            delta_h  33.500000           kcal

#121  Fe3OH4+5
3 Fe+2 + 4 H2O = H4O4Fe3+5 + 4 H+ + 3 e-
            logk       -45.400002
            delta_h  44.299999           kcal

#122  FeCl+2
 Cl- + Fe+2 = FeCl+2 +  e-
            logk       -11.550000
            delta_h  15.600000           kcal
```

```
#123  FeCl2+
2 Cl- + Fe+2 = FeCl2+ +  e-
        logk      -10.900000
        delta_h  10.000000         kcal

#124  FeCl3
3 Cl- + Fe+2 = FeCl3 +  e-
        logk      -11.900000
        delta_h  10.000000         kcal

#125  FeSO4+
 SO4-2 + Fe+2 = FeSO4+ +  e-
        logk      -9.110000
        delta_h  13.910000         kcal

#126  FeSO42-
2 SO4-2 + Fe+2 = O8FeS2-1 +  e-
        logk      -7.610000
        delta_h  14.600000         kcal

#127  FeHPO4+
 PO4-3 + Fe+2 + H+ = FeHPO4+ +  e-
        logk      4.740000
        delta_h  12.230000         kcal

#128  FeH2P+2
 PO4-3 + Fe+2 + 2 H+ = H2O4FeP+2 +  e-
        logk      11.950000
        delta_h  5.480000          kcal

#129  FeF+2
 F- + Fe+2 = FeF+2 +  e-
        logk      -6.800000
        delta_h  12.700000         kcal

#130  FeF2+
2 F- + Fe+2 = FeF2+ +  e-
        logk      -2.200000
        delta_h  14.700000         kcal

#131  FeF3
3 F- + Fe+2 = FeF3 +  e-
        logk      0.970000
        delta_h  15.400000         kcal

#132  UHPO4+2
 U+4 + PO4-3 + H+ = UHPO4+2
        logk      24.400000
        delta_h  0.000000          kcal

#133  UHPO42
 U+4 + 2 PO4-3 + 2 H+ = H2O8P2U
        logk      46.689999
        delta_h  0.000000          kcal

#134  UHPO43-2
 U+4 + 3 PO4-3 + 3 H+ = H3O12P3U-2
        logk      67.739998
        delta_h  0.000000          kcal
```

```
#135  MnOH+
 Mn+2 + H2O = MnOH+ +  H+
            logk       -10.590000
            delta_h  14.400000          kcal

#136  MnOH3-
 Mn+2 + 3 H2O = H3O3Mn-1 + 3 H+
            logk       -34.799999
            delta_h  0.000000           kcal

#137  MnCl+
 Cl- + Mn+2 = MnCl+
            logk       0.607000
            delta_h  0.000000           kcal

#138  MnCl2
2 Cl- + Mn+2 = MnCl2
            logk       0.041000
            delta_h  0.000000           kcal

#139  MnCl3-
3 Cl- + Mn+2 = MnCl3-
            logk       -0.305000
            delta_h  0.000000           kcal

#140  MnHCO3+
 CO3-2 + Mn+2 + H+ = MnHCO3+
            logk       11.600000
            delta_h  -3.604000          kcal

#141  MnSO4
 SO4-2 + Mn+2 = MnSO4
            logk       2.260000
            delta_h  2.170000           kcal

#142  Mn(NO3)2
2 NO3- + Mn+2 = Mn(NO3)2
            logk       0.600000
            delta_h  -0.396000          kcal

#143  MnF+
 F- + Mn+2 = MnF+
            logk       0.850000
            delta_h  0.000000           kcal

#144  Mn+3
 Mn+2 = Mn+3 +  e-
            logk       -25.507000
            delta_h  25.760000          kcal

#145  MnO4-2
 Mn+2 + 4 H2O = MnO4-2 + 8 H+ + 4 e-
            logk       -118.440002
            delta_h  150.020004         kcal

#146  MnO4-
 Mn+2 + 4 H2O = MnO4- + 8 H+ + 5 e-
            logk       -127.823997
            delta_h  176.619995         kcal
```

```
#147  Pu(CO3)5
 Pu+4 + 5 CO3-2 = Pu(CO3)5-6
            logk       44.500000
            delta_h  0.000000            kcal

#148  PuO2CO3-
 Pu+4 + CO3-2 + 2 H2O = PuO2CO3- + 4 H+ +  e-
            logk       -9.000000
            delta_h  0.000000            kcal

#149  UH2PO42+2
 U+4 + 2 PO4-3 + 4 H+ = UH2(PO4)2+2 + H2
            logk       47.105999
            delta_h  0.000000            kcal

#150  AlOH+2
 Al+3 + H2O = AlOH+2 +  H+
            logk       -4.990000
            delta_h  11.900000          kcal

#151  Al(OH)2+
 Al+3 + 2 H2O = Al(OH)2+ + 2 H+
            logk       -10.100000
            delta_h  0.000000            kcal

#152  Al(OH)3
 Al+3 + 3 H2O = Al(OH)3 + 3 H+
            logk       -16.000000
            delta_h  0.000000            kcal

#153  Al(OH)4-
 Al+3 + 4 H2O = Al(OH)4- + 4 H+
            logk       -23.000000
            delta_h  44.060001          kcal

#154  AlSO4+
 SO4-2 + Al+3 = AlSO4+
            logk       3.020000
            delta_h  2.150000            kcal

#155  Al(SO4)2-
2 SO4-2 + Al+3 = Al(SO4)2-
            logk       4.920000
            delta_h  2.840000            kcal

#156  AlF+2
 F- + Al+3 = AlF+2
            logk       7.010000
            delta_h  0.000000            kcal

#157  AlF2+
2 F- + Al+3 = AlF2+
            logk       12.750000
            delta_h  20.000000          kcal

#158  AlF3
3 F- + Al+3 = AlF3
            logk       17.020000
            delta_h  2.500000            kcal
```

```
#159  AlF4-
4 F- + Al+3 = AlF4-
        logk      19.719999
        delta_h   0.000000        kcal

#163  BaOH+
 Ba+2 + H2O = BaOH+ +  H+
        logk      -13.358000
        delta_h   15.095000       kcal

#165  SrOH+
 Sr+2 + H2O = SrOH+ +  H+
        logk      -13.178000
        delta_h   14.495000       kcal

#170  H3SiO4-
 H4SiO4 = H3SiO4- +  H+
        logk      -9.929000
        delta_h   8.936000        kcal
        -a_e      6.368000        -0.016346       -3405.899902

#171  H2SiO4-2
 H4SiO4 = H2SiO4-2 + 2 H+
        logk      -21.617001
        delta_h   29.716999       kcal
        -a_e      39.478001       -0.065927       -12355.099609

#172  SiF6-2
6 F- + H4SiO4 + 4 H+ = SiF6-2 + 4 H2O
        logk      30.180000
        delta_h   -16.260000      kcal

#181  LiSO4-
 Li+ + SO4-2 = LiSO4-
        logk      0.640000
        delta_h   0.000000        kcal

#182  CsOH
 Cs+ + H2O = CsOH +  H+
        logk      -14.000000
        delta_h   0.000000        kcal

#183  CsCO3-1
 Cs+ + CO3-2 = CsCO3-
        logk      0.000000
        delta_h   0.000000        kcal

#184  CsSO4-1
 Cs+ + SO4-2 = CsSO4-
        logk      0.300000
        delta_h   0.000000        kcal

#185  CsCl
 Cs+ + Cl- = CsCl
        logk      0.500000
        delta_h   0.000000        kcal

#186  PuCO33
 Pu+4 + 3 CO3-2 + e- = O9C3Pu-3
```

```
        logk       31.500000
        delta_h  0.000000         kcal

#187  PuSO4+
 Pu+4 + SO4-2 + e- = PuSO4+
        logk       19.900000
        delta_h  -9.850000        kcal

#188  Pu(HSO4)
 Pu+4 + 2 SO4-2 + e- + 2 H+ = H2O8S2Pu+1
        logk       25.700001
        delta_h  0.000000         kcal

#189  Pu(SO4)2
 Pu+4 + 2 SO4-2 + e- = Pu(SO4)2-
        logk       22.000000
        delta_h  0.000000         kcal

#190  PuCl+2
 Pu+4 + Cl- + e- = PuCl+2
        logk       18.000000
        delta_h  0.000000         kcal

#191  PuCl2+
 Pu+4 + 2 Cl- + e- = PuCl2+
        logk       12.000000
        delta_h  0.000000         kcal

#192  PuCO3+2
 Pu+4 + CO3-2 = PuCO3+2
        logk       19.100000
        delta_h  0.000000         kcal

#193  Pu(CO3)2
 Pu+4 + 2 CO3-2 = Pu(CO3)2
        logk       33.099998
        delta_h  0.000000         kcal

#194  Pu(CO3)3
 Pu+4 + 3 CO3-2 = Pu(CO3)3-2
        logk       42.299999
        delta_h  0.000000         kcal

#195  Pu(CO3)4
 Pu+4 + 4 CO3-2 = Pu(CO3)4-4
        logk       45.000000
        delta_h  0.000000         kcal

#196  BaSO4
 SO4-2 + Ba+2 = BaSO4
        logk       10.920000
        delta_h  0.000000         kcal

#197  PuO2+2
 Pu+4 + 2 H2O = PuO2+2 + 4 H+ + 2 e-
        logk       -34.900002
        delta_h  68.339996        kcal

#198  PuO2+
 Pu+4 + 2 H2O = PuO2+ + 4 H+ +  e-
```

```
            logk       -18.600000
            delta_h  46.240002          kcal

#199  Pu+3
 Pu+4 + e- = Pu+3
            logk       17.000000
            delta_h  -13.300000          kcal

#200  PuO2OH+
 Pu+4 + 3 H2O = PuO2OH+ + 5 H+ + 2 e-
            logk       -40.200001
            delta_h  65.800003          kcal

#201  PuO2OH2
 Pu+4 + 4 H2O = H2O4Pu + 6 H+ + 2 e-
            logk       -47.400002
            delta_h  0.000000             kcal

#202  PuO2OH3-
 Pu+4 + 5 H2O = H3O5Pu-1 + 7 H+ + 2 e-
            logk       -59.400002
            delta_h  0.000000             kcal

#203  PuO22OH2
2 Pu+4 + 6 H2O = H2O6Pu2+2 + 10 H+ + 4 e-
            logk       -75.199997
            delta_h  123.870003          kcal

#204  PuO23OH5
3 Pu+4 + 11 H2O = H5O11Pu3+1 + 17 H+ + 6 e-
            logk       -116.699997
            delta_h  171.600006          kcal

#205  PuO2CO3
 Pu+4 + CO3-2 + 2 H2O = PuO2CO3 + 4 H+ + 2 e-
            logk       -25.700001
            delta_h  0.000000             kcal

#206  PuO2CO32
 Pu+4 + 2 CO3-2 + 2 H2O = O8C2Pu-2 + 4 H+ + 2 e-
            logk       -20.100000
            delta_h  73.800003          kcal

#207  PuO2CO33
 Pu+4 + 3 CO3-2 + 2 H2O = O11C3Pu-4 + 4 H+ + 2 e-
            logk       -17.500000
            delta_h  0.000000             kcal

#208  PuO2Cl+
 Pu+4 + Cl- + 2 H2O = PuO2Cl+ + 4 H+ + 2 e-
            logk       -34.799999
            delta_h  0.000000             kcal

#209  PuO2Cl2
 Pu+4 + 2 Cl- + 2 H2O = PuO2Cl2 + 4 H+ + 2 e-
            logk       -35.400002
            delta_h  0.000000             kcal

#210  PuO2SO4
 Pu+4 + SO4-2 + 2 H2O = PuO2SO4 + 4 H+ + 2 e-
```

```
	logk	-31.700001
	delta_h	72.199997	kcal

#211  PuO2OH
 Pu+4 + 3 H2O = PuO2OH + 5 H+ +  e-
	logk	-28.299999
	delta_h	49.400002	kcal

#212  PuO2(OH)
 Pu+4 + 4 H2O = H2O4Pu-1 + 6 H+ +  e-
	logk	-37.599998
	delta_h	0.000000	kcal

#213  PuO2Cl
 Pu+4 + Cl- + 2 H2O = PuO2Cl + 4 H+ +  e-
	logk	-99.000000
	delta_h	0.000000	kcal

#214  Pu(OH)+3
 Pu+4 + H2O = Pu(OH)+3 +  H+
	logk	-0.900000
	delta_h	11.500000	kcal

#215  Pu(OH)2+
 Pu+4 + 2 H2O = Pu(OH)2+2 + 2 H+
	logk	-2.200000
	delta_h	17.799999	kcal

#216  Pu(OH)3+
 Pu+4 + 3 H2O = Pu(OH)3+ + 3 H+
	logk	-5.100000
	delta_h	23.100000	kcal

#217  Pu(OH)4
 Pu+4 + 4 H2O = Pu(OH)4 + 4 H+
	logk	-10.540000
	delta_h	26.100000	kcal

#218  Pu(OH)5-
 Pu+4 + 5 H2O = Pu(OH)5- + 5 H+
	logk	-99.000000
	delta_h	0.000000	kcal

#219  Pu2(OH)2
2 Pu+4 + 2 H2O = Pu2(OH)2+6 + 2 H+
	logk	-1.000000
	delta_h	0.000000	kcal

#220  Pu2(OH)3
2 Pu+4 + 3 H2O = Pu2(OH)3+5 + 3 H+
	logk	-2.000000
	delta_h	0.000000	kcal

#221  Pu2(OH)4
2 Pu+4 + 4 H2O = Pu2(OH)4+4 + 4 H+
	logk	-3.000000
	delta_h	0.000000	kcal

#222  Pu2(OH)5
2 Pu+4 + 5 H2O = Pu2(OH)5+3 + 5 H+
```

```
    logk      -7.000000
    delta_h  0.000000        kcal

#223  PuSO4+2
 Pu+4 + SO4-2 = PuSO4+2
    logk      5.720000
    delta_h  3.000000        kcal

#224  Pu(SO4)2
 Pu+4 + 2 SO4-2 = Pu(SO4)2
    logk      10.250000
    delta_h  13.310000       kcal

#225  Pu(SO4)3
 Pu+4 + 3 SO4-2 = Pu(SO4)3-2
    logk      11.500000
    delta_h  0.000000        kcal

#226  PuCl+3
 Pu+4 + Cl- = PuCl+3
    logk      1.670000
    delta_h  0.000000        kcal

#227  PuCl2+2
 Pu+4 + 2 Cl- = PuCl2+2
    logk      0.550000
    delta_h  0.000000        kcal

#228  PuCl3+
 Pu+4 + 3 Cl- = PuCl3+
    logk      -0.390000
    delta_h  0.000000        kcal

#229  Pu(OH)+2
 Pu+4 + H2O + e- = Pu(OH)+2 +  H+
    logk      10.000000
    delta_h  118.900002      kcal

#230  Pu(OH)2+
 Pu+4 + 2 H2O + e- = Pu(OH)2+ + 2 H+
    logk      0.000000
    delta_h  0.000000        kcal

#231  Pu(OH)3
 Pu+4 + 3 H2O + e- = Pu(OH)3 + 3 H+
    logk      -9.500000
    delta_h  0.000000        kcal

#232  Pu(OH)4-
 Pu+4 + 4 H2O + e- = Pu(OH)4- + 4 H+
    logk      -20.000000
    delta_h  0.000000        kcal

#233  Pu3(OH)5
3 Pu+4 + 5 H2O + 3 e- = Pu3(OH)5+4 + 5 H+
    logk      18.000000
    delta_h  0.000000        kcal

#234  Pu2(OH)2
2 Pu+4 + 2 H2O + 2 e- = Pu2(OH)2+4 + 2 H+
```

```
            logk       20.000000
            delta_h  0.000000            kcal

#235  PuCO3+
 Pu+4 + CO3-2 + e- = PuCO3+
            logk       23.500000
            delta_h  0.000000            kcal

#236  Pu(CO3)2
 Pu+4 + 2 CO3-2 + e- = Pu(CO3)2-
            logk       28.000000
            delta_h  0.000000            kcal

#237 ThOH+3
 Th+4 + H2O = ThOH+3 + H+
            logk       -3.300000
            delta_h  0.000000            kcal

#238 Th(OH)2+2
 Th+4 + 2 H2O = Th(OH)2+2 + 2 H+
            logk       -8.600000
            delta_h  0.000000            kcal

#239 Th(OH)3+
 Th+4 + 3 H2O = Th(OH)3+ + 3 H+
            logk       -14.200000
            delta_h  0.000000            kcal

#240 Th(OH)4
 Th+4 + 4 H2O = Th(OH)4 + 4 H+
            logk       -19.400000
            delta_h  0.000000            kcal

#241 ThH2PO4+3
 Th+4 + H2PO4- =  ThH2PO4+3
logk       8.767925
            delta_h  0.000000            kcal

#242 Th(H2PO4)2+2
 Th+4 + 2 H2PO4- =  Th(H2PO4)2+2
logk       14.366643
            delta_h  0.000000            kcal

#242 Th(H2PO4)3+
 Th+4 + 3 H2PO4- =  Th(H2PO4)3+
logk       21.764480
            delta_h  0.000000            kcal

#242 Th(H2PO4)4
 Th+4 + 4 H2PO4- =  Th(H2PO4)4
logk       28.810081
            delta_h  0.000000            kcal

#UCl+3            586
            U+4 + Cl- = UCl+3
            log_k                  1.72
            delta_h  -4.54 kcal

#USO4+2           587
            U+4 + SO4-2 = USO4+2
```

```
            log_k                 6.58
            delta_h  1.9 kcal

#U(SO4)2          588
            U+4 + 2SO4-2 = U(SO4)2
            log_k                 10.5
            delta_h  7.8 kcal

#U(CO3)4-4         589
            U+4 + 4CO3-2 = U(CO3)4-4
            log_k                 32.9

#U(CO3)5-6         590
            U+4 + 5CO3-2 = U(CO3)5-6
            log_k                 34.0
            delta_h  20.0 kcal

#300  Im-
 Im + e- = Im-
            logk      20.000000
            delta_h  0.000000           kcal

#301  Ip+
 Ip = Ip+ + e-
            logk      20.000000
            delta_h  0.000000           kcal

#CO2     could be used instead of H2CO3
CO3-2 + 2 H+ = CO2 + H2O
      log_k         16.681
      delta_h -5.738  kcal
      -analytic      464.1965      0.09344813  -26986.16    -165.75951     2248628.9

#238  Rx(-)
 Rx + e- = Rx-
            logk      5.000000
            delta_h  0.000000           kcal

#239  Rx(+)
 Rx = Rx+ +  e-
            logk      -5.000000
            delta_h  0.000000           kcal

#
PHASES
#

UO2(am)
            UO2 + 2.0 H2O = U+4 + 4.0 OH-
            log_k                 -54.5

CO2(g)
            CO2 = CO2
            log_k                 -1.468
            delta_h  -4.776 kcal
            -analytical          108.3865     0.01985076   -6919.53     -40.45154     669365.0

Calcite
O3CaC = 1.00 CO3-2 + 1.00 Ca+2
            logk      0.470000
```

```
            delta_h  0.580000         kcal
            -a_e     13.543000        -0.040100         -3000.000000

Dolomite
O6CaMgC2 = 1.00 Ca+2 + 1.00 Mg+2 + 2.00 CO3-2
            logk     7.020000
            delta_h  0.290000         kcal

Siderite
O3FeC = 1.00 Fe+2 + 1.00 CO3-2
            logk     0.550000
            delta_h  0.328000         kcal

Rhodochr
O3MnC = 1.00 Mn+2 + 1.00 CO3-2
            logk     0.410000
            delta_h  0.079000         kcal

Strontit
O3SrC = 1.00 Sr+2 + 1.00 CO3-2
            logk     0.250000
            delta_h  0.690000         kcal

Gypsum
H4O6CaS = 1.00 Ca+2 + 1.00 SO4-2 + 2.00 H2O
            logk     0.602000
            delta_h  0.028000         kcal

Celestit
O4SrS = 1.00 Sr+2 + 1.00 SO4-2
            logk     0.465000
            delta_h  0.470000         kcal

Barite
O4BaS = 1.00 Ba+2 + 1.00 SO4-2
            logk     0.976000
            delta_h  28.000000        kcal

Hydroxap
HO13Ca5P3 +  4.00 H+ = 1.00 H2O + 3.00 HPO4-2 + 5.00 Ca+2
            logk     0.421000
            delta_h  6.155000         kcal

Fluorite
CaF2 = 1.00 Ca+2 + 2.00 F-
            logk     0.960000
            delta_h  71.000000        kcal

Chalcedy
O2Si +  2.00 H2O = 1.00 H4SiO4
            logk     0.523000
            delta_h  615.000000       kcal

Quartz
O2Si +  2.00 H2O = 1.00 H4SiO4
            logk     0.006000
            delta_h  22.000000        kcal

Gibbsite
H3O3Al +  3.00 H+ = 1.00 Al+3 + 3.00 H2O
```

```
            logk     77.000000
            delta_h  2.800000          kcal

#Kaolinit
#H4O3Al2Si2 +  7.00 H2O = 2.00 H+ + 2.00 H4SiO4 + 2.00 Al(OH)4-
#           logk     6.921000
#           delta_h  9.150000          kcal

Sepiolit
H7O11.50Mg2Si3 +  4.50 H2O = 2.00 Mg+2 + 3.00 H4SiO4 + 4.00 OH-
            logk     0.079000
            delta_h  0.532000          kcal

Hematite
O3Fe2 +  6.00 H+ = 2.00 Fe+3 + 3.00 H2O
            logk     0.008000
            delta_h  0.485000          kcal

Goethite
HO2Fe +  3.00 H+ = 1.00 Fe+3 + 2.00 H2O
            logk     0.500000
            delta_h  4.480000          kcal

FeOH3a
H3O3Fe +  3.00 H+ = 1.00 Fe+3 + 3.00 H2O
            logk     0.891000
            delta_h  9.400000          kcal

Pyrite
FeS2 +  2.00 e- + 2.00 H+ = 1.00 Fe+2 + 2.00 HS-
            logk     8.480000
            delta_h  1.300000          kcal

Fes_ppt
FeS +  1.00 H+ = 1.00 Fe+2 + 1.00 HS-
            logk     0.915000
            delta_h  0.000000          kcal

Vivianit
H16O16Fe3P2 = 3.00 Fe+2 + 2.00 PO4-3 + 8.00 H2O
            logk     6.000000
            delta_h  0.000000          kcal

#PCO2
#O2C +  1.00 H2O = 1.00 O2C
#           logk     0.466000
#           delta_h  0.708000          kcal
#
#O2_gaS
#O2_aq = 1.00 O2
#           logk     0.960000
#           delta_h  0.844000          kcal

#H2_gaS
#H2_aq = 1.00 H2
#           logk     0.150000
#           delta_h  0.759000          kcal

UOH4
H4O4U +  4.00 H+ = 1.00 U+4 + 4.00 H2O
```

logk 0.600000
delta_h 0.000000 kcal

Pu(OH)4S
H4O4Pu + 4.00 H+ = 1.00 Pu+4 + 4.00 H2O
logk 0.400000
delta_h -15.400000 kcal

Pu(OH)2C
H2O5CPu + 2.00 H+ = 1.00 Pu+4 + 2.00 H2O + 1.00 CO3-2
logk -25.000000
delta_h 0.000000 kcal

PuO2S
O2Pu + 4.00 H+ = 1.00 Pu+4 + 2.00 H2O
logk -7.400000
delta_h -12.400000 kcal

CaUO4
O4CaU + 4.00 H+ = 1.00 UO2+2 + 2.00 H2O + 1.00 Ca+2
logk 15.000000
delta_h -31.480000 kcal

MgUO4
O4MgU + 4.00 H+ = 1.00 UO2+2 + 2.00 H2O + 1.00 Mg+2
logk 23.400000
delta_h -47.990002 kcal

Na2U2O7
O7Na2U2 + 6.00 H+ = 2.00 UO2+2 + 3.00 H2O + 2.00 Na+
logk 25.000000
delta_h -43.160000 kcal

Na2UO4
O4Na2U + 4.00 H+ = 1.00 UO2+2 + 2.00 H2O + 2.00 Na+
logk 31.559999
delta_h -43.419998 kcal

UO2CO3
O5CU = 1.00 UO2+2 + 1.00 CO3-2
logk 13.800000
delta_h -5.570000 kcal

Fechlori
H24O18Fe5Al2Si3 + 16.00 = 2.00 Al+3 + 5.00 Fe+2 + 3.00 H4SiO4 + 6.00 H2O
logk 45.549999
delta_h -121.209999 kcal

mONtmrl2
H2.01O12.01Mg0.11Na0.12Al2.33Si3.67 + 2.67 H2O + 7.33 H+ = 0.12 Na+ + 0.11 Mg+2 + 2.33 Al+3 + 3.67 H4SiO4
logk 5.700000
delta_h -96.000000 kcal

Epidote
HO13Ca2Al3Si3 + 13.00 H+ = 1.00 H2O + 2.00 Ca+2 + 3.00 Al+3 + 3.00 H4SiO4
logk 0.243000
delta_h 8.360000 kcal

Chlorite

```
H8O18Mg5Al2Si3 + 16.00 H+ = 6.00 H2O + 5.00 Mg+2 + 2.00 Al+3 + 3.00 H4SiO4
        logk      2.030000
        delta_h   55.259998      kcal

Coffinit
O4SiU +  4.00 H+ = 1.00 U+4 + 1.00 H4SiO4
        logk      10.000000
        delta_h   -164.699997    kcal


END
```