



LPNORM: A LINEAR PROGRAMMING NORMATIVE ANALYSIS CODE

PATRICE DE CARITAT,^{1*} JOHN BLOCH,^{2†} and IAN HUTCHEON¹

¹Department of Geology and Geophysics, University of Calgary, 2500 University Drive N.W., Calgary, Alberta, Canada T2N 1N4 and ²Institute of Sedimentary and Petroleum Geology, Geological Survey of Canada, 3303-33rd St. N.W., Calgary, Alberta, Canada T2L 2A7

(Received 27 April 1993; accepted 25 October 1993)

Abstract—The computer code LPNORM implements the mathematical method of linear programming to calculate the mineralogical makeup of mineral mixtures, such as rock, sediment, or soil samples, from their bulk geochemical composition and from the mineralogical (or geochemical) composition of the contained minerals. This method simultaneously solves the set of linear equations governing the distribution of oxides into these minerals, subject to an objective function and a set of basic constraints. LPNORM allows the user to specify what minerals will be considered for normative analysis, what their composition is (in terms of mineral formula or geochemical composition), and whether to maximize mineral abundances, minimize slack variables (oxides that can not be accounted for), or do both at once in the objective function. Independent knowledge about the abundance of one or several of the minerals in the sample can be entered as additional equality or inequality constraints. Trial-and-error approach enables the user to “optimize” the composition of one or a few of the contained minerals. Results of comparative tests, highlighting the efficiency, as well as the shortcomings, of LPNORM are presented.

Key Words: Normative analysis, Norms, Modes, Linear programming, Mineralogy, Geochemistry, Clay minerals.

INTRODUCTION

The goal of normative analysis is to determine the mineralogy of rocks from their bulk chemical composition. A *norm* is a calculated inventory of mineral abundances in a rock, and is accurate when these abundances approach or equal the actual mineral amounts, collectively referred to as the *mode*. Although alternative methods of quantification of a rock's mineralogy, including point-counting, image analysis, X-ray diffraction (XRD) and electron beam techniques, may work adequately under certain conditions, they may present prohibitive disadvantages in terms of time, cost, or efficacy when applied to fine-grained material, such as mud, shale, fine-grained metamorphic rocks, and soils.

Calculating or estimating the mineral composition of rocks from bulk chemical analyses long has been a concern of the disciplines of geochemistry and mineralogy (see Cross and others, 1902). The first undertakings of normative analysis were directed towards igneous rocks, and made use of the CIPW norms (Cross and others, 1902; Kelsey, 1965). This procedure consists of distributing the elements present into a series of hypothetical minerals not necessarily

present in the rock, following a crystallization sequence derived from petrographic observations and constrained thermodynamically. Although widely used for igneous rock classification schemes and petrogenetic studies, CIPW norms have limited application to sedimentary and pedologic material.

The application of normative analysis to sedimentary rocks has been hampered by the occurrence of minerals with variable compositions, particularly clay minerals, as well as organic matter and amorphous material. Combining XRD and chemical data, pioneering studies by Imbrie and Poldervaart (1959), Miesch (1962), and Nicholls (1962) were concerned with the determination of the mineralogy of clay-bearing rocks.

EXISTING NORMATIVE ANALYSIS COMPUTER CODES AND LPNORM

A few computer codes implementing normative analysis specifically for clay-bearing rocks have been published in recent years: for example, SEDNORM (Cohen and Ward, 1991) and PELNORM (Merodio, Spalletti, and Bertone, 1992). There are three main drawbacks to these programs.

The first drawback has to do with mineral compositions. Although these methods allow for some variation in clay mineral compositions, the variability generally is limited to choices between a few

*Present address: Geological Survey of Norway, P.O. Box 3006-Lade, N-7002 Trondheim, Norway.

†Present address: Seacal Modus, 2617 Cutler Ave. N.E., Albuquerque, NM 87106, U.S.A.

embedded compositions. For instance, in SEDNORM, the Ca:Na and/or Mg:Fe ratios of smectite can be set to 0:1, 1:1, or 1:0. These rigid rules are unlike the natural, continuous chemical and structural variability of clays and many other minerals, and, therefore, drastically limit the usefulness of these methods in practical applications.

Secondly, these programs comprise a built-in, fixed list of minerals to be considered for each normative analysis. The only exception to this is for K-bearing minerals. In SEDNORM, the distribution of K between muscovite/illite and feldspar can be set to 0:1, 1:1, or 1:0, essentially leaving the user to decide whether to have muscovite/illite or feldspar in the rock, or to have both, but then with the restriction that these two phases contain equal amounts of K, that is, one-half the K in the rock each. Similarly, PELNORM allows four variations, depending on whether the presence of smectite is or is not "desired", and on whether K₂O is to be distributed in orthoclase only or in orthoclase and illite. The suite of minerals considered in each normative analysis thus basically is fixed, and can not be adapted to particular applications. Therefore, these programs may return a positive normative amount for one or several minerals that are not actually present in the analyzed rock, whereas they will not consider minerals that are not in the built-in list, but that may be present in the samples.

Thirdly, these codes distribute the different analyzed elements in a structured and inflexible order, generally starting with distributing P into apatite, then S into pyrite and/or gypsum, etc. Although this is straightforward and acceptable in situations where there is only one mineral containing a given element, arbitrary choices must be made when an element occurs in two or more minerals in the same rock (e.g. K may occur in K-feldspar, illite, muscovite, and biotite in sedimentary rocks). The methods presented by Wright and Doherty (1970), Pearson (1978), Banks (1979), and Slaughter (1989) overcome the problem of the order in which oxides are allocated to minerals by solving simultaneous equations to derive mineral norms. Unfortunately, computer codes implementing these methods have not been made available publicly.

In this paper, we present a method, and its implementation as the FORTRAN computer code LPNORM, designed for normative analysis of any type of rock (igneous, metamorphic, or sedimentary), or other consolidated or unconsolidated mineral aggregate (bricks, concrete, ceramic, soil, mud, clay fractions, etc.). This procedure differs from available computer software in that it combines three advantageous characteristics. The first notable characteristic of LPNORM is that it makes use of the mathematical technique known as linear programming or linear optimization, which seeks the "best" numerical solution to the set of (linear) equations governing the distribution of oxides into minerals.

This approach was proposed originally by Greenwood (1967) to resolve petrological problems, and is particularly useful in the normative analysis of rocks and other mineral assemblages. The second beneficial characteristic of LPNORM is that it does not rely on any rigid presupposition as to the composition of the mineral phases, but instead requires that each mineral formula be explicitly specified. Finally, the list of minerals considered in each normative analysis by LPNORM is user-specified to best reflect what is known about the actual mineralogical composition of the rock.

Before describing in detail the code LPNORM and the principles behind it, it is important to stress that neither LPNORM, nor any other normative analysis method, should be used as a "black box" with the expectation that an accurate solution will be automatically generated. The accuracy of the result of any normative analysis is dependent upon the amount of information one has about the rock and its contained minerals.

LINEAR PROGRAMMING AND LPNORM

Mineral abundances

LPNORM distributes the analyzed oxides according to a set of (linear) equations, which are solved simultaneously using the mathematical method known as linear programming or linear optimization. This method is described fully in any linear programming textbook (e.g. Hadley, 1962; Gass, 1975; Bunday, 1984), and is, accordingly, only briefly introduced here. A linear programming problem can be posed as follows.

Maximize the objective function z :

$$z = \sum_{j=1}^m X_j, \quad (1)$$

subject to a set of n constraints (linear inequalities) of the general form:

$$b_i \geq \sum_{j=1}^m a_{i,j} X_j, \quad (i = 1, n), \quad (2)$$

and to a set of m basic constraints (linear inequalities):

$$X_j \geq 0, \quad (j = 1, m), \quad (3)$$

where, in the present context, X_j is the abundance (in wt%) of mineral j in the rock, b_i in the analytic proportion (in wt%) of oxide of i in the rock, and $a_{i,j}$ is the weight ratio of oxide of i in mineral j and is calculated according to:

$$a_{i,j} = \frac{MW_i}{MW_j} \times \frac{S_{[i],j}}{S_{[i],i}}, \quad (4)$$

where MW_i and MW_j are the molecular weights of oxide of i and of mineral j , respectively, and $S_{[i],j}$ and $S_{[i],i}$ are the stoichiometric coefficients of element i in mineral j and of element i in oxide of i , respectively.

Equation (1) dictates that the total mineral abundance be maximized. (Note that linear programming also can be used to minimize an objective function.) Ideally, if all elements had been analyzed for with absolute accuracy, and if the presence and exact composition of all phases in the rock were known, z should be equal to 100 wt%; in practice, however, z usually is smaller than 100 wt%. Equation (2) represents greater-than-or-equal-to relationships (' \geq ') accounting for the fact that a proportion of the analyzed oxides potentially may be tied up either in minerals not considered, or in amorphous or organic phases. In general, the constraints can be a mixture of ' \leq ', ' $=$ ' or ' \geq ' variety. Banks (1979) showed ways to incorporate analytical error on the oxide proportion measurements (b_i 's) into linear programming, but these are not implemented currently in LPNORM. Equation (3) represents a set of basic constraints requiring that the abundance of all normative minerals be nonnegative.

In LPNORM, the linear programming problem is modified somewhat from the previously given generalizations. In order to avoid ' \geq ' relationships, or mixtures of inequalities and equalities, all oxide distribution equations [Eq. (2)] are transformed into equalities by introducing slack variables on the right-hand sides. The objective function also is altered to take these slack variables into account. Thus, the linear optimization problem solved by LPNORM is as follows. Maximize the objective function z :

$$z = \sum_{j=1}^m X_j - \sum_{i=1}^n \lambda_i, \quad (5)$$

subject to a set of n constraints of the general form:

$$b_i = \sum_{j=1}^m a_{i,j} X_j + \lambda_i, \quad (i = 1, n), \quad (6)$$

and to a set of $m + n$ basic constraints:

$$X_j \geq 0, \quad (j = 1, m); \quad \lambda_i \geq 0, \quad (i = 1, n), \quad (7)$$

where λ_i is the slack variable for oxide of i , and all other symbols are defined as previously. Thus, in essence, a slack variable, also referred to as a "residual", represents the amount of any oxide that can not be accounted for, given the bulk rock geochemistry and mineral compositions provided. Therefore, the sum of the slack variables is an indication of how well the mineral compositions fit the rock's composition.

The objective function [Eq. (5)] maximizes mineral abundances and minimizes slack variables. Another type of objective function, which has yielded excellent results in our experience with LPNORM, consists in only minimizing the slack variables:

$$z = \sum_{i=1}^n \lambda_i. \quad (8)$$

The set of equations in (6) and (7) is solved simultaneously by the Simplex Method, which is detailed

in any linear programming textbook. Geometrically speaking, the set of linear equations [Eqs. (6) and (7)] define a volume, which usually is closed and convex ("convex hull") in the N -dimensional space considered. The objective function has a range of values across this volume, and the final solution among the infinite number of possible solutions is the one that maximizes the objective function.

In order to illustrate how mineral abundances are determined by LPNORM, let us consider the following simple, theoretical example: a fictional rock containing 10 wt% quartz, 20 wt% kaolinite, 35 wt% chlorite, and 35 wt% illite has a calculated composition of 47.28 wt% SiO_2 , 22.22 wt% Al_2O_3 , 15.07 wt% Fe_2O_3 , 5.33 wt% MgO , and 3.15 wt% K_2O , giving an anhydrous total of 93.05 wt%. XRD results indicate that the rock contains quartz, kaolinite, illite, and chlorite. The composition of the two first phases reasonably can be considered to be fixed (quartz = SiO_2 , kaolinite = $\text{Al}_2\text{Si}_2\text{O}_5(\text{OH})_4$), and the compositions of the two latter minerals are assumed to be known in this simple example (illite = $\text{K}_{0.76}\text{Fe}_{0.31}\text{Mg}_{0.28}\text{Al}_{1.97}\text{Si}_{3.46}\text{O}_{10}(\text{OH})_2$, chlorite = $\text{Fe}_1\text{Mg}_2\text{Al}_2\text{Si}_3\text{O}_{10}(\text{OH})_8$). The linear programming problem at hand is stated thus: maximize the objective function:

$$z = q + k + i + c - \lambda_{\text{SiO}_2} - \lambda_{\text{Al}_2\text{O}_3} - \lambda_{\text{Fe}_2\text{O}_3} - \lambda_{\text{MgO}} - \lambda_{\text{K}_2\text{O}}, \quad (9)$$

where q , k , i , and c represent the abundance (in wt%) of quartz, kaolinite, illite, and chlorite in the rock, subject to:

$$47.28 = 1.0000q + 0.4655k + 0.5221i + 0.2771c + \lambda_{\text{SiO}_2} \quad (10)$$

$$22.22 = 0.3950k + 0.2522i + 0.1568c + \lambda_{\text{Al}_2\text{O}_3} \quad (11)$$

$$15.07 = 0.0622i + 0.3683c + \lambda_{\text{Fe}_2\text{O}_3} \quad (12)$$

$$5.33 = 0.0284i + 0.1239c + \lambda_{\text{MgO}} \quad (13)$$

$$3.15 = 0.0899i + \lambda_{\text{K}_2\text{O}} \quad (14)$$

and to:

$$q \geq 0, k \geq 0, i \geq 0, c \geq 0, \lambda_{\text{SiO}_2} \geq 0, \lambda_{\text{Al}_2\text{O}_3} \geq 0, \lambda_{\text{Fe}_2\text{O}_3} \geq 0, \lambda_{\text{MgO}} \geq 0, \lambda_{\text{K}_2\text{O}} \geq 0. \quad (15)$$

The solution of this linear programming problem, determined using LPNORM, is:

$$q = 10(9.98) \text{ wt}\%, k = 20(19.99) \text{ wt}\%, i = 35(35.00) \text{ wt}\%, c = 35(35.04) \text{ wt}\%, \quad (16)$$

which is the correct solution. All slack variables are equal to zero in this example.

The $a_{i,j}$ coefficients for the first equation of this example [Eq. (10)] were calculated as follows:

$$a_{\text{SiO}_2, \text{quartz}} = \frac{MW_{\text{SiO}_2}}{MW_{\text{quartz}}} \times \frac{S_{\text{Si, quartz}}}{S_{\text{Si, SiO}_2}} = \frac{60.08}{60.08} \times \frac{1.00}{1.00} = 1.0000, \quad (17)$$

$$a_{\text{SiO}_2, \text{kaolinite}} = \frac{MW_{\text{SiO}_2}}{MW_{\text{kaolinite}}} \times \frac{S_{\text{Si, kaolinite}}}{S_{\text{Si, SiO}_2}} \\ = \frac{60.08}{258.16} \times \frac{2.00}{1.00} = 0.4655, \quad (18)$$

$$a_{\text{SiO}_2, \text{illite}} = \frac{MW_{\text{SiO}_2}}{MW_{\text{illite}}} \times \frac{S_{\text{Si, illite}}}{S_{\text{Si, SiO}_2}} \\ = \frac{60.08}{398.18} \times \frac{3.46}{1.00} = 0.5221, \quad (19)$$

$$a_{\text{SiO}_2, \text{chlorite}} = \frac{MW_{\text{SiO}_2}}{MW_{\text{chlorite}}} \times \frac{S_{\text{Si, chlorite}}}{S_{\text{Si, SiO}_2}} \\ = \frac{60.08}{650.42} \times \frac{3.00}{1.00} = 0.2771, \quad (20)$$

and so forth for the other equations.

Note that the objective function can be modified to maximize mineral abundances only, or minimize the slack variables only, instead of doing both together, as in Equation (9). Also note that any oxygen and hydrogen necessary for making up some minerals is added as needed, and taken into account in the final mineral weight proportions calculated.

Mineral compositions

The simple, theoretical example presented here would correspond to a situation where all mineral compositions are known exactly, all minerals present in the rock have been identified, and the rock, which contains no amorphous material, was analyzed fully with absolute accuracy. Obviously, this will never be the situation in any real world normative analysis. Usually there will be uncertainty about the composition of one or more minerals identified to be present in the rock. Further, many minerals may show a range of compositions within a single sample. Errors in bulk rock geochemical analyses, and those derived from the misinterpretation of XRD patterns, also should be assessed carefully by each user.

As alluded to before, the "goodness-of-fit" between the bulk geochemistry of a sample and the combination of mineral compositions and abundances that make up that sample, may be indicated by the sum of all the slack variables. However, a low or nil slack variable sum does not necessarily prove that the calculated norm is accurate: erroneous mineral compositions may yield low residuals, but generally the resulting norm will not be credible to the vigilant analyst, as it will not be consistent with other available information.

Many clay minerals contain small amounts of Na, Ca, Ti, etc. In minerals where these are minor constituents, slight overestimation of their abundance within these minerals may limit drastically the calculated percentage of that mineral. Further, in the situation of XRF analysis of Na₂O at least, experimental accuracy and precision usually are low. For these reasons, it may be wise to ignore these elements

in a mineral formula, which, in turn, may inflate the residual sum.

Minerals belonging to solid solutions, which are abundant in natural rocks and soils, can be dealt with by using only the end-member compositions in LP-NORM. The resulting end-member mixture then can be reported as such. Alternatively, a single phase can be reported, the composition of which is the weighted average of the end-members, or combinations of the possible intermediate compositions can be used. Once again, it is independent knowledge about the sample and the constituent minerals that determines what alternative is the most appropriate. For instance, if LPNORM returns 2.5 wt% calcite (CaCO₃), 1.5 wt% magnesite (MgCO₃), and 1 wt% siderite (FeCO₃), these results can be reworked to say that the rock contains 5 wt% carbonate (of average composition Ca_{0.50}Fe_{0.20}Mg_{0.30}CO₃), or 1 wt% calcite + 1.5 wt% dolomite (Ca_{0.50}Mg_{0.50}CO₃) + 1 wt% siderite, or ankerite and/or Fe-calcite can be invoked, and so forth. Specific knowledge about the composition of the rock-forming minerals obtained by XRD, EDX, electron-microprobe, analytical TEM, etc. must be integrated at this stage.

Sedimentary rocks may contain detrital illite and authigenic illite, or detrital chlorite and authigenic chlorite. The composition of these two mineral generations usually is different. If the composition of one of these generations can be determined, by electron microprobe on the detrital phase for example, normative analysis may help constrain the abundances of these two phases.

LPNORM can be used to improve one's estimate about the composition of one or a few mineral phases by the "trial-and-error" approach. This technique will be used in the subsequent section on sensitivity analysis, but it is worth stressing at this stage that the number of phases with uncertain composition should be kept to a strict minimum, and, therefore, that additional information about the rock and about some of the contained minerals may have to be obtained independently if one is to have any confidence in the resulting normative calculation.

SENSITIVITY ANALYSIS

In order to illustrate how it is possible to constrain the composition of one or a few minerals in a rock, let us conduct a theoretical experiment as was done for the example detailed previously. Consider a fictitious rock containing 10 wt% quartz (SiO₂), 20 wt% kaolinite (Al₂Si₂O₅(OH)₄), 10 wt% illite (K_{0.76}Fe_{0.31}Mg_{0.28}Al_{1.97}Si_{3.46}O₁₀(OH)₂), and 60 wt% chlorite (Fe₃Mg₂Al₂Si₃O₁₀(OH)₈). The exact bulk rock chemical composition is calculated to be: 41.16 wt% SiO₂, 19.83 wt% Al₂O₃, 22.72 wt% Fe₂O₃, 7.72 wt% MgO, and 0.90 wt% K₂O, giving an anhydrous total of 92.33 wt%. A series of calculations can be conducted in which the composition of two clay-mineral phases, namely illite and chlorite, is

deliberately modified. Normative analyses then are conducted using these inaccurate compositions, in effect simulating flawed initial guesses about these compositions. Observing how these mistaken compositions affect the calculated norm is part of the extensive sensitivity analysis we performed, and also provides a method to improve the inaccurate initial mineral compositions, as will be shown next.

In the experiment reported here, the Fe# (Fe/[Fe + Mg]) of illite and chlorite are altered deliberately and simultaneously from their correct values of 0.53 and 0.60, respectively. The calculations use combinations of erroneously high chlorite Fe# with high illite Fe#, and low chlorite Fe# with low illite Fe# (Fig. 1), or high chlorite Fe# with low illite Fe#, and low chlorite Fe# with high illite Fe# (Fig. 2). These clearly show that, in well-constrained situations at least, it is possible to determine precisely the Fe# of chlorite and illite in only a few trial-and-error attempts, by monitoring the effects these incorrect compositions have on the calculated norm. In both situations (Figs. 1 and 2), the wrong Fe#'s result in conspicuously low chlorite abundances and totals, and in somewhat less marked high kaolinite and quartz abundances, whereas the illite abundance remains unaffected. In addition, the residuals display a distinct minimum at the correct chlorite and illite Fe#, and change composition from one side of the minimum to the other (in the present situation from Fe to Mg oxide). During any trial-and-error process, this change in composition of the residual generally will indicate that the correct solution has just been exceeded, and

this can be used to narrow down any further calculation.

Another experiment was conducted to see if the K-content of illite could be "optimized" by using LPNORM using the same trial-and-error approach. In this situation, the fictitious rock contains 20 wt% quartz (SiO_2), 27 wt% kaolinite ($\text{Al}_2\text{Si}_2\text{O}_5(\text{OH})_4$), 38 wt% illite ($\text{K}_{0.82}\text{Fe}_{0.60}\text{Mg}_{0.16}\text{Al}_{1.74}\text{Si}_{3.46}\text{O}_{10}(\text{OH})_2$), and 15 wt% chlorite ($\text{Fe}_3\text{Mg}_1\text{Al}_2\text{Si}_{3.5}\text{O}_{10}(\text{OH})_8$). The exact bulk rock chemical composition is calculated to be: 56.88 wt% SiO_2 , 21.32 wt% Al_2O_3 , 10.08 wt% Fe_2O_3 , 1.55 wt% MgO , and 3.60 wt% K_2O , giving an anhydrous total of 93.43 wt%. Starting initial compositions for illite and chlorite were selected at random, simulating a real-life situation, and after a few initial iterations, the Fe# of chlorite and illite was constrained with sufficient confidence, following the approach outlined previously. Next, the K-content of illite was changed, and the effect on calculated norms was monitored (Fig. 3). Across the range of tested K-content values (going from 0.60 to 0.90 K/ $\text{O}_{10}(\text{OH})_2$), mineral totals remained constant at ~ 100 wt%, illite abundances fell from 51 to 35 wt%, whereas all other norms increased (kaolinite from 21 to 29 wt%, quartz from 17 to 21 wt%, and chlorite from 11 to 16 wt%). These variations show no break in slope (maxima or minima). The residuals are negligible in all instances (< 0.06 wt%), but do exhibit a minimum and a change in composition.

These results indicate that linear programming normative analysis is sensitive to changes in mineral compositions. The trial-and-error normative

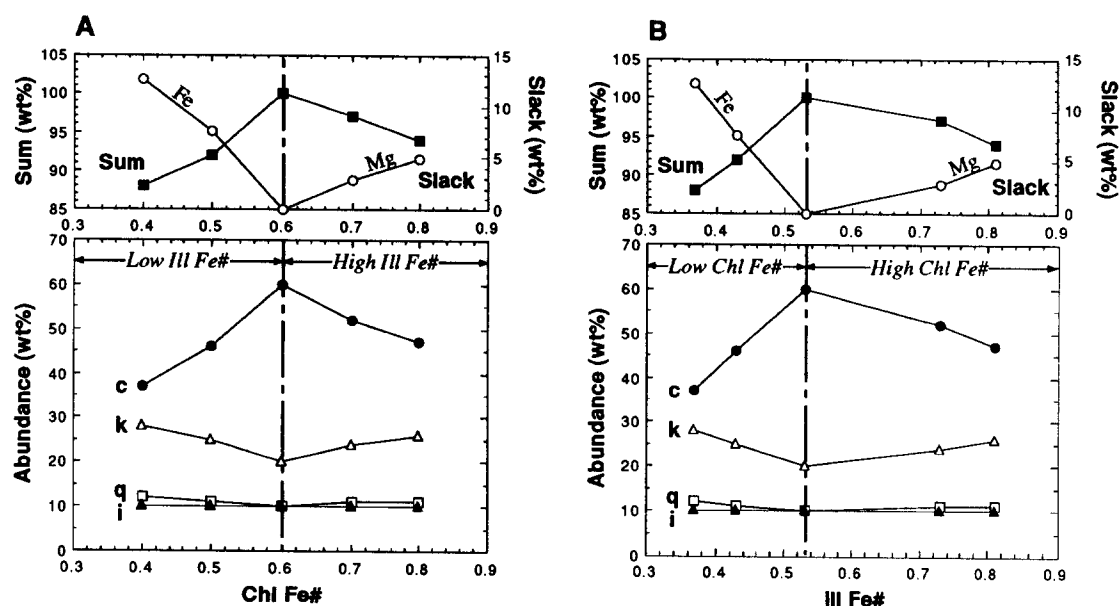


Figure 1. LPNORM results of sensitivity analysis: A, abundances versus chlorite Fe#; B, abundances versus illite Fe#. Fe# of chlorite and illite were modified to be simultaneously too high or too low. Bold, vertical dashed line indicates correct solution. "Fe" and "Mg" indicate what oxide is in excess (slack variable) on either side of solution. Symbols: "c" is for chlorite, "k" for kaolinite, "q" for quartz, and "i" for illite abundance; "Sum" is for total mineral abundance. Some of these LPNORM runs included constraint: chlorite abundance ≥ 10 wt%, or chlorite abundance ≥ 20 wt%.

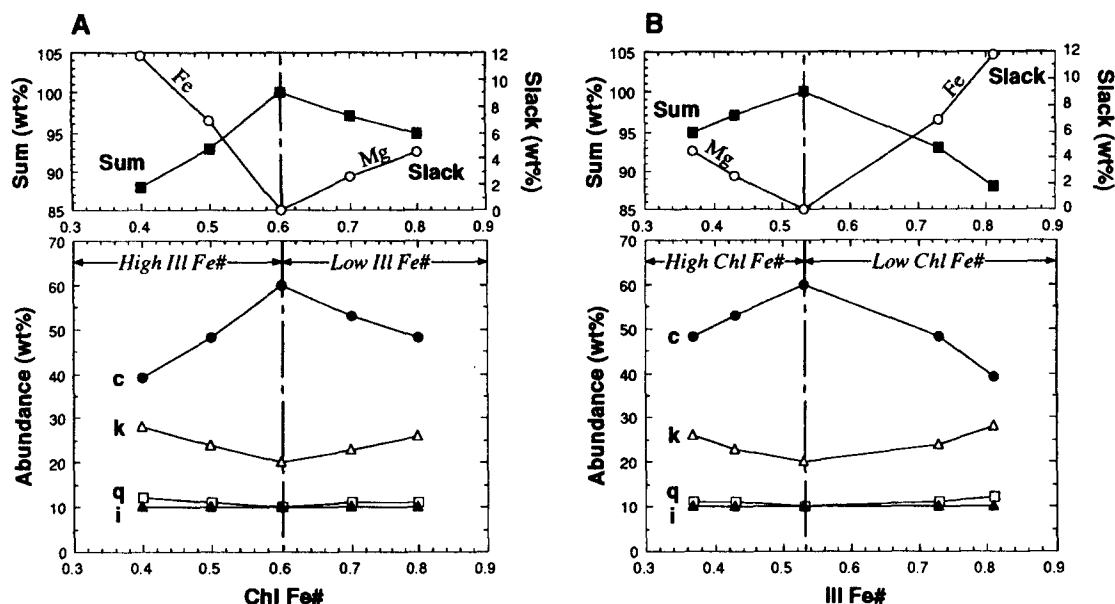


Figure 2. LPNORM results of sensitivity analysis: A, abundances versus chlorite Fe#; B, abundances versus illite Fe#. Fe# of chlorite and illite were modified to be high and low, or low and high. Some of these LPNORM runs included constraint: chlorite abundance ≥ 10 wt%, or chlorite abundance ≥ 20 wt%. (See Fig. 1 for label descriptions.)

approach is not as effective in constraining the K-content of illite as in determining the Fe# of illite and chlorite, as there is only one K-bearing mineral present whereas there are two Fe- and Mg-bearing phases in the rock. Also, the Si/Al of minerals is difficult to constrain when quartz and kaolinite are present, because the abundances of these two phases will adjust to dampen any SiO_2 and/or Al_2O_3 residuals. Therefore, it may be necessary to rely on additional, independent information about mineral

compositions and abundances to obtain a credible whole rock norm. Finally, Banks (1979) showed that for real rocks and real analyses (with associated errors) there exists a relatively large range of norms that satisfy the bulk rock composition at the 95% confidence level, even if errors in mineral compositions are ignored. Precise chemical analyses of the bulk sample thus are essential for obtaining a reasonable norm.

TESTING THE METHOD

The calculations presented thus far in this paper were all based on theoretical mixtures of minerals with predetermined and fixed compositions. It is imperative to test this method with actual data from real mineral mixtures, in which errors are introduced during bulk sample chemical analysis and while estimating individual mineral compositions, and where the presence of unsuspected, or amorphous and therefore XRD-invisible, phases cannot be irrevocably discounted. Calculation results are presented next in order of increasing mineralogical complexity: first, LPNORM results for a relatively coarse-grained rock (amphibolite) are compared with published modes; second, a series of clay mixtures analyzed by XRD and X-ray fluorescence (XRF) and subjected to normative analysis are presented; third, LPNORM output for a mudstone that has become a *de facto* yardstick for normative analysis in recent years is offered; fourth, mode and norm for a carbonate-altered lithic sandstone are compared. Finally, a recent application of the program for calculating shale norms is reported.

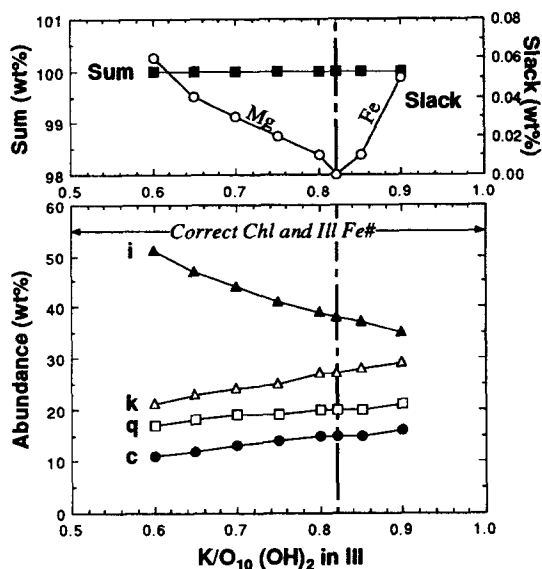


Figure 3. LPNORM results of sensitivity analysis: abundances versus K-content in illite (per $\text{O}_{10}(\text{OH})_2$) in Ill. Some of these LPNORM runs included constraint: chlorite abundance ≥ 5 wt%. (See Fig. 1 for label descriptions.)

Table 1. Three Valley Gap amphibolite samples: comparison of mode and norm

Mineral	Sample BR-2		Sample BR-3		Sample BR-4		Sample BR-5	
	Mode	LPNORM	Mode	LPNORM	Mode	LPNORM	Mode	LPNORM
Quartz	9	7	11	2	0	6	10	0
Plagioclase	28	30	36	35	36	37	27	32
Amphibole	52	51	44	46	47	46	55	59
Clinopyroxene	9	11	8	16	16	10	8	10
Orthopyroxene	0	0	1	0	0	0	0	0
Apatite	1	0	0	0	0	0	0	0
Chlorite	1	1	0	0	0	0	0	0
Sum	100	100	100	99	100	100	100	101

Modes in vol% and norms in wt% (mineral amounts rounded off to the closest integer).

Comparison of vol% and wt% justified on the basis of similar mineral specific weights.

Modes from Nicholls, Stout, and Ghent (1991).

Mineral compositions from Ghent and others (1977), and Nicholls, Stout, and Ghent (1991).

Three Valley Gap amphibolite

Ghent and others (1977) and Nicholls, Stout, and Ghent (1991) published whole rock chemical analyses and electron microprobe analyses of minerals for amphibolite boudin samples collected near Three Valley Gap in British Columbia, Canada. Nicholls, Stout, and Ghent (1991) presented mineral modes for four of these Three Valley Gap samples. These modes were obtained from automated electron microprobe counts, as detailed in Nicholls and Stout (1986) and Nicholls, Stout, and Ghent (1991). LPNORM was run for these samples on the basis of the chemical probe analysis data for the various minerals. Table 1 summarizes the comparison of mineral modes and calculated norms, also illustrated in Figure 4. The objective function for all these runs was to maximize mineral abundances and minimize slack variables. The correlation between mode and norm generally is satisfactory, and the few discrepancies, such as for quartz in samples BR-3, 4, and 5, could be explained by the compositional heterogeneity of these amphibolite boudins (Nicholls, Stout, and Ghent, 1991) as well as the presence of sporadic quartz veins (J. Nicholls, pers. comm., 1993), in addition to errors as discussed previously.

Clay standard mixtures

The next test was to compare LPNORM results with known proportions of clay mineral standards mixed together. The individual standards used were K3 kaolinite (Weaver and Pollard, 1973), IMT-1 illite (Hower and Mowatt, 1966), Upton (WY) smectite (Ross and Hendricks, 1945), and Chester (VT) chlorite (Foster, 1962). These standard clays were analyzed by XRD (to check their purity) and by XRF (to determine their chemistry) prior to admixing them. Table 2 summarizes the resulting clay standard compositional data. Four different clay standard mixtures were prepared by admixing various weights of these clay standards. These mixtures then were analyzed also by XRD and XRF (Table 3), and these results were used as input for normative analysis using LPNORM. All the runs were performed with the

same objective function, namely to maximize mineral abundances and minimize slack variables. Smectite and illite compositions were idealized by ignoring the small amounts of Na, Ca, and Ti contained, for reasons presented earlier. Results of the normative analysis, shown in Table 4 and Figure 5, indicate a reasonable match between weighted modes and calculated norms for samples CSM-5 and CSM-6, which contain high amounts of smectite, and a poorer agreement for samples CSM-7 and CSM-8, which contain lower amounts of smectite and higher amounts of kaolinite and illite.

Bersham mudstone

Nicholls (1962) presented chemical data for an Upper Carboniferous mudstone sample collected at a depth of about 1260 ft (≈ 380 m) from a core drilled near Bersham, U.K. Nicholls (1962) performed normative analysis, using a series of "normative" mineral formulae (some of these are similar to end-member compositions of solid solutions, and are labeled "illite-1", "illite-2", and "illite-3", and "chlorite-1" and "chlorite-2" herein, the others are considered to be pure phases). The chemical data (Table 5) and mineral formulae from Nicholls (1962) are used as input for LPNORM, and our results are compared to Nicholls' (1962) results and Cohen and Ward's (1991) SEDNORM results (Table 6). The LPNORM results, using the minimization of slack variables as objective function, compare favorably with the original data from Nicholls (1962) (Fig. 6), except that LPNORM predicted 4 wt% more kaolinite ($\text{Al}_4\text{Si}_4\text{O}_{10}(\text{OH})_8$) and 6 wt% less "illite-3" ($\text{Al}_2\text{Si}_4\text{O}_{10}(\text{OH})_2$). This reflects the relatively comparable composition of these two phases, which therefore are more or less interchangeable. Because "illite-3" has a higher SiO_2 content (~ 67 wt%) than kaolinite (~ 46 wt%), transferring 4% mass from the kaolinite to the "illite-3" phase, would result in a decrease of about 2–3 wt% free SiO_2 , and therefore quartz.

Bowen sandstone

Cohen and Ward (1991) performed normative analysis with SEDNORM (their tables 3 and 4) on a

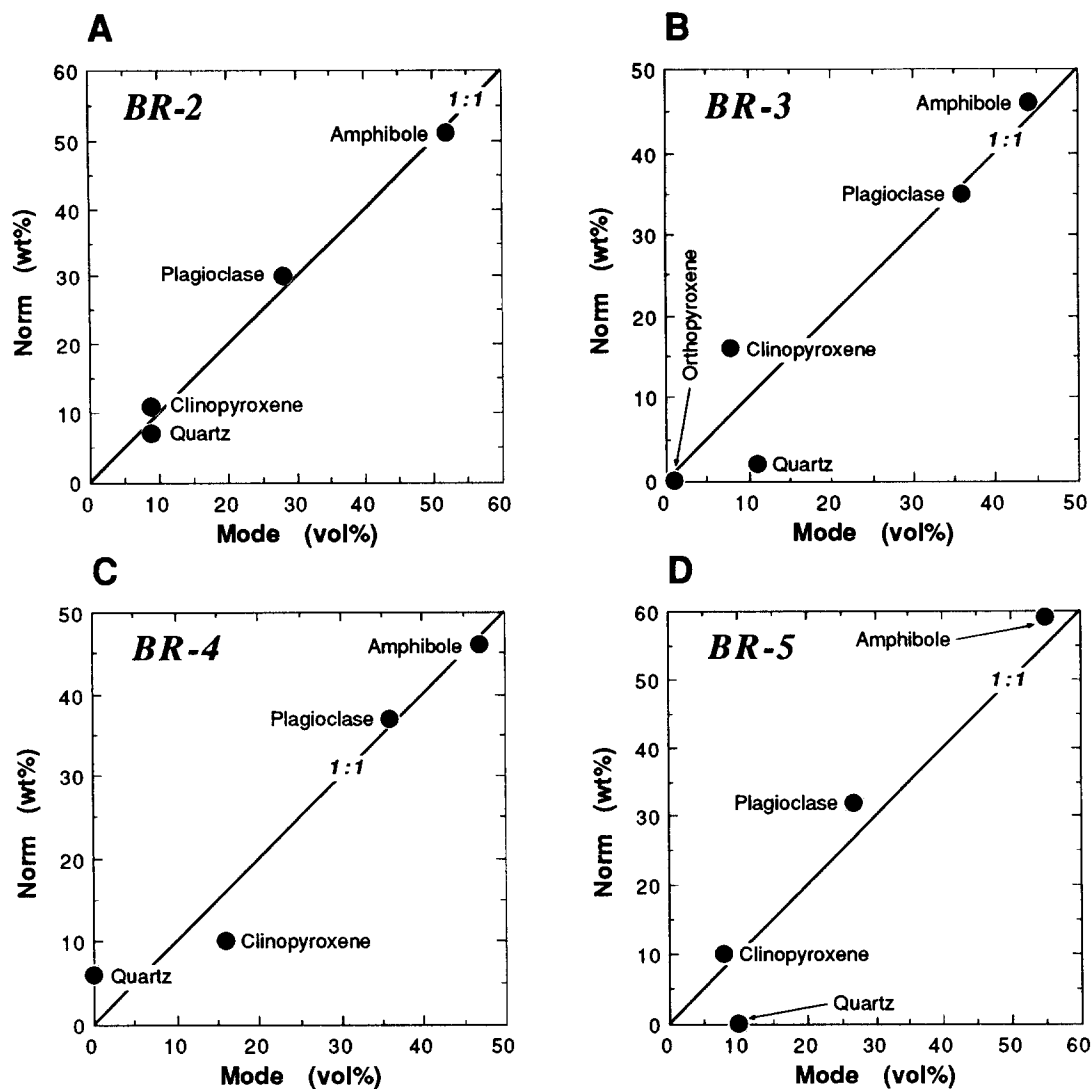


Figure 4. Norm versus mode plot for samples BR-2 (A), BR-3 (B), BR-4 (C), and BR-5 (D) from Three Valley Gap. Diagonal line labeled "1:1" represent location of the perfect, 1:1 correlation between norm and mode. See text for discussion.

Table 2. Clay mineral standards: XRF compositions

Oxide	K3 Kaolinite	IMT-1 Illite	Upton, WY Smectite	Chester, VT Chlorite
SiO ₂	45.95	48.35	56.76	20.13
Al ₂ O ₃	38.79	24.10	19.86	20.93
Fe ₂ O ₃ *		7.68	3.98	15.85
MgO		2.96	2.72	32.20
CaO		0.35	0.43	
MnO				
K ₂ O		8.50	0.11	
Na ₂ O		1.99	3.28	
P ₂ O ₅				
TiO ₂			0.15	
LOI	14.89	6.43	12.70	10.09
Total	99.63	100.36	99.99	99.20

All values in wt% (LOI: loss on ignition).

Illite and chlorite compositions corrected for the presence of ~4.5 wt% quartz and ~4.7 wt% magnetite, respectively, in clay standard.

*Total Fe reported as Fe₂O₃.

Table 3. Clay standard mixtures: XRF compositions

Oxide	Sample CSM-5	Sample CSM-6	Sample CSM-7	Sample CSM-8
SiO ₂	51.49 ± 0.30	48.70 ± 0.34	45.85 ± 0.37	51.23 ± 0.44
Al ₂ O ₃	22.83 ± 0.41	24.11 ± 0.39	25.61 ± 0.53	25.53 ± 0.37
Fe ₂ O ₃ *	5.17 ± 0.11	5.92 ± 0.11	6.85 ± 0.16	5.76 ± 0.08
MgO	3.78 ± 0.12	4.63 ± 0.10	5.82 ± 0.17	2.83 ± 0.17
CaO	0.37 ± 0.01	0.33 ± 0.01	0.29 ± 0.01	0.33 ± 0.01
MnO	0.04 ± 0.00	0.05 ± 0.00	0.08 ± 0.00	0.03 ± 0.01
K ₂ O	1.67 ± 0.03	2.33 ± 0.02	3.10 ± 0.05	5.04 ± 0.05
Na ₂ O	1.83 ± 0.37	1.87 ± 0.45	0.71 ± 0.42	0.54 ± 0.50
P ₂ O ₅	0.09 ± 0.03	0.11 ± 0.02	0.12 ± 0.02	0.21 ± 0.04
TiO ₂	0.75 ± 0.02	0.99 ± 0.02	1.33 ± 0.02	1.04 ± 0.02
LOI	12.00	10.95	10.20	7.45
Total	100.02	99.99	99.96	99.99

All values in wt%.

Reported compositions are the average and ISD of duplicate or triplicate XRF analyses performed on three glass beads for each mixture (except LOI).

*Total Fe reported as Fe₂O₃.

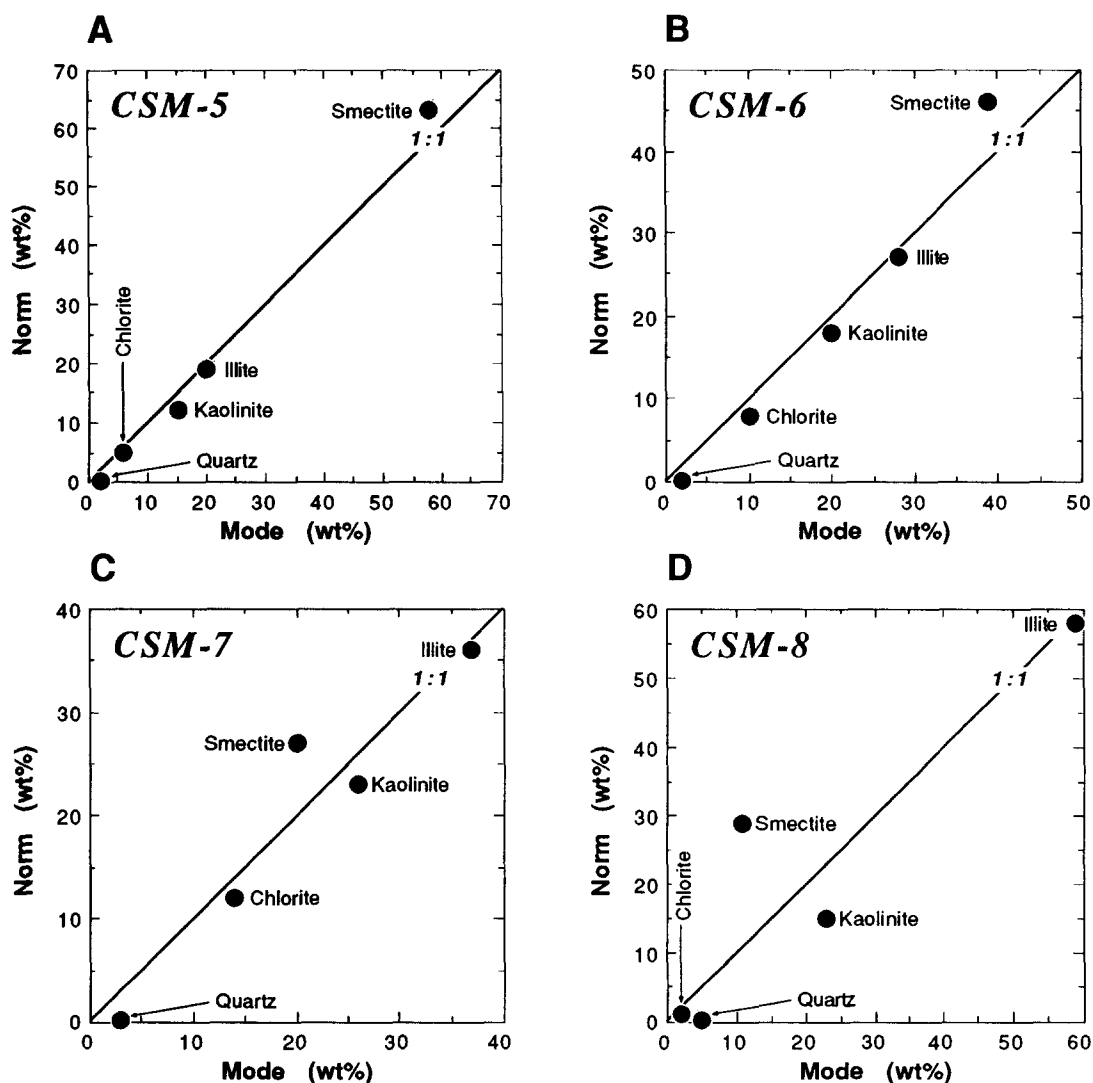


Figure 5. Norm versus mode plot for samples CSM-5 (A), CSM-6 (B), CSM-7 (C), and CSM-8 (D). See text for discussion.

Table 4. Clay standard mixtures: comparison of mode and norm

Mineral	Sample CSM-5		Sample CSM-6		Sample CSM-7		Sample CSM-8	
	Mode	LPNORM	Mode	LPNORM	Mode	LPNORM	Mode	LPNORM
Quartz	2	0	2	0	3	0	5	0
Chlorite	6	5	10	8	14	12	2	1
Kaolinite	15	12	20	18	26	23	23	15
Illite*	20	19	28	27	37	36	59	58
Smectite†	58	63	39	46	20	27	11	29
Sum	100	100	100	100	100	101	100	102

Modes and norms in wt% (mineral amounts rounded off to the closest integer).

*Illite composition free of Na and Ca.

†Smectite composition free of Na, Ca, and Ti.

carbonate-altered lithic sandstone sample from the Bowen Basin, Queensland (Australia), for which they also reported optical modes. LPNORM was applied to that sample (Tables 7 and 8), with the objective function to minimize the slack variables, and results show a good correlation with the mode (Fig. 7). Given the amount of (inorganic) CO₂ reported (11.2 wt%), it is impossible to have 34% carbonates, as reported in the mode (Cohen and Ward, 1991, table 4). Indeed, using the carbonate with lowest CO₂ content, siderite (~38 wt%), among the carbonates reported to be present, it is evident that the maximum carbonate proportion allowable is 29 wt%. This illustrates the usefulness of complementing any modal analysis by normative analysis to confirm that the modes are at least consistent with the geochemistry. The discrepancy for the "kaolinite + muscovite" amounts is difficult to comment upon without distinct modes for each of these two phases. Obviously, the abundance of muscovite is governed by the K₂O content of the sample (1.20 wt%), which imposes a muscovite abundance between 10 wt% (for 1.00 K/O₁₀(OH)₂) and 13 wt% (for 0.75 K/O₁₀(OH)₂). Unless there are large quantities of free (?amorphous) Si and Al in the sample, or the analysis lacked precision, it is inescapable that there must be a high kaolinite content (26 or 23 wt%, respectively, for these two extreme muscovite compositions). Thus, the sum of the abundances of kaolinite and illite will be about 36 wt%.

Belle Fourche Formations shales

de Caritat and others (1994) presented a study of the mineralogy and geochemistry of the Cenomanian Belle Fourche Formation, Western Canada Sedimentary Basin. The Belle Fourche Formation is a regionally extensive shale unit within the lower Colorado Group in this foreland basin. Bulk rock geochemistry of shale and bentonite core samples were determined from XRF analysis, Rock-Eval pyrolysis, and elemental C and S determinations. Clay mineral compositions were determined from XRF analysis, and elemental C and S determinations, on thirteen <0.2 or <0.1 μ m, quasimonomineralic clay-fraction separates from shale samples. LPNORM was used to derive the mineralogical composition of the shale and

bentonite samples, using semiquantitative constraints from XRD data. Resulting norms were taken to be accurate within 20% relative. Composition of the shale matrix was calculated by removing the contributions of the silt fraction (quartz, feldspars, etc.) from the bulk shale geochemistry, on the basis of the normative abundances of these minerals. The results agree well with the available compositions of the thirteen clay-fraction separates (Fig. 8), suggesting that reasonable clay-mineral formulae had been used (see de Caritat and others, 1994, for further discussion). Overestimating the abundance of kaolinite, that is setting aside too much Si and Al for this clay mineral, resulted in an overly celadonic composition (Fe- and Mg-enriched) for a few illite/smectites (Fig. 9), although the possibility that these celadonic compositions are real can not be altogether discounted at this stage.

CONCISE USER'S GUIDE

LPNORM consists of three main parts: (1) mineral compositional data entry; (2) bulk sample composition entry; and (3) computation of the stoichiometric coefficients, setting up of the linear equations, and solving them by the Simplex Method.

Table 5. Composition of Bersham mudstone

Oxide	Bersham mudstone
SiO ₂	62.56
Al ₂ O ₃	20.60
Fe ₂ O ₃ *	3.55
MgO	1.25
CaO	0.34
MnO	0.02
K ₂ O	3.35
Na ₂ O	0.62
P ₂ O ₅	0.17
TiO ₂	0.94
CO ₂	0.90
S	0.02
Total	94.32

All values in wt% (LOI not reported).

Data from Nicholls (1962).

*FeO and Fe₂O₃ (from Nicholls, 1962) combined as Fe₂O₃.

Mineral data input

Minerals to be used for normative analysis must have a name, the ten first characters of which are used in the program only, and the name needs not be any formal or official name. Mineral data can be entered in two different formats: either as mineral formulae, or as a listing of oxides. To enter mineral formulae, the companion computer program MINPUT can be used to create the mineral data file MINLIST.DAT that will be used by LPNORM, or that file can be directly edited. Elements taken into account are, in order of increasing atomic weight: H, C, N, O, F, Na, Mg, Al, Si, P, S, Cl, K, Ca, Ti, Mn, Fe, and Ba; provision was made for OH to be treated as though it was an element by itself, because it is a constituent of clay-mineral formulae.

To enter mineral geochemistry data, or the $a_{i,j}$ coefficients (which are equivalent to the oxide content in wt% divided by 100), approximate mineral formulae can be used and the $a_{i,j}$ coefficients can be modified interactively at the beginning of the program. Entering mineral compositional data as $a_{i,j}$ coefficients (i.e. oxides from chemical or microprobe analyses), rather than as mineral formulae, is recommended where at all possible, given that the construction of any mineral formula from a chemical analysis introduces rounding errors and relies on subjective choices. The $a_{i,j}$ coefficients, whether calculated by LPNORM on the basis of the MINLIST.DAT file or interactively entered during LPNORM execution, are stored in the AIJ.DAT file. If the user desires to modify directly the AIJ.DAT file, this can be achieved using any available editor. This file can be read in directly during LPNORM execution.

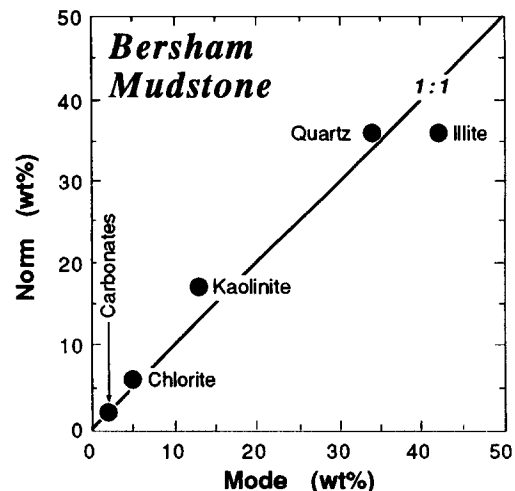


Figure 6. Norm versus mode plot for Bersham mudstone sample. See text for discussion.

Oxide data input

The oxide data list, which is the catalog of oxide abundances in the bulk sample(s) under investigation, is stored in a file called OXLIST.DAT. Similar to the procedure for entering mineral data briefly described in the previous section, the companion program OXINPUT can be used, or the OXLIST.DAT file can be edited prior to running LPNORM. The oxides listed in this file are: SiO_2 , Al_2O_3 , Fe_2O_3 , MgO , Na_2O , CaO , K_2O , TiO_2 , P_2O_5 , MnO , and inorganic CO_2 ; in addition, sulfur is entered as elemental S.

Because a batch of similar samples (from the same outcrop, core, stratigraphic unit, etc.) may have a similar mineralogy (but different mineral abundances), LPNORM reads one list of minerals

Table 6. Bersham mudstone: comparison of norms from Nicholls (1962), from SEDNORM (Cohen and Ward, 1991), and from LPNORM (this study)

Mineral	Norm*	SEDNORM†	LPNORM‡	Mineral composition
Quartz	34	31	36	SiO_2
Illite-1	28	NA	28	$\text{K}_2\text{Al}_6\text{Si}_6\text{O}_{20}(\text{OH})_4$
Illite-2	8	NA	8	$\text{Na}_2\text{Al}_6\text{Si}_6\text{O}_{20}(\text{OH})_4$
Illite-3	6	NA	0	$\text{Al}_4\text{Si}_8\text{O}_{20}(\text{OH})_4$
Chlorite-1	2	NA	3	$\text{Fe}_{12}\text{Si}_8\text{O}_{20}(\text{OH})_{16}$
Chlorite-2	3	NA	3	$\text{Mg}_{12}\text{Si}_8\text{O}_{20}(\text{OH})_{16}$
Kaolinite	13	22	17	$\text{Al}_4\text{Si}_4\text{O}_{10}(\text{OH})_8$
Calcite	0	0	0	CaCO_3
Magnesite	0	1	0	MgCO_3
Siderite	2	1	2	FeCO_3
Apatite	0	0	0	$\text{Ca}_5(\text{PO}_4)_3(\text{OH})$
Anatase	1	1	1	TiO_2
Hematite (or "excess" Fe_2O_3)	1	0	0	Fe_2O_3
Pyrite	0	0	0	FeS_2
Water	1	NA	NA	H_2O
Sum	100	100	99	

Norms in wt% (mineral amounts rounded off to the closest integer).

*"Argillaceous norm" from Nicholls (1962, tables 2 and 3).

†SEDNORM data from Cohen and Ward (1991, table 5).

‡LPNORM data (this study).

§"Smectite" used in Cohen and Ward (1991) instead of chlorite.

Table 7. Composition of Bowen Basin carbonate-altered sandstone

Oxide	Bowen sandstone
SiO ₂	52.5
Al ₂ O ₃	15.5
Fe ₂ O ₃ *	4.1
MgO	1.8
CaO	8.9
MnO	0.1
K ₂ O	1.2
Na ₂ O	0.8
P ₂ O ₅	0.2
TiO ₂	0.8
CO ₂	11.2
S	0.02
Total	97.12

All values in wt% (LOI not reported).

Data from Cohen and Ward (1991).

*Total Fe reported as Fe₂O₃.

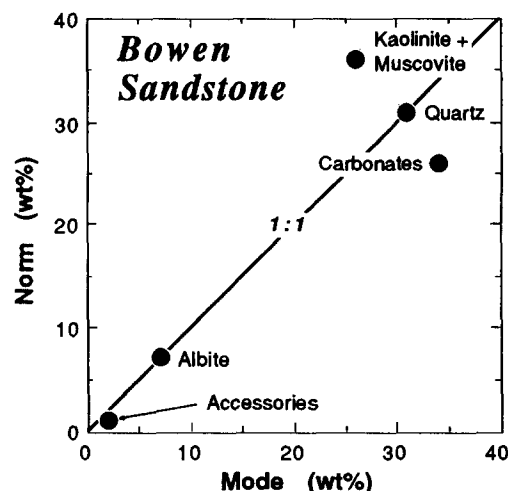


Figure 7. Norm versus mode plot for Bowen sandstone sample. See text for discussion.

Table 8. Bowen sandstone: comparison of modes with norms from SEDNORM (Cohen and Ward, 1991), and from LPNORM (this study)

Mineral	Mode*	SEDNORM†	LPNORM‡	Mineral composition
Quartz	31	32	31	SiO ₂
Albite	7	10	7	NaAlSi ₃ O ₈
Kaolinite	NA	27	26	Al ₂ Si ₄ O ₁₀ (OH) ₈
Muscovite	NA	5	10	KAl ₃ Si ₃ O ₁₀ (OH) ₂
Calcite	NA	15	16	CaCO ₃
Magnesite	NA	4	4	MgCO ₃
Siderite	NA	6	6	FeCO ₃
Anatase	NA	1	1	TiO ₂
Pyrite	NA	NA	0	FeS ₂
Hematite	NA	0	0	Fe ₂ O ₃
Sum	100	99	100	

Mode in vol%, norms in wt% (mineral amounts rounded off to the closest integer).

*Optical mode from Cohen and Ward (1991, table 4).

†SEDNORM data from Cohen and Ward (1991, table 4).

‡LPNORM data (this study).

§Collectively reported as "accessory minerals" in Cohen and Ward (1991).

(MINLIST.DAT), and uses it for up to 100 consecutive samples listed in the OXLIST.DAT file.

Both the MINLIST.DAT and the OXLIST.DAT files are read as formatted files during execution of LPNORM, and it therefore is imperative to respect the correct number of spaces between data entry, etc. In doubt, use can be made of the MINPUT and OXINPUT companion programs.

Running LPNORM

Once the MINLIST.DAT and OXLIST.DAT files, and, if applicable the additional AIJ.DAT file, are ready, execution of LPNORM can begin. During execution, LPNORM overwrites (or creates, if this is the first run) the output files LPNORM.OUT and LPNORM.LOG, as well as the LPNORM.LP file. The first two files contain information pertinent to the conditions under which the user ran the program as well as the results generated, and differ only in the amount of detail recorded, with the former being more concise than the latter. LPNORM.LP saves the "input tableau", which is the proper name

in linear programming parlance for the matrix containing the objective function and the set of linear equations, in a format acceptable by the linear programming code LP_SOLVE. This computer program in C-language (M. Berkelaar, Eindhoven University of Technology, The Netherlands) was used to test the robustness of the Simplex subroutines of Press and others (1986) implemented in LPNORM, and therefore the LPNORM.LP file should not be of any use to most LPNORM users. For the purpose of normative analysis, which usually requires a small input tableau of less than 15 rows and less than 30 columns, the Simplex subroutines of Press and others (1986) performed well, after minor modification.

The first task accomplished by LPNORM is to compute the $a_{i,j}$ coefficients from equations similar to Equations (17)–(20), based on the mineral molecular weights and mineral stoichiometric coefficients calculated from mineral formulae (MINLIST.DAT), and the stored oxide molecular weights and oxide stoichiometric coefficients. If the $a_{i,j}$ coefficients are

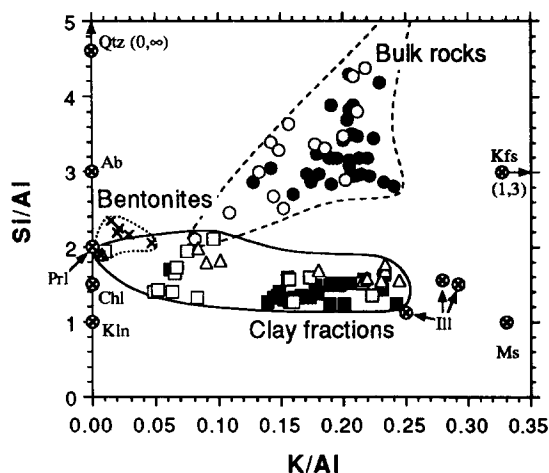


Figure 8. Si/Al versus K/Al diagram showing position of bulk shale samples (open and solid circles), bentonites (crosses), analyzed clay fractions (triangles), and clay fractions calculated with LPNORM (open and solid squares). Open and solid symbols are for immature ($T_{\max} \leq 435^\circ\text{C}$) and mature ($T_{\max} > 435^\circ\text{C}$) samples, respectively. Position of individual minerals quartz (Qtz), albite (Ab), pyrophyllite (Prl), chlorite (Chl), kaolinite (Kln), K-feldspar (Kfs), and muscovite (Ms), as well as the composition of three different illites (Ill) are indicated by crossed circles. See de Caritat and others (1994) for further discussion.

directly read in from the AIJ.DAT file, these data of course are used for subsequent computation.

Next, the objective function is devised from a selection of five options, of which the following three are the most useful for normative analysis: minimize slack variables (option 1), maximize mineral abundances (option 2), or simultaneously minimize slack variables and maximize mineral abundances (option 4). Option 1 should be used first, as it may yield the best results. Depending on which option was selected

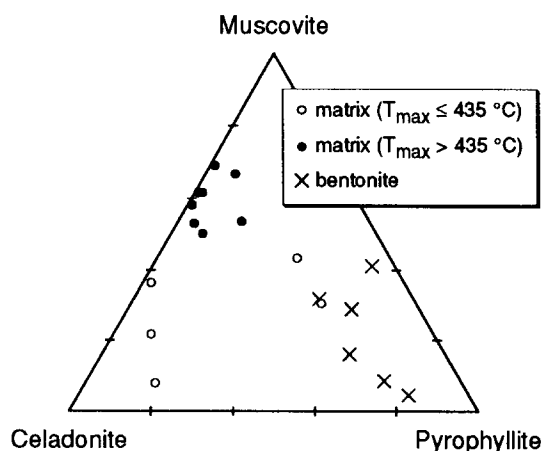


Figure 9. Muscovite, celadonite, pyrophyllite ternary diagram showing calculated compositions of illite/smectite from shale matrix, both immature ($T_{\max} \leq 435^\circ\text{C}$) and mature ($T_{\max} > 435^\circ\text{C}$), and bentonite samples from Belle Fourche Formation. Three celadonitic compositions probably result from overestimation of kaolinite content in those clay fractions, resulting in illite/smectite overly enriched in Fe and Mg. See de Caritat and others (1994) for further discussion.

for the objective function, the user may have the choice of minimizing and/or maximizing the abundance of one or several minerals, and these changes are embedded directly into the objective function. The minimization is effective, mathematically speaking, and results in a zero abundance for the selected mineral(s); it can be used when a sample from a series of bulk rock compositions is encountered that does not contain a mineral otherwise present in the other samples from the same batch. Maximization of one or a few minerals may be used when a given sample is known (e.g. from XRD information) to be dominated by one or a few mineral phases.

The input tableau then is constructed on the basis of the oxide amounts from OXLIST.DAT, and the $a_{i,j}$ coefficients calculated from the MINLIST.DAT data file or read from the AIJ.DAT file. This results in a matrix similar to that of Equations (10)–(14).

Additional constraints, in the form of equality or inequality linear equations, also can be taken into account, and are appended to the input tableau as supplementary rows, one for each new constraint. The number of constraints, and the number of minerals involved in each of them, must be entered, before the constraint(s) can be specified. There are a few syntax rules to be followed. (1) The mineral names must match exactly those saved in the MINLIST.DAT file (if they do not, the user will be asked to type them again). (2) There must be a (free-formatted) coefficient in front of every mineral in each constraint, even if this coefficient is one; this coefficient must include the appropriate sign (although '+' can be omitted, being the default sign). (3) Each coefficient and following mineral name must be separated by a blank space. (4) All coefficient-mineral couples must be on the left-hand side on the constraint; a (free-formatted) number must be on the right-hand side. (5) The operand can be '<', '=', or '>', which is interpreted as ' \leq ', '=', or ' \geq ', respectively. (6) Additionally, mineral names containing mathematical symbols, such as K-feldspar or illite/smectite, which include '-' (minus sign) and '/' (divide sign) respectively, will cause program failure. (7) Finally, mineral names and operands must be written in single quotes (e.g. 1 'quartz' '>' 10). Given these rules, the following constraints, containing 2, 3, and 2 minerals, respectively, are valid:

1 'quartz' + 2.5 'kaolinite' '>' 15.5, meaning

$$1 \times \text{quartz} + 2.5 \times \text{kaolinite} \geq 15.5 \text{ wt\%}, \quad (21)$$

1 'albite' 1 'orthoclase' 1 'anorthite' '=' 45,

$$\text{meaning } 1 \times \text{albite} + 1 \times \text{orthoclase} + 1 \times \text{anorthite} = 45 \text{ wt\%} \quad (22)$$

+ 1 'illite' - 3 'chlorite' '<' 0, meaning

$$1 \times \text{illite} - 3 \times \text{chlorite} \leq 0 \text{ wt\%},$$

meaning $1 \times \text{illite} \leq 3 \times \text{chlorite}$,

$$\text{meaning } \text{illite/chlorite} \leq 3. \quad (23)$$

Conversely, the following two constraints would not be acceptable:

1 'Kfeldspar' '>' 10 'smectite', (right-hand

side can not contain any minerals), (24)

'biotite' '>' 5, (coefficient missing for biotite). (25)

The set of linear equations subject to the objective function now is solved simultaneously by linear programming, using the well-established computational procedure termed the Simplex Method, which was developed in the 1950s by George Dantzig (e.g. see Dantzig, 1956). LPNORM (Appendix 1) uses a modified version of the Simplex algorithm presented in Press and others (1986).

The solution, in terms of wt% of the various minerals, then is returned to the screen and saved in the LPNORM.OUT and LPNORM.LOG files. The results are given, and should only be used, as integers, but for the benefit of the user they also are returned with two decimal places (to allow monitoring of small variations), although these decimal places are not significant in our opinion. The mineral sum and the objective function value are given next. Finally, a verification table indicates how much of the various oxides has been used up, as well as the sum of the unused oxides (slack variables).

Execution of LPNORM is rapid in terms of CPU time, and in real time is limited mainly by its interactive nature. Written in FORTRAN 77, this program was compiled and run on SUN workstations during development, and the distribution version has been modified to compile and execute adequately on any PC, taking advantage of a mathematical coprocessor if available. The source code and example files are available upon submission of a 3 1/2 in. diskette to the first author.

CONCLUSION

The computer code LPNORM performs normative analysis from a bulk chemical analysis and from the composition of contained minerals. This code can be used for any type of mineral mixture, such as soils, concrete, ceramic, sediments, or rocks. LPNORM implements the mathematical method of linear programming, and makes use of the Simplex Method to solve simultaneously the set of linear equations governing the distribution of all listed oxides into these minerals, subject to an objective function and a set of basic linear constraints. The user can specify the list of minerals to be considered for normative analysis, and what their composition is (mineral formulae or geochemical compositions). The objective function can be set to the maximization of mineral abundances, the minimization of slack variables, or the simultaneous maximization of mineral abundances and minimization of slack variables. Independently acquired information about the absolute or relative abundance of one or several of the minerals in the

sample can be entered as additional equality or inequality constraints. A trial-and-error approach enables the user to "optimize" the composition of one or a few of the contained minerals. Results of comparative tests show that LPNORM yields acceptable results in many situations, and represents an improvement upon semiquantitative normative methods such as XRD. Thus, any attempt to quantify mineralogical abundances by optical, image analysis, or XRD methods should be complemented by chemical analysis and linear programming normative analysis. LPNORM is an efficient, quick, and therefore economical, method to obtain first-order quantitative norms from geochemical data, and does not violate the normative "golden rule", namely that the more one knows about a sample and its constituent minerals, the more one can rely on the results from its normative analysis.

Acknowledgments—This study was supported financially by a Natural Sciences and Engineering Research Council of Canada strategic research grant. We wish to thank Maurice Shevalier for numerous rescue operations from crashing SUNs and intergalactic networks. Linear discussions with Terry Gordon enhanced the first author's understanding of optimization. As always, we are indebted to the Dudes of Diagenesis for keeping us on the right track. Colin Harvey and an anonymous referee are thanked for their assistance.

REFERENCES

- Banks, R., 1979, The use of linear programming in the analysis of petrological mixing problems: *Contrib. Mineral. Petrol.*, v. 70, no. 3, p. 237–244.
- Bunday, B. D., 1984, *Basic linear programming*: Edward Arnold Ltd, London, 163 p.
- Caritat, P. de, Bloch, J. D., Hutcheon, I. E., and Longstaffe, F. J., 1994, Compositional trends of a Cretaceous foreland basin shale (Belle Fourche Formation, Western Canada Sedimentary Basin): diagenetic and depositional controls: *Clay Minerals*.
- Cohen, D., and Ward, C. R., 1991, SEDNORM—a program to calculate a normative mineralogy for sedimentary rocks based on chemical analysis. *Computers & Geosciences*, v. 17, no. 9, p. 1235–1253.
- Cross, W., Iddings, J. P., Pirsson, L. V., and Washington, H. S., 1902, A quantitative chemico-mineralogical classification and nomenclature of igneous rocks: *Jour. Geology*, v. 10, no. 6, p. 555–690.
- Dantzig, G. B., 1956, *The Simplex Method*: Rand Corp. Rept., v. P-891, unpaginated.
- Foster, M. D., 1962, Interpretation of the composition and a classification of the chlorites: *U.S. Geol. Survey Prof. Paper*, v. 414-A, 33 p.
- Gass, S. I., 1975, *Linear programming: methods and applications*: McGraw-Hill Book Co., New York, 406 p.
- Ghent, E. D., Nicholls, J., Stout, M. Z., and Rottenfusser, B., 1977, Clinopyroxene amphibolite boudins from Three Valley Gap, British Columbia: *Can. Mineralogist*, v. 15, no. 3, p. 269–282.
- Greenwood, H. J., 1967, The N-dimensional tie-line problem: *Geochim. Cosmochim. Acta*, v. 31, no. 4, p. 465–490.
- Hadley, G. F., 1962, *Linear programming*: Addison-Wesley, Reading, Massachusetts, 520 p.
- Hower, J., and Mowatt, T. C., 1966, The mineralogy of illites and mixed-layer illite/montmorillonites: *Am. Mineral.*, v. 51, no. 5–6, p. 825–854.

- Imbrie, J., and Poldervaart, A., 1959, Mineral compositions from chemical analysis of sedimentary rocks: *Jour. Sed. Pet.*, v. 29, no. 4, p. 588–595.
- Kelsey, C. H., 1965, Calculation of the CIPW norm: *Min. Magazine*, v. 34, p. 276–282.
- Merodio, J. C., Spalletti, L. A., and Bertone, L. M., 1992, A FORTRAN program for the calculation of normative composition of clay minerals and pelitic rocks: *Computers & Geosciences*, v. 18, no. 1, p. 47–61.
- Miesch, A. T., 1962, Computing mineral compositions of sedimentary rocks from chemical analyses: *Jour. Sed. Pet.*, v. 32, no. 2, p. 217–225.
- Nicholls, G. D., 1962, A scheme for recalculating the chemical analysis of argillaceous rocks for comparative purposes: *Am. Mineralogist*, v. 47, no. 1–2, p. 34–46.
- Nicholls, J., and Stout, M. Z., 1986, Electron beam analytical instruments and the determination of modes, spatial variations of minerals and textural features of rocks in polished sections: *Contrib. Mineral. Petrol.*, v. 94, no. 3, p. 395–404.
- Nicholls, J., Stout, M. Z., and Ghent, E. D., 1991, Characterization of partly-open-system chemical variations in clinopyroxene amphibolite boudins, Three Valley Gap, British Columbia, using Thompson space calculations: *Can. Mineralogist*, v. 29, no. 4, p. 633–653.
- Pearson, M. J., 1978, Quantitative clay mineralogical analyses from the bulk chemistry of sedimentary rocks: *Clays and Clay Minerals*, v. 26, no. 6, p. 423–433.
- Press, W. H., Flannery, B. P., Teukolsky, S. A., and Vetterling, W. T., 1986, *Numerical recipes: the art of scientific computing*: Cambridge Univ. Press, 818 p.
- Ross, C. S., and Hendricks, S. B., 1945, Minerals of the montmorillonite group, their origin and relation to soils and clays: U.S. Geol. Survey Prof. Paper, v. 205-B, p. 23–72.
- Slaughter, M., 1989, Quantitative determination of clays and other minerals in rocks, in Pevear, D. R., and Mumpton, F. A., eds., *Quantitative mineral analysis of clays: The Clay Minerals Society Workshop Lecture Series*, v. 1, p. 120–151.
- Weaver, C. E., and Pollard, L. D., 1973, *The chemistry of clay minerals. Developments in Sedimentology Series*, v. 15: Elsevier, New York, 213 p.
- Wright, T. L., and Doherty, P. C., 1970, A linear programming and least squares computer method for solving petrological mixing problems: *Geol. Soc. America Bull.*, v. 81, no. 7, p. 1995–2008.

APPENDIX 1

Program Listing

Note: the following listing contains copyrighted subroutines (SIMPLX, SIMP1, SIMP2, SIMP3) from Press and others (1986) reproduced herein by permission.*

```

      program lpnorm
      =====
      c
      c
      implicit      none
      integer      MMAX
      parameter     (MMAX=100)
      integer      stox(12),nel(30),mi,ne,oxmax,mimax,iflag1
&                ,iflag2,i,j,k,l,n,icount,icol,irow,lmin,ICASE
&                ,jsam,m1,m2,m3,NP,MP,m,ncon,nc,iabun(30),isumin
&                ,dmi,di,iflag3,iflag4,dmimax
      real         oxide(12),mwox(12),verify(12),diff(12)
&                ,EPS,fac(30),sumin,totdiff
      real*8       stoich(20,30),st,mw,molw(30),sumox,aij(30,12)
&                ,aijsum(30),abun(30),chbal
      real         A(MMAX,MMAX)
      integer      IZROV(MMAX),IPOSV(MMAX)
      character*1  sign,answer1,answer2,answer3
      character*2  symbol(20,30),sy,element(12),sign2
      character*10 mineral(30),mincon(30),minmin,minmax,name(100)
      character*58 sample(100),heading
      c
      data mwox / 60.08,101.96,159.69,40.30,61.98,56.08,94.20
&              ,79.90,141.94,70.94,44.01,32.06 /
      data element / 'Si','Al','Fe','Mg','Na','Ca','K','Ti','P'
&                  , 'Mn','C','S' /
      data stox / 1,2,2,1,2,1,2,1,2,1,1,1 /
      oxmax=12
      c
      open (7,file='oxlist.dat',status='old',err=2)
      open (8,file='minlist.dat',status='old',err=4)
      open (10,file='lpnorm.out',status='unknown')
      open (11,file='lpnorm.lp',status='unknown')
      open (12,file='aij.dat',status='unknown')
      open (20,file='lpnorm.log',status='unknown')
      goto 6
      c

```

*Copyright © 1986, 1992 Numerical Recipes Software. Reproduced by permission, from the book *Numerical Recipes: The Art of Scientific Computing*, published by Cambridge University Press.

```

2      continue
      write (*,*)
      write (*,*) '*** Cannot open OXLIST.DAT file'
      goto 100
4      continue
      write (*,*)
      write (*,*) '*** Cannot open MINLIST.DAT file'
      goto 100
6      continue
c
      rewind 7
      rewind 8
      rewind 10
      rewind 11
      rewind 12
      rewind 20
c
      write (*,504)
      write (10,506)
      write (20,506)
c
c *****
c read 'minlist.dat' file
      read (8,*)
c
      mi=1
      write (10,*) '*** Computing mineral data'
      write (20,*) '*** Computing mineral data'
      write (10,508)
      write (20,508)
8      continue
      mimax=mi-1
      read (8,500,end=12) mineral(mi),
&(symbol(ne,mi),stoich(ne,mi),ne=1,20)
c
      do ne=20,1,-1
        if (symbol(ne,mi).eq.' ') then
          nel(mi)=ne-1
        endif
      end do
c
c check if any oxide occurs more than once in any mineral
c formula; if so, add up the stoichiometries for those oxides
      iflag1=0
      do ne=1,nel(mi)
        do j=1,nel(mi)-ne
          if (symbol(ne,mi).eq.symbol(ne+j,mi).and.
&          stoich(ne+j,mi).ne.0) then
            stoich(ne,mi)=stoich(ne,mi)+stoich(ne+j,mi)
            stoich(ne+j,mi)=0.0
            iflag1=1
          endif
        end do
      end do
c
c if the previous loop was accessed, we must clean up the slate
      if (iflag1.eq.1) then
        icount=0
        do ne=1,nel(mi)
          if (stoich(ne,mi).eq.0.0) then
            icount=icount+1
          endif
        end do
c
        do ne=1,nel(mi)-icount
          k=1
10         if (stoich(ne,mi).eq.0.and.stoich(ne+k,mi).ne.0) then
            stoich(ne,mi)=stoich(ne+k,mi)
            symbol(ne,mi)=symbol(ne+k,mi)

```



```

        stoich(ne+k,mi)=0
      endif
      k=k+1
      if (ne+k.le.nel(mi)) goto 10
    end do
    nel(mi)=nel(mi)-icount
  endif
C
  mw=0
  chbal=0.
  do ne=1,nel(mi)
    sy=symbol(ne,mi)
    st=stoich(ne,mi)
C
C
C
    CALL MOLWEIGHT (sy,st,mw,chbal)
C
C
C
    if (mw.eq.0) then
      write (*,*) '*** molecular weight not computed'
      write (*,*) '*** charge balance not computed'
      write (10,*) '*** molecular weight not computed'
      write (10,*) '*** charge balance not computed'
      write (20,*) '*** molecular weight not computed'
      write (20,*) '*** charge balance not computed'
      goto 100
    endif
  end do
  molw(mi)=mw
C
  write (10,510) mi,mineral(mi),molw(mi),nel(mi),chbal,
&(symbol(ne,mi),stoich(ne,mi),ne=1,nel(mi))
  write (20,510) mi,mineral(mi),molw(mi),nel(mi),chbal,
&(symbol(ne,mi),stoich(ne,mi),ne=1,nel(mi))
C
  mi=mi+1
  goto 8
12  continue
C
  write (10,512)
  write (20,512)
C
C *****
C calculate the stoichiometric coefficients a(i,j)
14  continue
    write (*,520)
    read (*,'(a1)') answer1
C
    if ((answer1.eq.'a').or.(answer1.eq.'A').or.
& (answer1.eq.'b').or.(answer1.eq.'B').or.
& (answer1.eq.'c').or.(answer1.eq.'C').or.
& (answer1.eq.'d').or.(answer1.eq.'D')) then
      write (20,522) answer1
      write (20,*)
    endif
C
C if answer1 is 'a' ~~~~~~
C
    if ((answer1.eq.'a').or.(answer1.eq.'A')) then
      write (20,*) '*** Computing a(i,j) coefficients'
      do mi=1,mimax
        aijsum(mi)=0
        do ne=1,nel(mi)
          do i=1,oxmax
            if (symbol(ne,mi).eq.element(i)) then
              aij(mi,i)=(mwox(i)*stoich(ne,mi))/(molw(mi)*stox(i))
              if (element(i).eq.'S') then
                write (20,534) element(i),mineral(mi),aij(mi,i)
              endif
            endif
          end do
        end do
      end do
    endif
  end if

```



```

      aijsum(mi)=0
      iflag4=0
      do ne=1,nel(mi)
        do i=1,omax
          if (symbol(ne,mi).eq.element(i)) then
            aij(mi,i)=(mwox(i)*stoich(ne,mi))/(molw(mi)*stox(i))
            if ((iflag3.eq.0).and.(iflag4.eq.0)) then
              if (element(i).eq.'S') then
                write (*,530) element(i),mineral(mi),aij(mi,i)
              else
                write (*,528) element(i),mineral(mi),aij(mi,i)
              endif
              write (*,524)
              read (*,'(a1)') answer2
            endif
          c
          c if answer2 is not 'yes', or 'no', or 'end', or 'next'
          c
          if ((answer2.ne.'y').and.(answer2.ne.'Y').and.
            & (answer2.ne.'n').and.(answer2.ne.'N').and.
            & (answer2.ne.'e').and.(answer2.ne.'E').and.
            & (answer2.ne.'x').and.(answer2.ne.'X')) then
            answer2='n'
          endif
          c
          c if answer2 is 'yes'
          c
          if ((answer2.eq.'y').or.(answer2.eq.'Y')) then
            write (*,526)
            read (*,*) aij(mi,i)
            if (element(i).eq.'S') then
              write (20,530) element(i),mineral(mi),aij(mi,i)
            else
              write (20,528) element(i),mineral(mi),aij(mi,i)
            endif
            write (20,*) ' (User-specified value)'
            if (element(i).eq.'S') then
              write (12,538) i,mi,element(i),mineral(mi)
            & ,aij(mi,i)
            &
            else
              write (12,542) i,mi,element(i),mineral(mi)
            & ,aij(mi,i)
            &
            endif
          c
          c if answer2 is 'no'
          c
          else if ((answer2.eq.'n').or.(answer2.eq.'N')) then
            if (element(i).eq.'S') then
              write (20,534) element(i),mineral(mi),aij(mi,i)
            else
              write (20,532) element(i),mineral(mi),aij(mi,i)
            endif
            if (element(i).eq.'S') then
              write (12,538) i,mi,element(i),mineral(mi)
            & ,aij(mi,i)
            &
            else
              write (12,542) i,mi,element(i),mineral(mi)
            & ,aij(mi,i)
            &
            endif
          c
          c if answer2 is 'end'
          c
          else if ((answer2.eq.'e').or.(answer2.eq.'E')) then
            if (iflag3.eq.1) then
              if (element(i).eq.'S') then
                write (*,534) element(i),mineral(mi)
            & ,aij(mi,i)
                else
                  write (*,532) element(i),mineral(mi)
            & ,aij(mi,i)
                endif
            &
            endif
          c

```

```

endif
  if (element(i).eq.'S') then
    write (20,534) element(i),mineral(mi)
    ,aij(mi,i)
  &
  else
    write (20,532) element(i),mineral(mi)
    ,aij(mi,i)
  &
  endif
  if (element(i).eq.'S') then
    write (12,544) i,mi,element(i),mineral(mi)
    ,aij(mi,i)
  &
  else
    write (12,542) i,mi,element(i),mineral(mi)
    ,aij(mi,i)
  &
  endif
  iflag3=1
c
c if answer2 is 'next'
c
  else if ((answer2.eq.'x').or.(answer2.eq.'X')) then
    if (iflag4.eq.1) then
      if (element(i).eq.'S') then
        write (*,534) element(i),mineral(mi),aij(mi,i)
      else
        write (*,532) element(i),mineral(mi),aij(mi,i)
      endif
    endif
    if (element(i).eq.'S') then
      write (20,534) element(i),mineral(mi),aij(mi,i)
    else
      write (20,532) element(i),mineral(mi),aij(mi,i)
    endif
    if (element(i).eq.'S') then
      write (12,544) i,mi,element(i),mineral(mi)
      ,aij(mi,i)
    &
    else
      write (12,542) i,mi,element(i),mineral(mi)
      ,aij(mi,i)
    &
    endif
    iflag4=1
  endif
  aijsum(mi)=aijsum(mi)+aij(mi,i)
endif
end do
end do
write (*,546) aijsum(mi)
write (20,546) aijsum(mi)
end do
goto 22
endif
c
c if answer1 is 'd' ^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
c
  if ((answer1.eq.'d').or.(answer1.eq.'D')) then
    goto 100
  endif
c
c if answer1 is none of the above (a,b,c,d) ^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
c
  if ((answer1.ne.'a').or.(answer1.ne.'A').or.
    & (answer1.ne.'b').or.(answer1.ne.'B').or.
    & (answer1.ne.'c').or.(answer1.ne.'C').or.
    & (answer1.ne.'d').or.(answer1.ne.'D')) then
    goto 14
  endif
c
22 continue
c
c *****
c read 'oxlist.dat' file
  read (7,*)

```

```

jsam=1
c
24 continue
c
  read (7,548,end=100) sample(jsam)
  read (7,*,end=100) (oxide(i),i=1,oxmax)
c
  write (*,502) (mineral(n),n=1,mimax)
  write (10,502) (mineral(n),n=1,mimax)
  write (20,502) (mineral(n),n=1,mimax)
c
  do i=1,oxmax
    sumox=sumox+oxide(i)
  end do
c
  write (*,550) sample(jsam)
  write (10,550) sample(jsam)
  write (20,550) sample(jsam)
  write (*,554) (oxide(i),i=1,oxmax),sumox
  write (10,554) (oxide(i),i=1,oxmax),sumox
  write (20,554) (oxide(i),i=1,oxmax),sumox
c
c *****
c prepare the initial tableau
  N=mimax+oxmax
  M1=0
  M2=0
  M3=oxmax
  M=M1+M2+M3
  NP=100
  MP=100
26 continue
c
c reset matrix to zero
  do i=1,MP
    IZROV(i)=0
    IPOSV(i)=0
    do j=1,NP
      A(i,j)=0
    end do
  end do
c
c ***** OBJECTIVE FUNCTION *****
c
c write the objective function (z=a01x1+a02x2+...+a0NxN);
c options are
c 1) to minimize the sum of the slack variables,
c 2) to maximize the mineral abundances,
c 3) to minimize the mineral abundances,
c 4) to do both 1) and 2) together, or
c 5) to do both 1) and 3) together
c
28 continue
  write (*,568)
  read (*,570,err=28) iflag2
  if (iflag2.eq.0) then
    goto 100
  else if (iflag2.eq.9) then
    goto 52
  else if (iflag2.eq.1) then
    write (10,*)
    write (20,*)
    write (10,*) '*** Obj.fct.: minimize slack variables'
    write (20,*) '*** Obj.fct.: minimize slack variables'
    do icol=2,mimax+1
      A(1,icol)=0
    end do
    do icol=mimax+2,mimax+oxmax+1
      A(1,icol)=-1
    end do
  else if (iflag2.eq.2) then

```

```

write (10,*)
write (20,*)
write (10,*) '*** Obj.fct.: maximize mineral abundances'
write (20,*) '*** Obj.fct.: maximize mineral abundances'
do icol=2,mimax+1
  A(1,icol)=1
end do
do icol=mimax+2,mimax+oxmax+1
  A(1,icol)=0
end do
else if (iflag2.eq.3) then
write (10,*)
write (20,*)
write (10,*) '*** Obj.fct.: minimize mineral abundances'
write (20,*) '*** Obj.fct.: minimize mineral abundances'
do icol=2,mimax+1
  A(1,icol)=-1
end do
do icol=mimax+2,mimax+oxmax+1
  A(1,icol)=0
end do
else if (iflag2.eq.4) then
write (10,*)
write (20,*)
write (10,*) '*** Obj.fct.: minimize slack variables AND
&'maximize mineral abundances'
write (20,*) '*** Obj.fct.: minimize slack variables AND
&'maximize mineral abundances'
do icol=2,mimax+1
  A(1,icol)=1
end do
do icol=mimax+2,mimax+oxmax+1
  A(1,icol)=-1
end do
else if (iflag2.eq.5) then
write (10,*)
write (20,*)
write (10,*) '*** Obj.fct.: minimize slack variables AND
&'minimize mineral abundances'
write (20,*) '*** Obj.fct.: minimize slack variables AND
&'minimize mineral abundances'
do icol=2,mimax+1
  A(1,icol)=-1
end do
do icol=mimax+2,mimax+oxmax+1
  A(1,icol)=-1
end do
else
  goto 28
endif
endif
c
c now, the '=' constraints (oxmax of them: one for each oxide)
c rows 2->oxmax+1: constraints
do irow=2,oxmax+1
  A(irow,1)=oxide(irow-1)
  do mi=2,mimax+1
    A(irow,mi)=-aij(mi-1,irow-1)
  end do
  do j=mimax+2,mimax+oxmax+1
    A(irow,j)=0
    if (irow.ge.2) then
      A(irow,mimax+irow)=-1
    endif
  end do
end do
c
c allow to minimize and/or maximize the abundance of a mineral
c (compatible with the type of objective function chosen)
if ((iflag2.eq.1).or.(iflag2.eq.2).or.(iflag2.eq.4)) then
30  continue

```

```

write (*,606)
read (*,608,err=30) minmin
if (minmin.eq.' ') goto 32
do k=1,30
  if (minmin.eq.mineral(k)) then
    write (10,610) minmin
    write (20,610) minmin
    A(1,k+1)=-1
    goto 30
  endif
end do
write (*,612) minmin
goto 30
endif
32 continue
if ((iflag2.eq.1).or.(iflag2.eq.3).or.(iflag2.eq.5)) then
34 continue
write (*,614)
read (*,616,err=34) minmax
if (minmax.eq.' ') goto 36
do k=1,30
  if (minmax.eq.mineral(k)) then
    write (10,618) minmax
    write (20,618) minmax
    A(1,k+1)=1
    goto 34
  endif
end do
write (*,620) minmax
goto 34
endif
36 continue
c
c ***** ADDITIONAL CONSTRAINTS *****
c
c allow for additional constraints (interactive)
c
38 continue
ncon=0
write (*,602)
read (*,'(i2)',err=38) ncon
if (ncon.eq.0) goto 46
if (ncon.le.-1) goto 38
do i=oxmax+2,oxmax+2+ncon-1
  do j=1,mimax+oxmax+1+ncon
    A(i,j)=0
  end do
  nc=i-oxmax-1
40 continue
write (*,628) nc
read (*,'(i2)',err=40) lmin
if (lmin.le.0) goto 40
42 continue
write (*,604) nc
read (*,*,err=42) (fac(l),mincon(l),l=1,lmin),sign,A(i,1)
do l=1,lmin
  do k=1,30
    if (mincon(l).eq.mineral(k)) then
      A(i,k+1)=-fac(l)
      goto 44
    endif
  end do
  write (*,*) 'Mineral name not recognized; try again'
  goto 42
44 continue
end do
c
if (sign.eq.'<') then
  sign2='=<'
  A(i,mimax+i)=-1

```

```

        else if (sign.eq.'=') then
            sign2=='=='
            A(i,mimax+i)=0
        else if (sign.eq.'>') then
            sign2=='>='
            A(i,mimax+i)=1
        else
            goto 100
        endif
        write (*,*) ('+',fac(1),'*',mincon(1),l=1,lmin),sign2,A(i,1),
&' wt%'
        write (10,*) '*** Constraint: ',
&('+',fac(1),'*',mincon(1),l=1,lmin),sign2,A(i,1),' wt%'
        write (20,*) '*** Constraint: ',
&('+',fac(1),'*',mincon(1),l=1,lmin),sign2,A(i,1),' wt%'
        end do
c
46      continue
c
c fill in area right of the original matrix with zeros
      do i=1,oxmax+1
          do j=mimax+oxmax+2,mimax+oxmax+1+ncon
              A(i,j)=0
          end do
      end do
c
c write initial tableau to log file ('lpnorm.log')
      write (20,*)
      write (20,572) (mineral(mi),mi=1,mimax),(element(i),i=1,oxmax)
      write (20,574) A(1,1),(A(1,j),j=2,mimax+oxmax+1+ncon)
      do i=2,oxmax+1
          write (20,576) element(i-1),A(i,1),
&          (A(i,j),j=2,mimax+oxmax+1+ncon)
      end do
      do i=oxmax+2,oxmax+1+ncon
          write (20,574) A(i,1),(A(i,j),j=2,mimax+oxmax+1+ncon)
      end do
c
c write initial tableau to 'lpnorm.lp' file (to be used with lp_solve
c if one wishes to check the accuracy of the linear programming results)
      do i=1,mimax
          name(i)=mineral(i)
      end do
      do i=mimax+1,mimax+oxmax
          name(i)=element(i-mimax)
      end do
      write (11,578) (A(1,j),name(j-1),j=2,mimax+oxmax+1)
      write (11,580)
      do i=2,oxmax+1
          write (11,582) (-A(i,j),name(j-1),j=2,mimax+oxmax+1)
          write (11,584) A(i,1)
      end do
      write (11,586) (name(j),j=1,mimax)
      write (11,580)
      write (11,588) (name(j),j=mimax+1,mimax+oxmax)
      write (11,580)
      do i=oxmax+2,oxmax+2+ncon-1
          write (11,622) fac(1),mincon(1)
          if (lmin.gt.1) then
              write (11,624) (fac(1),mincon(1),l=2,lmin)
          endif
          write (11,626) sign,A(i,1)
          write (11,580)
      end do
c
c call up the subroutine implementing the Simplex method (Copyright:
c Cambridge University Press. Source: Numerical Recipes, the Art of
c Scientific Computing; by Press et al.)
      M3=M3+ncon
      M=M1+M2+M3
      N=N+ncon

```



```

EPS=1.E-6
ICASE=2
write (20,*)
write (20,*) '*** Initial EPS value is: ',EPS
c
c
c
CALL SIMPLX(A,M,N,NP,M1,M2,M3,ICASE,IZROV,IPOSV,EPS)
c
c
c
write (*,*)
write (20,*)
if (ICASE.eq.0) then
  write (*,*) '*** A finite solution was found for sample ',
& sample(jsam)
  write (10,*) '*** A finite solution was found for sample ',
& sample(jsam)
  write (20,*) '*** A finite solution was found for sample ',
& sample(jsam)
else if (ICASE.eq.1) then
  write (*,*) '*** The objective function is ',
& 'unbounded for sample ',sample(jsam)
  write (10,*) '*** The objective function is ',
& 'unbounded for sample ',sample(jsam)
  write (20,*) '*** The objective function is ',
& 'unbounded for sample ',sample(jsam)
else if (ICASE.eq.-1) then
  write (*,*) '*** No solution satisfies the given ',
& 'constraints for sample ',sample(jsam)
  write (10,*) '*** No solution satisfies the given ',
& 'constraints for sample ',sample(jsam)
  write (20,*) '*** No solution satisfies the given ',
& 'constraints for sample ',sample(jsam)
else
  write (*,*) 'ICASE error'
  write (10,*) 'ICASE error'
  write (20,*) 'ICASE error'
  goto 100
endif
write (*,*)
write (20,*)
c
c output of linear optimization
if (ICASE.ne.0) goto 48
sumin=0
write (10,*)
do k=1,mimax
  abun(k)=0
end do
do i=2,oxmax+1
  k=IPOSV(i-1)
  if (k.le.mimax) then
    abun(k)=A(i,1)
    iabun(k)=int(abun(k)+0.50)
    write (*,558) iabun(k),abun(k),mineral(k)
    write (10,558) iabun(k),abun(k),mineral(k)
    write (20,558) iabun(k),abun(k),mineral(k)
    sumin=sumin+abun(k)
    isumin=int(sumin+0.50)
  endif
end do
c
c include minerals from additional constraints
if (ncon.gt.0) then
  do i=oxmax+2,oxmax+1+ncon
    k=IPOSV(i-1)
    abun(k)=A(i,1)
    iabun(k)=int(abun(k)+0.50)
    write (*,560) iabun(k),abun(k),mineral(k)
  end do

```

```

        write (10,560) iabun(k),abun(k),mineral(k)
        write (20,560) iabun(k),abun(k),mineral(k)
        sumin=sumin+abun(k)
        isumin=int(sumin+0.50)
    end do
endif
write (*,562) isumin,sumin,A(1,1)
write (10,564) isumin,sumin,A(1,1)
write (20,564) isumin,sumin,A(1,1)
c
c To print the output tableau to log file, de-comment the following 4 line
c   write (20,514) (IZROV(J),J=1,M)
c   do i=1,oxmax+1+ncon
c       write (20,566) IPOSV(i-1),(A(i,j),j=1,mimax+oxmax+1+ncon)
c   end do
c
c verify output
write (10,*)
write (20,*)
write (*,590)
write (10,590)
write (20,590)
totdiff=0
do i=1,oxmax
    verify(i)=0
    do j=1,mimax
        if (aij(j,i).ne.0.and.abun(j).ne.0) then
            verify(i)=verify(i)+aij(j,i)*abun(j)
        endif
    end do
    diff(i)=oxide(i)-verify(i)
    totdiff=totdiff+diff(i)
    write (*,592) element(i),verify(i),oxide(i),diff(i)
    write (10,592) element(i),verify(i),oxide(i),diff(i)
    write (20,592) element(i),verify(i),oxide(i),diff(i)
end do
write (*,594) totdiff
write (10,594) totdiff
write (20,594) totdiff
c
48    continue
c
c rerun this sample under different conditions?
50    continue
    write (*,596)
    read (*,598,err=50) answer3
    if ((answer3.eq.'y').or.(answer3.eq.'Y')) then
        write (20,*)
        write (20,600)
        goto 26
    else if ((answer3.eq.'n').or.(answer3.eq.'N')) then
        goto 52
    else
        goto 48
    endif
c
c next sample, if any
52    continue
    jsam=jsam+1
    goto 24
c
c terminate execution
100   continue
c
    write (*,700)
    write (20,702)
c
    close (7)
    close (8)
    close (10)

```

```

close (11)
close (12)
close (20)

c
  stop

c
c formats
500  format (a10,1x,20(a2,f5.2,.,3x))
502  format (/,80('='),/,/,', MINERALS: ',30(a10,.,1x))
504  format (5(/),2(80('='),/,),2(/),32x,'Program LPNORM',3(/),21x,
&'Copyright 1992,1993 Patrice de Caritat',3(/),2(80('='),/))
506  format (80('='),/,
&'Program LPNORM - University of Calgary - ',
&'Copyright 1992,1993 Patrice de Caritat',/,
&7x,'[Ref: Caritat, Bloch and Hutcheon, Computers ',
&'& Geosciences, 1993]',/,80('='),/)
508  format (1x,'#',1x,'MINERAL',48x,'MOL.WT.',1x,
&'#EL.',1x,'*CH.BAL.',/,80('_'),/)
510  format (i2,':',a10,1x,43(' '),f7.2,1x,i3,3x,f8.4,/,
&5x,20(a2,f5.2,.,3x))
512  format (/,
&'(*CH.BAL. based on following valences: -2:O -1:F,S,Cl,OH ',
&' +1:H,Na,K',/,10x,' +2:Mg,Ca,Mn,Fe,Ba +3:Al +4:C,Si,Ti ',
&' +5:N,P)',/,80('_'))
514  format (/,30x,'OUTPUT TABLEAU',/,13x,100(i2,.,4x))
516  format (1x,a2,' in ',a10,' = ',f8.6)
518  format (1x,a2,' oxide in ',a10,' = ',f8.6)
520  format (/, ' Select option (a), (b), (c) or (d)',/,
&4x,'a: calculate a(i,j) coefficients from MINLIST.DAT file',/,
&4x,'b: read a(i,j) coefficients from AIJ.DAT file',/,
&4x,'c: enter a(i,j) coefficients from keyboard',/,
&4x,'d: quit LPNORM now',/,
&4x,'Your choice... ', $)
522  format (/, ' Select option (a), (b), (c) or (d)',/,
&4x,'a: calculate a(i,j) coefficients from MINLIST.DAT file',/,
&4x,'b: read a(i,j) coefficients from AIJ.DAT file',/,
&4x,'c: enter a(i,j) coefficients from keyboard',/,
&4x,'d: quit LPNORM now',/,
&4x,'Your choice... ',a1)
524  format (' ...change value? (y/n/e/x): ', $)
526  format (8x,' Enter new value: ', $)
528  format (1x,a2,' oxide in ',a10,' = ',f8.6,$)
530  format (1x,a2,' in ',a10,' = ',f8.6,$)
532  format (1x,a2,' oxide in ',a10,' = ',f8.6)
534  format (1x,a2,' in ',a10,' = ',f8.6)
536  format (/, ' Reading a(i,j) coeff. from AIJ.DAT file for ',
&'sample: ',/,1x,a58)
538  format (i3,1x,i3,1x,a2,' in ',a10,' = ',f8.6)
540  format (i3,1x,i3,1x,a2,10x,a10,3x,f8.6)
542  format (i3,1x,i3,1x,a2,' oxide in ',a10,' = ',f8.6)
544  format (i3,1x,i3,1x,a2,' in ',a10,' = ',f8.6)
546  format (19x,' SUM = ',f8.6,'____')
548  format (a58)
550  format (/, ' SAMPLE ',a58)
552  format (a58)
554  format (1x,' SiO2 Al2O3 Fe2O3 MgO Na2O CaO K2O TiO2 ',
&'P2O5 MnO inCO2 S SUM',/,1x,12(f5.2,1x),f6.2)
556  format (22x,a58)
558  format (1x,i3,' (',f6.2,')',', wt% ',a10)
560  format (1x,i3,' (',f6.2,')',', wt% ',a10,' (constraint)')
562  format (30('-'),/,1x,i3,' (',f6.2,')',', wt% total ',/,8x,f6.2,
&' wt% objective function',/,)
564  format (30('-'),/,1x,i3,' (',f6.2,')',', wt% total ',/,8x,f6.2,
&' wt% objective function')
566  format ('x',i2,':',100(f6.2,:))
568  format (/, ' Select type of objective function (option 1-5) ',
&'(0: exit; 9: skip this sample)',/,
&4x,'1: minimize slack variables',/,
&4x,'2: maximize mineral abundances',/,
&4x,'3: minimize mineral abundances',/,
&4x,'4: minimize slack variables AND maximize mineral abundances',/,

```

```

&4x,'5: minimize slack variables AND minimize mineral abundances',/
&4x,'Your choice... ', $)
570 format (i1)
572 format (30x, 'INPUT TABLEAU',/, 8x, 30(1x, a4, :), 12(a2, 3x))
574 format (2x, f6.2, 100(f5.2, :))
576 format (a2, f6.2, 100(f5.2, :))
578 format (100(f9.6, 1x, a10, :, ' + '), $)
580 format (';')
582 format (100(f9.6, 1x, a10, :, ' + '), $)
584 format (' = ', f9.6, ';')
586 format ('SUM_MINERALS = ', 100(a10, :, ' + '), $)
588 format ('SUM_SLACK = ', 100(a10, :, ' + '), $)
590 format (/, 2x, 30('-'), ' VERIFY OUTPUT ', 31('-'),/, 1x, '|', 76x, '|')
592 format (1x, '|', 3x, a2, ': ', f6.2, ' wt% used of the initial ',
&f6.2, ' wt%', ' (diff.= ', f6.2, ' wt%)', 5x, '|')
594 format (1x, '|', 76x, '|',/, 1x, '|', 47x, 'total diff.= ',
&f6.2, ' wt%', 6x, '|',/, 1x, '|', 76x, '|',/, 2x, 76('-'),/,/)
596 format (/, ' Do you want to rerun this sample (y/n)? ', $)
598 format (a1)
600 format (80('='))
602 format (/, ' How many constraints do you want to add? ', $)
604 format (' Enter constraint ', i2, ': ', $)
606 format (/, ' *** Minimize the abundance of... ', $)
608 format (a10)
610 format (/, '*** Minimize the abundance of... ', a10)
612 format (' Unable to identify mineral ', a10, ' ... try again')
614 format (' *** Maximize the abundance of... ', $)
616 format (a10)
618 format ('*** Maximize the abundance of... ', a10)
620 format (' Unable to identify mineral ', a10, ' ... try again')
622 format (f9.6, 1x, a10, $)
624 format (100(:, ' + ', f9.6, 1x, a10), $)
626 format (1x, a1, 1x, f9.6, $)
628 format (' Number of minerals involved in constraint ', i2, ': ', $)
700 format (/,/, 80('='),/,/, 32x, 'Quitting LPNORM',/,/, 80('='),/,/)
702 format (80('='),/, 32x, 'Quitting LPNORM',/, 80('='))
c
      end
c
c -----
c ----- SUBROUTINES -----
c -----
c
      SUBROUTINE MOLWEIGHT (sy, st, mw, chbal)
      implicit none
      integer anum, chge
      real*8 aw(60), st, mw, chbal
      character*2 sy
c
c atomic weights for H,C,N,O,F,Na,Mg,Al,Si,P,S,Cl,
c K,Ca,Ti,Mn,Fe,Ba,OH
      data
&aw(1), aw(6), aw(7), aw(8), aw(9), aw(11)
&,aw(12),aw(13), aw(14),aw(15), aw(16),aw(17)
&,aw(19),aw(20),aw(22),aw(25),aw(26),aw(56),aw(60)/
&1.0079,12.011,14.0067,15.9994,18.9984,22.9898
&,24.305,26.9815,28.086,30.9738,32.06, 35.453
&,39.102,40.08, 47.90, 54.938,55.847,137.34,17.0073/
c
c Molecular weight computation (assuming Fe = +2 & S = -1)
c
      if (sy.eq.'H') then
        anum=1
      else if (sy.eq.'C') then
        anum=6
      else if (sy.eq.'N') then
        anum=7
      else if (sy.eq.'O') then
        anum=8
      else if (sy.eq.'F') then
        anum=9

```

```

else if (sy.eq.'Na') then
  anum=11
else if (sy.eq.'Mg') then
  anum=12
else if (sy.eq.'Al') then
  anum=13
else if (sy.eq.'Si') then
  anum=14
else if (sy.eq.'P') then
  anum=15
else if (sy.eq.'S') then
  anum=16
else if (sy.eq.'Cl') then
  anum=17
else if (sy.eq.'K') then
  anum=19
else if (sy.eq.'Ca') then
  anum=20
else if (sy.eq.'Ti') then
  anum=22
else if (sy.eq.'Mn') then
  anum=25
else if (sy.eq.'Fe') then
  anum=26
else if (sy.eq.'Ba') then
  anum=56
else if (sy.eq.'OH') then
  anum=60
else
  write (*,*) '**** syntax error in element ',sy,' '
  write (10,*) '**** syntax error in element ',sy,' '
  write (20,*) '**** syntax error in element ',sy,' '
  mw=0
  return
endif
mw=mw+aw(anum)*st
c
c Charge balance computation (assuming Fe = +2 & S = -1)
c
  if ((sy.eq.'H').or.(sy.eq.'Na').or.(sy.eq.'K')) then
    chge=+1
  else if ((sy.eq.'Mg').or.(sy.eq.'Ca').or.(sy.eq.'Mn').or.
&      (sy.eq.'Fe').or.(sy.eq.'Ba')) then
    chge=+2
  else if ((sy.eq.'Al')) then
    chge=+3
  else if ((sy.eq.'C').or.(sy.eq.'Si').or.(sy.eq.'Ti')) then
    chge=+4
  else if ((sy.eq.'N').or.(sy.eq.'P')) then
    chge=+5
  else if ((sy.eq.'F').or.(sy.eq.'Cl').or.(sy.eq.'S').or.
&      (sy.eq.'OH')) then
    chge=-1
  else if ((sy.eq.'O')) then
    chge=-2
  else
    return
  endif
  chbal=chbal+chge*st
c
c
c
c -----
c
c
c
c
SUBROUTINE SIMPLX(A,M,N,NP,M1,M2,M3,ICASE,IZROV,IPOSV,EPS)
implicit none
integer MMAX
PARAMETER(MMAX=100)
real A(MMAX,MMAX),eps,q1,bmax

```

```

integer      IZROV(MMAX), IPOSV(MMAX), L1(MMAX), L2(MMAX), L3(MMAX)
&epscount,k,nl2,i,ir,kp,ip,M12,is,kh,M,M1,M2,M3,n,NP,ICASE,nl1
write (20,*)
write (20,*) 'M=',M,' M1=',M1,' M2=',M2,' M3=',M3
epscount=0
IF(M.NE.M1+M2+M3)PAUSE 'Bad input constraint counts.'
NL1=N
DO 11 K=1,N
  L1(K)=K
  IZROV(K)=K
11 CONTINUE
NL2=M
DO 12 I=1,M
  IF(A(I+1,1).LT.0.)PAUSE 'Bad input tableau.'
  L2(I)=I
  IPOSV(I)=N+I
12 CONTINUE
DO 13 I=1,M2
  L3(I)=1
13 CONTINUE
IR=0
IF(M2+M3.EQ.0)GO TO 30
IR=1
DO 15 K=1,N+1
  Q1=0.
  DO 14 I=M1+1,M
    Q1=Q1+A(I+1,K)
14 CONTINUE
  A(M+2,K)=-Q1
15 CONTINUE
10 CALL SIMP1(A,M+1,L1,NL1,0,KP,BMAX)
IF(BMAX.LE.EPS.AND.A(M+2,1).LT.-EPS) THEN
c
  EPS=EPS*10
  write (*,*)
  write (*,*) '*** SIMPLX - EPS value increased to: ',EPS
  write (20,*)
  write (20,*) '*** SIMPLX - EPS value increased to: ',EPS
c
  if (EPS.le.1E-3) goto 161
c
  ICASE=-1
  RETURN
161 continue
ELSE IF (BMAX.LE.EPS.AND.A(M+2,1).LE.EPS) THEN
  M12=M1+M2+1
  IF (M12.LE.M) THEN
    DO 16 IP=M12,M
      IF (IPOSV(IP).EQ.IP+N) THEN
        CALL SIMP1(A,IP,L1,NL1,1,KP,BMAX)
        IF (BMAX.GT.0.)GO TO 1
      ENDIF
16 CONTINUE
    ENDIF
    IR=0
    M12=M12-1
    IF (M1+1.GT.M12)GO TO 30
    DO 18 I=M1+1,M12
      IF (L3(I-M1).EQ.1) THEN
        DO 17 K=1,N+1
          A(I+1,K)=-A(I+1,K)
17 CONTINUE
        ENDIF
18 CONTINUE
      GO TO 30
    ENDIF
  CALL SIMP2(A,N,NP,L2,NL2,IP,KP,Q1,EPS,epscount)
  IF (IP.EQ.0) THEN
    ICASE=-1
    RETURN
  ENDIF

```

```

1      CALL SIMP3(A,M+1,N,IP,KP)
      IF(IPOSV(IP).GE.N+M1+M2+1) THEN
          DO 19 K=1,NL1
              IF(L1(K).EQ.KP)GO TO 2
19         CONTINUE
2         NL1=NL1-1
          DO 21 IS=K,NL1
              L1(IS)=L1(IS+1)
21        CONTINUE
      ELSE
          IF(IPOSV(IP).LT.N+M1+1)GO TO 20
          KH=IPOSV(IP)-M1-N
          IF(L3(KH).EQ.0)GO TO 20
          L3(KH)=0
      ENDIF
      A(M+2,KP+1)=A(M+2,KP+1)+1.
      DO 22 I=1,M+2
          A(I,KP+1)=-A(I,KP+1)
22        CONTINUE
20        IS=IZROV(KP)
          IZROV(KP)=IPOSV(IP)
          IPOSV(IP)=IS
          IF(IR.NE.0)GO TO 10
30        CALL SIMP1(A,0,L1,NL1,0,KP,BMAX)
          IF(BMAX.LE.0.) THEN
              ICASE=0
              RETURN
          ENDIF
      CALL SIMP2(A,M,N,MP,NP,L2,NL2,IP,KP,Q1,EPS,epscount)
      CALL SIMP2(A,N,NP,L2,NL2,IP,KP,Q1,EPS,epscount)
      IF(IP.EQ.0) THEN
          ICASE=1
          RETURN
      ENDIF
      CALL SIMP3(A,M,N,IP,KP)
      GO TO 20
      END
c
c
c -----
c
c
      SUBROUTINE SIMP1(A,MM,LL,NLL,IABF,KP,BMAX)
      implicit none
      integer MMAX
      PARAMETER(MMAX=100)
      real A(MMAX,MMAX),TEST,bmax
      integer LL(MMAX),nll,iabf,kp,k,MM
      KP=LL(1)
      BMAX=A(MM+1,KP+1)
      DO 11 K=2,NLL
          IF(IABF.EQ.0) THEN
              TEST=A(MM+1,LL(K)+1)-BMAX
          ELSE
              TEST=ABS(A(MM+1,LL(K)+1))-ABS(BMAX)
          ENDIF
          IF(TEST.GT.0.) THEN
              BMAX=A(MM+1,LL(K)+1)
              KP=LL(K)
          ENDIF
11        CONTINUE
      RETURN
      END
c
c
c -----
c
c
      SUBROUTINE SIMP2(A,N,NP,L2,NL2,IP,KP,Q1,EPS,epscount)
      implicit none
      integer MMAX,NP

```

```

PARAMETER(MMAX=100)
real      A(MMAX,NP), q1,q,q0,qp,eps
integer    L2(MMAX),nl2,ip,kp,i,ii,k,N,epscount
IP=0

C
epscount=epscount+1
if (epscount.ge.1000) then
    EPS=EPS*10
    epscount=0
    write (*,*)
    write (*,*) '*** SIMP2 - EPS value increased to: ',EPS
    write (20,*)
    write (20,*) '*** SIMP2 - EPS value increased to: ',EPS
endif

C
DO 11 I=1,NL2
    IF(A(L2(I)+1,KP+1).LT.-EPS)GO TO 1
11 CONTINUE
RETURN
1  Q1=-A(L2(I)+1,1)/A(L2(I)+1,KP+1)
    IP=L2(I)
DO 13 I=I+1,NL2
    II=L2(I)
    IF(A(II+1,KP+1).LT.-EPS)THEN
        Q=-A(II+1,1)/A(II+1,KP+1)
        IF(Q.LT.Q1)THEN
            IP=II
            Q1=Q
        ELSE IF (Q.EQ.Q1) THEN
            DO 12 K=1,N
                QP=-A(IP+1,K+1)/A(IP+1,KP+1)
                Q0=-A(II+1,K+1)/A(II+1,KP+1)
                IF(Q0.NE.QP)GO TO 2
12 CONTINUE
2  IF(Q0.LT.QP) IP=II
            ENDIF
        ENDIF
13 CONTINUE
RETURN
END

C
C
C -----
C
C
SUBROUTINE SIMP3(A, I1,K1,IP,KP)
implicit none
integer    MMAX
PARAMETER(MMAX=100)
real      A(MMAX,MMAX),piv
integer    il,k1,ip,kp,ii,kk
PIV=1./A(IP+1,KP+1)
DO 12 II=1,I1+1
    IF(II-1.NE.IP)THEN
        A(II,KP+1)=A(II,KP+1)*PIV
        DO 11 KK=1,K1+1
            IF(KK-1.NE.KP)THEN
                A(II,KK)=A(II,KK)-A(IP+1,KK)*A(II,KP+1)
            ENDIF
11 CONTINUE
        ENDIF
12 CONTINUE
DO 13 KK=1,K1+1
    IF(KK-1.NE.KP)A(IP+1,KK)=-A(IP+1,KK)*PIV
13 CONTINUE
A(IP+1,KP+1)=PIV
RETURN
END

```


APPENDIX 2

*Mineral and Oxide Input Files: Bersham Mudstone Example**Mineral input file*

```

MINERAL      COMPOSITION Sample Bersham mdst (Nicholls, 1962)
quartz       Si 1.00   O  2.00
ill1         K  2.00   Al 6.00   Si 6.00   O 20.00   OH 4.00
ill2         Na 2.00   Al 6.00   Si 6.00   O 20.00   OH 4.00
ill3         Al 4.00   Si 8.00   O 20.00   OH 4.00
kaolinite    Al 2.00   Si 2.00   O  5.00   OH 4.00
chl1         Fe12.00  Si 8.00   O 20.00   OH16.00
chl2         Mg12.00  Si 8.00   O 20.00   OH16.00
calcite      Ca 1.00   C  1.00   O  3.00
magnesite    Mg 1.00   C  1.00   O  3.00
siderite     Fe 1.00   C  1.00   O  3.00
apatite      Ca 5.00   P  3.00   O 12.00   OH 1.00
pyrite       Fe 1.00   S  2.00
anatase      Ti 1.00   O  2.00

```

Oxide input file

```

SiO2  Al2O3  Fe2O3   MgO  Na2O   CaO   K2O   TiO2   P2O5   MnO  inCO2   S
Bersham mdst (Nicholls, 1962)
62.56 20.60  3.55  1.25  0.62  0.34  3.35  0.94  0.17  0.02  0.90  0.02

```

APPENDIX 3

Log File: Bersham Mudstone Example

```

=====
Program LPNORM - University of Calgary - Copyright 1992,1993 Patrice de Caritat
[Ref: Caritat, Bloch and Hutcheon, Computers & Geosciences, 1993]
=====

```

```

*** Computing mineral data

```

#	MINERAL	MOL.WT.	#EL.	*CH.BAL.
1:	quartz	60.08	2	0.0000
	Si 1.00 O 2.00			
2:	ill1	796.63	5	0.0000
	K 2.00 Al 6.00 Si 6.00 O 20.00 OH 4.00			
3:	ill2	764.40	5	0.0000
	Na 2.00 Al 6.00 Si 6.00 O 20.00 OH 4.00			
4:	ill3	720.63	4	0.0000
	Al 4.00 Si 8.00 O 20.00 OH 4.00			
5:	kaolinite	258.16	4	0.0000
	Al 2.00 Si 2.00 O 5.00 OH 4.00			
6:	chl1	1486.96	4	0.0000
	Fe12.00 Si 8.00 O 20.00 OH16.00			
7:	chl2	1108.45	4	0.0000
	Mg12.00 Si 8.00 O 20.00 OH16.00			
8:	calcite	100.09	3	0.0000
	Ca 1.00 C 1.00 O 3.00			
9:	magnesite	84.31	3	0.0000
	Mg 1.00 C 1.00 O 3.00			
10:	siderite	115.86	3	0.0000
	Fe 1.00 C 1.00 O 3.00			
11:	apatite	502.32	4	0.0000
	Ca 5.00 P 3.00 O 12.00 OH 1.00			
12:	pyrite	119.97	2	0.0000
	Fe 1.00 S 2.00			
13:	anatase	79.90	2	0.0000
	Ti 1.00 O 2.00			

```

(*CH.BAL. based on following valences:  -2:O  -1:F,S,Cl,OH  +1:H,Na,K
+2:Mg,Ca,Mn,Fe,Ba  +3:Al  +4:C,Si,Ti  +5:N,P)

```

Select option (a), (b), (c) or (d)

a: calculate a(i,j) coefficients from 'minlist.dat' file

b: read a(i,j) coefficients from 'aij.dat' file

c: enter a(i,j) coefficients from keyboard

d: quit LPNORM now

Your choice... a

*** Computing a(i,j) coefficients

```
Si oxide in quartz      = 0.999920
                        SUM = 0.999920_____
K  oxide in ill1        = 0.118249
Al oxide in ill1        = 0.383969
Si oxide in ill1        = 0.452508
                        SUM = 0.954726_____
Na oxide in ill2        = 0.081083
Al oxide in ill2        = 0.400156
Si oxide in ill2        = 0.471584
                        SUM = 0.952824_____
Al oxide in ill3        = 0.282974
Si oxide in ill3        = 0.666971
                        SUM = 0.949945_____
Al oxide in kaolinite   = 0.394947
Si oxide in kaolinite   = 0.465446
                        SUM = 0.860393_____
Fe oxide in chl1        = 0.644363
Si oxide in chl1        = 0.323237
                        SUM = 0.967600_____
Mg oxide in chl2        = 0.436284
Si oxide in chl2        = 0.433613
                        SUM = 0.869897_____
Ca oxide in calcite     = 0.560300
C  oxide in calcite     = 0.439708
                        SUM = 1.000008_____
Mg oxide in magnesite   = 0.477974
C  oxide in magnesite   = 0.521976
                        SUM = 0.999950_____
Fe oxide in siderite    = 0.689173
C  oxide in siderite    = 0.379867
                        SUM = 1.069041_____
Ca oxide in apatite     = 0.558208
P  oxide in apatite     = 0.423852
                        SUM = 0.982060_____
Fe oxide in pyrite      = 0.665558
S  oxide in pyrite      = 0.534480
                        SUM = 1.200038_____
Ti oxide in anatase     = 1.000015
                        SUM = 1.000015_____
```

=====

```
MINERALS: quartz      ill1      ill2      ill3      kaolinite  chl1      chl2
calcite      magnesite siderite  apatite   pyrite     anatase
```

SAMPLE Bersham mdst (Nicholls, 1962)

```
SiO2 Al2O3 Fe2O3  MgO  Na2O  CaO  K2O  TiO2  P2O5  MnO  inCO2  S  SUM
62.56 20.60 3.55  1.25  0.62  0.34  3.35  0.94  0.17  0.02  0.90  0.02  94.32
```

*** Obj.fct.: minimize slack variables

INPUT TABLEAU

```
quar ill1 ill2 ill3 kaol chl1 chl2 calc magn side apat pyri anat  Si  Al
Fe  Mg  Na  Ca  K   Ti  P   Mn  C   S
0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00-1.00-
1.00-1.00-1.00-1.00-1.00-1.00-1.00-1.00-1.00-1.00-1.00
Si 62.56-1.00-0.45-0.47-0.67-0.47-0.32-0.43 0.00 0.00 0.00 0.00 0.00 0.00-1.00 0.00
0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00
Al 20.60 0.00-0.38-0.40-0.28-0.39 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00-1.00
0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00
Fe 3.55 0.00 0.00 0.00 0.00 0.00 0.00-0.64 0.00 0.00 0.00-0.69 0.00-0.67 0.00 0.00
0.00-1.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00
```

```

Mg  1.25 0.00 0.00 0.00 0.00 0.00 0.00 0.00-0.44 0.00-0.48 0.00 0.00 0.00 0.00 0.00 0.00
0.00-1.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00
Na  0.62 0.00 0.00-0.08 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00
0.00 0.00-1.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00
Ca  0.34 0.00 0.00 0.00 0.00 0.00 0.00 0.00-0.56 0.00 0.00-0.56 0.00 0.00 0.00 0.00
0.00 0.00 0.00-1.00 0.00 0.00 0.00 0.00 0.00 0.00
K   3.35 0.00-0.12 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00
0.00 0.00 0.00 0.00-1.00 0.00 0.00 0.00 0.00 0.00
Ti  0.94 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00-1.00 0.00 0.00
0.00 0.00 0.00 0.00 0.00-1.00 0.00 0.00 0.00 0.00
P   0.17 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00-0.42 0.00 0.00 0.00 0.00
0.00 0.00 0.00 0.00 0.00 0.00-1.00 0.00 0.00 0.00
Mn  0.02 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00
0.00 0.00 0.00 0.00 0.00 0.00 0.00-1.00 0.00 0.00
C   0.90 0.00 0.00 0.00 0.00 0.00 0.00 0.00-0.44-0.52-0.38 0.00 0.00 0.00 0.00 0.00
0.00 0.00 0.00 0.00 0.00 0.00 0.00-1.00 0.00
S   0.02 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00-0.53 0.00 0.00 0.00
0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00-1.00

```

*** Initial EPS value is: 1.000000E-06

M= 12 M1= 0 M2= 0 M3= 12

*** SIMP2 - EPS value increased to: 1.000000E-05

*** A finite solution was found for sample
Bersham mdst (Nicholls, 1962)

```

36  ( 36.01) wt% quartz
17  ( 16.87) wt% kaolinite
 3  (  3.19) wt% chl1
 3  (  2.87) wt% chl2
 8  (  7.65) wt% ill2
 0  (  0.21) wt% calcite
28  ( 28.33) wt% ill1
 1  (  0.94) wt% anatase
 0  (  0.40) wt% apatite
 2  (  2.13) wt% siderite
 0  (  0.04) wt% pyrite
-----
99  ( 98.63) wt% total
    -0.02 wt% objective function

```

```

----- VERIFY OUTPUT -----
|
| Si: 62.56 wt% used of the initial 62.56 wt% (diff.= 0.00 wt%)
| Al: 20.60 wt% used of the initial 20.60 wt% (diff.= 0.00 wt%)
| Fe:  3.55 wt% used of the initial  3.55 wt% (diff.= 0.00 wt%)
| Mg:  1.25 wt% used of the initial  1.25 wt% (diff.= 0.00 wt%)
| Na:  0.62 wt% used of the initial  0.62 wt% (diff.= 0.00 wt%)
| Ca:  0.34 wt% used of the initial  0.34 wt% (diff.= 0.00 wt%)
| K :  3.35 wt% used of the initial  3.35 wt% (diff.= 0.00 wt%)
| Ti:  0.94 wt% used of the initial  0.94 wt% (diff.= 0.00 wt%)
| P :  0.17 wt% used of the initial  0.17 wt% (diff.= 0.00 wt%)
| Mn:  0.00 wt% used of the initial  0.02 wt% (diff.= 0.02 wt%)
| C :  0.90 wt% used of the initial  0.90 wt% (diff.= 0.00 wt%)
| S :  0.02 wt% used of the initial  0.02 wt% (diff.= 0.00 wt%)
|
|                                     total diff.= 0.02 wt%
|
-----

```

=====

Quitting LPNORM

=====